Escuela Politécnica Superior de Jaén

Grado en Ingeniería Informática

**UNIVERSIDAD DE JAÉN**
*Escuela Politécnica Superior de Jaén*

Trabajo Fin de Grado

# Ensembles dinámicos de modelos machine learning para regresión

**Alumno:** Marta Molina Pérez

Tutor: Prof. Dña. Mª Dolores Pérez Godoy

Prof. D. Antonio Jesús Rivera Rivas

Dpto: Informática

UNIVERSIDAD DE JAÉN

D./Dª Prof. Dña. Mª Dolores Pérez Godoy y D./Dª Prof. D. Antonio Jesús Rivera Rivas, tutor(es) del Trabajo Fin de Grado titulado: **Ensembles dinámicos de modelos machine learning para regresión**, que presenta Marta Molina Pérez, autoriza(n) su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, julio de 2022

El estudiante

**Firmado por MOLINA PEREZ MARTA - ***0694** el día 15/07/2022 con un certificado emitido por AC FNMT Usuarios**

Marta Molina Pérez

Los tutores

PEREZ GODOY MARIA DOLORES - 75093149N

Firmado digitalmente por PEREZ GODOY MARIA DOLORES - 75093149N
Fecha: 2022.07.15 14:07:39 +02'00'

Prof. Dña. Mª Dolores Pérez Godoy

RIVERA RIVAS ANTONIO JESUS - 26015863B

Firmado digitalmente por RIVERA RIVAS ANTONIO JESUS - 26015863B
Fecha: 2022.07.15 13:38:15 +02'00'

Prof. D. Antonio Jesús Rivera Rivas

**Agradecimientos**

A mis padres, Vicente y Mari Seba, por darme la oportunidad de estudiar la carrera que quería, por haberme enseñado a confiar en mí y nunca tirar la toalla, por hacerme ver que puedo con todo; por hacer de mí la persona que hoy soy.

A Agustín, por apoyarme cada día, por vivir conmigo cada uno de los malos y buenos momentos, por estar ahí día y noche a pie de cañón conmigo. Por haber tenido la paciencia de aguantar los baches y por los buenos, muy buenos, momento vividos sobre todo, gracias por ser mi apoyo incondicional.

A mi hermana Cristina y mis abuelos, por confiar en mí, por sufrir conmigo y alegrarse por mis logros casi más que yo misma. Por apoyarme cada día, demostrarme que puedo con lo que me proponga y por sentise tan orgullosos de mí.

A mis tutores, Loli y Antonio. Gracias por estar siempre disponibles para lo que fuera, por tenderme una mano siempre que lo he necesitado y por haberme enseñado tanto.

Este Trabajo de Fin de Grado se lo dedico a mi abuelo Marcos Pérez Bautista. Abuelo, tu “chiquitilla´´ lo ha conseguido. De ti he aprendido lo que es la paciencia, el trabajo duro y la constancia.

Desde donde estés, sé que te sientes muy orgulloso de mí.

# Tabla de contenidos

# Lista de figuras

# Lista de tablas

# Lista de algoritmos

# Capítulo 1

# INTRODUCCIÓN, OBJETIVOS Y PLANIFICACIÓN DEL PROYECTO.

En este capítulo se presenta la especificación del proyecto, contextualizando mediante una breve introducción y seguidamente, explicando la motivación que da lugar a la realización del mismo. Además, se exponen los objetivos a cumplir en el proyecto y por último se comenta la estructura del documento y planificación temporal que se he llevado a cabo para el desarrollo del mismo.

## 1.1. Introducción

Cada vez son más los datos que se generan y a mayor velocidad. Este fenómeno se da por varias casuísticas. En primer lugar, vivimos en la sociedad de la información, donde cada una de las áreas de la vida cotidiana está ya integrada con la tecnología, utilizando dispositivos conectados durante el trabajo, haciendo deporte e incluso durante el entretenimiento. Esto es posible gracias al llamado Internet de la cosas (IoT - Internet of Things). La IoT se refiere a la interconexión en red de todos los objetos cotidianos, que a menudo están equipados con algún tipo de inteligencia. Evans [2011].

Por lo tanto, todos estos dispositivos conectados como ordenadores, tablets, relojes inteligentes e incluso aspiradoras con wifi, junto con la actividad de los usuarios, son los culpables de la generación diaria de millones y millones de datos. Por otro lado, gracias a los últimos avances tecnológicos, el acceso y almecanamiento de datos ya no es un problema y las empresas lo aprovechan para almacenar todos los datos que son generados.
Como podemos observar en la imagen, 1.1 la cantidad de datos que se genera en un solo

minuto a través de diversas aplicaciones, es abismal.

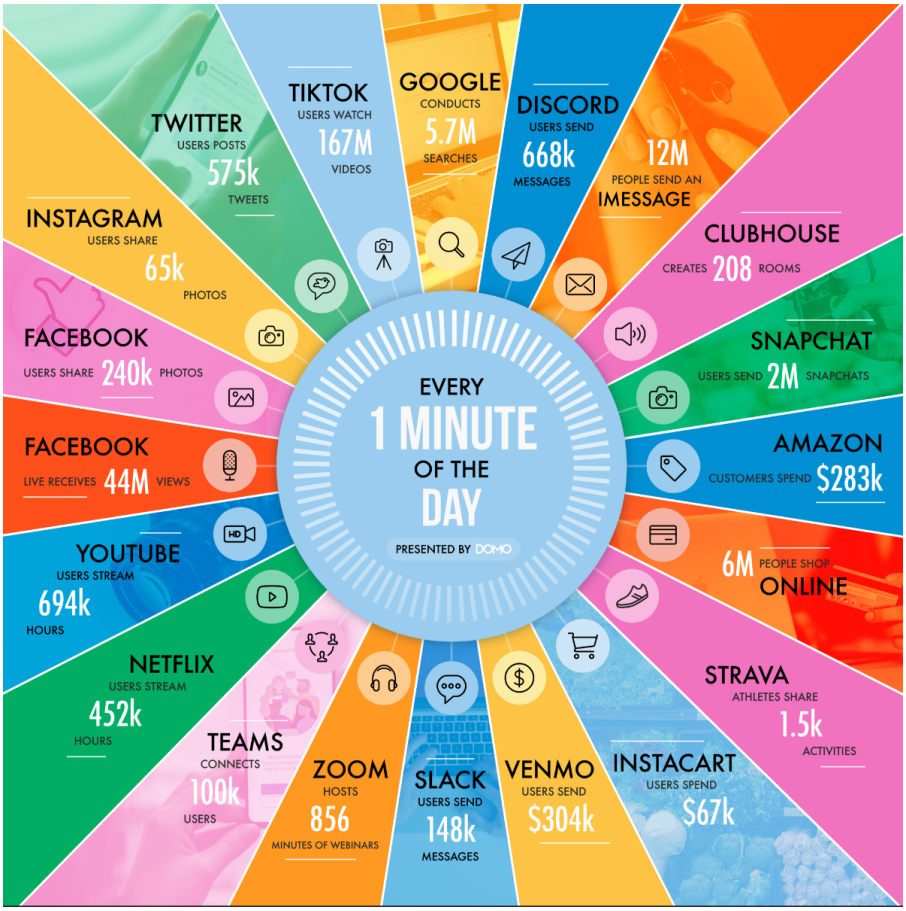


Figura 1.1: Cantidad de datos medios por aplicación generada cada minuto (2021) Fuente : Domo.es

Pero, **¿por qué resultan tan interesantes estos datos?**

La pregunta tiene una sencilla respuesta y es que dichos datos contienen gran cantidad de información y conocimiento. Conocimiento que muestra el comportamiento humano y que puede ser aprovechado para la toma de decisiones de las empresas.

Sin embargo, el proceso para obtenerlo no es tan fácil.Aquí juega un papel muy importante la Ciencia de Datos. Esta es un área fundamental para la explotación de datos y la generación de conocimiento. Entre los objetivos que persigue se encuentra la búsqueda de modelos que describan patrones y comportamientos a partir de los datos con el fin de tomar decisiones o hacer predicciones García et al. [2018].

En los últimos años, la Ciencia de Datos ha experimentado un gran auge en empresas de diverso índole, ya que su aplicación puede llevar a la obtención de conocimiento de cualquier ámbito. Así, puede ser utilizada para sectores como la banca, el marketing, la

medicina, los seguros, la biología, etcétera. Esto será posible siempre y cuando se tengan datos de calidad referentes al campo de aplicación.

Adentrándonos en la Ciencia de Datos, el paso más determinante es la creación de un modelo capaz de aprender de los datos (ya procesados) y la revelación de patrones novedosos e inesperados. De esto se encarga en concreto el proceso de Minería de Datos. Dentro de este proceso encontramos distintas maneras de proceder según el objetivo del problema, la naturaleza del mismo y la función objetivo que se persigue. (Véase Sección 2.3).Por lo que, cada vez que se afronta un nuevo problema, se trata de un reto distinto que debe abordar el científico de datos, donde deben entrar en juego tanto sus conocimientos en computación , estadística y matemática, como su imaginación y creatividad, para ser capaz de dar un enfoque novedoso y aportar valor al proceso.

## 1.2. Motivación

Debido a la explicada importancia de la Ciencia de Datos y el auge de implantación en las distintas empresas, este Trabajo Fin de Grado se enmarca dentro de dicha disciplina, concretándose en la tarea predictiva de la regresión. Esta tarea es la encargada de la predicción de muestras cuyas salidas son variables numéricas. Este enfoque es de gran importancia debido a su gran aplicación a problemas de cualquier ámbito. Además, una de las ventajas es no tener que dar ninguna lista acotada de salidas posibles, como sí que pasa en clasificación. Por lo tanto, es una técnica más versátil y a su vez compleja.

Una de las técnicas más prometedoras que se han estudiado teórica y empíricamente para la tarea predictiva es el *ensemble* (Véase Sección 3). Concretamente, destacan los resultados que se están obteniendo con los ensembles dinámicos. Esta es una técnica que se basa en la combinación de predicciones de los mejores estimadores base, seleccionados según la muestra de prueba Ko et al. [2008]. Este tipo de ensemble ha demostrado en diversos artículos que disminuye el error con respecto a las técnicas clásicas (Sección 2.3.3) tales como vecinos más cercanos para regresión, árboles de decisión para regresión, o máquina de soporte de vectores, entre otros, además de superar a los ensembles estáticos convencionales. Sin embargo, esta técnica ha sido mucho más explotada para el campo de la clasifiación, dejando a un lado la tarea de regresión.

Ante esta situación, se ha decidido indagar, estudiar e implementar una librería nueva para problemas de regresión utilizando las bases de la técnica de Selección Dinámica de Ensembles.

Para dar forma a esta librería, se ha realizado una ardua revisión bibliográfica con el objetivo de obtener los conceptos básicos de esta técnica novedosa y realizar un estudio del estado del arte de la misma.

## 1.3. Objetivos

Los objetivo a alcanzar durante la realización de este Trabajo Fin de Grado son los siguientes:

1. Estudiar los fundamentos *data science*, *minería de datos* y *machine learning*.

2. Conocer los fundamentos de las técnicas de *ensembles*.

3. Realizar una búsqueda y estudio bibliográfico sobre los modelos de *ensembles dinámicos*.
   existentes, en general en el área de minería predictiva y en particular en el área de regresión.

4. Desarrollo de métodos de ensembles dinámicos aplicados a solucionar problemas de regresión.

5. Experimentación y análisis de los resultados obtenidos por los modelos.

## 1.4. Estructura del documento

En esta sección se detalla la estructura del documento que nos acontece.

Este se organiza principalmente en capítulos, subdivididos a su vez en secciónes y subsecciones:

- En el capítulo 1, se presenta una breve introducción para poner al lector en antecedentes antes de comenzar con el resto del documento. También aparecen en este

capítulo, la motivación de la que surge el presente proyecto, la explicación de lo estructura del documento y la planificación temporal del mismo.

- En el capítulo 2, se explica el marco teórico en el que se encuentra el tema principal del Proyecto. En concreto, se exponen los conceptos de Ciencia de Datos y Minería de Datos, estudiando en mayor profundidad la tarea de regresión.

- Durante el capítulo 3 se introduce la técnica de ensembles que da lugar seguidamente a una explicación detallada del enfoque dinámico de dicha técnica.
- En el capítulo 4 se detallan los procesos de análisis, diseño y desarrollo de la librería. Se describe aquí cada decisión tomada para la implementación y los métodos que se deciden desarrollar.
- Durante el capítulo 5 se presentan las pruebas realizadas y los resultados obtenidos para la comprobación de los métodos implementados.
- Para finalizar, el documento se cierra con el capítulo 6, donde se realiza una breve conclusión final.

## 1.5. Planificación temporal del proyecto

El trabajo aquí expuesto se ha desarrollado entre el día 1 de Noviembre de 2021 y el día de julio de 2022. En este intervalo de tiempo, se han empleado más de 300 horas (equivalentes a 12 crédito ECTS).

Para la correcta planificación del proyecto, la primera tarea a realizar es la identificación de las tareas en las que se dividen el proyecto, de las dependencias entre ellas y los tiempos de duración.

En la imagen 1.2 se muestra la tabla en la que se especifican las tareas, la duración, la fecha de inicio, la fecha de finalización y las tareas predecesoras de cada una. A partir de dicha tabla, se obtiene un diagrama de Gantt en el que se visualiza la planificación llevada a cabo para la realización de este Trabajo Fin de Grado.

El diagrama de Gantt obtenido se presenta en varias imagenes debido a que representarlo en una única página no sería práctico, pues sería inteligible. Por lo tanto en las imagenes 1.3, 1.4, 1.5, 1.6 y 1.7 se presenta secuencialmente el diagrama de Gantt.

| | Tarea | Duración (días) | Comienzo | Fin | Predecesora |
|---|---|---|---|---|---|
| 1 | Estudio conceptos básicos sobre ensembles dinámicos. | 14d | 1/11/2021 | 15/11/2021 | |
| 2 | Estudio del estado del arte de ED para clasifiación | 28d | 16/11/2021 | 14/12/2021 | 1 |
| 3 | Estudio del estado del arte de ED para regresión/series temporales | 24d | 15/12/2021 | 8/01/2022 | 1 |
| 4 | Introducción a librería deslib | 13d | 9/01/2022 | 22/1/22 | 2 |
| 5 | Análisis de la biblioteca a desarrollar | 28d | 23/1/22 | 20/2/22 | 3,4 |
| 6 | Diseño de la biblioteca a desarrollar | 22d | 21/2/22 | 15/3/22 | 5 |
| 7 | Desarrollo de la biblioteca. | 50d | 16/3/22 | 5/5/2022 | 6 |
| 8 | Pruebas y experimentación. | 13d | 6/5/22 | 19/05/2022 | 7 |
| 9 | Análisis de resultados. | 11d | 20/05/2022 | 31/5/22 | 8 |
| 10 | Desarrollo de la documentación | 41d | 1/06/2022 | 12/7/2022 | 9 |
| 11 | Revisión y finalización. | 2d | 13/7/2022 | 15/7/22 | 10 |

Figura 1.2: Planificación temporal de la realización del Proyecto.

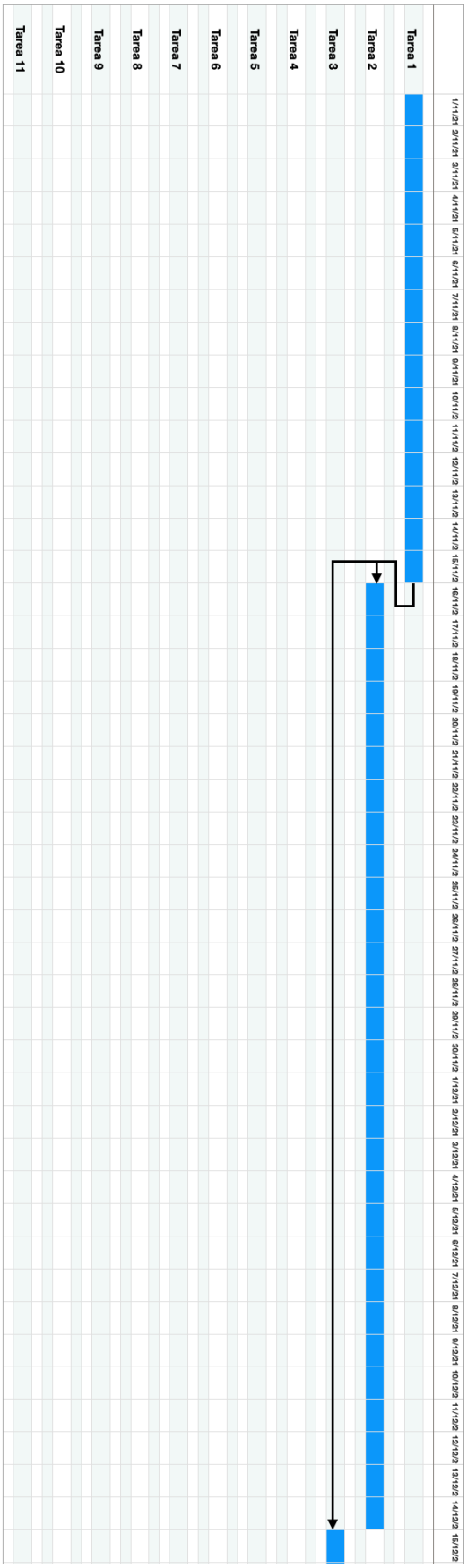


Figura 1.3: Planificación temporal de la realización del Proyecto. Diagrama de Gantt (I). Fuente: Elaboración propia.

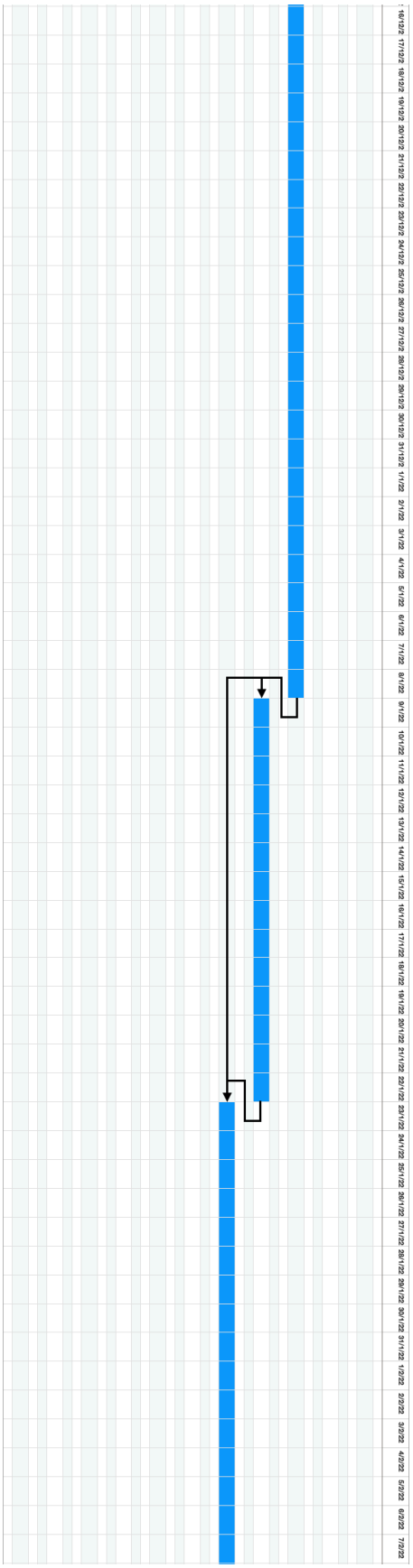

Figura 1.4: Planificación temporal de la realización del Proyecto. Diagrama de Gantt (II). Fuente: Elaboración propia.

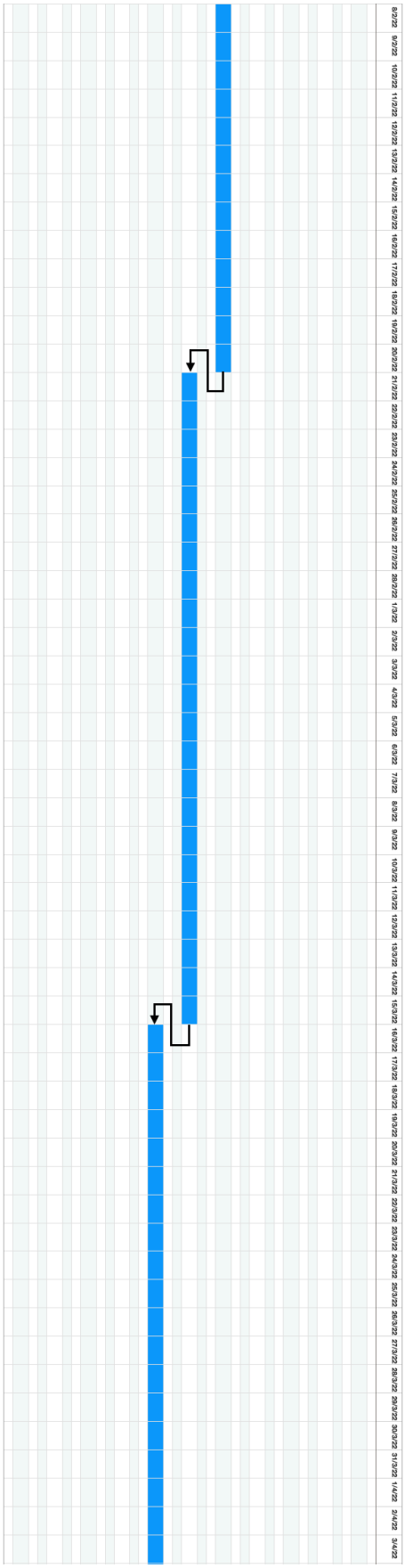

Figura 1.5: Planificación temporal de la realización del Proyecto. Diagrama de Gantt (III). Fuente: Elaboración propia.

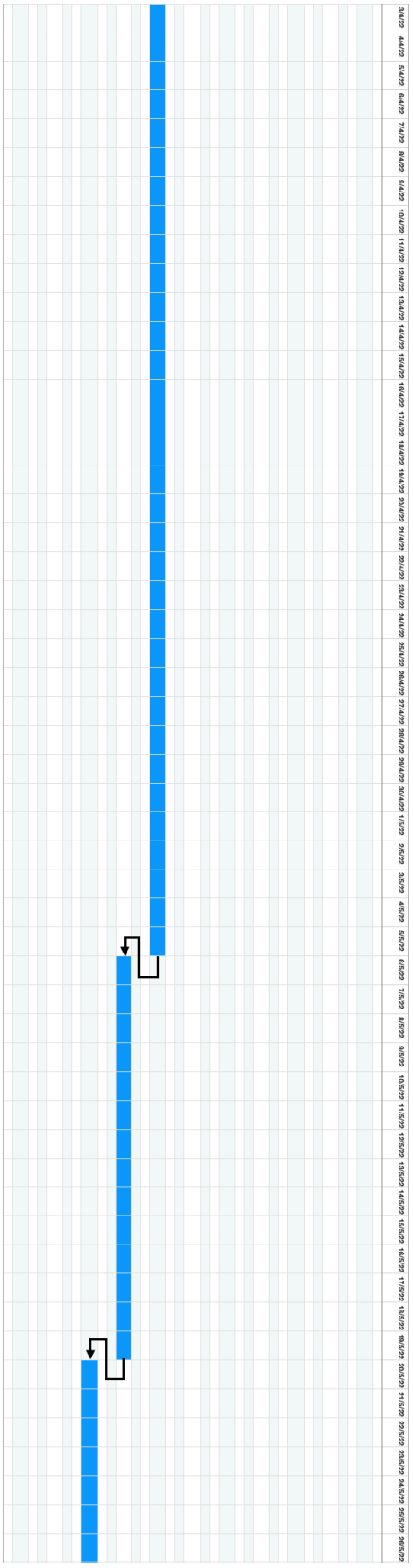

Figura 1.6: Planificación temporal de la realización del Proyecto. Diagrama de Gantt (IV). Fuente: Elaboración propia.

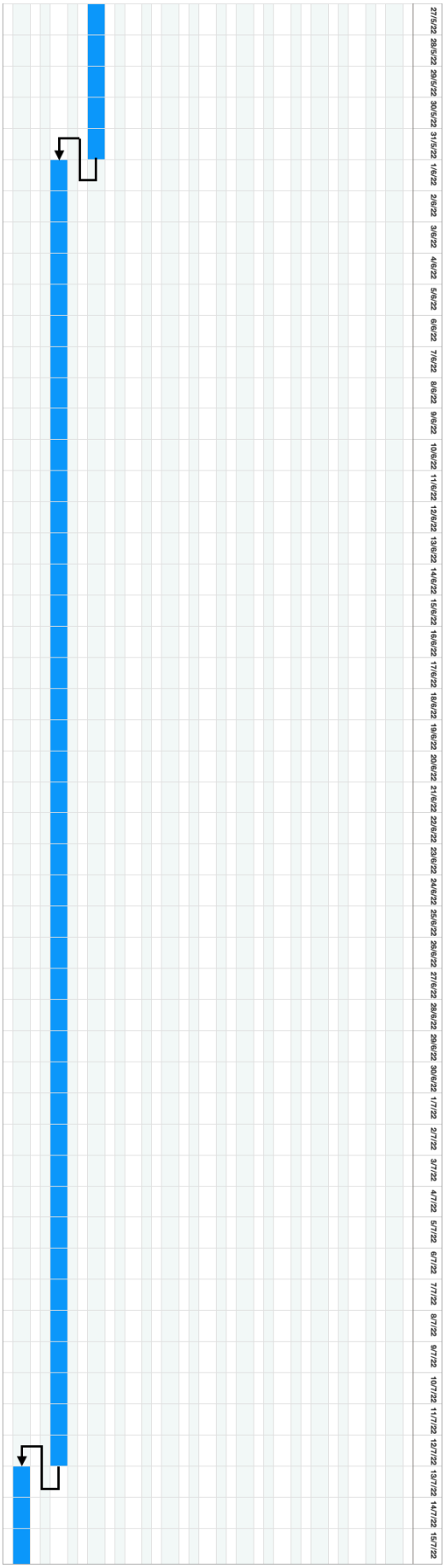

Figura 1.7: Planificación temporal de la realización del Proyecto. Diagrama de Gantt (V). Fuente: Elaboración propia.

# Capítulo 2

# ANTECEDENTES

## 2.1. Introduction

Durante este capítulo se pondrá en antecedente toda la terminología y marco teórico que engloba al presente trabajo. Para ello, se ha realizado un estudio de las disciplinas más relacionadas, que son las que se exponen a continuación. Cabe destacar que es necesaria su aclaración debido a que los distintos ámbitos expuestos en los siguientes puntos suelen ser confundidos y a veces, sus definiciones se fusionan debido a las pequeñas diferencias que las separan.

## 2.2. Ciencia de datos

La Ciencia de Datos o Data Science Dhar [2013] es un campo interdisciplinar que une la estadística, la minería de datos, el aprendizaje automático, y la analítica predictiva para poder extraer conocimiento a partir de diferentes fuentes de datos. Así pues, es un ámbito del conocimiento que engloba las habilidades asociadas al procesamiento de datos. (Figura 2.1)

El término de Ciencia de Datos fue por primera vez acuñado en el año 2002. Apareció gracias a que el *'International Council for Science: Committee on Data for Science and Technology' (CODATA)* creó el *Data Science Journal* cod. En esta revista se trataban temas como la descripción de los problemas de sistemas de datos, su publicación en Internet y los problemas legales de estos, etc. Sin embargo, araigó más fuerza en 2003, cuando la Universidad de Columbia empezó a publicar *The Journal of Data Science* jds la cual ofreció una plataforma en la que los profesionales del ámbito podían intercambiar ideas y presentar

diferentes perspectivas.

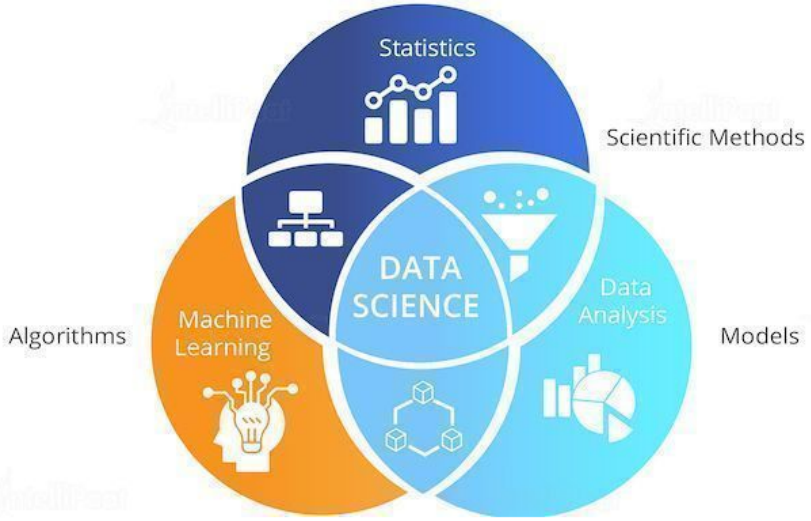


Figura 2.1: Disciplinas que forman parte de la Ciencia de Datos. Fuente :ecointeligencia.com

Dentro de las tareas de un científico de datos se pueden incluir el desarrollo de estrategias para analizar datos, la preparación y procesamiento de datos, la exploración, el análisis y la visulaización de ellos, además de ser capaz de construir e implementar modelos de *Machine learning* (Mitchell and Mitchell [1997]).

Por lo tanto, debe ser un profesional que domine las ciencias matemáticas, estadísticas, de la computación y analíticas, además de poseer conocimentos de programación.

El matemático José Antonio Guerrero, considerado uno de los mejores ciéntificos de datos (alcanzó el primer puesto en Kaggle[1]), describe a la personalidad de científico de Datos de la siguiente manera :

> *“ Un científico de Datos es una persona con fundamentos en matemáticas, estadística y métodos de optimización, con conocimientos en lenguajes de programación y que además tiene una experiencia práctica en el análisis de datos reales y la elaboración de modelos predictivos.*
> *De las tres características quizás la más difícil es la tercera; no en vano la modelización de los datos se ha definido en ocasiones como un arte. Aquí no hay reglas de oro, y cada conjunto de datos es un lienzo en blanco."*

[1] Plataforma web que reúne la comunidad Data Science más grande del mundo, con más de 536 mil miembros activos en 194 países, recibe más de 150 mil publicaciones por mes, que brindan todas las herramientas y recursos más importantes para progresar al máximo en data science.A. [2021]

## 2.3. Minería de datos

El término de Minería de Datos (Han et al. [2011]) hace referencia al proceso de aplicación de técnicas de *Machine Learning* sobre grandes cantidades de datos, con el fin de descubrir patrones o tendencia útiles y previamente desconocidas.

La Minería de Datos se enmarca en el KDD (Knowledge Discovery from Databases), como una fase de este, precismanete la más importante. Sin embargo, informal y erróneamente se asocian ambos términos haciendo referencia a la definición del KDD (Fayyad et al. [1996a]).

### 2.3.1. KDD

El proceso del KDD Fayyad et al. [1996a]) o descubrimiento de conocimiento en base de datos, se define como : Proceso no trivial de la identificación de patrones válidos, nuevos, potencialmente usables y comprensibles a partir de datos.

> "Non-trivial process of identifying valid, novel, potentially useful and ultimately comprehensible patterns from data." Fayyad et al. [1996b]

El KDD es un proceso global de descubrimiento del conocimiento de bases de datos, con especial ímpetu en la búsqueda de patrones comprensibles. El proceso se compone del uso de la base de datos, algoritmos para la obtención de patrones (Minería de Datos en sentido estricto) y evaluación de resultados y elección de los modelos que se consideren conocimiento. Es un proceso iterativo e interactivo; a veces es necesario realizar el proceso más de una vez para encontrar conocimiento de calidad, además de necesitar expertos en el dominio que se está tratando para la validación del conocimiento extraído, para evitar incongruencias.

El KDD, como ya se ha introducido previamente, consta de varias etapas, como se detallan en la figura 2.2 y se detallan a continuación :

- **Integración y recopilación de datos:**
  Esta etapa consiste en un primer contacto con los datos, la comprensión del dominio de ellos, la realización de análisis exploratorios de los datos y extracción de conocimiento a priori, entre otros. La principal ventaja de la previa familizarización con el dominio del problema y conocimiento a priori es la disminución del espacio de soluciones, lo que supone una eficiencia mayor para el resto del proceso.

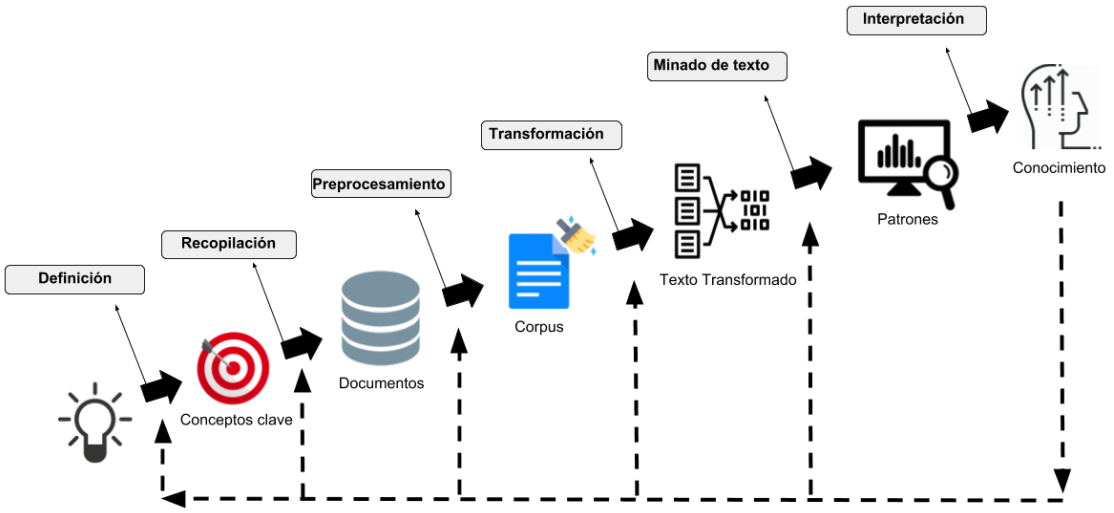


Figura 2.2: Etapas KDD. Fuente:linkedin.com

Otro de los problemas que se solventan en esta etapa es la unificación y agrupamiento de datos de diferente naturaleza. Es decir, imponer un formato común a todos los datos procedentes de distintas fuentes para que sea más fácil manejarlos, eliminar evidencias y reducir datos duplicados.

- **Selección de datos, limpieza y transformación:**
  Etapa que se encarga de preprocesar los datos que se van a introducir al algoritmo de aprendizaje. Se revisan los datos para asegurar la calidad de estos. Esto es un factor que afecta directamente a los resultados obtenidos.

  Así, la calidad del conocimiento descubierto tras este proceso no solo depende del algoritmo utilizado en la fase de Minería de Datos, sino también de los datos que se utilizan en él. Los datos son la materia prima para que el algoritmo aprenda correctamente. Si estos no presentan calidad, los resultados dados por el algoritmo tampoco la tendrán.

  Este proceso engloba tres tareas fundamentales: En primer lugar la limpieza de datos, que consiste en el tratamiento de valores perdidos, identificación y manejo de datos anómalos y reducción de ruido e inconsistencias; en segundo lugar, la transformación de los datos, adecuándolos a la manera que tendrá de tratarlo el algoritmo que se utilizará en la siguiente etapa. Algunas de las transformaciones más comunes son la discretización, construcción de nuevos atributos derivados de otros ya existentes, normalización, agregación o generalización de datos; y por último, la reducción de dimensionalidad, que consiste en la utilización de técnicas para reducir el volumen de datos, pero sin perder calidad. Es decir, se pretende reducir la dimensionalidad sin

que esto afecte negativamente al conocimiento obtenido. Para ello se pueden utilizar técnicas de reducción de atributos o de reducción de instancias.

- **Selección y aplicación de algoritmos de Minería de Datos.**
  Etapa de construcción del modelo de Minería de Datos que se utilizará para el proceso. En esta etapa, es importante escoger un modelo que funcione bien para el problema que se está tratando. Normalmente, se hace un estudio previo de los algoritmos disponibles y se elige la técnica más apropiada.

  Para la construcción del modelo se debe realizar una partición de los datos disponibles. Se crean dos subconjuntos de datos, uno será el conjunto de entrenamiento y el otro el de test. El conjunto de entrenamiento se utiliza para entrenar el modelo, para que aprenda de los datos. Después con el conjunto de test, se evalua el modelo. Con esto podemos realizar pruebas para encontrar la mejor opción de modelo de minería de datos a utilizar.

- **Evaluación, interpretación y visualización de los resultados.**
  En un primer lugar, la evaluación de los modelos creados en la etapa anterior, se realiza en esta fase. Para ello, se utiliza el subconjunto de test. Para elegir qué modelos son más válidos se deben seguir los siguientes criterios: Precisión, comprensibilidad y novedad. Fayyad et al. [1996a] Existen distintas técnicas de evaluación como la validación cruzada o el bootstrapping Kohavi et al. [1995], entre otras.

  La interpretación de los mejores modelos probados ayuda a la selección del mejor modelo final.

- **Difusión y utilización del nuevo conocimiento extraído en el proceso.**
  En esta última etapa es necesaria la difusión del modelo construido mediante la elaboración de informes para su distribución, la utilización de dicho conocimiento de manera independiente y por último, la incorporación a sistemas ya existentes.

### 2.3.2. Tareas de Minería de Datos

Las tareas de la Minería de Datos se clasifican en función del objetivo o resultado que se quiere obtener del problema a abordar con el algoritmo de Minería de Datos. Existen dos enfoques de tareas en la Minería de Datos, como se muestra en la Figura 2.3, que recoge un resumen de la clasificación de las tareas de Minería de Datos.

**Tareas predictivas**

El modelo debe ser capaz de generar una salida, **predicción**, por cada una de las muestras.

Dentro de este tipo de tareas, encontramos otra subdivisión basada en el tipo de dato a predecir generado por el modelo.

- Clasificación : La predicción debe ser de tipo categórico, es decir, el modelo debe predecir a qué clase, de un conjunto acotado, pertenece cada muestra.
- Regresión : La variable objetivo es del tipo continua.
- Predicción de series temporales : En este caso, la variable objetivo es de tipo continuo también, pero la diferencia con la tarea anterior es que en esta, hay una componente temporal que hay que tener en cuenta para la predicción.

**Tareas descriptivas**

El modelo generado debe mostrar relaciones entre variables o instancias, además de otro tipo de información descriptiva de las muestras del modelo. Algunos de los ejemplos más conocidos son :

- Reglas de asociación: El descubrimiento de reglas que asocian atributos de un conjunto de datos, nos brinda información valiosa sobre las relaciones entre las características. Gracias a este enfoque, se obtiene información descriptiva en forma de reglas.

- Agrupamiento: Consiste en agrupar en $n$ grupos todas las instancias del conjunto de datos. Cada instancia pertenecerá al grupo más cercano, es decir, el que contenga los ejemplos más parecidos. Se debe minimizar la similitud entre grupos y maximizar dentro de cada grupo.

### 2.3.3. Técnicas de Minería de Datos

La Minería de Datos hace uso de técnicas del *Machine Learning* o aprendizaje automático para crear sus modelos.
El aprendizaje automático o Machine Learning se puede definir como el proceso de aprendizaje de una máquina. Es decir, cómo pasa de tener unos datos de entrada a obtener los

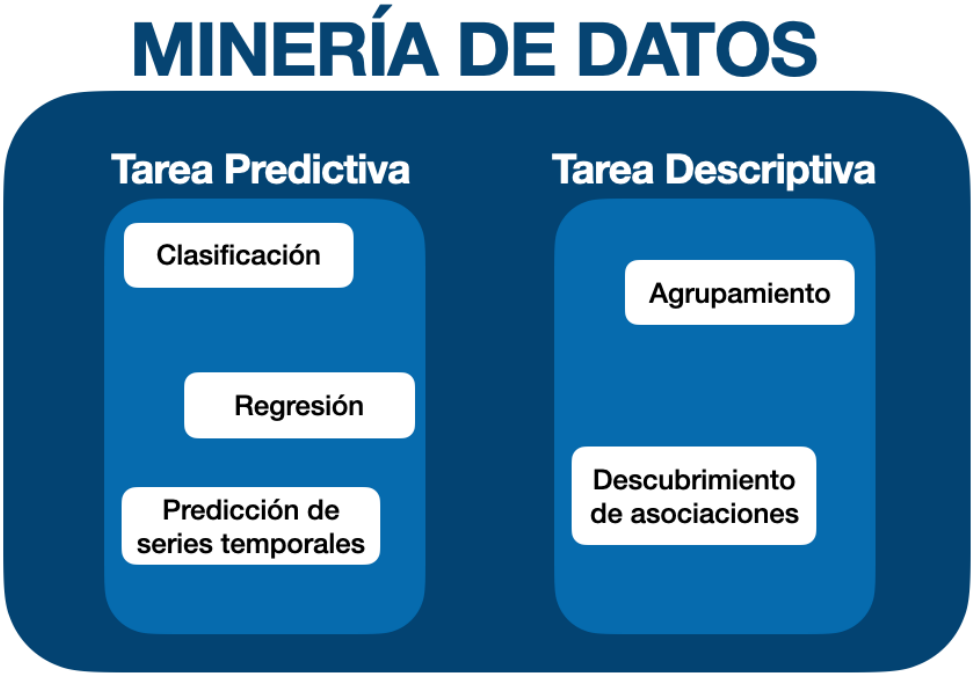


Figura 2.3: Principales tareas de Minería de Datos. Fuente: Elaboración propia.

resultados asociados a estos datos. Mitchell and Mitchell [1997] Para poder resolver este tipo de problemas, el aprendizaje automático se sirve de tres componentes fundamentales:

1. Una tarea T ( Sección 2.3.2),

2. Una medida de rendimiento (como por ejemplo, el error que comete), y

3. Una experiencia o previo aprendizaje.

Así, una máquina, a partir de experiencia previa (datos de entrenamiento), será capaz de realizar una tarea cuyo rendimiento es medible.

El aprendizaje automático se puede utilizar para emplearlo a distintos tipos de aprendizaje. A continuación se expone una taxonomía de ellos.

**Paradigmas de aprendizaje**

Los paradigmas de aprendizaje se dividen en función de la información disponible que tiene un algoritmo de *Machine Leraning* a la hora de su entrenamiento, según se explican

en Charte Ojeda and Charte Luque [2020]. Veamos las diferencias:

- **Aprendizaje supervisado** Durante este aprendizaje, los algorimtos disponen de las salidas reales de los datos, Así pues, se puede comparar la salida generada por el algoritmo con la salida verdadera. Normalmente, el propio algoritmo genera ajustes para corregir sus fallos en función de si la predicción ha sido satisfactoria o no. Este tipo de aprendizaje es el utilizado para tareas de predicción.
- **Aprendizaje no supervisado** En este caso, el modelo no es capaz de saber si está realizando bien o no la tarea, si está generando salidas acertadas. Por lo tanto, este tipo de aprendizaje está más orientado a la descripción de los datos que a la predicción, ya que con este tipo de modelos seremos capaces de analizar la distribución, varianza, o grado de concurrencia entre variables.
- **Aprendizaje semisupervisado** Este tipo de aprendizaje, como bien su nombre indica, se encuentra a caballo entre los dos anteriores. Esto significa que, solo tendrá la información completa, es decir, tendrá disponible la salida real, de algunas muestras, pero no de todas. Este tipo de aprendizaje está cogiendo cada vez más fuerza debido a la gran cantidad de recursos económicos, de tiempo y humanos que supone el etiquetado de instancias, sobretodo para grandes volúmenes de datos.
- **Aprendizaje por refuerzo** En este caso, no se le suministran las salidas como tal, pero sí que se le otorga un premio más o menos grande en función de cómo de acertado está siendo con las predicciones.

Algunas de las técnicas de Minería de Datos más conocidas son las siguientes (Mining [2006]):

- Redes Neuronales: Las redes neuronales artificiales están basadas en el funcionamiento de las redes de neuronas biológicas. Las redes neuronales están compuestas de nodos dispuestos en varias capas. Las capas suelen agruparse en tres: una capa de entrada (con tantos nodos como variables de entrada), una o varias capas ocultas y una capa de salida (con uno o varios nodos que representan la salida). Su funcionamiento se basa en propagar los ejemplos desde los nodos de entrada, pasando de una neurona a la neurona de la capa siguiente, hasta llegar a la capa de salida el resultado final. Los resultados se obtienen gracias a las ponderaciones de cada interconexión entre neuronas. (Fig. 2.4)

  Ejemplos destacados son el Perceptrón Multicapa y Redes Neuronales Convolucionales (Matich [2001]).

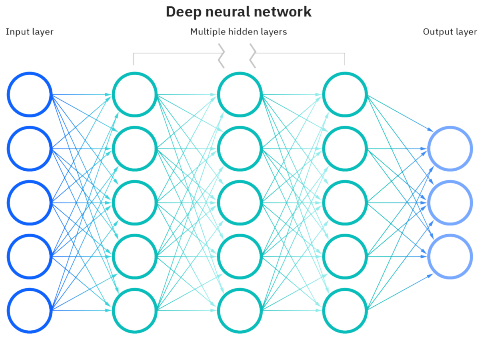


Figura 2.4: Ejemplo de Red Neuronal. Fuente: ibm.com.

- Árboles de decisión: Los algoritmos que dan lugar a un árbol de decisión son aquellos que clasifican la información disponible generando un árbol. Cada nodo del árbol es una variable y cada posible valor de la variable genera una rama en dicho nodo. Por último, los nodos hoja contendrán el valor de salida para las muestras que llegan hasta ellas. La manera de predecir una nueva muestra es recorriendo el árbol, escogiendo las ramas según los valores de la muestra hasta llegar a un nodo hoja. Todas las muestras que llegan a un nodo hoja tendrán la misma salida. Esto es fácil si los atributos son categóricos, pues cada valor da lugar a una rama. Sin embargo, para atributos numéricos,lo más habitual es establecer rangos y cada uno de ellos genera una rama distinta. (Fig. 2.5)
  Algunos de los más destacados son el C4.5, ID3 o CART (Singh and Gupta [2014]).

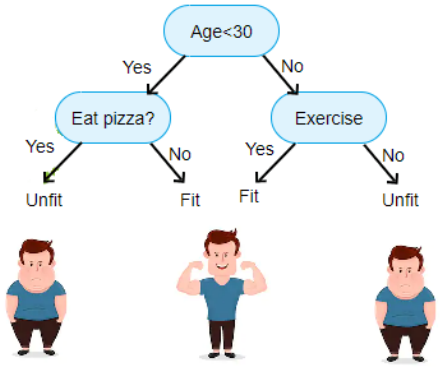


Figura 2.5: Ejemplo de árbol de decisión. Fuente: fhernand.github.io.

- Clustering: Los algoritmos de *clustering*, como el KMeans o KMedoids (Arora et al. [2016]), se encargan de agrupar las distintas instancias en un número predefinido de grupos o *clusters* durante el entrenamiento. Una vez definidos dicho grupos, la evaluación de una nueva muestra se realiza mediante el cálculo de distancias al punto de referencia de cada grupo. El grupo que se encuentre más cercano a la instancia de prueba, será al que se asigna. (Fig. 2.6)

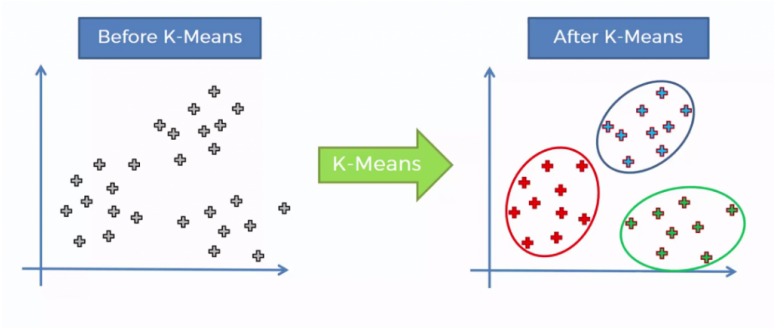


Figura 2.6: Ejemplo de clustering. Fuente: adrimelus.com.

- Modelos basados en instancias: Realmente este tipo de algoritmo no construye como tal un modelo, sino que cada vez que una muestra nueva debe ser predicha, este algoritmo se encarga de calcular la salida a partir de la salida de las instancias de entrenamiento más cercanas. Por lo tanto, este algoritmo no tiene una fase de construcción previa a partir de los datos de entrenamiento. Por ello, este tipo de aprendizaje también es denominado *lazy learning* Aha [2013]. (Fig. 2.7)
  El modelo basado en instancias por excelencia es el K-Nearest-Neighbours o KNN (Guo et al. [2003])

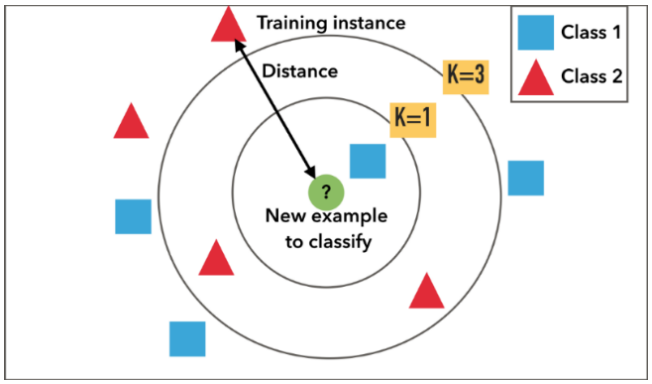


Figura 2.7: Ejemplo de modelo basado en instancias. Fuente: sxcco.com.

- Máquina de soporte de vectores: El SVM o Máquina de soporte de vectores (Noble [2006]) construye un hiperplano (o conjunto de hiperplanos) en un espacio de dimensionalidad muy alta (o incluso infinita) que puede ser utilizado en problemas de clasificación o regresión. En clasifiación, cada hiperplano separa una clase de otra, por lo que cuanto mejor se separen las clases, es decir, cuanto mejor se defina el hiperplano, más alta será la probabilidad de realizar una correcta clasificación. (Fig. 2.8)

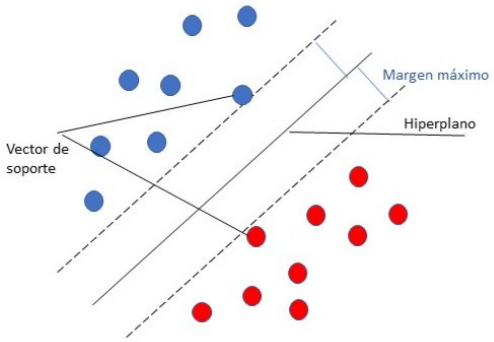


Figura 2.8: Ejemplo SVM. Fuente: ml2projects.com.com.

## 2.4. Regresión

La regresión o predicción numérica, como ya se ha aclarado en el anterior epígrafe, es una técnica utilizada para la tarea predictiva de variables numéricas. Se le dedica una sección debido a que esta técnica ha sido estudiada con mayor detalle, ya que la implementación de la librería está enfocada a la resolución de este tipo de problemas.

### 2.4.1. Qué es un problema de regresión.

Un problema de regresión tiene como objetivo predecir el valor numérico de una variables a partir de valores de otras variables, es decir, aprender o aproximar la función que establece la relación entre los valores de los atributos de una muestra y el valor de salida. Las variable a predecir se denomina variable dependiente, mientras que las demás serán independientes. La variables independientes en estos problemas son todas, o casi todas, numéricas.
Algunos ejemplos de problemas de regresión :

- ¿Cuánto combustible gasta en media un coche en carretera, sabiendo su peso, cilindrada, potencia, etc ?
- ¿Cuánto paga una persona de seguro médico al año según características como su peso, edad, hábitos de tabaquismo e Índice Basal Medio?
- ¿Qué capacidad debe tener un depósito de agua para una casa sabiendo el consumo medio de la vivienda, la temperatura, cuántas personas viven, etcétera?

### 2.4.2. Evaluación de modelos de regresión.

A diferencia de la clasificación, en estos problemas la evaluación de los algoritmos de *Machine Learning* no se basan en la cantidad de aciertos o errores que este comete, ya que medir cuánto de preciso o bueno está siendo un regresor, suele verse algo más como cuánto se está acercando, en la recta real, la predicción que hace con respecto a la salida real de cada muestra.

En clasificación se tiene un conjunto finito con $X$ posibles valores de salida, mientras que para regresión, esa salida puede ser cualquier valor de la recta real. Por ejemplo, lo que para clasificación binaria podría ser detectar si algo es blanco o negro, teniendo dos posibles opciones de salida, para regresión, los predictores no eligirían entre blanco o negro, sino que habría una escala de grises en los que se moverían las salidas de los predicciones.

Por ello, se miden de manera distinta. Para regresión, se mide normalmente el error cometido al aproximar el conjunto de valores $y = y_1, y_2, ...y_n$ por su estimación $\hat{y} = \hat{y}_1, \hat{y}_2, ...\hat{y}_n$. Los errores más comúnmente utilizados son los que se exponen a continuación.

1. **ECM o MSE** (Error Cuadrático Medio - *Mean Squared Error*) : Ecuación 4.4
   Este tipo de error es el utilizado por excelencia. Este error mide el promedio del cuadrado del error que comete el estimador. El valor obtenido será la cantidad de error que introduce el estimador al predecir las muestras. Por lo tanto, a mayor valor, peor es el estimador.

   $$ECM(y_i, \hat{y}_i) = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n} \tag{2.1}$$

   donde $y_i$ corresponde a la salida real del ejemplo $i$, $\hat{y}_i$ se refiere al valor estimado de $i$.

2. **EMA o MAE** (Error Medio Absoluto - *Mean Absolute Error*) : Ecuación 2.2
   Este tipo de error se caracteriza por su fácil interpretación. Básicamente, se define como la diferencia absoluta media entre los valores predichos y los valores reales de un fenómeno. Por ello, este tipo de error es muy utilizado. Al igual que el anterior, a mayor valor obtenido por un estimador, peor será.

   $$MAE(y_i, \hat{y}_i) = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n} \tag{2.2}$$

   donde $y_i$ corresponde a la salida real del ejemplo $i$, $\hat{y}_i$ se refiere al valor estimado de $i$.

3. **EPMA o MAPE** (Error porcentual medio absoluto - *Mean Absolute Percentage Error*): Este error mide el porcentaje de error medio absoluto que comete un estimador sobre un conjunto de salidas. Su cálculo corresponde a la ecuación 2.3

$$MAPE(y_i, \hat{y}_i) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \frac{|y_i - \hat{y}_i|}{\max(\epsilon, |y_i|)} \tag{2.3}$$

donde $y_i$ es la salida real del ejemplo $i$, $\hat{y}_i$ se refiere al valor estimado de $i$ y $\epsilon$ es un número arbitrario, pequeño pero estrictamente positivo, para evitar resultados indefinidos cuando $y_i$ es cero.

4. **RECM o RMSE** (Raíz del error cuadrático medio - *Root Mean Square Error*): Este mide la raíz del error cuandrático medio (Véase Sección 2.4.2). Se calcula siguiendo la fórmula 2.4

$$RECM(y_i, \hat{y}_i) = \sqrt{ECM(y_i, \hat{y}_i)} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{2.4}$$

donde $y_i$ corresponde a la salida real del ejemplo $i$, $\hat{y}_i$ se refiere al valor estimado de $i$.

### 2.4.3. Algunos algoritmos de *Machine Learning* para regresión.

Se presentan a continuación algunas de las técnicas de *Machine Learning* o aprendizaje automático utilizadas en problemas de regresión. Algunos son comunes a las de clasificación, como el KNN o el Perceptrón Multicapa, pero haciendo algunas modificaciones. Sin embargo, existen otros que no son aplicables a ambos tipos de técnicas de Minería de Datos predictiva.

Algunos de los algoritmos utilizados para regresión clasificados por tipos son :

- **De Minería de Datos: (Riquelme Santos et al. [2006])**
  - Modelos basados en instancias. Algoritmos como vecinos más cercanos, que calculan la salida de la muestra de prueba realizando la media de las salidas de los $K$ vecinos más cercanos.

- Modelos basados en redes neuronales, cuya capa de salida debe ser de una sola neurona.
- Árboles de regresión, donde cada hoja es un valor numérico concreto.
- Árboles de modelos, donde los nodos hoja de los árboles no continen un valor numérico, sino una ecuación de regresión local para una partición del espacio concreta.
- Máquina de Soporte de Vectores para regresión.
- Algoritmos basados en ensembles, como el Bagging o RandomForest enfocados a regresión.

- **Matemáticos: Peláez [2016]**

  - Regresión lineal, la forma más sencilla y utilizada para este tipo de problemas. Reducida a una sola variable independiente, $x$, la forma sería : Eq. 2.5

$$y = a + b \cdot x \tag{2.5}$$

    Estos coeficientes pueden obtenerse facilmente mediante el método de los mínimos cuadrados.

  - Regresión exponencial: Parecida al anterior, pero resuelve $a$ y $b$ tomando logaritmos.

# Capítulo 3

# Ensembles

Los ensembles (Rokach [2010]) son técnicas muy utilizadas en la actualidad gracias a la mejora que suponen en precisión de los modelos de Machine Learning. Es una práctica que está en desarrollo y cada vez son más los estudios que se centran en las distintas variantes de ensembles que se pueden crear.

## 3.1. Definición

Los ensembles son agrupaciones de modelos de *Machine Learning*. Así pues, un ensemble estará compuesto de varios modelos base del tipo árboles decisión, máquina de soporte de vectores, redes neuronales, etcétera. Es decir, para la decisión final, no solo se tendrá en cuenta la decisión de un solo modelo, sino que esta será la combinación de salidas de los modelos que forman parte del ensemble (Figura 3.1).

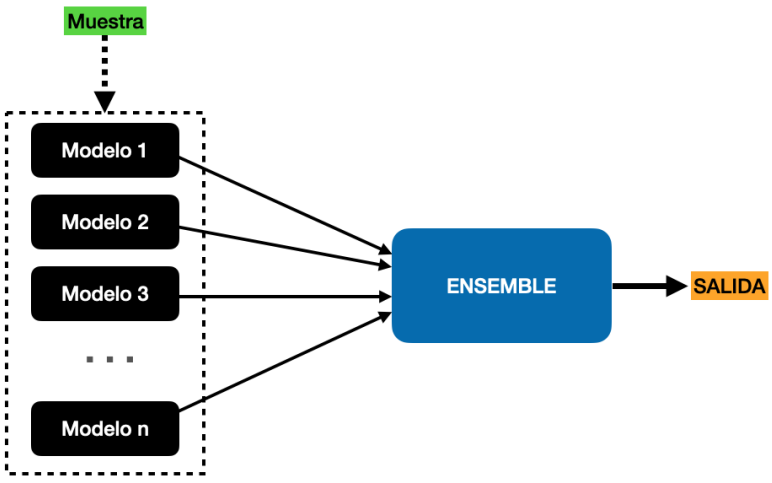


Figura 3.1: Funcionamiento Ensemble Básico. Fuente: Elaboración propia.

La razón de ser de esta estrategia radica en la mayor generalización que se crea al tener varios modelos que funcionan de manera distinta. Así los errores que cometen cada uno de los modelos que forman parte del ensemble, tienden a compensarse.(Figura 3.2). En la literatura de este área, se han demostrado mejoras utilizando esta técnica de combinación de modelos frente a un modelo individual. Por ello, es una de las técnicas más utilizadas para obtener mejores resultados en los procesos de *Machine Learning*.

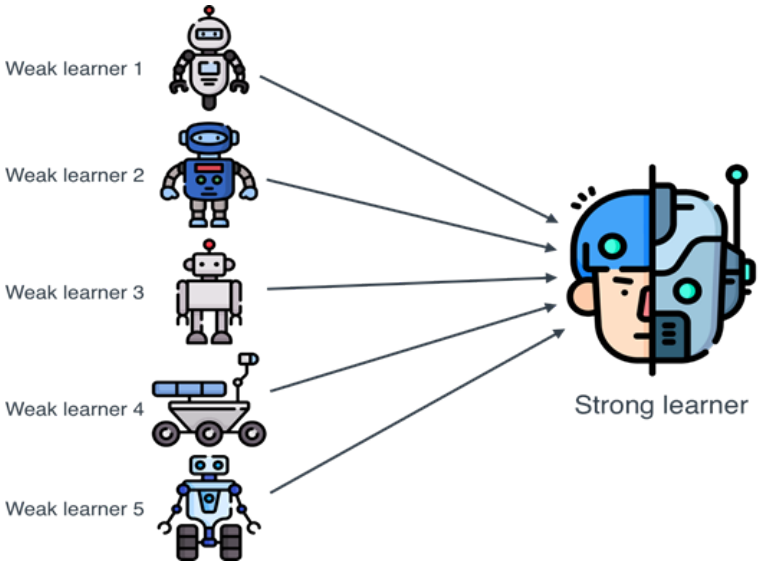


Figura 3.2: La combinación de modelos débiles dan lugar a un modelo fuerte.
Fuente: livebook.manning.com

Veámoslo con un ejemplo más aplicable a nuestro día a día: Imagine que tiene que tomar la decisión de someterse a una operación porque un médico lo ha considerado. Lo más probable, es que antes de hacerlo, pregunte a varios médicos para saber la opinión de más expertos y así, decidir qué hacer, ya que la operación puede traer consecuencia. Por lo tanto, se quedará más tranquilo si toma la decisión apoyándose en la opinión de un mayor número de expertos, pues será más acertada y con mayor probabilidad estará tomando la decisión correcta. Si todos le dicen que sí, está claro que la decisión es hacerlo; sin embargo, la duda llega cuando unos dicen que sí y otros que no. Aquí, es donde en las técnicas de ensemble de *Machine Learning* se aplican distintos enfoques para tomar una decisión. Por ejemplo, habrá veces que la opinión de uno de los expertos tendrá más peso por su experiencia y conocimiento, sobre la opinión de los demás, o habrá ocasiones en las que todos aparentemente tienen el mismo nivel de conocimiento y por lo tanto, misma capacidad de decisión.

Una vez entendido el concepto que lleva a cabo la técnica de ensemble, se exponen las distintas variantes o grupos en los que se pueden clasificar los métodos estáticos más utilizados: *Voting*, *Stacking*, *Bagging* y *Boosting* (Odegua [2019])

- **Voting:** Es el sistema más fácil de entender. Básicamente, los modelos que forman parte del ensemble tienen la misma capacidad de decisión. Cada uno realiza su predicción y a continuación, si nos encontramos en un problema de clasificación, se cuentan los votos de las clases y se devuelve como salida la clase más votada; si por el contrario se trata de un problema de regresión, la salida será la media de las predicciones de los regresores base.

- **Stacking:** Supone una evolución del anterior. Este tipo de ensemble se basa en la idea de *«apilar»* modelos. Con apilar modelos, nos referimos a utilizar las salidas de unos modelos como entradas de los siguientes. Así, por ejemplo, un ensemble de cuatro predictores dará lugar a cuatro salidas distintas (o iguales) proviniente cada una de un predictor. Estas cuatro salidas serán las entradas de otro modelo (o meta-modelo), al contrario del caso anterior que se hace un conteo o la media, según el tipo de problema, y con esto ya se obtendría la predicción final. A partir de las salidas de los modelos del ensemble, el meta-modelo será el encargado de obtener la salida final.

- **Bagging:** Este enfoque, a diferencia de los dos anteriores, se centra más en la manera de generar los modelos base. A partir de un mismo modelo base, se genera el ensemble mediante entrenamientos distintos. Para ello, se hace una subdivisión aleatoria del conjunto de entrenamiento. Cada subconjunto generado de la división del conjunto de entrenamiento se utilizan para entrenar un modelo. Así pues, cada modelo estará entrenado con distintos subconjuntos de datos.

- **Boosting:** En boosting, cada modelo pretende mejorar los errores cometidos por modelos anteriores. Por ejemplo, en un problema de clasifiación, el primer modelo tratará de encontrar las relaciones entre los valores de los atributos y la clase. Tras este, el siguiente modelo intentará sopesar los fallos que haya tenido el primero. Para ello, se la da un mayor peso a las muestras que han sido clasificadas incorrectamente. y así, sucesivamente con los siguientes modelos, hasta que se alcanza un umbral de tolerancia de error.

## 3.2. Ensembles dinámicos

En primer lugar, se explican las distintas fases generales por las que debe pasar el proceso y, a continuación, se expone el proceso de selección dinámica, donde se detallan cada una de las etapas y las distintas vías por las que se puede optar en cada una de ellas, ya que se pueden crear distintas combinaciones entre etapas. Se verá más claro durante la subsección dedicada a ello a continuación.

Para poder redactar de la forma más correcta dicho proceso, ha sido necesario hacer una previa recopilación y estudio de los cada uno de los avances realizados en el ámbito gracias a una amplia búsqueda bibliográfica realizada.

### 3.2.1. Introducción a la selección dinámica en ensembles

El principal concepto introductorio al enfoque de la Selección Dinámica en ensembles es saber que los modelos predictores base son escogidos en el momento, es decir, dependiendo de la muestra de entrada. A esto también se le llama proceso *on-line*. Por lo tanto, como se ha podido intuir, para la selección dinámica lo primero que se necesita es un conjunto base de predictores.

La razón de ser de esta técnica novedosa es que no todos los modelos son expertos en la predicción de todas las muestras, sino que cada uno lo es en una región de competencia diferente dentro del espacio de características. Por lo tanto, existirán modelos que serán muy buenos para predecir una muestra, pero que para otra muestra diferente, resulte no serlo.

Por consiguiente, la clave de este enfoque es realizar la selección de predictores de la manera más correcta, es decir, escogiendo aquellos que realmente sean los mejores para la muestra de entrada. Normalmente, los pasos comunes que se llevan a cabo en la Selección Dinámica son los siguientes:

-**Definición de la región de competencia de la muestra dentro del espacio de búsqueda**. Es decir, la región de competencia de una muestra estará formada por muestras parecidas en cuanto a los valores de los atributos. Existen diferentes enfoques capaces de definirla, como por ejemplo, el KNN o el Clustering.

-**Estimación del nivel de competencia para cada modelo base**. Es el momento en el

que todos los predictores son evaluados en la región de competencia definida en el paso anterior. Por lo tanto, se obtiene cómo de bueno es cada predictor para las muestras de la región de competencia, es decir, las más parecidas a la muestra que se está evaluando. Con esta información, se obtiene también en este paso el umbral que se utiliza en la selección (Paso siguiente).

-**Selección de los mejores predictores**: todos aquellos cuyo nivel de competencia en dicha región local hayan superado el umbral preestablecido en el paso anterior.

### 3.2.2. Clasificación de las técnicas de Selección Dinámica.

Según la taxonomía propuesta en Cruz et al. [2018], las técnicas de selección dinámica se puede clasificar de diferentes maneras, dependiendo del aspecto a tener en cuenta para ello :

-Según enfoque de selección : Existen dos vertientes. Por un lado, se encuentran las técnicas que solo seleccionan un modelo base, el mejor, para la predicción. Por otro lado, existen técnicas que seleccionan un subconjunto de modelos base, los $n$ mejores del conjunto base, para realizar la predicción de la muestra.

-Según método para definir la región de competencia, que es donde se estimará el nivel de competencia de los modelos base : Se pueden utilizar distintos métodos como el KNN, Clustering, Perfiles de salida, etc (se ve más adelante con mayor detalle).

-Según los criterios de selección utilizados para estimar el nivel de competencia de los modelos base: podemos utilizar métodos tales como el Ranking, la Precisión, el error cometido, etc.

### 3.2.3. Etapas

Como se expone en Kumar and Kumar [2012], las fases de los ensembles dinámicos, realmente, son heredadas de los llamados Sistemas de multiclasificadores/sistemas de multiregresores. Por lo tanto las etapas son las mismas. Se dividen fundamentalmente en las siguientes :

- **Etapa de generación:** En esta etapa se determina el conjunto de modelos base, donde debe existir diversidad en el conjunto, ya que a mayor diversidad, más espacio de búsqueda quedará cubierto por un buen predictor. Además, se busca que no solo sean distintos, si no también precisos en al menos alguna región del espacio de características. Existen dos maneras de clasificar esta etapa. Se denomina generación homogénea o ensemble homogéneo a aquel cuyos modelos base han sido creados bajo el mismo algoritmo; por el contrario, será heterogéneo. Además, se enumeran en la publicación Duin [2002] las diversas maneras con las que poder generar un conjunto de modelos base de manera efectiva:

  - **Distintas inicializaciones**, por ejemplo en redes neuronales, la asignación de pesos.
  - **Diferentes parámetros**, como por ejemplo, para el algoritmo KNN, diferentes K generan diferentes modelos.
  - **Diferentes arquitecturas**, por ejemplo para el modelo MLP (MultiLayer Perceptron), o perceptrón multicapa, se pueden generar modelos distintos combinando distintos número de capas y número de nodos en cada capa.
  - **Distintos modelos**, es decir, un ensemble o conjunto de predictores heterogéneo. Se utiliza un algoritmo de Machine Learning distintos para cada modelo a construir.
  - **Diferentes conjuntos de entrenamiento**, es decir, distinto subconjuto de entrenamiento para cada modelo.
  - **Diferentes conjuntos de características.**

- **Etapa de selección:** Se selecciona el mejor o los mejores predictores, según el enfoque de selección que siga. Esta etapa no aparece en todos los algoritmos de multiclasificadores o multiregresores, pero sí en todos los referidos a la selección dinámica, por eso se añade en esta taxonomía. Esta estapa puede ser estática o dinámica. En la estática, el subconjunto de los mejores modelos o el mejor, se realiza durante la etapa de entrenamiento y es igual para cualquier tipo de muestra. Sin embargo, la dinámica cada vez que entra una nueva muestra, se debe realizar la selección de los mejores (o el mejor) predictor en base a dicha muestra. Por ello, se denomina dinámico. La imagen 3.3 muestra la diferencia entre el enfoque dinámico y el estático. El estático, mediante la evaluación de los modelos en el conjunto de evaluación, escoge los $M$ mejores y son los que se utilizan para todas las muestras a predecir, mientras que en el dinámico, cada muestra nueva, genera un nuevo ensemble, debido a que se evaluan en la región de competencia que esta nueva muestra determina.

- **Etapa de agregación:** Fase en la que los resultados obtenidos de cada predictor escogido son agregados. La agregación se genera mediante una función de agregación,

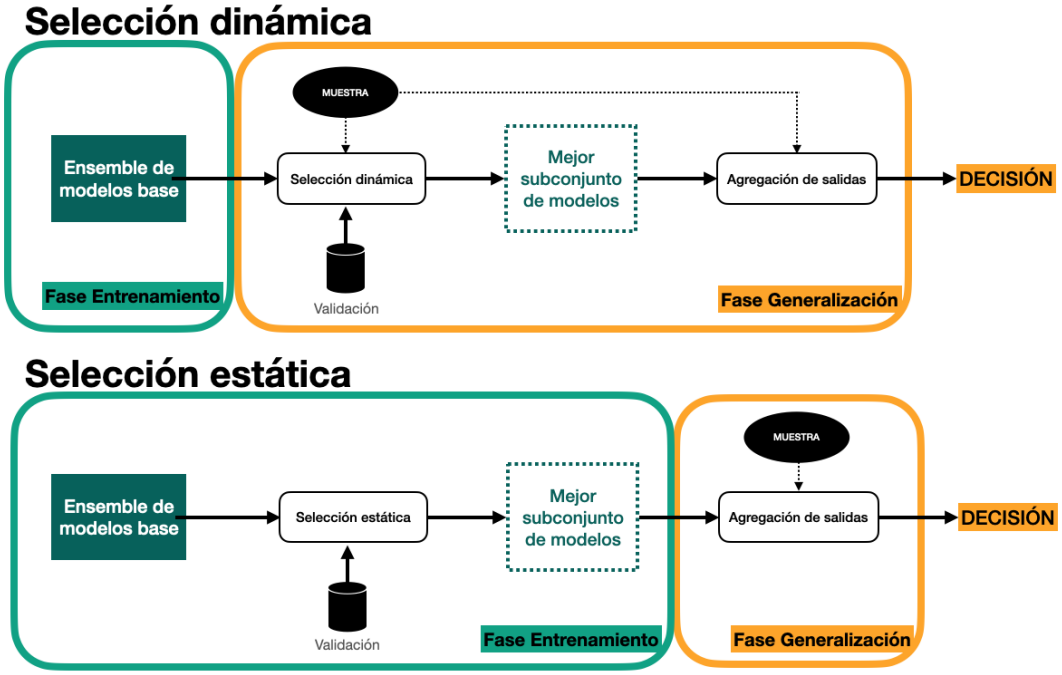


Figura 3.3: Comparativa selección dinámica frente a estática.Fuente: Elaboración propia.

cuya salida será la predicción final. La función de agregación puede ser, por ejemplo, votación por mayoría para clasificación, media ponderada para regresión o incluso un meta modelo que decida, a partir de las predicciones del ensemble, la predicción final.

La etapa más importante en este caso, será la de selección, que como se ha explicado anteriormente (epígrafe 3.2.1) a su vez esta compuesta de subetapas. A continuación se explica con mayor detalle el proceso de selección dinámica.

### 3.2.4. Proceso de selección dinámica

En la selección dinámica en concreto, el proceso se compone de tres pasos y, por lo tanto, de tres decisiones a tomar para la clasificación de una muestra. Primero, elegir con qué método se va a obtener la región de competencia. Segundo, qué criterio de selección se seguirá (Individual o grupal). Por último, si se va a escoger solo al mejor clasificador o un conjunto de los mejores.

En la figura 3.4 se detallan las tres fases de manera más visual y entendible. En primer lugar, se calcula la región de competencia de la nueva muestra. Después, todos los modelos base del ensemble son evaluados en dicha región. Por último, se evaluan los resultados obtenidos por los modelos en la región de competencia y se eligen los mejores.

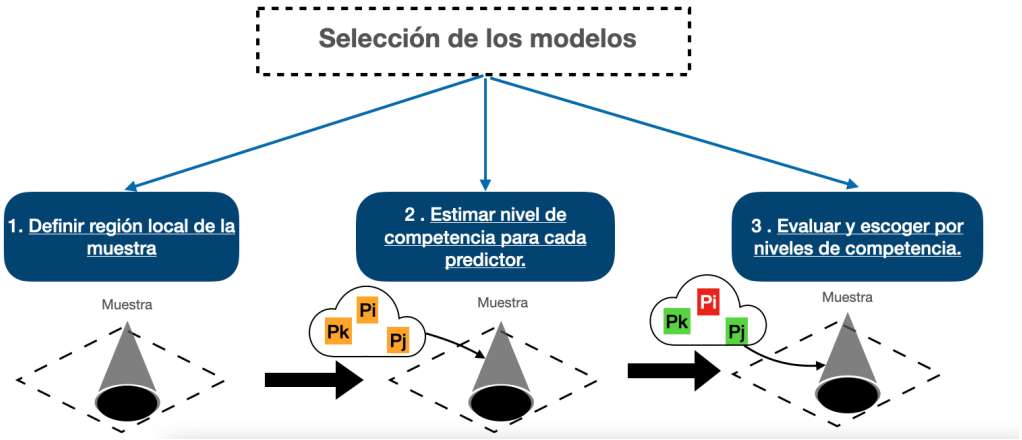


Figura 3.4: Etapas en la Selección dinámica.Fuente: Elaboración propia.

Veamos con más detalle las opciones disponibles para elegir en cada paso :

**Definición de la región de competencia.**

El objetivo de definir una región de competencia es encontrar las muestras (de las cuales tenemos sus salidas reales disponibles) más parecidas a la muestra de prueba. Así, sobre estas muestras parecidas, al tener sus salidas reales, podemos medir el rendimiento o nivel de competencia de todos los regresores base. Entonces, se induce que los regresores que mejor funcionen para estas muestras, serán también los que mejor predigan la muestra de prueba.

Concretar cómo se va a definir la región de competencia es una decisión muy importante, pues el rendimiento de todas las técnicas de Selección Dinámica de ensembles son muy sensibles a esta distribución de la región. Es más, varios estudios como Duin [2002] indican que solo con la mejora de este paso, el rendimiento de las técnicas de Selección Dinámica incrementan significativamente. En dicho artículo proponen dos mejoras en el paso de la definición de la región de competencia y mediante experimentación, demuestran que el rendimiento de los ensembles dinámicos está directamente relacionado con el cálculo de la región de competencia.

Lo más importante para esta paso es tener un conjunto de muestras etiquetadas. Este conjunto se denomina DSEL (Dynamic Selection Dataset). El DSEL puede ser el propio conjunto de entrenamiento o un conjunto de validación. También pueden aparecer solapados el DSEL y el conjunto de entrenamiento, sobre todo para casos en los que no existan

muchos datos etiquetados. En este conjunto se buscan las instancias más parecidas a la muestra nueva, es decir, la región de competencia.

Siguiendo con la taxonomía propuesta en Cruz et al. [2018], algunos de los métodos que se pueden utilizar para determinar la región de competencia son los siguientes:

**- Clustering:**
El proceso de definición de la región local mediante Clustering es el siguiente:

1. Definición de los clusters en DSEL.
2. Estimar la competencia de cada clasificador para todos los clusters.

De esta manera, cada cluster tendrá, después de la fase de entrenamiento, el subconjunto de modelos del ensemble que mejor se comporten en las muestras que lo componen.

Como podemos ver, una de las ventajas de este método es que la estimación de los niveles de competencia de los predictores se realiza durante la etapa de entrenamiento. Esto que hace que el tiempo de predicción de una nueva muestra con este modelo sea menor. Por lo tanto, solo se mediría de manera *on-line* la distancia a los centroides de los cluster, eligiendo el más cercano.

**- KNN:**
El proceso de definición de la región de competencia mediante el K-NN se basa, precisamente, en que la región local de la muestra nueva será la definida por ella y sus k vecinos más cercanos.

Este método tiene como ventaja que su estimación es más precisa que el Clustering, ya que las regiones locales son dinámicas y no predispuestas por clusters, y, por lo tanto, la selección de predictores para cada muestra será distinta. Sin embargo, este presenta un mayor coste computacional, pues para estimar los niveles de competencia de cada clasificador, se tiene que medir la distancia de todo el DSEL con respecto a la muestra de consulta.

**- Modelo de la función potencial:**
Para implementar este modelo, hace falta tener en cuenta todo el DSEL, ya que lo usa completo para estimar los niveles de competencia para cada estimador. Este pondera las salidas de los estimador en base a la distancia euclídea de cada dato del DSEL con res-

pecto la muestra de consulta utilizando una función potencial, como puede ser una función potencial Gaussiana.

La gran desventaja que presenta es el coste computacional, ya que usa todo el DSEL para la estimación de los niveles de competencia.

No obstante, tiene una gran ventaja con respecto al método visto anteriormente, ya que elimina la necesidad de definir el tamaño de la vecindad a priori. Esto se debe a que su función potencial se usa para reducir o aumentar la influencia de cada punto del DSEL en función de su distancia euclídea.

**- Perfiles de salida:**
Se basa en el comportamiento del conjunto de predictores sobre las muestras. Se utilizan las predicciones de los modelos como información. Un aspecto muy importante de este método es que la muestra nueva y el conjunto DSEL son transformados a perfiles de salida. Los perfiles de salida se crean de la siguiente manera:

$$\tilde{x}_j = \{\tilde{x}_{j,1}, \tilde{x}_{j,2}, ..., \tilde{x}_{j,m}\}$$

, donde $x_{j,i}$, es la decisión tomada por el estimador $e_i$ para la instancia $x_j$.

Es decir, el perfil de salida de una muestra es el vector compuesto por la predicción de cada modelo para dicha muestra. Así, cada posición del vector indica el predictor y el contenido del vector en dicha posición, la predicción de ese modelo para la muestra.

La región de competencia es estimada mediante la similitud entre el perfil de salida de $x_j$ y los perfiles de salida de todas las muestras del DSEL. El conjunto de perfiles de salida más similares ($\varphi$) se usa para estimar el nivel de competencia de los clasificadores base.

**Estimación del nivel de competencia de cada predictor.**

Existen dos enfoques para medir y estimar los niveles de competencia de los estimadores base: Individual y grupal.

Veamos sus principales diferencias y en qué se basan cada uno:

**- Individual:** Define el rendimiento o nivel de competencia de cada predictor de manera independiente al resto. Es decir, el rendimiento del modelo $P_i$ se calcula de manera individual, sin tener en cuenta los demás modelos base. Algunos tipos que siguen este criterio de selección son el Ranking, la Precisión, etc.

**- Grupal:** Define el rendimiento teniendo en cuenta la interacción entre los predictores del ensemble. Algunas medidas que siguen este enfoque son la Diversidad o la Ambigüedad. Estas medidas están relacionadas con la noción de relevancia, es decir, si un modelo funciona bien en conjunto con otros modelos base.

**Enfoque de evaluación y selección de estimadores por niveles de competencia.**

En este paso lo que se elige es si solo se va a utilizar el mejor estimador para predecir la salida o se va a considerar un conjunto de ellos.

**DCS** / **DRS:** Dynamic Classifier Selection / Dynamic Regressor Selection, es decir, solo se tendrá en cuenta para la decisión un clasificador / regresor. El escogido será el que mayor nivel de compencia haya obtenido.

**DES:** Dynamic Ensemble Selection, es decir, se tendrá en cuenta un ensemble o conjunto de estimadores. Los escogidos serán los que hayan obtenido un determinado nivel de competencia. Este vendrá determinado por un umbral. Por consiguiente, sus resultados han de ser agregados para la predicción de la muestra de $x_j$.

Para concluir el capítulo, cabe destacar que el concepto de la Selección Dinámica puede ser aplicado tanto a problemas de clasifiación como a problemas de regresión, siendo estos últimos los menos estudiados por el momento. Por lo tanto, en la búsqueda bibliográfica, se encuentran sobre todo artículos de Selección Dinámica de Clasificadores. Sin embargo, en este trabajo nos vamos a centrar, como ya se adelantaba, en problemas de regresión.

# Capítulo 4

# Análisis, diseño y desarrollo de la libreria

Para poder desarrollar la librería de la manera más correcta, el primer paso es realizar un análisis del campo de estudio y descubrir cuáles son los requisitos que el usuario espera de la librería que se va a proponer. Con este fin, se he realizado una extensa revisión bibliográfica para así poner en antecedente la temática, ver cuáles han sido los últimos avances y, fundamentalmente, para sostener cada una de las decisiones tomadas en el diseño y desarrollo de la librería.

Cabe destacar la poca literatura que existe sobre selección dinámica de regresores. Casi todos los artículos y publicaciones se centran en el campo de la clasificación. Esta carencia, junto con la evidencia de ser una técnica que ha demostrado empírica y teóricamente mejorar a la versión estática, han sido las principales motivaciones para realizar este trabajo. Seguidamente, se explica el diseño propuesto para dar forma al desarrollo de la librería y, por último, el desarrollo de esta.

## 4.1. Análisis

Este paso es crucial para una buena toma de decisiones en la implementación de la librería.

Para guiar el diseño y desarrollo de la librería, el primer paso es identificar los requisitos software que requiere. Es decir, qué es lo que el usuario espera de la librería, qué es lo que le va a ofrecer.

Una vez enumerados estos, será el momento de indagar e investigar cómo cumplir con ellos. Esto se estudia en el análisis de requisitos y el diseño.

### 4.1.1. Requisitos

Los requisitos software detectados para la librería a desarrollar son los siguientes:

- Implementar una librería capaz de llevar a cabo el proceso completo de la técnica de ensembles dinámicos, incluyendo etapa de generación, selección dinámica y agregación.
- Proporcionar una librería capaz de soportar cualquier combinación de regresores base elegida por el usuario.
- Poner a disposición una librería que implemente una alternativa eficaz para la creación del ensemble si el usuario no introduce ningún regresor base.
- Crear una librería que implemente distintos métodos de selección dinámica de ensembles.
- Proporcionar al usuario la libertad de poder elegir cómo se va a llevar a cabo cada etapa del proceso de Selección Dinámica.
- Generar distintos métodos para calcular la región de competencia y poder elegir entre ellas.
- Proporcionar al usuario elegir la dimensionalidad de la región de competencia.
- Suministrar distintos métodos para calcular los niveles de competencia de los regresores base.
- Crear distintos métodos para seleccionar los mejores regresores base.

- Proporcionar distintas funciones de agregación y que el usuario pueda elegir con cuál se agregarán las salidas de los regresores base.
- Implementar la librería de la manera más robusta posible para soportar cualquier combinación de las posibles elecciones de los usuarios para cada etapa del proceso (cálculo de la región de competencia, cálculo de los niveles de competencia de los regresores y selección de los mejores regresores).
- Dar al usuario la opción de definir el porcentaje del conjunto de entrenamiento que se destina al conjunto DSEL.
- Dar la posibilidad de poder solapar conjunto de entrenamiento y conjunto DSEL.
- Proporcionar una manera fácil y entendible para el usuario de utilizar la librería.
- Parametrizar todas aquellas decisiones que corran por parte del usuario.

Obtenido el conocimiento de los requisitos, a continuación se lleva a cabo un análisis de los mismos.

### 4.1.2. Análisis de requisitos

Para llevar un orden claro, el análisis se hará de manera secuencial de acuerdo a las etapas de la selección dinámica de ensembles. El análisis se basa en la recopilación de información referente al punto que se está tratando y la extracción de ideas y fundamentos en los que se basará el diseño y desarrollo posteriores de la librería.

#### Generación de regresores base

En la generación de regresores base, queda claro que lo ideal es tener modelos precisos y diversos, como se justifica en Sagi and Rokach [2018]. Esto significa que, no solo sean buenos o consigan buenos resultados, si no que entre ellos deben ser lo más diferentes posibles, aunque parezca características contrapuestas.

Como ya se ha visto en el capítulo anterior, existen diversas maneras de generar modelos diferentes. La manera más utilizada durante la literatura de la Selección dinámica es el Bagging, ya que consigue buenos resultados. Con el bagging, conseguimos distintos modelos base, pero todos parten del mismo algoritmo de Machine Learning. Por lo tanto,

como también se busca que el conjunto base sea los más diverso, es decir, que los regresores sean lo más diferentes posibles para abarcar más regiones de competencia de manera precisa (porque recordemos, cuanto más diversos, más probabilidad de que cada uno sea experto en un subconjunto distinto de espacio de características), sería conveniente realizar el conjunto base a partir de diferentes algoritmos de aprendizaje.

Como conclusión, se toman dos ideas como fundamento. La primera, el usuario podrá elegir los modelos base de los que partirá el conjunto base final utilizado para la Selección Dinámica. A partir de ellos, se realizará un proceso de Bagging a cada uno de ellos. Por lo tanto, se diversificará doblemente el conjunto base; primero porque los algoritmos de aprendizaje serán distintos, y segundo, porque cada uno de ellos dará lugar a $M$ modelos basados en el mismo algoritmo de aprendizaje pero entrenados con diferentes subconjuntos aleatorios del conjunto de entrenamiento. Tal que:

$$algoritmo_i = modelo_{i1}, modelo_{i2}, ..., modelo_{i,j}, ..., modelo_{i,m}$$

donde $i$ indica el número de algoritmo y $j$ el modelo generado en el proceso de Bagging a partir del algoritmo $i$. Así pues, cada proceso de Bagging da lugar a $M$ modelos, que se diferencian entre sí por el conjuntos de entrenamiento.

Este tipo de ensemble se denomina heterogéneo (Distintos algoritmos de aprendizaje).

La segunda idea es, si el usuario no introduce ningún tipo de ensemble, la biblioteca internamente será capaz de generar un conjunto de modelos base. Haciendo referencia a los buenos resultados obtenidos en los artículos Zyblewski et al. [2021], Mousavi et al. [2018] en los que realizan ensembles homogéneos a partir de Bagging, este mismo procedimiento es el que se seguirá en este caso. Por consiguiente, si el usuario no introduce ningún ensemble, se generará automáticamente, mediante Baggin, un ensemble homogéneo (mismo algoritmo de aprendizaje). En la Figura 4.1 se indica el procedimiento de generación del ensemble. En primer lugar se ve cómo un usuario propone como ensemble cuatro algoritmos distintos y, cuando la librería los recibe, cada uno de ellos se somete a un proceso de bagging. En la misma figura, a continuación, se muestra la siguiente casuística en la que el usuario no introduce ningún algoritmo.

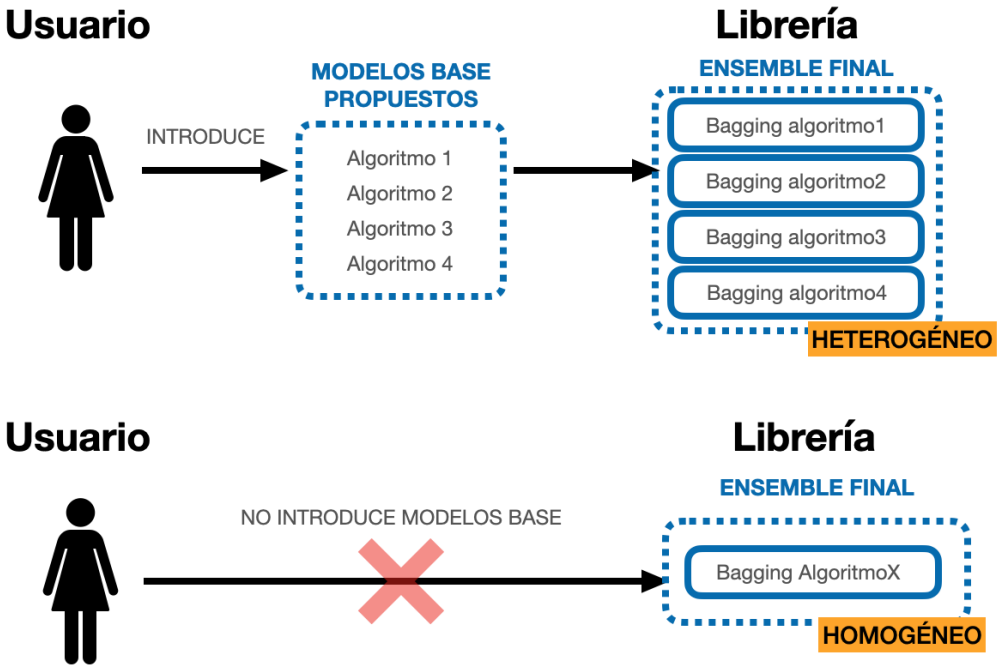


Figura 4.1: Generación del ensemble. Fuente: Elaboración Propia.

### Selección Dinámica

- **Preprocesamiento.**

  Previamente, cabe destacar la gran cantidad de trabajos en los que se investiga cómo mejorar la Selección Dinámica para datos imbalanceados (problemas de clasificación). Muchas propuestas están apareciendo en esta vertiente, ya que cada vez son más los datos no balanceados que se tienen en los problemas reales. Algunos de los trabajos que han concluído en mejoras son Wu et al. [2021] o Hou et al. [2020], en ambos se proponen modelos híbridos basados en la combinación de la Selección Dinámica con preprocesamiento de las instancias mediante técnicas como el remuestreo de clases minoritarias para eliminar desbalanceos.

  No obstante, para regresión no tenemos este problema de desbalanceo como tal, pero sí que nos hace reflexionar sobre la importancia de la distribución de los datos de entrenamiento para este tipo de problemas. Ya que para la seleccion dinámica esto es clave. Si los regresores base escogidos son los que mejor funcionan en la región de competencia de la muestra de entrada, es evidente que cuanto más distribuídos estén los datos de entrenemiento o DSEL (si no son el mismo), más probabilidad de calcular una región de competencia real de la muestra. Si por el contrario, todos los ejemplos del DSEL se encuentran agrupados en un mismo espacio de características, algunas nuevas muestras no estarán bien representadas por ninguno, o casi ninguno, de los ejemplos en los que van a ser evaluados los regresores. Y por lo tanto, no sabremos

si realmente son competentes para la muestra nueva.

Por lo tanto, la definición del DSEL, por inducción, cuanto más grande sea, es decir, más instancias recoja, más probabilidad de encontrar un conjunto de muestras parecidas verdaderamente a la muestra de prueba. Aunque, como siempre, todo dependerá del problema. Sin embargo, aquí también entra en juego el tiempo de ejecución. A mayor DSEL, mayor será el tiempo que el algoritmo necesita para calcular las distancias a todos los puntos, porque el número de puntos aumenta.

Por consiguiente, una de las primeras consideraciones que se deben tener en cuenta para realizar una Selección Dinámica correcta es definir las subdivisiones de los datos en datos de entrenamiento y DSEL mediante algunas experimentaciones previas. En consecuencia, se toma la decisión de incorporar como parámetro (elegible por el usuario) el porcentaje de DSEL que se utilizará para la Selección Dinámica.

Una vez tratado este punto de preprocesamiento que resulta bastante interesante, sigamos por la definición de la región de competencia.

- **Región de competencia**

  La región de competencia, como ya se mencionaba antes, tiene relación directa con el rendimiento final del ensemble. Es decir, cuanto mejor se defina esta, mejores resultados tendrá nuestro sistema de ensemble dinámico.

  Sin embargo, no son muchos los trabajos que se centran en mejorar el cálculo de la región de competencia. Esto podría justificarse porque la región de competencia es dependiente del problema. En artículos como Moura et al. [2020] y Silva et al. [2021], se corrobora esta afirmación, por lo que, encontrar una manera universal y común de calcular la región de competencia para cualquier problema de clasificación o regresión, resulta bastante complicado. Así pues, lo más lógico parece ser experimentar previamente con el datasets que se esté utilizando para ver si se comporta mejor un método u otro.

  Como conclusión de esta primera parte, podemos resaltar que un solo método para calcular la región de competencia no nos dará siempre los mejores resultados. Por ello, se toma la decisión de implementar distintas maneras de escoger la región de competencia entre las cuales el usuario podrá elegir. En concreto, se utilizará KNN, clustering y perfiles de salida.

- **KNN**, además de ser un modelo basado en instancias que por su propia definición es un buen candidato para obtener la región de competencia, es el método escogido como base en diversos trabajos como Zhang et al. [2019], Kinal and Woźniak [2020], en los cuales se proponen mejoras para la selección dinámica, ambos hacen uso de métodos de Selección Dinámica basados en KNN para el cálculo de la región de competencia para Clasifiación, que propone Cruz et al. [2020] en su artículo e implementa como librería en Python.

- El **clustering** se decide incorporar también debido a que, gracias al agrupamiento del DSEL previo, para datasets grandes podría suponer una mejora en tiempo, ya que se calculan los ensemble más competentes en la parte de entrenamiento. Por ello, también se añade.

- Y por último, los **perfiles de salida** se tendrán en cuenta, ya que cambia la perspectiva de los dos anteriores. En este, no se miden distancias entre instancias, sino que se basa en en el comportamiento de los regresores con las muestras. Así, incorporando este método de definición de la región de competencia, se medirá la similitud de comportamiento de los regresores entre instancias.

Este último, no es implementado por ningún método visto en la revisión bibliográfica, solo es propuesto en Cruz et al. [2018] en la revisión de la taxonomía de la Selección Dinámica de Calsificadores.

- **Cálculo del nivel del competencia**

  El siguiente punto a analizar es el cálculo de los niveles de competencia.

  En un primer lugar, se analizan distintas maneras de medir la bondad de cada regresor para cada muestra dentro de la región de competencia y luego realizar algún cálculo de error, como en Moura et al. [2020] que expecifica que para problemas de regresión, el cálculo de los errores es lo más adecuado.

  Sin embargo, durante la revisión bibliográfica aparece el término *Dynamic weighting*, que significa ponderación dinámica. Específicamente, como se explica en Mendes-Moreira et al. [2009], Mendes-Moreira et al. [2015], se basa en la ponderación dinámica de las salidas generadas por los ensembles. Esto, a priori, debería afectar a las decisiones de diseño que se llevarán a cabo para la agregación. Pero, no obstante, se decide incluir en esta sección y se toma la decisión de incorporar algunas ideas de la ponderación dinámica al cálculo del nivel de competencia.

Como bien se explica en Moura et al. [2021], cada muestra de consulta genera una región de competenca con $K$ muestras del DSEL. Cada una de las muestras de dicha región se encuentra a una distancia de la muestra de prueba. El peso de cada una de ellas es calculado y utilizado para la ponderación de los errores que se cometen en cada punto por cada regresor. Estos pesos se calculan en base a la distancia que existe entre la muestra de prueba y cada un de las que forman parte de la región de competencia, tal que 4.1:

$$d_k = \frac{\frac{1}{dist_k}}{\sum_{j=1}^{K}(\frac{1}{dist_j})} \tag{4.1}$$

donde $dist_k$ es la distancia entre la muestra de prueba y la muestra $k$ de la región de competencia.

A continuación, en el artículo, dichas distancias son utilizadas para ponderar las salidas de los regresores a la hora de la agregación.

Para esta librería se toma esta idea, pero en lugar de realizar esta ponderación solo para la parte de agregación de salidas, se aprovecha este enfoque para el cálculo de los niveles de competencia de los regresores.

Por lo tanto, una vez se calculan las ponderaciones de los puntos del DSEL, que están basadas en la inversa de la distancia de cada punto con respecto a la muestra de test, tendríamos el vector $d = d_1, d_2, ..., d_k$. Con estas ponderaciones ahora se pueden calcular los errores medios de los regresores en la región de competencia penalizando más los errores que se cometen en muestras más cercanas (por ello se utiliza la inversa de la distancia).

Así, mediante la fórmula 4.2 , se pondera el error que se esté calculando del regresor en el punto $k$ de la región de competencia y luego se suman todos estos errores de la manera que indica la ecuación 4.3 para así obtener el nivel de competencia total del regresor.

$$e\prime_{r,k} = d_k * error_{r,k} \tag{4.2}$$

$$e\prime_r = \sum_{j=0}^{K} e\prime_{r,j} \tag{4.3}$$

, donde $error_{r,k}$ es el error que comete el regresor $r$ en la predicción de la muestra $k$ de la región de competencia , $e\prime_{r,k}$ es el error ponderado de $r$ en el punto $k$ y $e\prime_r$ es la suma ponderada de los errores del regresor $r$, que se utilizará como nivel de competencia para el regresor $r$.

Tomando en cuenta la decisión anterior, solo queda determinar las medidas de error que se van a implementar (la que se va a usar como $error_{r,k}$ en Ec.4.2). Moura et al. [2021] propone distintas medidas para evaluar a los regresores, de las cuales para la implementación nos quedaremos con:

- **ECM (Error Cuadrático Medio) :** Ecuación 4.4
  Este tipo de error es el utilizado por excelencia. Este error mide el promedio del cuadrado del error que comete el estimador. El valor obtenido será la cantidad de error que introduce el estimador al predecir las muestras. Por lo tanto, a mayor valor, peor es el estimador.

$$ECM = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n} \tag{4.4}$$

- **Suma de errores absolutos.** Calcula la suma de los errores absolutos cometidos entre las salidas reales y las predichas por un mismo regresor. Es calculada siguiendo la fórmula 4.5.

$$SEA_r = \sum_i = 1^N |x_i - x\prime_{i,r}| \tag{4.5}$$

  , donde $N$ es el número de muestras a predecir, $x_i$ el valor real de salida de la muestra y $x\prime_{i,r}$, el valor estimador para la muestra $i$ por el regresor $r$.

- **Varianza.** No es un error, pero se incluye para tener otra medida que se diferencie más de las demás y estudiar su comportamiento. Además, para esta medida, no se utilizará la ponderación de las muestras del DSEL. Por lo tanto, con esta se estima el nivel de competencia de cada regresor solo teniendo en cuenta la varianza de sus salidas en los distintos puntos del DSEL, atendiendo a la ecuación:

$$m_r = Var(t\prime_1, t\prime_2, ..., t\prime_k) \tag{4.6}$$

, donde $m_r$ será el nivel de competencia para el regresor $r$ y $t\prime_k$, el valor estimado por el regresor $r$ para la muestra $k$ perteneciente a la región de competencia de la muestra de prueba.

Con estas, se tienen tres medidas bastante diferentes para evaluar a los regresores. Para acabar esta sección, cabe destacar que, debido a las medidas en los que están basados los niveles de competencia a calcular, lo ideal es que en la selección de los regresores se escojan aquellos que tengan menores niveles de competencia. Ya que tanto para el error como para la varianza, un valor bajo indica que el regresor funciona bien en la región en la que se está estudiando.

- **Selección de los mejores regresores.**

  Por último, para la selección de los mejores regresores, lo habitual es establecer algún tipo de umbral utilizando la información de los niveles de competencia obtenidos en el paso anterior.

  Como métodos más habituales para elegir un umbral dentro de varios valores, encontramos las media aritmética. Es quizás la medida más utilizada y fácil de implementar. Esta se añadirá a la librería por su fácil comprensión y para considerarla como selección base.

  Además de la media aritmética, una de las herramientas más utilizadas en estadística para calcular un umbral a partir de varios datos, es la desviación típica. Por definición, la desviación típica estudia la dispersión media de una variable (Pita Fernández and Pértega Díaz [1997]). Por lo tanto, otro umbral podría definirse como la media más la desviación típica. Se decide sumarla para aumentar el rango de aceptación de regresores en la selección y aumentar así el número de regresores que tomen parte de la decisión final con respecto al umbral definido solo con la media. Ya que utilizar un número mayor de modelos para la predicción final suele dar mejores resultados tanto para ensembles estáticos como para dinámicos, pues como su propia definición indica, cuantas más *opiniones* se tengan, más acertada será la predicción. Aunque esa premisa a veces no se cumple y por ello aquí se implementan varios umbrales, para poder tener varias opciones ya que dependerá de los datos y los regresores base que funcione un umbral mejor que otro.

  Por último, también se implementa otro umbral que vendrá definido por la fórmula del punto medio. Dados los niveles de competencia y siendo $n_{max}$ el máximo nivel de competencia obtenido y $n_{min}$ el mínimo, el punto medio se calcula tal que :

$$umbral\_puntoMedio = \frac{n_{max} + n_{min}}{2} \tag{4.7}$$

Este umbral no es tan sensible al comportamiento de los datos. Solo tiene en cuenta el rango en el que se definen los niveles de comeptencia y calcula su punto medio, que es distinto a la media.

### Agregación

La última etapa, la etapa de agregación, como se indicaba ya en el análisis del cálculo de los niveles de competencia, puede estar compuesta por una integración de salidas ponderada, que es lo que realmente se propone en Moura et al. [2021], pero que para esta librería se ha tomado como inspiración para el cálculo de los niveles de competencia. Por lo tanto, como los niveles de competencia ya están ponderados y realmente nos dicen cómo de *bien* funcionan en la región de competencia cada estimador, tiene sentido utilizar estos niveles de competencia para ponderar las salidas. Así, cuanto mejor funcione un regresor en la región de competencia, más peso tendrá su predicción con respecto a los demás.

Para la ponderación, es necesaria la normalización de los niveles de competencia para obtener el peso de cada regresor en una misma escala. Los niveles de competencia se normalizan a escala [0-1] para tener pesos más manejables (Ec.4.8). Hay que tener en cuenta que en la normalización se resta a uno ( *1* - ) porque los niveles de competencia más pequeños son los que más deben pesar en la decisión (ya que son los que menos error han obtenido) y los más grandes, los que pesan menos para la toma de decisión final.
Obtenidos los pesos normalizados, es momento de agregar las salidas de los regresores de manera ponderada como en la Ec.4.9 se indica.

$$n\prime_r = 1 - \frac{n_r - n_{min}}{n_{max} - n_{min}} \tag{4.8}$$

, donde $n_r$ es el nivel de competencia del regresor $r$, $n\_min$ el nivel de competencia mínimo de los regresores seleccionados y $n\_max$ el nivel de competencia máxima de los regresores seleccionados.

$$p_j = \sum_{i=1}^{R} n\prime_i * p_{i,j} \tag{4.9}$$

, donde $p_j$ es la predicción final para la muestra de prueba $j$, $R$ es el número de regresores que forman parte del ensemble escogido durante el proceso de Selección Dinámica, $nl_i$ es el nivel de competencia del regresor $i$ y $p_{i,j}$ es la predicción del regresor $i$ para la muestra de test $j$.

Por otro lado, también se incluye en la librería la agregación mediante media aritmética, ya que es la más básica y se pretende realizar una librería base, por lo que esta agregación no puede faltar.

Por útlimo, pero no menos importante, la biblioteca Deslib para Python propuesta en Cruz et al. [2018] y especificada en la última revisión Cruz et al. [2020], es la única que se ha encontrado referente a la selección dinámica de ensembles. Sin embargo, esta está enfocada a clasificación. La mayoría de los métodos que se implementan en ella, son dificiles de extrapolar a la regresión. Además, se observa que los métodos implementados ya vienen con una configuración fija. Es decir, un método tendrá predefinida la manera de calcular la región de competencia (por ejemplo, con KNN), el cálculo del nivel de competencia (por ejemplo, por Ranking) y el umbral de selección (por ejemplo la media). Como consecuencia, no da opción al usuario a combinar libremente las distintas opciones de las tres etapas.

Por lo tanto, se propone con esta librería que se va a implementar en este proyecto dar la posibilidad de combinar de manera libre cada unos de los métodos de cada subtarea.

## 4.2. Diseño

La idea es crear un diseño de la librería lo más sencillo posible y parametrizable. Por ello, gracias al análisis realizado en la sección anterior, se toma la decisión de crear una única clase que contenga el cálculo de la región de competencia en todas sus variantes, el cálculo de los niveles de competencia y la selección. Es decir, una clase dedicada al proceso de Selección Dinámica. A esta la llamaremos ***Base***.

Las distintas variantes dentro de las subetapas de la selección dinámica, serán elegibles por el usuario. Es decir, estarán parametrizadas. Así el usuario podrá elegir la combinación que quiera. También estarán parametrizados aquí el número de muestras que se tomaran para crear la región de competencia, el procentaje de entrenamiento que se va a usar para la el conjunto DSEL y un booleano que controlará si el DSEL se solapa con el conjunto de entrenamiento.

A partir de esta clase, habrá otra que herede todos estos métodos y parámetros. Esta clase básicamente llama a un método u otro según el parámetro introducido por el usuario referente a la selección dinámica. A esta la llamaremos ***DynamicEnsemble***. Con esta clase los usuarios crearán los objetos que implementan un ensemble dinámico.

Por otro lado, se creará un archivo donde se implementen funciones útiles como las funciones de agregación y funciones para el preprocesamiento de los datasets. Las funciones de agregación serán utilizadas por la clase *DynamicEnsemble* para la fase de agregación del ensemble dinámico. Sin embargo, las funciones referentes al preprocesamiento de datos, se crearán para facilitar al usuario la etapa de carga y preprocesamiento de los datos previa a la utilización de un ensemble dinámico.

Veámoslo de manera más clara mediante un diagrama de clases.

### 4.2.1. Diagrama de clases

Se presenta a continuación el diagrama de clases que se han implementado para la librería. (Figura 4.2)

El diagrama es muy breve y esa es la intención. Uno de los objetivos principales es realizar una librería fácil y manejable para el usuario.

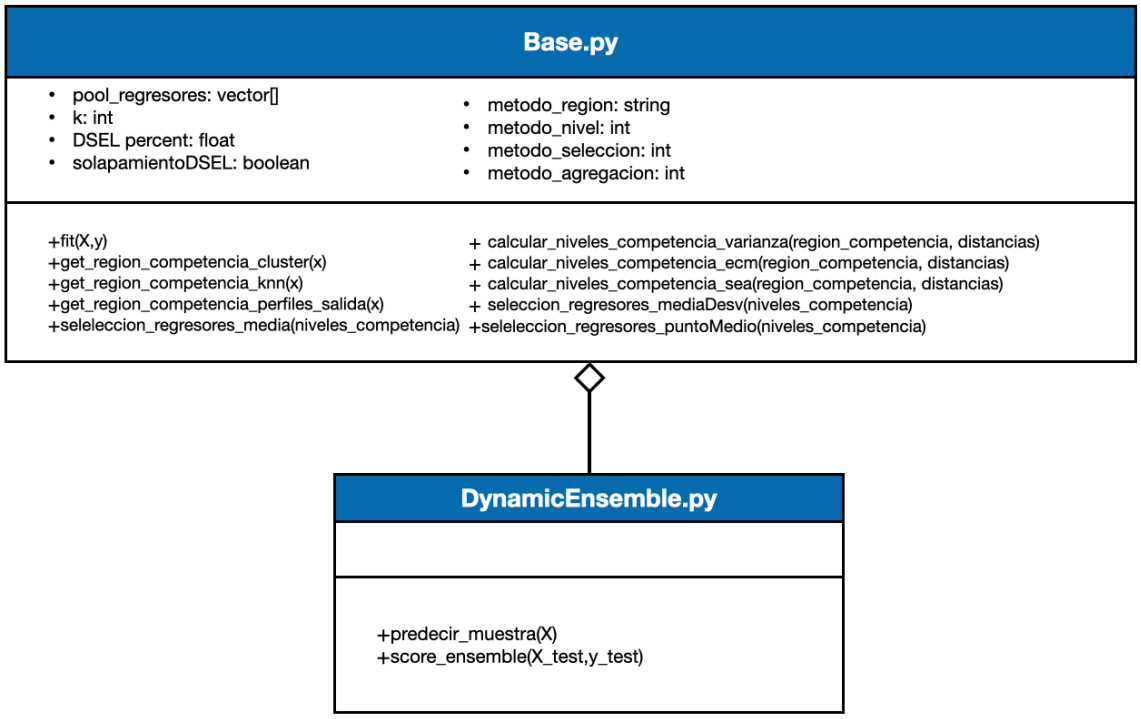


Figura 4.2: Diagrama de clases implementadas. Fuente: Elaboración propia

La idea principal ha sido crear una clase Base de la cual heredan todos los métodos que se creen en un futuro. Por ello, se incluyen en esta clase todos los métodos que llevan a cabo las distintas fases de la selección dinámica, además de la fase de generación.

La clase Base no se va a utilizar directamente por el usuario. El usuario utilizará la clase DynamicEnsemble para crear instancias. En los objetos creados a partir de dicha clase, mediante los parámetros del constructor o mediante lo métodos *setters*, el usuario será capaz de seleccionar cómo calcular la región de competencia, los niveles de competencia de los regresores, la selección de los mejores regresores y la agregación.

Realmente, la clase DynamicEnsemble se puede visualizar como un método de métodos. Es decir, a partir de esta obtenemos muchas combinaciones de submétodos, donde cada combinación se puede considerar un método distintos. Este método de métodos, y por consiguiente, todos los métodos que salen de cada combinación, se pone a disposición del usuario mediante esta clase.

Por lo tanto, la clase *Base* servirá para utilizarse de base, valga la redundancia, en implementaciones de métodos futuros.

**Clase *Base.py***

Para clarificar qué implementa y para qué sirven los parámetros y funciones de la clase *Base*, se exponen a continuación una explicación más detallada de cada uno de ellos.

- **Parámetros** Los parámetros de esta clase son los siguientes:
  - *n_regresores:* Número de regresores base.
  - *pool_regresores:* Es un vector que contiene los regresores base que se van a utilizar durante el proceso. Estos regresores pueden ser introducidos por el usuario o no. Si son introducidos por el usuario, se realiza un Bagging a cada uno de ellos para dar mayor diversidad al conjunto base. Si por el contrario el usuario no introduce ningún conjunto de regresores, la librería es capaz de generar uno realizando de nuevo Bagging.
  - *k:* Es un número entero que indica el número de muestras del DSEL que formarán parte de la región de competencia. Es decir, la dimensionalidad de la región de competencia.
  - *DSEL_percent:* Número flotante en el que se indica el porcentaje de datos del conjunto de entrenamiento que será utilizado para el conjunto DSEL.
  - *solapamientoDSEL:* Booleano que permite al usuario definir si se realiza solapamiento entre los subconjuntos de entrenamiento y DSEL. Si es *True* significa que existirá solapamiento, concretamente el conjunto de entrenamiento final será del 100 % y el DSEL será el porcentaje introducido por el parámetro *DSEL_percent*, repitiendo las muestras que se van a utilizar de entrenamiento de los regresores. Si por el contraria es *False*, no se realiza solapamiento entre conjuntos de datos de entrenamiento y DSEL, es decir, serían disjuntos.
  - *metodo_region:* Cadena en la que se especifica cómo calcular la región de competencia. Existen tres posibles valores: "knn", que utiliza el algoritmo KNN para definir la región de competencia (K vecinos más cercano), “cluster", que utiliza el algoritmo KMeans para calcular el cluster al que pertenece la nueva muestra, y "perfiles", que utiliza los perfiles de salida para encontrar las instancias más similares.
  - *metodo_nivel:* Entero con el cual se indica cómo calcular el nivel de competencia de los regresores. Si este parámetro es 0, entonces los niveles de competencia

se calcularán mediante la medida de varianza. Si es 1, serán calculados mediante el error cuadrático medio ponderado por el peso de cada muestra (Sección 4.1.2). Y si es 2, entonces serán obtenidos mediante la suma de errores absolutos ponderado por el peso de cada muestra.

- *metodo_seleccion:* Siguiendo la misma filosofía que el anterior parámetro, este es un entero con el que se indica cómo seleccionar los mejores regresores: si es 0, el umbral de selección será la media; si es 1, el umbral de selección será la media más la desviación; y si es 2, el umbral de selección se definirá como el punto medio del intervalo de niveles de competencia.

- *metodo_agregación:* Al igual que las dos anteriores, se trata de un número entero con el que se indica cómo agregar las salidas del ensemble. Si el parámetro está a 0, se realizará la agregación mediante media aritmética de las salidas de los regresores escogidos. Sin embargo, si el parámetro está a 1, se realizará la agregación mediante media ponderada.

- **Métodos**

  Los métodos que son necesarios para el correcto funcionamiento de la clase son los siguientes:

  - *fit(X,y):* Función en la que se realiza el entrenamiento del ensemble. Si cuando este método es llamado el usuario no ha incorporado el ensemble, es en este método donde se crea el ensemble homogéneo a partir de un proceso de Bagging. Sin embargo, si existe un ensemble que ha sido introducido por el usuario, entonces es cuando cada regresor base propuesto por el usuario es sometido a un proceso de Bagging. Se realiza también durante este método la partición del conjunto de entrenamiento y el DSEL, ya que X e y, que son los parámteros de este método, hacen referencia al conjunto de entrenamiento completo, antes de realizar la partición. Es necesaria hacerla aquí ya que con el nuevo conjunto de entrenamiento (subconjunto de X) es con el que se van a entrenar los distintos regresores base. Por último, en este método también se entrena el algoritmo con el que se va a calcular la región de competencia (solo si se utiliza knn o clustering).

  - *get_region_competencia_cluster(x):* Con este método se pretende obtener la región de competencia mediante clustering. En ella, el parámetro $x$ será la muestra a partir de la cual se debe calcular la región de competencia. En el caso del clustering, se utilizará el algoritmo de scikit-Learn *KMeans* entrenado con el conjunto

de DSEL previamente, para calcular el cluster que pertenece. Este método devuelve los puntos del DSEL que se encuentran en el mismo cluster que la muestra de prueba.

- *get_region_competencia_knn(x):* Al igual que en el anterior, en este se calcula la región de competencia, pero ahora mediante el algoritmo de scikit-Learn *KNearestNeighbours* también entrenado anteriormente con el conjunto DSEL. Esta función devuelve los puntos del DSEL que forman parte de la región de competencia de la muestra $x$ y las distancias a dichos puntos.

- *get_region_competencia_perfiles(x):* Igual que en las anteriores, se calcula la región de competencia de la muestra de prueba $x$. Sin embargo, ahora no se utiliza un algoritmo ya implementado, si no que se implementará la propia lógica de elección de puntos del DSEL mediante el cálculo de la similitud de los perfiles de salida.

- *calcular_niveles_competencia_varianza(region_competencia):* Una de las maneras de calcular lo niveles de competencia de los regresores es mediante la medida de varianza. Este método calcula justo eso. El parámetro que necesita es el de la región de competencia, ya que será en ella donde se evaluen los regresores base. En base a la varianza en las predicciones que obtenga un regresor, se calcula su nivel de competencia. Esta función devuelve los niveles de competencia de todos los regresores.

- *calcular_niveles_competencia_ecm(region_competencia, distancias):* Función que también calcula los niveles de competencia pero esta vez mediante el error cuadrático medio. Además, este error está ponderado por los pesos de los puntos del DSEL, que están relacionados con la inversa de la distancia. Estos pesos también son calculados dentro de esta función. Este método, al igual que el anterior, devuelve los niveles de competencia de los regresores base.

- *calcular_niveles_competencia_sea(region_competencia, distancias):* Última función referida al cálculo de los niveles de competencia. Esta es exactamente igual que la anterior, pero en lugar de calcular el error cuadrático medio, se calcula la suma de los errores absolutos. También devuelve los niveles de competencia de los regresores base.

- *seleccionar_regresores_media(niveles_competencia):* Este método genera el umbral con el que se definirá la cota superior de los niveles de competencia escogi-

dos igual a la media de los niveles de competencia de todos los regresores. Los niveles de competencia deben ser introducidos por parámetro, que es a partir de los que se va a calcular el umbral. Seguidamente se hace la selección de los que se quedan por debajo del umbral y se devuelve el vector con los regresores seleccionados y otro vector con los niveles de competencia de los regresores escogidos.

- *seleccionar_regresores_mediaDesv(niveles_competencia):* Misma filosofía que el método anterior, pero ahora el umbral se definirá mediante la media más la desviación típica de los niveles de competencia.

- *seleccionar_regresores_puntoMedio(niveles_competencia):* Idéntica a las dos anteriores excepto en la definición del umbral. Este umbral se calculará mediante la fórmula del punto medio de un intervalo. El intervalo vendrá dado por el máximo y el mínimo de los niveles de competencia. Estos se suman y de divide entre dos. Así se obtiene el punto medio del intervalo que no es lo mismo que la media.

**Clase *DynamicEnsemble.py***

Esta clase implementa todos los métodos posibles mediante la configuración de los parámetros. Mediante los valores de los parámetros escogidos por el usuario, realiza una llamada u otra a los métodos que hereda de la clase *Base* en función del valor de cada parámetro con el que se controla el cálculo de la región de competencia (*metodo_region*), el que controla cómo calcular el nivel de competencia de los regresores base (*metodo_nivel*) y el que controla el método de selección de los regresores (*metodo_selección*). Además, es aquí donde incorpora algunas funciones que, como se ha dicho anteriormente, se consideran de apoyo y se encuentran en un fichero aparte. Es decir, la agregación de las salidas. Al igual que antes, según el valor escogido por el usuario en el parámetro (*metodo_agregación*), este hará una llamada a una función u otra de entre las implemntadas como agregación. Veámos entonces los distintos parámetros y métodos que implementa esta clase.

- **Parámetros** Los parámetros de esta clase son los mismos que los de la clase anterior, ya que los hereda. No tiene ningún parámetro adicional.

- **Métodos**

  - *predecir_muestra(x):* En este método es donde realiza las llamadas a las funciones de cálculo de región de competencia, niveles de competencia y selección de

los mejores niveles de comeptencia mediante la comprobación de los distintos valores de los parámetros. Como parámetro de entrada del método solo tenemos $x$, que es la muestra de prueba de la que se quiere predecir su salida. Por lo tanto, este método devuelve la predicción de la muestra $x$ dada por el ensemble dinámico con las características seleccionadas por el usuario mediante los parámetros de clase.

- *score_ensemble(X_test, y_test):* Esta función básicamente calcula y devuelve el porcentaje de error absoluto medio y la raíz del error cuadrático medio que comete el ensemble dinámico en las predicciones de las muestras de test X_text (se le pasa por parámetro). Además, se pasa por parámetro el vector de las salidas reales de las muestras de test (y_test) para poder calcular el error. Para calcular las predicciones de las muestra, se llama a la función anterior.

### 4.2.2. Métodos

Como ya se ha reiterado en varias ocasiones, para cada subetapa de la Selección Dinámica, existen distintas opciones que el usuario puede indicar mediante un parámetro. Es decir, habrá un parametro mediante el cual se elija el método de la región de competencia, otro en el que se indique cómo medir los niveles de competencia de los regresores y otro para seleccionar el método de selección de los mejores regresores. Además, otro parámetro será el encargado de determinar de que manera se realiza la agregación de salidas. Como se ve en la imagen, 4.3, cada etapa (rectángulos azules) sería un parámetro en el cual se podría elegir entre las opciones que cuelgan de dicho rectángulo.

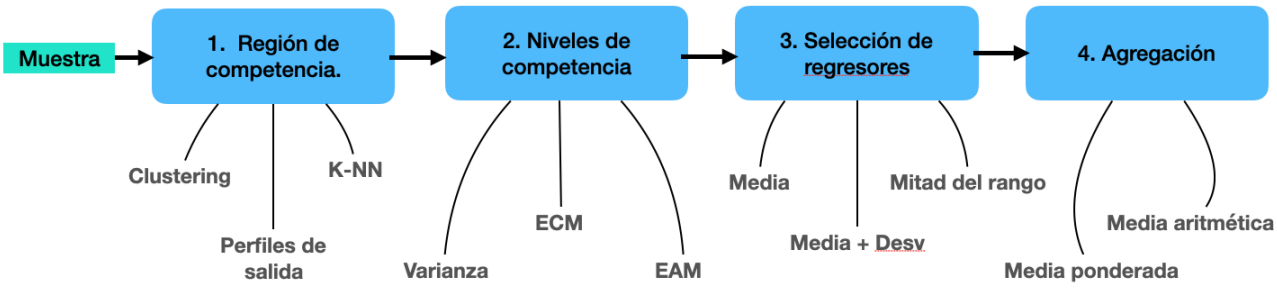


Figura 4.3: Parámetros elegibles y sus distintas opciones. Funete: Elaboración Propia.

La elección de cada parámetro debe ser independiente de los demás. Es decir, debe

ser posible crear cualquier combinación de valores de los parámetros. Por consiguiente, la librería debe ser lo suficientemente robusta como para soportar cualquier combinación y cumplir con las expectativas expuestas en esta fase.

#### Combinaciones

Las distintas combinaciones de parámetros son cincuenta y cuatro en total. Es lo mismo que decir que existen cincuenta y cuatro métodos o maneras distinas de calcular la predicción a partir de un ensemble dinámico con esta librería. Por lo tanto, los métodos (combinaciones) que se implementan son los que se enumeran en la tabla 4.1, diferenciados entre sí por el cálculo de la región de competencia, el cálculo de los niveles de competencia, la selección de los regresores y la manera de agregar las salidas de los regresores elegidos.

Todos estos métodos pueden ser utilizados por los usuarios a través de la clase DynamicEnsemble mediante la configuración de sus parámetros.

| Método | Región compet. | Nivel compet. | Selección regresores | Agregación |
|---|---|---|---|---|
| Método 1 | KNN | Varianza | Media | Media aritmética |
| Método 2 | KNN | Varianza | Media | Media ponderada |
| Método 3 | KNN | Varianza | Media+DesvTíp | Media aritmética |
| Método 4 | KNN | Varianza | Media+DesvTíp | Media ponderada |
| Método 5 | KNN | Varianza | MAX+MIN/2 | Media aritmética |
| Método 6 | KNN | Varianza | MAX+MIN/2 | Media ponderada |
| Método 7 | KNN | ECM | Media | Media aritmética |
| Método 8 | KNN | ECM | Media | Media ponderada |
| Método 9 | KNN | ECM | Media+DesvTíp | Media aritmética |
| Método 10 | KNN | ECM | Media+DesvTíp | Media ponderada |
| Método 11 | KNN | ECM | MAX+MIN/2 | Media aritmética |
| Método 12 | KNN | ECM | MAX+MIN/2 | Media ponderada |
| Método 13 | KNN | SEA | Media | Media aritmética |
| Método 14 | KNN | SEA | Media | Media ponderada |
| Método 15 | KNN | SEA | Media+DesvTíp | Media aritmética |
| Método 16 | KNN | SEA | Media+DesvTíp | Media ponderada |
| Método 17 | KNN | SEA | MAX+MIN/2 | Media aritmética |
| Método 18 | KNN | SEA | MAX+MIN/2 | Media ponderada |
| Método 19 | Clustering | Varianza | Media | Media aritmética |
| Método 20 | Clustering | Varianza | Media | Media ponderada |
| Método 21 | Clustering | Varianza | Media+DesvTíp | Media aritmética |
| Método 22 | Clustering | Varianza | Media+DesvTíp | Media ponderada |
| Método 23 | Clustering | Varianza | MAX+MIN/2 | Media aritmética |
| Método 24 | Clustering | Varianza | MAX+MIN/2 | Media ponderada |
| Método 25 | Clustering | ECM | Media | Media aritmética |
| Método 26 | Clustering | ECM | Media | Media ponderada |
| Método 27 | Clustering | ECM | Media+DesvTíp | Media aritmética |
| Método 28 | Clustering | ECM | Media+DesvTíp | Media ponderada |
| Método 29 | Clustering | ECM | MAX+MIN/2 | Media aritmética |
| Método 30 | Clustering | ECM | MAX+MIN/2 | Media ponderada |
| Método 31 | Clustering | SEA | Media | Media aritmética |
| Método 32 | Clustering | SEA | Media | Media ponderada |
| Método 33 | Clustering | SEA | Media+DesvTíp | Media aritmética |
| Método 34 | Clustering | SEA | Media+DesvTíp | Media ponderada |
| Método 35 | Clustering | SEA | MAX+MIN/2 | Media aritmética |
| Método 36 | Clustering | SEA | MAX+MIN/2 | Media ponderada |
| Método 37 | Perfiles de salida | Varianza | Media | Media aritmética |
| Método 38 | Perfiles de salida | Varianza | Media | Media ponderada |
| Método 39 | Perfiles de salida | Varianza | Media+DesvTíp | Media aritmética |
| Método 40 | Perfiles de salida | Varianza | Media+DesvTíp | Media ponderada |
| Método 41 | Perfiles de salida | Varianza | MAX+MIN/2 | Media aritmética |
| Método 42 | Perfiles de salida | Varianza | MAX+MIN/2 | Media ponderada |
| Método 43 | Perfiles de salida | ECM | Media | Media aritmética |
| Método 44 | Perfiles de salida | ECM | Media | Media ponderada |
| Método 45 | Perfiles de salida | ECM | Media+DesvTíp | Media aritmética |
| Método 46 | Perfiles de salida | ECM | Media+DesvTíp | Media ponderada |
| Método 47 | Perfiles de salida | ECM | MAX+MIN/2 | Media aritmética |
| Método 48 | Perfiles de salida | ECM | MAX+MIN/2 | Media ponderada |
| Método 49 | Perfiles de salida | SEA | Media | Media aritmética |
| Método 50 | Perfiles de salida | SEA | Media | Media ponderada |
| Método 51 | Perfiles de salida | SEA | Media+DesvTíp | Media aritmética |
| Método 52 | Perfiles de salida | SEA | Media+DesvTíp | Media ponderada |
| Método 53 | Perfiles de salida | SEA | MAX+MIN/2 | Media aritmética |
| Método 54 | Perfiles de salida | SEA | MAX+MIN/2 | Media ponderada |

Tabla. 4.1: Métodos a implementar.

## 4.3. Desarrollo

Durante esta sección, se pretende dar forma a todas las decisiones tomadas en los pasos anteriores. Para ello, se detallarán en los siguientes puntos tanto la implmentación, como las características hardware y software con las que se ha llevado a cabo el la misma. Además de nombrar las librerías de apoyo ya existentes que se han utilizado.

### 4.3.1. Lenguaje de programación y entorno de desarrollo

El lenguaje de programación con el que se ha implementado la librería ha sido **Python**. Este es un lenguaje de programación de alto nivel que se caracteriza por su versatilidad y facilidad de uso. Además, es un lenguaje multiplataforma de código abierto, lo que significa que es gratuito. Este está tomando cada vez más fuerza en el campo de la Minería de Datos, Ciencia de Datos y Big Data, y por ello, cada vez son más los recursos y librerías implementadas en este lenguaje de programación dentro de este ámbito.Canales Luna [2022]

Como entorno de desarrollo se ha utilizado Visual Studio Code. Esta decisión ha sido muy personal, ya que la principal razón por la que se ha elegido este entorno de desarrollo ha sido por comodidad y la experiencia previa en la utilización del mismo. Entre otras características, destaca la manera tan visual y clara en la que realiza la vista de las variables durante el *debugging*.

### 4.3.2. Librerías de apoyo

Para la realización de la librería, se utilizan algunos métodos y clases de otras librerías ya implementadas en Python.

Concretamente, para los regresores base se utilizan los algoritmos para regresión implementados en **Scikit-Learn**. También, el proceso de Bagging interno que se realiza para la generación de los ensembles, es el implementado por la clase BaggingRegressor de esta misma librería. Esta clase genera un Bagging a los regresores que se le introducen por parámetro creando una subdivisión del conjunto de entrenamiento con reemplazo. Es decir, los subconjuntos creados no son disjuntos, porque una misma instancia puede aparecer en varias particiones. También se utilizan de esta librería algunas funciones como *check_X_y()*, o *check_is_fitted()*, para la comprobación de ciertos aspectos del proceso de constucción

de un modelo.

Por otro lado, también se utiliza la librería ***Numpy***, ***Pandas*** y ***Scipy***, para el uso de métodos estadísticos como el cálculo de los errores , para la utilización de dataframes y volcado de datos, y para el uso de funciones para el cálculo de las distancias, respectivamente.

### 4.3.3. Implementación

Las distintas clases expuestas en el diseño, junto con todos sus parámetros y métodos han sido implementados en Python en la librería de manera satisfactoria.

Se detallan a continuación todas las decisiones de implementación concretas que se han llevado a cabo.

Durante esta sección, también se presentan algunas de las funciones implementadas en la librería mediante pseudocódigo. Solo aquellas cuyo procedimiento puede resultar más complicado de entender.

Además, como ya sabemos por el diseño, cada manera de calcular la región de competencia, niveles de competencia y la selección ha sido implementada mediante un método distinto. Sin embargo, dentro de las que se refieren a la misma parte del proceso, por ejemplo al cálculo de la región de competencia, tienen un funcionamiento parecido. Por ello, para cada etapa, se mostrará mediante el pseudocódigo el proceso de manera más general, la parte común a todos los métodos destinado a dicha etapa o las parte de implementación más complicadas de entender (como el cálculo de los perfiles de salida para la región de competencia).

Igualmente, se comentarán y aclararán los cambios para cada método distinto dentro de cada etapa.

#### Cálculo de la región de competencia

Existen tres métodos destinados a este cálculo. En concreto :

- *get_region_competencia_knn(x):* Este realiza una llamada al método *.predict()* del al-

goritmo de KNN de Scikit Learn que ha sido previamente entrenado con el conjunto DSEL. Dicha función devuelve dos vectores: un vector que contiene los puntos del DSEL seleccionados y otro con las distancias a dichos puntos.

- *get_region_competencia_cluster(x):* Este método se ha implementado siguiendo un procedimiento similar al anterior. Se utiliza el método *.predict()* de la clase KMeans de Scikit-Learn. Este devuelve el cluster al que pertenece la muestra $x$. Sin embargo, se pretende obtener la misma salida que el anterior: dos vectores, uno con los puntos del DSEL seleccionados y otro con las distancias. Esta parte se ha implementado de la sigiuente manera: Una vez obtenido el cluster al que pertenece la muestra de prueba, se recorre el DSEL buscando y almacenando todos los puntos que pertenecen al mismo cluster. Así obtenemos los puntos de la región de competencia. Para obtener las distancia, se calcula la distancia euclídea entre cada punto de la región de competencia y la muestra de prueba y son almacenadas. Para calcular la distancia euclçidea se ha hecho uso de la funcion *distance.euclidean(punto_region_competencia, muestra_prueba)* de la librería *Scipy*.

- *get_region_competencia_perfiles(x):* Este es el método más diferente con respecto a los dos anteriores. Este método ha sido implementado desde cero, sin utilizar algoritmos de aprendizaje como el KNN o KMeans. Por lo tanto, para comprender mejor cómo se han implementado los diferentes pasos dentro de este método, se presenta el siguiente pseudocódigo 1:

**Algoritmo 1:** Cálculo de la región de competencia mediante perfiles de salida.

```
Result: región_competencia, distancias
perfil_salida_muestraPrueba[] <– predecir_muestra_regresores_base(x);
vector_distancias <– ∅
perfiles_salida_DSEL[][] <– predecir_DSEL_regresores_base;
for punto en DSEL do
    vector_distancias[punto] <– distancia_euclidea(perfil_salida_muestraPrueba,
     perfiles_salida_DSEL);
end
umbral_distancias <– media(vector_distancias)
for i desde 0 hasta len(DSEL) do
    if umbral_distancias mayor que vector_distancias[i] then
        region_competencia <– DSEL[i];
    end
end
distancias <– distancia_euclídea (x, region_competencia)
```

Como se puede observar en el pseudocódigo 1, primero se calculan los perfiles de salida de la muestra de prueba $x$. A continuación, se calculan los perfiles de salida de los puntos del DSEL. Un perfil de salida de una muestra será un vector donde se

almecenan todas las predicciones de los regresores base de dicha muestra. A continuación, se calcula la distancia euclídea entre los perfiles de salida de la muestra de prueba y los perfiles de salida de todos los puntos del DSEL. Con estas distancias es con lo que se mide la similitud. A mayor distancia, menor similitud y viceversa. Obtenidas las distancias, se considera un umbral para la selección de los más similares. Concretamente, se escogen los puntos que se encuentren por debajo de este. Por último, se calculan las distancias euclídea de los puntos de la región de competencia y la muestra de prueba mediante la misma funcion de *Scipy* utilizada en los dos métodos anteriores. Por último, esta función devuelve el vector con los puntos de la región de competencia y el vector de distancias.

**Cálculo de los niveles de competencia de los regresores base**

Para esta etapa se han implementado otros tres métodos especificados en el diseño. Concretamente, son *calcular_niveles_competencia_varianza()*, *calcular_niveles_competencia_ecm()* y *calcular_niveles_competencia_sea()*, donde como ya sabemos, lo único que los diferencia entre sí es el error o medida que se implementa para medir la bondad de cada regresor. Además, en el primero no se utilizan pesos para ponderar los errores cometidos en los puntos por cada regresor.

Veámos el pseudocódigo de la función *calcular_niveles_competencia_ecm()*:

**Algoritmo 2:** Cálculo de los niveles de competencia mediante ECM

```
Result: niveles_competencia
error_regresor_punto[][] <– ∅
niveles_competencia[] <– ∅
suma_distancias_inversas <– suma_inversas(distancias)
for regresor en pool_regresores: do
    for punto en region_competencia do
        error_regresor_punto[regresor][punto]<–
          calcular_ecm(regresor.predict(punto), y_real[punto]));
    end
end
for i desde 0 hasta n_regresores): do
    for j desde 0 hasta n_puntos_region_competencia): do
        ponderacion_punto <–
          calcular_ponderacion(distancia[j],suma_distancias_inversas)
          errores_ponderados[j] <– ponderacion_punto * error_regresor_punto[i][j]
    end
    nivel_competencia[i] <– suma(errores_ponderados[i])
end
```

Donde las función *calcular_ponderacion()* implementa lo que se especifica en la ecuación 4.1

Cogiendo de base el pseudocódigo 2 , solo habría que cambiar la función *calcular_ecm()* por *calcular_sea* o *calcular_varianza()* para generar la implementación de los otros dos métodos. Aunque hay que tener en cuenta que para la varianza se eliminaría la parte donde se calculan los pesos de las muestras en base a la inversa de la distancia.

### Selección de los regresores más prometedores

Por último, durante esta etapa se pueden elegir tres maneras de proceder. No obstante, todas son similares a la hora de implementarlas. El pseudocódigo común para las tres es el siguiente:

**Algoritmo 3:** Selección de los regresores más prometedores.

```
Result: niveles_competencia
umbral_seleccion <– calcular_umbral();
niveles_competencia_seleccionados[] <– ∅
regresores_seleccionados[] <– ∅
contador_seleccionados <– 0
for i en len(niveles_competencia) do
    if umbral_selección mayor o igual a nivel_competencia[i] then
        regresores_seleccionados[contador_seleccionados]<–pool_regresores[i]
        niveles_competencia_seleccionados[] <– niveles_competencia[i]
        contador_seleccionados <– contador_seleccionados + 1
    end
end
```

Como podemos ya intuir, la única diferencia entre los métodos que implementan la selección de regresores para el ensemble dinámico es la manera de definir la funcion *calcular_umbral()*. Para el método *seleccionar_regresores_media()*, el umbral será definido con la media de los niveles. En *seleccionar_regresores_mediaDesv()* el umbral debe calcularse con la media más la suma de la desviación típica, por lo que en este se implementa el cálculo de la desviación típica de los niveles de competencia haciendo uso de la función *.std()* de la librería *Numpy*. Por último, para determinar el umbral calculado con el punto medio, se implementa la fórmula 4.2.

# Capítulo 5

# PRUEBAS Y RESULTADOS

En el capítulo que nos acontece, se exponen las distintas pruebas y experimentaciones que se han hecho para comprobar la funcionalidad de la librería.

Se explican los distintos experimentos que se han llevado a cabo, los datasets o conjuntos de datos utilizados para las pruebas, los métodos con los que se ha probado y los distintos valores que se le han asignado a los parámetros para cada prueba.

## 5.1. Marco experimental

Para probar la funcionalidad de la librería y comprobar si las técnicas implementadas cumplen con las expectativas de mejora expuestas durante el documento, se realizan múltiples experimentos con distintos datasets, parámetros y métodos. No obstante, para las pruebas finales se van a presentar solo las más interesantes, de las que se pueden extraer las conclusiones más relevantes.

Cabe destacar que se ha utilizado un escalado de las variables a todos los datasets, ya que se trabaja con distancias y es muy importante en este caso escalar la entrada. Si no, las pruebas darían resultados espúrios.

### 5.1.1. Datos utilizados

Para las pruebas finales, se ha decidido experimentar con tres conjuntos de datos distintos para probar las diferentes funcionalidades e intentar obtener unas conclusiones que no están sesgadas a un solo conjunto de datos.

Por ello, se consideran 3 conjuntos de datos para la experimentación, para obtener información más generalizada sobre el comportamiento de los ensembles dinámicos.

Concretamente los datasets son:

- Conjunto de datos 1: **50 Startups**.
  Este conjunto de datos para regresión cuenta con 50 instancias. Cada una de ellas presenta 4 atributos y la salida. Los atributos de este dataset se corresponden a los siguientes:

  1. *RD Spend*: Gasto en investigación y desarrollo (unidad: dolares)
  2. *Administration*: gasto en administración (unidad: dolares)
  3. *Marketing Spend*: Gasto en marketing (unidad: dolares)
  4. *State*: estado - referente a la localización (cadena de texto).

  La variable de salida es el *Profit*, es decir, el beneficio obtenido en dolares (variable de tipo decimal). Por lo tanto, con este dataset se pretenderá predecir el beneficio que

obtendrá una empresa del tipo *Startup* conociendo su gasto en investigación y desarrollo, en administración y en marketing y el lugar donde se encuentra la empresa.

- Conjunto de datos 2: **Student Marks**.
  Este es un conjunto de datos de regresión que pretende predecir la nota que va a obtener un alumno según el tiempo que le dedica y las asignaturas que estudia a la vez. Por lo tanto, cada instancia de este dataset tendrá 2 atributos de entrada:

  1. *Number_courses*: Número de asignaturas que cursa a la vez (entero).
  2. *time_study*: Tiempo medio diario que dedica al estudio (unidad: horas).

  La variable de salida es *mark*, que es la nota final que obtiene el alumno. La salida es de tipo decimal.

  El dataset cuenta con 100 muestras o instancias.

- Conjunto de datos 3: **Real State Price**.
  Por último, se escoge un conjunto de datos para regresión más grande que los dos anteriores. Este presenta 414 ejemplos con 7 variables, incluyendo la variable objetivo. La información de los atributos de entrada es la siguiente:

  1. X1: Fecha de la transacción (por ejemplo, 2013.250=Marzo de 2013, 2013.500=Junio de 2013, etc.)
  2. X2: Tiempo que lleva construida la casa (unidad: años).
  3. X3: Distancia a la estación de metro más cercana (unidad: metros)
  4. X4: El número de tiendas de conveniencia cercanas a pie (entero)
  5. X5: Latitud, coordenada geográfica (unidad: grados)
  6. X6: Longitud, coordenada geográfica (unidad: grados)

  La salida *y* es el coste de la casa, cuya unidad es dolares por metro cuadrado (número decimal).

  Con este conjunto de datos se pretende estimar el valor real de una casa sabiendo la fecha de la transacción, el tiempo que lleva la casa construida, la distancia a la estación de metro más cercana, etcétera.

Todos estos datasets son descargados de la plataforma `Kaggle.com`

### 5.1.2. Métodos comparados.

Se ha realizado una comparación de todos los métodos que se han implementado para la selección dinámica de ensembles para regresión (54 en total) con respecto a su equivalente ensemble estático.

El ensemble estático tendrá el funcionamiento básico. Las salidas se agregan por media aritmética. Los ensembles tanto para dinámico como estático, son los mismos.

Se han realizado dos tipos de comparaciones:

- En un primer lugar, creando un ensemble heterogéneo a partir de seis regresores. Cada uno de estos genera realmente 10 regresores más, ya que internamente se realiza un proceso de *Bagging* de ellos y para los experimentos se ha decidido que sean 10 regresores los que se generen a partir de cada proceso de Bagging así, se obtienen sesenta regresores base.

  Para la generación del ensemble estático se realiza exactamente el mismo proceso de Bagging para que los resultados obtenidos sean comparables. Así, se comparará el error cometido por el ensemble estático con respecto al error cometido por los 54 métodos de ensembles dinámicos que se han implementado.

- En segundo lugar, de la misma manera, se hace otra comparación pero esta vez con un ensemble homogéneo. Esto significa que todos los regresores base parten del mismo algoritmo de regresión. Además, para crear el ensemble, se sigue el mismo proceso de Bagging, ya automatizado en los métodos de selección dinámica, tanto para el ensemble dinámico como para el estático. Dicho ensemble se ha creado con sesenta regresores provinientes del proceso de Bagging del modelo base elegido.

### 5.1.3. Parámetros fijados para las pruebas.

Para los experimentos, hay una serie de parámetros que han quedado fijados para todos ellos. Tanto para los ensembles dinámicos homogéneos como para los heterogéneos, los parámetros fijos en los métodos de Selección dinámica son los que se presentan en la siguiente tabla 5.1:

| Parámetro | Valor |
|---|---|
| Nº regresores base | 60 |
| Región de competencia ($k$) - KNN | 7 |
| Nº clusters ($k$) - Clustering | 7 |

Tabla. 5.1: Parámetros fijos para ensemble dinámicos durante la experimentación.

En la tabla 5.1, las dos últimas columnas hacen referencia a un mismo parámetro $k$. No obstante, si el método que se está utilizando para el cálculo de la región de competencia es *KNN*, entonces el parámetro $k$ significará el número de vecinos a escoger. Sin embargo, si se utiliza *Clustering*, el parámetro $k$ define de número de grupos. Por último, para el cálculo de la región de competencia mediante perfiles de salida no hace falta indicar ningún valor de $k$, ya que los puntos del DSEL escogidos para la región de competencia se seleccionan según un umbral de similitud (Véase Sección 4.1.2) , por lo que en este caso este parámetro no se utiliza. Cabe destacar que estos parámetros se han escogido en función de ciertas pruebas previas hechas, pero pueden ser configurados por el usuario igual que otra gran cantidad de ellos.

Con respecto a los ensembles heterogéneos, tanto dinámicos como estáticos, se fija en la tabla 5.2 el conjunto base de regresores que será utilizado para todas las pruebas. De igual manera, para los homogéneos se fija el modelo base como se muestra en la tabla 5.3.

Para formar este conjunto de algorimos base y fijarlo para todos los experimentos, se han elegido algunos de los algoritmos más utilizados en la literatura y que mejor han funcionado en pruebas iniciales y con configuraciones típicas.

| Algoritmos base - Heterogéneo |
|---|
| KNeighborsRegressor(n_neighbors=5) |
| KNeighborsRegressor(n_neighbors=3) |
| linear_model.Lasso(alpha=0.15) |
| linear_model.Lasso(alpha=0.4) |
| linear_model.Ridge(alpha=0.5) |
| linear_model.Ridge(alpha=0.2) |

Tabla. 5.2: Regresores base utilizados para ensembles heterogéneos durante la experimentación.

| Algoritmo base - Homogéneo |
|---|
| DecissionTreeRegressor() |

Tabla. 5.3: Regresor base utilizado para ensembles homogéneos durante la experimentación.

### 5.1.4. Criterios de comparación.

Con el objetivo de comparar los ensembles dinámicos implementados con su equivalente ensemble estático, se escogen las siguientes dos medidas de error:

- **MAPE** (Mean Absolute Percentage Error) (Véase Sección. 2.4.2)
- **RMSE** (Root Mean Square Error) (Véase Sección. 2.4.2)

Para ambos casos, lo ideal es estar cercano a cero. Si alguna de estas medidas es cero, significa que no se ha cometido ningún error de estimación en ninguna de las muestras estimadas.

Se seleccionan estas porque, como se ha visto durante la revisión bibliográfica, son las más utilizadas para evaluar de forma general el funcionamiento de las técnicas de estudio.

### 5.1.5. Validación

Para la validación de los modelos y, como consecuencia, el cálculo de los errores, los distintos conjuntos de datos se han particionado dando lugar a 5 particiones cada uno. Es decir, se ha utilizado validación cruzada con 5 particiones disjuntas (*5-folds Cross-Validation*), utilizando el 20 % de los datos para el conjunto de test y el 80 % restante para el entrenamiento.

## 5.2. Experimentación y análisis de los resultados

Las características de la experimentación son las siguientes:

- Las pruebas mostradas en esta sección se dividirán en experimentos. Cada experimento será referente a un conjunto de datos distinto.
- Dentro de cada experimento, además se harán tres pruebas cambiando los parámetros del porcentaje de DSEL utilizado y del solapamiento de DSEL con el de entrenamiento.

- Para cada prueba, se presentan dos tablas, una que presentará la comparación de los ensembles dinámicos con respecto a su estático en base al MAPE y otra en base al RMSE.

- Los diferentes parámetros que se utilizan en cada prueba (los mismos para todos los datasets o experimentos) son los que se presentan en la tabla 5.4.

| Prueba | %DSEL | solapamiento DSEL-TRAIN |
|---|---|---|
| Prueba 1 | 20 % | NO |
| Prueba 2 | 50 % | SÍ |
| Prueba 3 | 95 % | SÍ |

Tabla. 5.4: Parámetros de cada prueba.

- Por último, cabe resaltar que las tablas de presentación de resultados contienen algunos resultados en **negrita**. Esto significa que el método de ensemble dinámico implementado mejora al resultado obtenido por su correspondiente ensemble estático.

### 5.2.1. Experimento 1

En este experimento se exponen las pruebas que se han realizado con el dataset **50 Startups**.

Los resultados obtenidos con los ensembles estáticos (homogéneo y heterogéneo) son los que se presentan en la tabla 5.5. Estos resultados son los que se van a comparar con los obtenidos por los métodos dinámicos implementados.

Tabla. 5.5: Estáticos 50

| | **Estático** | |
|---|---|---|
| | **Heterogéneo** | **Homogéneo** |
| **MAPE** | 0,190630378 | 0,317424553 |
| **RMSE** | 14114,19191 | 24374,02316 |

Recordemos que, como en la tabla 5.4 se indica, en esta prueba se fijan los parámetros %DSEL a 20 y sin permitir solapamiento con conjunto de entrenamiento.

**Prueba 1**

En la tabla 5.6, se presentan los errores MAPE cometidos por los métodos dinámicos implementados sobre el conjunto de datos *50 Startups*. Presenta dos columnas de resultados: una para el ensemble heterogéneo y otra para el homogéneo.

Para visualizar de manera más sencilla el comportamiento de los distintos métodos dinámicos, se han realizado las gráficas 7.1 y 5.2, referentes al ensemble heterogéneo y homogéneo respectivamente. Básicamente, la línea azul es la que se crea a partir de la unión de los puntos que representan los errores MAPE que obtienen los distintos métodos de ensembles dinámicos implementados y la línea roja indica el valor MAPE obtenido por el ensemble estático equivalente. Estas gráficas son presentadas a modo orientativo, ya que en el eje de la X no se pueden representar todos los valores (específicamente, los 54 métodos implementados) y no se puede ver claramente a qué método pertenece cada punto.

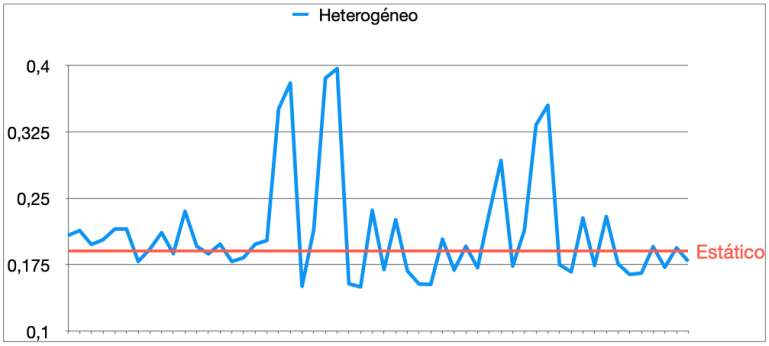


Figura 5.1: Gráfica MAPE con %DSEL = 20 HETEROGÉNEO. Fuente: Elaboración Propia.

Si nos centramos en el ensemble heterogéneo, en la tabla 5.6 vemos que en 23 métodos dinámicos se mejora el error con respecto a su versión estática.

Los tres mejores métodos en este caso han sido *knn-ecm-media-ponderada*, *knn-var-mediaDesv-media* y *knn-sea-media-ponderada*, siendo el mejor **knn-ecm-media-ponderada**. Con respecto a estos, destaca que en todos la región de competencia ha sido calculada mediante KNN.

Como se puede observar en la gráfica 7.1, de un vistazo vemos que muchos métodos dinámicos mejoran, es decir, se quedan por debajo del estático. Además, también se pueden resaltar los picos que hace la gráfica. Más o menos, los errores obtenidos por los distintos métodos, obtienen unos resultados similares, concentrado en un rango pequeño (entre 0,15 y 0,2 aproximadamente). Sin embargo, destacan los picos que hace la línea azul. Estos son

Tabla. 5.6: 50-MAPE-dsel20

| | Dinámico | |
|---|---|---|
| **Método** | **Heterogéneo** | **Homogéneo** |
| Cluster-var-media-media | 0,2080078018624584 | 0,32949279030135437 |
| Cluster-var-media-ponderada | 0,2135527034996303 | 0,3216779846561814 |
| Cluster-var-mediaDesv-media | 0,19782279765748115 | 0,3249384918132699 |
| Cluster-var-mediaDesv-ponderada | 0,20317432355233356 | 0,321901200152084 |
| Cluster-var-pMedio-media | 0,21531890587664898 | 0,3230209303001935 |
| Cluster-var-pMedio-ponderada | 0,21527355421832875 | 0,3217846055122965 |
| Cluster-ecm-media-media | **0,17862197746395975** | 0,32827579942254637 |
| Cluster-ecm-media-ponderada | 0,19259337254383693 | 0,3244364747174106 |
| Cluster-ecm-mediaDesv-media | 0,211187093630015 | 0,3192941679731424 |
| Cluster-ecm-mediaDesv-ponderada | **0,18746814180397225** | 0,3210647364990128 |
| Cluster-ecm-pMedio-media | 0,235313248655388 | 0,324233693730497 |
| Cluster-ecm-pMedio-ponderada | 0,19584055487780241 | 0,32285425273565 |
| Cluster-sea-media-media | **0,18726956055644634** | 0,3232118635601848 |
| Cluster-sea-media-ponderada | 0,1982811600671852 | 0,3232228279625637 |
| Cluster-sea-mediaDesv-media | **0,17858374000460314** | 0,3232371397624372 |
| Cluster-sea-mediaDesv-ponderada | **0,18292016146597373** | 0,3223639279049195 |
| Cluster-sea-pMedio-media | 0,19817888285967283 | 0,3227009711008877 |
| Cluster-sea-pMedio-ponderada | 0,20233315439792757 | 0,3227443913166889 |
| Knn-var-media-media | 0,3510756599179342 | 0,32097129707585925 |
| Knn-var-media-ponderada | 0,38017207563957545 | 0,31999159149743844 |
| Knn-var-mediaDesv-media | **0,15066257101545522** | 0,32482614145984956 |
| Knn-var-mediaDesv-ponderada | 0,21436002281954364 | 0,3220588368420982 |
| Knn-var-pMedio-media | 0,38568213969759213 | 0,3205690021967808 |
| Knn-var-pMedio-ponderada | 0,3965418266130926 | 0,32006146139126007 |
| Knn-ecm-media-media | **0,15349701000571342** | 0,3268563433149422 |
| Knn-ecm-media-ponderada | **0,14987399907087026** | 0,3231279225743354 |
| Knn-ecm-mediaDesv-media | 0,23649788497625987 | 0,3314423460147454 |
| Knn-ecm-mediaDesv-ponderada | **0,1694627861966863** | 0,3258166025354052 |
| Knn-ecm-pMedio-media | 0,22573275762496184 | 0,33013732043937344 |
| Knn-ecm-pMedio-ponderada | **0,16806474226795615** | 0,3250694311026581 |
| Knn-sea-media-media | **0,15305407074982105** | 0,3315665348797566 |
| Knn-sea-media-ponderada | **0,15275716963074848** | 0,32339162044557923 |
| Knn-sea-mediaDesv-media | 0,20387719874684587 | 0,3289480392629923 |
| Knn-sea-mediaDesv-ponderada | **0,1688769163761417** | 0,3244083379240075 |
| Knn-sea-pMedio-media | 0,1957898717249417 | 0,3326293481556807 |
| Knn-sea-pMedio-ponderada | **0,1715638798530022** | 0,3245873331049044 |
| Perfiles-var-media-media | 0,23365277164361564 | 0,31771226655170504 |
| Perfiles-var-media-ponderada | 0,29275252970001875 | 0,32008342551164815 |
| Perfiles-var-mediaDesv-media | **0,17329781182756637** | 0,32607004646332277 |
| Perfiles-var-mediaDesv-ponderada | 0,21384252933712178 | 0,3225463933251326 |
| Perfiles-var-pMedio-media | 0,33298976252677426 | 0,32485986010151646 |
| Perfiles-var-pMedio-ponderada | 0,3550793162646534 | 0,3225208827825198 |
| Perfiles-ecm-media-media | **0,17497606488565992** | 0,3222555665064824 |
| Perfiles-ecm-media-ponderada | **0,16690641332611705** | **0,31672663192081857** |
| Perfiles-ecm-mediaDesv-media | 0,22765654415441042 | 0,32491613473141695 |
| Perfiles-ecm-mediaDesv-ponderada | **0,1741438156238368** | 0,3198013923376231 |
| Perfiles-ecm-pMedio-media | 0,22931545687328625 | 0,3228963577332133 |
| Perfiles-ecm-pMedio-ponderada | **0,17584983490447645** | 0,3187336282035279 |
| Perfiles-sea-media-media | **0,1639649519804132** | 0,32130874399688275 |
| Perfiles-sea-media-ponderada | **0,16549729553250658** | **0,3151946493258864** |
| Perfiles-sea-mediaDesv-media | 0,19537291524409117 | 0,3264777820062583 |
| Perfiles-sea-mediaDesv-ponderada | **0,1720735671119084** | 0,3204511022850952 |
| Perfiles-sea-pMedio-media | 0,1940146584512379 | 0,32230424966538274 |
| Perfiles-sea-pMedio-ponderada | **0,17874055102738823** | **0,3171557750992178** |

Tabla. 5.7: 50-RMSE-dsel20

| | Dinámico | |
|---|---|---|
| **Método** | **Heterogéneo** | **Homogéneo** |
| Cluster-var-media-media | 14947,027204688802 | 26259,571000942447 |
| Cluster-var-media-ponderada | 15393,024700920185 | 25316,05164288101 |
| Cluster-var-mediaDesv-media | 14467,413224914771 | 25592,162635914447 |
| Cluster-var-mediaDesv-ponderada | 14640,24282224483 | 25332,46379259725 |
| Cluster-var-pMedio-media | 15571,73446282077 | 25333,876663089624 |
| Cluster-var-pMedio-ponderada | 15557,789848720047 | 25314,466527964523 |
| Cluster-ecm-media-media | 14580,08623191641 | 25809,488795399615 |
| Cluster-ecm-media-ponderada | 15940,826324190095 | 25632,112290553014 |
| Cluster-ecm-mediaDesv-media | 14574,317069554676 | 24845,63347507989 |
| Cluster-ecm-mediaDesv-ponderada | 14022,211565884634 | 25258,044586349824 |
| Cluster-ecm-pMedio-media | 16610,454631749806 | 25281,948469811578 |
| Cluster-ecm-pMedio-ponderada | 15194,377282403046 | 25411,078529336828 |
| Cluster-sea-media-media | 15563,894654539321 | 25462,772950377766 |
| Cluster-sea-media-ponderada | 16632,085154571127 | 25603,661322834912 |
| Cluster-sea-mediaDesv-media | 13090,981677623851 | 25207,545702714728 |
| Cluster-sea-mediaDesv-ponderada | 13796,84961961215 | 25361,174399370568 |
| Cluster-sea-pMedio-media | 17372,85525423401 | 25309,67426861621 |
| Cluster-sea-pMedio-ponderada | 17689,10980049115 | 25496,25967788374 |
| Knn-var-media-media | 25046,954698744135 | 25139,08602001086 |
| Knn-var-media-ponderada | 28379,009406417823 | 25125,407357514752 |
| Knn-var-mediaDesv-media | **10827,074614424779** | 25499,255712577127 |
| Knn-var-mediaDesv-ponderada | 15753,337414335325 | 25318,48338860932 |
| Knn-var-pMedio-media | 29381,808259199217 | 25184,3238638892 |
| Knn-var-pMedio-ponderada | 30343,582362362406 | 25180,454369219857 |
| Knn-ecm-media-media | **11823,025895957817** | 25519,26475525841 |
| Knn-ecm-media-ponderada | **11842,903614199084** | 25271,741478297565 |
| Knn-ecm-mediaDesv-media | 16901,736991471036 | 26616,85847984223 |
| Knn-ecm-mediaDesv-ponderada | **12622,045095664811** | 25732,634322453705 |
| Knn-ecm-pMedio-media | 17060,271511736013 | 26443,022785221743 |
| Knn-ecm-pMedio-ponderada | **12896,670772782094** | 25631,200078683538 |
| Knn-sea-media-media | **11723,330608553011** | 25835,485208372473 |
| Knn-sea-media-ponderada | **12159,206980958343** | 25374,03319036133 |
| Knn-sea-mediaDesv-media | 14196,322272273814 | 26049,138511623634 |
| Knn-sea-mediaDesv-ponderada | **12660,747946958927** | 25581,43924766776 |
| Knn-sea-pMedio-media | 14503,12467955409 | 25880,369809784905 |
| Knn-sea-pMedio-ponderada | **13370,445683388414** | 25396,357201907125 |
| Perfiles-var-media-media | 16511,739133978645 | 25090,35703183771 |
| Perfiles-var-media-ponderada | 21000,038784895212 | 25182,372538977383 |
| Perfiles-var-mediaDesv-media | **12325,103804657669** | 25629,706987129197 |
| Perfiles-var-mediaDesv-ponderada | 15516,104457049712 | 25342,198875563747 |
| Perfiles-var-pMedio-media | 20885,943532842706 | 25823,609653373773 |
| Perfiles-var-pMedio-ponderada | 23747,33004253555 | 25405,238165898758 |
| Perfiles-ecm-media-media | **13247,39076007859** | 25219,471056027676 |
| Perfiles-ecm-media-ponderada | **13226,910127768386** | **24298,21709** |
| Perfiles-ecm-mediaDesv-media | 15824,855455673645 | 25747,26808458125 |
| Perfiles-ecm-mediaDesv-ponderada | **13055,342694506748** | 25128,29125836062 |
| Perfiles-ecm-pMedio-media | 16512,6395819838 | 25354,6242483093 |
| Perfiles-ecm-pMedio-ponderada | **13407,051620895978** | 24955,572446811693 |
| Perfiles-sea-media-media | **12341,502119417579** | 25326,697152987323 |
| Perfiles-sea-media-ponderada | **13054,079505082633** | 24857,797646979234 |
| Perfiles-sea-mediaDesv-media | 14123,902427329152 | 25720,66460567376 |
| Perfiles-sea-mediaDesv-ponderada | **12938,32650120073** | 25187,68486455721 |
| Perfiles-sea-pMedio-media | 14633,263516513947 | 25153,480227429674 |
| Perfiles-sea-pMedio-ponderada | 14274,164636407711 | 24861,324635877787 |

errores de métodos que han funcionado peor. Diferenciamos cuatro picos. Esto podría estar sucediendo por el poco DSEL disponible que queda para definir la región de competencia, ya que recordemos que este datasets es pequeño.

Con respecto al ensemble homogéneo, los mejores métodos dinámicos han resultado ser *perfiles-sea-media-ponderada*, *perfiles-ecm-media-ponderada* y *perfiles-sea-pMedio - ponderada*, dando el mejor resultado **perfiles-ecm-media-ponderada**.

A diferencia del ensemble heterogéneo, en el caso del homogéneo no se han tenido resultados tan satisfactorios al utilizar los métodos dinámicos. Como se puede observar en la gráfica 5.2, la mayoría de los errores obtenidos por los métodos dinámicos se encuentran por encima del estático. Solo en tres ocasiones corta la recta de este. Sin embargo, el error más bajo de error lo consigue un método dinámico.

Como se puede observar, los resultados han sido muy ajustados. Esto se puede explicar debido a que el dataset sobre el que se está experimentando es muy pequeño. Esto supone que no se tengan muchos datos disponibles para entrenar a los regresores base cuando una parte de ellos es destinada al DSEL. En las siguientes pruebas podremos comprobar si el solapamiento y un aumneto del porcentaje de DSEL hace que funcionen mejor los métodos dinámicos.

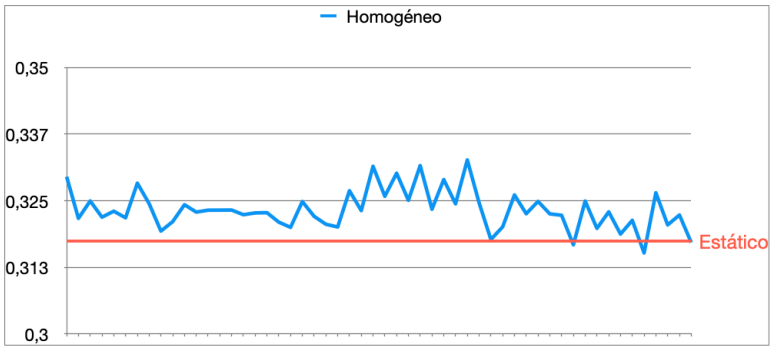


Figura 5.2: Gráfica MAPE con %DSEL = 20 HOMOGÉNEO Fuente: Elaboración Propia.

Destaca que, los mejores métodos dinámicos utilizan todos como calculo de la región de competencia el método de perfiles de salida y que en todos se agregan las salidas mediante media ponderada.

Por otro lado, la tabla 5.7, en la que se exponen los resultados RMSE de los métodos dinámicos implementados, presenta la misma estructura que la anterior. Lo único que cambia es el error con el que se está evaluando cada método representado en la tabla.

Si observamos los resultados obtenidos por los distintos métodos dinámicos con el ensemble dinámico heterogéneo recogidos en la tabla 5.7, los métodos que mejores resultados dan para este tipo de error son: *knn-var-mediaDesv-media*, *knn-sea-media-media* y *knn-ecm-media-media*. En concreto, el mejor es el obtenido por **knn-var-mediaDesv-media**.

Sin embargo, para el ensemble homogéneo, los mejores métodos son: *Perfiles-ecm-media-ponderada*, *Cluster-ecm-mediaDesv-media* y *Perfiles-sea-media-ponderada*.

Volvemos a ver en los resultados medidos con este error que, para estos casos concretos, con los parámetros y el dataset utilizados, los métodos que mejores resultados dan para el ensemble heterogéneo son los que calculan la región de competencia con KNN y para los homogéneos, la que la calculan mediante los perfiles de salida.

**Prueba 2**

En esta prueba, como se indica en la tabla 5.4, cambian dos parámetros con respecto a la prueba anterior. Concretmente: porcentaje de DSEL ahora es 50 y sí se permite el solapamiento con el conjunto de entrenamiento.

Los resultados de estas pruebas son presentados en las tablas 5.8 y 5.9, siguiendo exactamente el mismo esquema de presentación que en la prueba anterior: Ambas tablas presentan los errores obtenidos por los métodos dinámicos implementados para un ensemble heterogéneo y homogéneo, de forma separada. La única diferencia entre las tablas es el cálculo del error (MAPE y RMSE, respectivamente).

Para el análisis de resultados de esta prueba, se añaden gráficas similares a las del apartado anterior, ya que nos ayudan a darnos una visión general y rápida de cómo han mejorado o empeorado los métodos dinámicos al estático equivalente.

Gracias a la tabla 5.8, se observa que en 29 ocasiones, es decir, 29 de los métodos implementos, son capaces de ganar al ensemble estático. Esto supone un aumento con respecto a la prueba anterior.

Los tres mejores resultados son obtenidos por los siguientes métodos: *Perfiles-sea-media-ponderada*, *Perfiles-ecm-media-pond* y *Perfiles-sea-media-media*, siendo el mejor resultado el de **Perfiles-sea-media-ponderada**.

Tabla. 5.8: 50-MAPE-dsel50

| | Dinámico | |
|---|---|---|
| **Método** | **Heterogéneo** | **Homogéneo** |
| Cluster-var-media-media | 0,20721966506503472 | **0,3072920524808393** |
| Cluster-var-media-ponderada | 0,288317656363262 | **0,3073168343439929** |
| Cluster-var-mediaDesv-media | **0,15888125736168507** | **0,30831754478315776** |
| Cluster-var-mediaDesv-ponderada | **0,18993981576507182** | **0,30875310231708897** |
| Cluster-var-pMedio-media | 0,2186057021596691 | **0,30269975546456257** |
| Cluster-var-pMedio-ponderada | 0,23686281176371482 | **0,30590749462438993** |
| Cluster-ecm-media-media | **0,17083062717819744** | **0,308460192** |
| Cluster-ecm-media-ponderada | **0,16993165731303825** | **0,31053389405304566** |
| Cluster-ecm-mediaDesv-media | 0,205467671095895 | 0,318432984 |
| Cluster-ecm-mediaDesv-ponderada | **0,17689511153745563** | **0,3081682664888234** |
| Cluster-ecm-pMedio-media | 0,2399519206515246 | **0,30553153512667436** |
| Cluster-ecm-pMedio-ponderada | **0,1841150026181919** | **0,30868145994018087** |
| Cluster-sea-media-media | **0,17297881092715284** | **0,3030427273011346** |
| Cluster-sea-media-ponderada | **0,17340377242686067** | **0,30962795465925763** |
| Cluster-sea-mediaDesv-media | **0,18818307445291077** | **0,3065178427316186** |
| Cluster-sea-mediaDesv-ponderada | **0,17399930440699218** | **0,3096864577618751** |
| Cluster-sea-pMedio-media | 0,20622929215664315 | **0,30401743018630717** |
| Cluster-sea-pMedio-ponderada | **0,1846601111740796** | **0,30779652303282257** |
| Knn-var-media-media | 0,24117020786386645 | **0,30463924086187255** |
| Knn-var-media-ponderada | 0,2792700910573272 | 0,319854049 |
| Knn-var-mediaDesv-media | **0,16043303241710294** | **0,3066111189686137** |
| Knn-var-mediaDesv-ponderada | 0,19467376511229423 | **0,3055741066669005** |
| Knn-var-pMedio-media | 0,2626203726285686 | **0,3060691473335184** |
| Knn-var-pMedio-ponderada | 0,2862012183236714 | **0,30586913313294545** |
| Knn-ecm-media-media | **0,1605571421785621** | **0,3099735596042459** |
| Knn-ecm-media-ponderada | **0,15999464461841284** | **0,3095403851151365** |
| Knn-ecm-mediaDesv-media | 0,2159675771154094 | **0,3035487231428645** |
| Knn-ecm-mediaDesv-ponderada | **0,1695030720093152** | **0,30674515701297483** |
| Knn-ecm-pMedio-media | 0,24909831754068618 | **0,30435608289291133** |
| Knn-ecm-pMedio-ponderada | **0,17898881582458057** | **0,3064161555142097** |
| Knn-sea-media-media | **0,1590683318109438** | **0,30725875259357105** |
| Knn-sea-media-ponderada | **0,15985840772584764** | **0,30824010995390067** |
| Knn-sea-mediaDesv-media | 0,1931934454137585 | **0,309706552** |
| Knn-sea-mediaDesv-ponderada | **0,1692803487717343** | **0,30789052260969235** |
| Knn-sea-pMedio-media | 0,21177733022546344 | **0,30529656063796706** |
| Knn-sea-pMedio-ponderada | **0,17597318977315082** | 0,320529261 |
| Perfiles-var-media-media | 0,2427974004506205 | **0,3020359833366056** |
| Perfiles-var-media-ponderada | 0,2427974004506205 | **0,3020359833366056** |
| Perfiles-var-mediaDesv-media | **0,17787788573277372** | **0,3053861264059437** |
| Perfiles-var-mediaDesv-ponderada | 0,20252358167054757 | **0,314656076** |
| Perfiles-var-pMedio-media | 0,33664194563916416 | **0,31681544** |
| Perfiles-var-pMedio-ponderada | 0,3396576130176075 5 | 0,325695941 |
| Perfiles-ecm-media-media | **0,15914384820005534** | 0,31918859 |
| Perfiles-ecm-media-ponderada | **0,14857722460133974** | **0,3077726025111033** |
| Perfiles-ecm-mediaDesv-media | 0,2404361541330858 | **0,315891061** |
| Perfiles-ecm-mediaDesv-ponderada | **0,16759733482745634** | **0,30668240420256077** |
| Perfiles-ecm-pMedio-media | 0,25682812066145544 | **0,3088461011614465** |
| Perfiles-ecm-pMedio-ponderada | **0,17127888400239674** | **0,30679091021723875** |
| Perfiles-sea-media-media | **0,15253143030695995** | 0,31952908 |
| Perfiles-sea-media-ponderada | **0,14822026337749944** | 0,31902607 |
| Perfiles-sea-mediaDesv-media | 0,21092341842331147 | 0,318002778 |
| Perfiles-sea-mediaDesv-ponderada | **0,1638688026243465** | **0,3088186278301815** |
| Perfiles-sea-pMedio-media | 0,21193290848393326 | **0,30826262338392924** |
| Perfiles-sea-pMedio-ponderada | **0,16613625178006275** | 0,318337379 |

Tabla. 5.9: 50-RMSE-dsel50

| | Dinámico | |
|---|---|---|
| **Método** | **Heterogéneo** | **Homogéneo** |
| Cluster-var-media-media | 16596,976146133107 | 24727,690122516757 |
| Cluster-var-media-ponderada | 21194,13206005952 | 24502,339198828053 |
| Cluster-var-mediaDesv-media | **11472,866021173628** | 24563,60699530857 |
| Cluster-var-mediaDesv-ponderada | 14187,04244787468 | 24532,303153784702 |
| Cluster-var-pMedio-media | 17300,520558513108 | 24445,56227241993 |
| Cluster-var-pMedio-ponderada | 19328,578250916336 | **24303,70316076428** |
| Cluster-ecm-media-media | **13747,577479199806** | **24223,181661892664** |
| Cluster-ecm-media-ponderada | 15159,661407772282 | 24803,261808607083 |
| Cluster-ecm-mediaDesv-media | 14527,61906784576 | 24933,981646575307 |
| Cluster-ecm-mediaDesv-ponderada | 14193,467777213355 | 24431,415258536446 |
| Cluster-ecm-pMedio-media | 16448,464223993225 | 24376,92149590022 |
| Cluster-ecm-pMedio-ponderada | 14868,692532980844 | 24461,160819841298 |
| Cluster-sea-media-media | 14716,353205381401 | 24451,192790712423 |
| Cluster-sea-media-ponderada | 15706,869156409108 | 24761,808328279418 |
| Cluster-sea-mediaDesv-media | **13866,7084205742** | 24526,57439856443 |
| Cluster-sea-mediaDesv-ponderada | **14033,35596319139** | 24689,220273677965 |
| Cluster-sea-pMedio-media | 16066,221634657364 | **24317,191108924562** |
| Cluster-sea-pMedio-ponderada | 15710,087848040901 | 24438,51896157812 |
| Knn-var-media-media | 18652,43584269771 | **24045,429275917508** |
| Knn-var-media-ponderada | 21117,509274272274 | 24987,37889 |
| Knn-var-mediaDesv-media | **11867,933914598994** | 24383,042054730533 |
| Knn-var-mediaDesv-ponderada | 14438,111758944497 | **24230,076880155706** |
| Knn-var-pMedio-media | 20782,321718146042 | **24238,55532718951** |
| Knn-var-pMedio-ponderada | 22090,44772898395 | **24215,639690477296** |
| Knn-ecm-media-media | **13384,853094440952** | 24820,886758270608 |
| Knn-ecm-media-ponderada | **13526,776568885427** | 24508,903363309724 |
| Knn-ecm-mediaDesv-media | 16226,37091935133 | **23964,823528137** |
| Knn-ecm-mediaDesv-ponderada | **13630,691546309588** | **24281,459382771074** |
| Knn-ecm-pMedio-media | 17932,774655871584 | **24016,335472755152** |
| Knn-ecm-pMedio-ponderada | 14223,749823438658 | **24242,49519692733** |
| Knn-sea-media-media | **12769,307155079145** | 24490,50131521868 |
| Knn-sea-media-ponderada | **13128,176308897633** | **24366,378456943938** |
| Knn-sea-mediaDesv-media | 14638,252252107977 | **24214,2074032316** |
| Knn-sea-mediaDesv-ponderada | **13434,619873173378** | **24326,739435517076** |
| Knn-sea-pMedio-media | 16047,518469871371 | **24183,961761336985** |
| Knn-sea-pMedio-ponderada | 14196,256234976978 | 24385,08144 |
| Perfiles-var-media-media | 18020,325917527025 | **24015,580831347943** |
| Perfiles-var-media-ponderada | 18020,325917527025 | **24015,580831347943** |
| Perfiles-var-mediaDesv-media | **12840,504960770159** | **24013,163234834567** |
| Perfiles-var-mediaDesv-ponderada | 14877,285342720064 | **24045,033070986934** |
| Perfiles-var-pMedio-media | 22770,086647884913 | **24114,512568741968** |
| Perfiles-var-pMedio-ponderada | 23703,12611950174 | **24094,45791134235** |
| Perfiles-ecm-media-media | **12280,491501399509** | 24428,367050914472 |
| Perfiles-ecm-media-ponderada | **12028,695707475212** | **24264,785737760354** |
| Perfiles-ecm-mediaDesv-media | 16628,291148349137 | **24109,285040444007** |
| Perfiles-ecm-mediaDesv-ponderada | **12997,905708658422** | **24176,294842256622** |
| Perfiles-ecm-pMedio-media | 18128,553188364844 | 24607,706906943047 |
| Perfiles-ecm-pMedio-ponderada | **13327,024872651094** | **24273,648414547464** |
| Perfiles-sea-media-media | **12066,21770564939** | 24921,90624 |
| Perfiles-sea-media-ponderada | **11968,951413279461** | **24340,75048056104** |
| Perfiles-sea-mediaDesv-media | 15008,967110935273 | 24525,693861637665 |
| Perfiles-sea-mediaDesv-ponderada | **12857,201691306345** | 24417,55053573337 |
| Perfiles-sea-pMedio-media | 15322,890896421852 | 24392,011235686437 |
| Perfiles-sea-pMedio-ponderada | **13057,130567453492** | **24337,899073853234** |

Parece que con estos nuevos parámetros han resultado mejores los métodos que calculan la región de competencia mediante perfiles de salida parra ensembles heterogéneos, a diferencia de la prueba anterior.

Además, observando la gráfica 5.3, podemos que para el ensemble dinámico heterogéneo, los métodos dan resultados menos dispersos entre sí en esta prueba y que los picos observados en la prueba anterior, se han achatado, significando una mejora de dichos métodos.

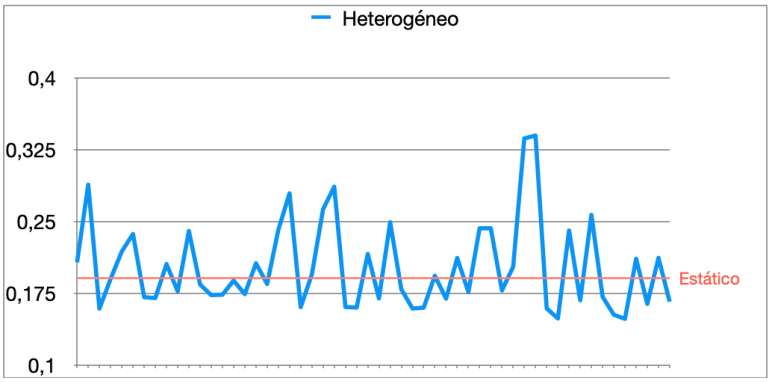


Figura 5.3: Gráfica MAPE con %DSEL = 50 HETEROGÉNEO Fuente: Elaboración Propia.

Con respecto al ensemble homogéneo, encontramos un gran mejora con respecto al experimento anterior, ya que solo tres eran los que conseguían mejores resultados que el estático y ahora son 45 métodos los que ganan al estático.

Precisamente, los tres mejores han resultaados ser: *Perfiles-var-media-media*, *Perfiles-var-media-ponderada* y *Cluster-var-pMedio-media*. Siendo **Perfiles-var-media-media** el método con error MAPE más bajo.

Si nos fijamos en la gráfica 5.4, vemos, de manera muy clara, cómo ahora prácticamente todos se encuentran por debajo del error obtenido con el ensemble estático.

Con respecto al RMSE, vemos en la tabla 5.9, que para el ensemble heterogéneo se obtienen los mejores resultados con los métodos *Cluster-var-mediaDesv-media*, *Knn-var-mediaDesv-media* y *Perfiles-sea-media-ponderada*, dando el primero el mejor resultado (**Cluster-var-mediaDesv-media**).

En esta misma tabla 5.9, los tres mejores resultados obtenidos para los ensembles homogéneos han sido los pertenecientes a los métodos dinámicos implementados siguientes:

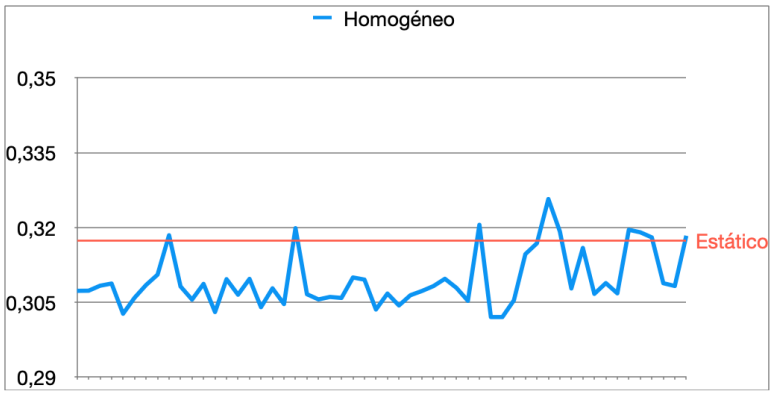


Figura 5.4: Gráfica MAPE con %DSEL = 50 HOMOGÉNEO Fuente: Elaboración Propia.

*Knn-ecm-mediaDesv-media*, *Perfiles-var-mediaDesv-media* y *Perfiles-var-media-media*, siendo **Knn-ecm-mediaDesv-media** el mejor, coincidiendo en la selección de mejores regresores (media más desviación típica) y en la agregación (media aaritmética), con el mejor del ensemble heterrogéneo para este tipo de error.

**Prueba 3**

En esta última prueba, se han realizado de nuevo cambios en los parámetros que controlan el porcentaje de DSEL y el solapamiento entre el conjunto de entrenamiento y DSEL. Específicamente, el porcentaje de DSEL se ha configurado al 95 % y sí se ha permitido solapamiento (como se indica en la tabla 5.4)

Los resultados adquiridos de esta prueba son los que se presentan en las tablas 5.10 y 5.11, con la misma estructura que en las dos pruebas anteriores. Cada tabla representa un tipo de error: MAPE y RMSE, respectivamente.

Si nos fijamos en la tabla 5.10, vemos que el MAPE más bajo para el ensemble heterogéneo es el obtenido por el método ***Perfiles-sea-media-pond***, al igual que en la prueba anterior.

Los dos siguientes mejores métodos para el ensemble dinámico heterogéneo resultan ser *Perfiles-ecm-media-ponderada*, *Perfiles-sea-media-media*, coincidiendo de nuevo en el segundo mejor método con la prueba anterior para el ensemble heterogéneo.

Sin embargo, si observamos la gráfica 5.5, vemos que a diferencia de la prueba anterior,

Tabla. 5.10: 50-MAPE-dsel95

| | Dinámico | |
|---|---|---|
| **Método** | **Heterogéneo** | **Homogéneo** |
| Cluster-var-media-media | 0,264535820029815 | 0,32344804595888255 |
| Cluster-var-media-ponderada | 0,31365849528897416 | **0,311094072884177** |
| Cluster-var-mediaDesv-media | **0,1519299306899282** | **0,30765178650901703** |
| Cluster-var-mediaDesv-ponderada | 0,20004129958095035 | **0,3080402281246428** |
| Cluster-var-pMedio-media | 0,2855978717708765 | **0,30884625535925025** |
| Cluster-var-pMedio-ponderada | 0,3134567826566057 | **0,3092854998584464** |
| Cluster-ecm-media-media | **0,15399058306602004** | **0,31375803515651357** |
| Cluster-ecm-media-ponderada | **0,16543484311426723** | **0,3120282753176213** |
| Cluster-ecm-mediaDesv-media | 0,24303622347419312 | **0,30890938079294583** |
| Cluster-ecm-mediaDesv-ponderada | **0,16908696050783642** | **0,31102179973328464** |
| Cluster-ecm-pMedio-media | 0,25407464410558583 | **0,2991311889810849** |
| Cluster-ecm-pMedio-ponderada | **0,17414420456897808** | **0,30868303214609644** |
| Cluster-sea-media-media | **0,14783847784242743** | **0,30023927243538557** |
| Cluster-sea-media-ponderada | **0,15255766312608102** | **0,30986085791786955** |
| Cluster-sea-mediaDesv-media | 0,19303370214591256 | **0,3113414748941997** |
| Cluster-sea-mediaDesv-ponderada | **0,16714979640766917** | **0,30982467086413046** |
| Cluster-sea-pMedio-media | 0,2067273288296562 | **0,3086852661211398** |
| Cluster-sea-pMedio-ponderada | **0,1718959242941087** | **0,3097714483890765** |
| Knn-var-media-media | 0,2170983033619942 | **0,30880187793231884** |
| Knn-var-media-ponderada | 0,2633153436340764 | **0,31160628943152907** |
| Knn-var-mediaDesv-media | **0,16133035571941962** | **0,30615859608014634** |
| Knn-var-mediaDesv-ponderada | 0,19442615063029905 | **0,30905230864811045** |
| Knn-var-pMedio-media | 0,2337744220844948 | **0,3070417510324942** |
| Knn-var-pMedio-ponderada | 0,26952126506976637 | **0,31055595662149027** |
| Knn-ecm-media-media | **0,15712959411470245** | **0,31326273584468844** |
| Knn-ecm-media-ponderada | **0,1571157486771917** | **0,31257650909229573** |
| Knn-ecm-mediaDesv-media | 0,20609517529784066 | **0,3099948387140745** |
| Knn-ecm-mediaDesv-ponderada | **0,1671079600712039** | **0,31143695737954913** |
| Knn-ecm-pMedio-media | 0,23581827418732945 | **0,3017639359043303** |
| Knn-ecm-pMedio-ponderada | **0,17532243380493778** | **0,3094079970936252** |
| Knn-sea-media-media | **0,15951937909785527** | **0,31253794992022554** |
| Knn-sea-media-ponderada | **0,1613910786624577** | **0,31119629104433494** |
| Knn-sea-mediaDesv-media | **0,1819354814430235** | **0,30013934041703834** |
| Knn-sea-mediaDesv-ponderada | **0,16719981276618756** | **0,3119338207297031** |
| Knn-sea-pMedio-media | 0,19460514305849727 | **0,30858049507864255** |
| Knn-sea-pMedio-ponderada | **0,17232415609179158** | **0,31063326000034347** |
| Perfiles-var-media-media | 0,22559199200066252 | **0,30970516304627604** |
| Perfiles-var-media-ponderada | 0,2445243624358837 | **0,3127561991319199** |
| Perfiles-var-mediaDesv-media | **0,1694271971156694** | **0,30608545415328364** |
| Perfiles-var-mediaDesv-ponderada | **0,1851709545019935** | **0,3077686280523663** |
| Perfiles-var-pMedio-media | 0,24493791937683343 | **0,30562359004376394** |
| Perfiles-var-pMedio-ponderada | 0,26340684372011636 | **0,30900303769752474** |
| Perfiles-ecm-media-media | **0,1410368639074758** | **0,3126829058905315** |
| Perfiles-ecm-media-ponderada | **0,13626815380151774** | **0,3090209430399336** |
| Perfiles-ecm-mediaDesv-media | 0,24538224627004496 | **0,3100110797610328** |
| Perfiles-ecm-mediaDesv-ponderada | **0,15836874428945252** | **0,3088172707112973** |
| Perfiles-ecm-pMedio-media | 0,2616144290456287 | **0,3071976105872375** |
| Perfiles-ecm-pMedio-ponderada | **0,16253379167437218** | **0,30867049652375805** |
| Perfiles-sea-media-media | **0,1386375289434091** | **0,30115490807502** |
| Perfiles-sea-media-ponderada | **0,13580879504242319** | **0,30758414423069086** |
| Perfiles-sea-mediaDesv-media | 0,21680060995655434 | **0,31382897644031527** |
| Perfiles-sea-mediaDesv-ponderada | **0,15715622899124723** | **0,3105769612646796** |
| Perfiles-sea-pMedio-media | 0,21103230333618367 | **0,3019125922349675** |
| Perfiles-sea-pMedio-ponderada | **0,153998233910019** | **0,3097237631865257** |

Tabla. 5.11: 50-RMSE-dsel95

| | Dinámico | |
|---|---|---|
| **Método** | **Heterogéneo** | **Homogéneo** |
| Cluster-var-media-media | 19794,32851010975 | 26003,635575478304 |
| Cluster-var-media-ponderada | 23520,412077927398 | 24900,034581370586 |
| Cluster-var-mediaDesv-media | **10947,615490344457** | 24501,61455626198 |
| Cluster-var-mediaDesv-ponderada | 14712,874568765339 | 24541,133112153842 |
| Cluster-var-pMedio-media | 18708,380302878995 | 25129,848518880102 |
| Cluster-var-pMedio-ponderada | 21970,39490695364 | 24670,501603184406 |
| Cluster-ecm-media-media | **12776,774479793448** | 25302,61234659113 |
| Cluster-ecm-media-ponderada | **13372,691758281337** | 24985,66879487913 |
| Cluster-ecm-mediaDesv-media | 16941,55006113684 | 24424,01531110498 |
| Cluster-ecm-mediaDesv-ponderada | **13124,577432092517** | 24819,01556875347 |
| Cluster-ecm-pMedio-media | 18402,926125198584 | 23694,269125833984 |
| Cluster-ecm-pMedio-ponderada | **13468,43394172956** | 24609,45382559502 |
| Cluster-sea-media-media | **12646,981020718018** | 24786,754522147698 |
| Cluster-sea-media-ponderada | **13110,65874741696** | 24837,991934074475 |
| Cluster-sea-mediaDesv-media | 14656,15742283851 | 24943,264669237076 |
| Cluster-sea-mediaDesv-ponderada | **13127,948636161804** | 24725,396290576457 |
| Cluster-sea-pMedio-media | 14842,515122086785 | 24594,018376092594 |
| Cluster-sea-pMedio-ponderada | **13386,093561416588** | 24747,364475596747 |
| Knn-var-media-media | 17011,694074643565 | 24717,51572638019 |
| Knn-var-media-ponderada | 20098,544308304794 | 24830,09809302493 |
| Knn-var-mediaDesv-media | **11849,010333273614** | **24343,06893170702** |
| Knn-var-mediaDesv-ponderada | 14378,650398380309 | 24542,070952265378 |
| Knn-var-pMedio-media | 17959,13817464513 | 24677,054600246174 |
| Knn-var-pMedio-ponderada | 20311,538387548022 | 24921,915376171288 |
| Knn-ecm-media-media | **12320,034779287374** | 24993,1027783464 |
| Knn-ecm-media-ponderada | **12598,421150322505** | 25025,0763413743 |
| Knn-ecm-mediaDesv-media | 15476,375708834657 | 24584,102608347173 |
| Knn-ecm-mediaDesv-ponderada | **13025,758060608552** | 24797,712762760682 |
| Knn-ecm-pMedio-media | 17074,26228560652 | **23787,355824584974** |
| Knn-ecm-pMedio-ponderada | **13419,062129842432** | 24574,712483526655 |
| Knn-sea-media-media | **12709,686312171998** | 25197,46516867155 |
| Knn-sea-media-ponderada | **13080,378422838849** | 25009,58661226048 |
| Knn-sea-mediaDesv-media | **13813,160622487712** | 24837,74815094215 |
| Knn-sea-mediaDesv-ponderada | **13172,785813214588** | 24847,09151457469 |
| Knn-sea-pMedio-media | 14808,620059264294 | 24478,02204121367 |
| Knn-sea-pMedio-ponderada | **13596,892181350733** | 24723,382019253324 |
| Perfiles-var-media-media | 16519,004922710494 | 24437,17602528335 |
| Perfiles-var-media-ponderada | 17749,7011221355 | 24716,97987547246 |
| Perfiles-var-mediaDesv-media | **12899,429532355374** | **24175,77802853944** |
| Perfiles-var-mediaDesv-ponderada | **14033,941894583288** | **24344,98867370623** |
| Perfiles-var-pMedio-media | 19332,62643070653 | **24190,290462713503** |
| Perfiles-var-pMedio-ponderada | 20495,105207609984 | 24563,059802723357 |
| Perfiles-ecm-media-media | **10895,825223536089** | 24466,438010693404 |
| Perfiles-ecm-media-ponderada | **10399,029613522089** | 24422,916982754738 |
| Perfiles-ecm-mediaDesv-media | 17463,85292966651 | **24371,372242205212** |
| Perfiles-ecm-mediaDesv-ponderada | **12004,03879849413** | **24367,312954075613** |
| Perfiles-ecm-pMedio-media | 18430,465112547845 | **24198,539788850245** |
| Perfiles-ecm-pMedio-ponderada | **12166,346508532433** | **24372,509351625158** |
| Perfiles-sea-media-media | **10356,961946528283** | 24586,225609448076 |
| Perfiles-sea-media-ponderada | **10164,064564370603** | 24483,728534045735 |
| Perfiles-sea-mediaDesv-media | 15411,542139465464 | 24661,54649664893 |
| Perfiles-sea-mediaDesv-ponderada | **12083,858987006588** | 24548,254132954025 |
| Perfiles-sea-pMedio-media | 14351,407068237764 | 24394,04706086321 |
| Perfiles-sea-pMedio-ponderada | **11515,864346403338** | 24441,073727577404 |

el peor error obtenido por un método dinámico es más bajo (mejor) que el peor de la prueba anterior y el mejor error es más bajo en esta ocasión que en la prueba anterior del MAPE para ensembles dinámicos heterogéneos.

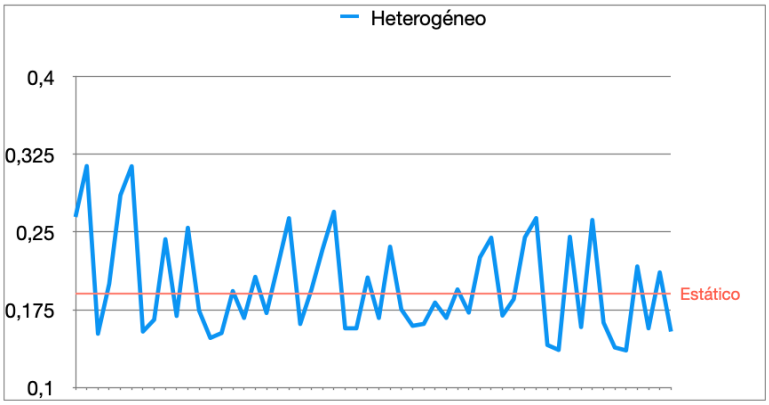


Figura 5.5: Gráfica MAPE con %DSEL = 95 HETEROGÉNEO Fuente: Elaboración Propia.

Con respecto a los resultados obtenidos para los ensembles homogéneos en la tabla 5.10, notamos que, en todos, excepto uno, de los métodos implementados (dinámicos) se obtienen mejores resultados que el estático.

Los mejores métodos en este caso son *Cluster-ecm-pMedio-media*, *Knn-sea-mediaDesv-media* y *Cluster-sea-media-media*, siendo **Cluster-ecm-pMedio-media** el que mejor resultadaos ha obtenido.

En la gráfica 5.6, se observa cómo los distintos métodos mejoran al estático casi siempre. Además, con esta configuración de parámetros se obtiene el mínimo error MAPE en ensembles homogénos probados en este dataset, es decir, en esta prueba se obtiene el menor valor de error para ensembles homogéneos con respecto a todas las pruebas anteriores de ensembles homogéneos.

Por último, cabe destacar que en la tabla 5.11 donde se representa el error RMSE obtenido por cada método, para el ensemble heterogéneo se obtienen los mejores resultados con los métodos *Perfiles-sea-media-ponderada*, *Perfiles-sea-media-media* y *Perfiles-ecm-media-ponderada*, mientras que para el homogéneo, se obtienen los mejores resultados con *Cluster-ecm-pMedio-media*, *Knn-ecm-pMedio-media* y *Perfiles-var-mediaDesv-media*.

Concretamente, el mejor método para el heterogéneo es **Perfiles-sea-media-ponderada** y para el homogéneo es **Cluster-ecm-pMedio-media**

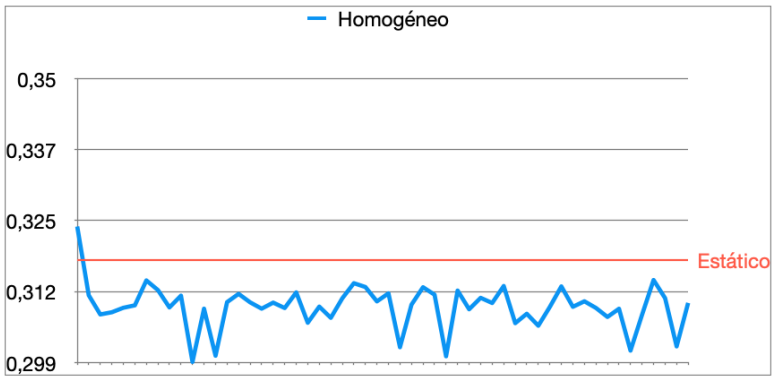


Figura 5.6: Gráfica MAPE con %DSEL = 50 HOMOGÉNEO Fuente: Elaboración Propia.

**Conclusiones del experimento 1**

Como conclusiones sacadas de este experimento, una de las más clara es que el dataset es muy pequeño, y en estos casos, un porcentaje de DSEL bajo no genera buenos resultados. Vemos durante las pruebas que a medida que el porcentaje de DSEL aumenta, para los ensembles homogéneos suponen una mejoría muy notoria. El ensemble heterogéneo también mejora cuando se incrementa el porcentaje de DSEL y se utiliza solapamiento.

Los resultados de las mejores puntuaciones de todas las pruebas) son: 0,135808 para el heterogéneo (frente a 0,19063 que es el error MAPE del estático heterogéneo) y 0,29913 para el homogéneo (frente a su versión estática que comete un MAPE de 0,317424553).

Los mejores resultados MAPE obtenidos por ambos ensembles (heterogéneo y homogéneo) se han conseguido durante el experimento número 3.

Por lo tanto, se puede concluir en que la mejor combinación de parámetros para ambos casos de ensembles es el porcentaje de DSEL al 95 % y permitiendo solapamiento.

### 5.2.2. Experimento 2

Durante esta sección, se exponen y analizan las pruebas realizadas con el dataset **Students-marks** siguiendo la misma metodología que en el experimento anterior.

Los resultados de error que consiguen los ensembles estáticos son lo que se muestran en la tabla 5.12. Con estos resultados se comparan los métodos implementados para ensembles dinámicos en las siguientes pruebas.

Tabla. 5.12: Estáticos SM

| | **Estático** | |
|---|---|---|
| **Método** | **Heterogéneo** | **Homogéneo** |
| MAPE | 0,130087382 | 0,057850645 |
| RMSE | 2,682719562 | 1,576180889 |

**Prueba 1**

Recordemos que durante esta prueba, el porcentaje del conjunto de entrenamiento destinado al DSEL es 20 % sin permitir solapamiento.

Las tablas 5.13 y 5.14 presentan los resultados obtenidos durante esta prueba medidos con el error MAPE y el RMSE, respectivamente.

Si observamos la gráfica 5.7, que representa de manera orientativa los errores obtenidos por cada método en comparación con el error obtenido por el estático (línea roja) para los ensembles heterogéneos, observamos que resultados de los métodos dinámicos se encuentran en su mayoría por debajo del obtenido por su equivalente ensemble estático. Esto significa, que en términos generales funciona mejor si el ensemble utiliza Selección Dinámica.

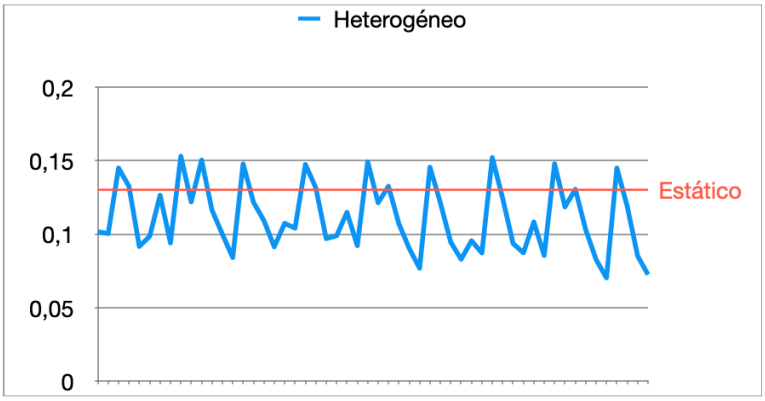


Figura 5.7: Gráfica MAPE con %DSEL = 20 HETEROGÉNEO Fuente: Elaboración Propia.

Estudiando la tabla 5.13, nos damos cuenta de que son 41 métodos dinámicos en ensembles heterogéneos los que mejoran el error MAPE del ensemble estático heterogéneo. El error más bajo lo consigue ***perfiles-sea-media-ponderada***, seguido de *perfiles-sea-pMedio-ponderada* y *knn-sea-media-ponderada*. Cabe destacar que los mejores métodos para este caso coinciden en que para el cálculo de los niveles de competencia se utiliza el error SEA ponderado (Véase Sección 4.1.2) y la media ponderada para la agregación de las salidas.

Tabla. 5.13: SM-MAPE-dsel20

| | Dinámico | |
|---|---|---|
| **Método** | **Heterogéneo** | **Homogéneo** |
| Cluster-var-media-media | **0,10158579455391688** | 0,06735646611899855 |
| Cluster-var-media-ponderada | **0,10037692632081643** | 0,07117965413570215 |
| Cluster-var-mediaDesv-media | 0,14490554877577222 | 0,07176508723703796 |
| Cluster-var-mediaDesv-ponderada | **0,1324703073131704** | 0,0710031539052219 |
| Cluster-var-pMedio-media | **0,09161356119822321** | 0,07019009625519654 |
| Cluster-var-pMedio-ponderada | **0,09849462307930705** | 0,07084826249027495 |
| Cluster-ecm-media-media | **0,12637720002750255** | 0,06745189030621443 |
| Cluster-ecm-media-ponderada | **0,09401038851767708** | 0,06219754224769201 |
| Cluster-ecm-mediaDesv-media | 0,1529156681466771 | 0,07512200139639087 |
| Cluster-ecm-mediaDesv-ponderada | **0,12187846548156453** | 0,06609493468286878 |
| Cluster-ecm-pMedio-media | 0,15027631052579332 | 0,07671727587432163 |
| Cluster-ecm-pMedio-ponderada | **0,11624704702708086** | 0,06514347286939662 |
| Cluster-sea-media-media | **0,09990580127623948** | 0,0644840534540217 2 |
| Cluster-sea-media-ponderada | **0,08404786243862619** | 0,06157921595640664 |
| Cluster-sea-mediaDesv-media | 0,1476208823062724 | 0,07273579743302486 |
| Cluster-sea-mediaDesv-ponderada | **0,12152112022930735** | 0,06464312212855959 |
| Cluster-sea-pMedio-media | **0,10919986229009322** | 0,06664404911822219 |
| Cluster-sea-pMedio-ponderada | **0,09136918745529232** | 0,062225177850069535 |
| Knn-var-media-media | **0,1072811179333684** | 0,07099023758797071 |
| Knn-var-media-ponderada | **0,10401412147569518** | 0,07016302731887955 |
| Knn-var-mediaDesv-media | 0,14729739377420364 | 0,07144173572766972 |
| Knn-var-mediaDesv-ponderada | 0,1314270862152389 | 0,06981590537957442 |
| Knn-var-pMedio-media | **0,09690738457248388** | 0,07192124275632811 |
| Knn-var-pMedio-ponderada | **0,09883394807126358** | 0,07018456807362992 |
| Knn-ecm-media-media | **0,11477950708772089** | 0,06427189770058454 |
| Knn-ecm-media-ponderada | **0,0921390608685211** | 0,06151427441841171 |
| Knn-ecm-mediaDesv-media | 0,14911784220544183 | 0,07534524817153788 |
| Knn-ecm-mediaDesv-ponderada | **0,1212973355981265** | 0,0669049698582416 |
| Knn-ecm-pMedio-media | 0,13247522229789815 | 0,07503073706691692 |
| Knn-ecm-pMedio-ponderada | **0,10703997805668442** | 0,06588223164878296 |
| Knn-sea-media-media | **0,09024555184066255** | 0,062231555949490966 |
| Knn-sea-media-ponderada | **0,07681023724124628** | 0,060374948372280454 |
| Knn-sea-mediaDesv-media | 0,14544626492945645 | 0,07198953984928788 |
| Knn-sea-mediaDesv-ponderada | **0,12117232570832548** | 0,0647081970047269 |
| Knn-sea-pMedio-media | **0,09445200418901924** | 0,06517255620005773 |
| Knn-sea-pMedio-ponderada | **0,0829025079975196** | 0,06198609937884057 |
| Perfiles-var-media-media | **0,09544403273970906** | 0,07230571679027471 |
| Perfiles-var-media-ponderada | **0,08714793569160309** | 0,07189227631484449 |
| Perfiles-var-mediaDesv-media | 0,15200625253294237 | 0,07077564686785125 |
| Perfiles-var-mediaDesv-ponderada | **0,1242624783196882** | 0,07064854145487008 |
| Perfiles-var-pMedio-media | **0,09359143007145565** | 0,06873222716100805 |
| Perfiles-var-pMedio-ponderada | **0,08725903841146414** | 0,06959692588056698 |
| Perfiles-ecm-media-media | **0,10826450544306328** | 0,06398906253110807 |
| Perfiles-ecm-media-ponderada | **0,08545132218218741** | 0,06169367914006933 |
| Perfiles-ecm-mediaDesv-media | 0,14775203573509615 | 0,07413753655517868 |
| Perfiles-ecm-mediaDesv-ponderada | **0,11847222880668395** | 0,0665619349261698 |
| Perfiles-ecm-pMedio-media | 0,13042770487880656 | 0,07362362401018552 |
| Perfiles-ecm-pMedio-ponderada | **0,10296878176608126** | 0,06604366556378449 |
| Perfiles-sea-media-media | **0,0826317093155087** | 0,06182248493349864 |
| Perfiles-sea-media-ponderada | **0,0702468934228606** | 0,0601508671652345 |
| Perfiles-sea-mediaDesv-media | 0,14487342769516848 | 0,07022214556988526 |
| Perfiles-sea-mediaDesv-ponderada | **0,11877196377150981** | 0,06425179054763065 |
| Perfiles-sea-pMedio-media | **0,08512811213475316** | 0,06513483204809374 |
| Perfiles-sea-pMedio-ponderada | **0,07243165189777186** | 0,06190295168954979 |

Tabla. 5.14: SM-RMSE-dsel20

| | Dinámico | |
|---|---|---|
| **Método** | **Heterogéneo** | **Homogéneo** |
| Cluster-var-media-media | 2,6872412648359187 | 1,8488123510450714 |
| Cluster-var-media-ponderada | 2,817663684740876 | 1,8710361117710548 |
| Cluster-var-mediaDesv-media | 2,950544547754414 | 1,9700470780812034 |
| Cluster-var-mediaDesv-ponderada | 2,8778727931485886 | 1,8626170883083826 |
| Cluster-var-pMedio-media | **2,401921074103841** | 1,9910821698597263 |
| Cluster-var-pMedio-ponderada | **2,6758847166212103** | 1,8652730718764423 |
| Cluster-ecm-media-media | 2,72852591747784 | 1,8028160067345467 |
| Cluster-ecm-media-ponderada | **2,2792641037687438** | 1,697396033256376 |
| Cluster-ecm-mediaDesv-media | 3,1785893466308743 | 2,009375721250527 |
| Cluster-ecm-mediaDesv-ponderada | **2,6365285766167412** | 1,7520009650030068 |
| Cluster-ecm-pMedio-media | 3,1782203037562176 | 2,0063131818348756 |
| Cluster-ecm-pMedio-ponderada | **2,5978800733219467** | 1,7412205203895408 |
| Cluster-sea-media-media | **2,3982880675004923** | 1,7504677021197996 |
| Cluster-sea-media-ponderada | **2,1672627504685673** | 1,68897100170769 |
| Cluster-sea-mediaDesv-media | 3,040112449372492 | 1,861427424942849 |
| Cluster-sea-mediaDesv-ponderada | **2,6187081517810205** | 1,7229235396892464 |
| Cluster-sea-pMedio-media | **2,5612729932472957** | 1,8341002840022007 |
| Cluster-sea-pMedio-ponderada | **2,2595880740323024** | 1,7060809877554797 |
| Knn-var-media-media | 2,862309645855931 | 1,8617546026591245 |
| Knn-var-media-ponderada | 2,888465884881767 | 1,8446170442477061 |
| Knn-var-mediaDesv-media | 2,9866265624144726 | 1,9297413615692853 |
| Knn-var-mediaDesv-ponderada | 2,9594191030408514 | 1,8617883703133777 |
| Knn-var-pMedio-media | **2,599777084314913** | 1,9012323882026512 |
| Knn-var-pMedio-ponderada | 2,7594454515260955 | 1,8623628595868589 |
| Knn-ecm-media-media | **2,588286183871367** | 1,7769003759864272 |
| Knn-ecm-media-ponderada | **2,2691464602632405** | 1,7159840173056948 |
| Knn-ecm-mediaDesv-media | 3,0856279519378464 | 1,9575388499207995 |
| Knn-ecm-mediaDesv-ponderada | **2,6353814017132295** | 1,7649110244983457 |
| Knn-ecm-pMedio-media | 2,9830808685595125 | 1,987830646477029 |
| Knn-ecm-pMedio-ponderada | **2,5321668901719847** | 1,7780468938681508 |
| Knn-sea-media-media | **2,2612081698699287** | 1,7252826729385966 |
| Knn-sea-media-ponderada | **2,0845721065259646** | 1,6875461433977315 |
| Knn-sea-mediaDesv-media | 2,99821865629585 | 1,8689348846807257 |
| Knn-sea-mediaDesv-ponderada | **2,615332991876259** | 1,7287564926191006 |
| Knn-sea-pMedio-media | **2,436776973057269** | 1,811832782022811 |
| Knn-sea-pMedio-ponderada | **2,253006105567247** | 1,713715567869243 |
| Perfiles-var-media-media | 2,707873262703213 | 1,9246295334775543 |
| Perfiles-var-media-ponderada | **2,6383894115390527** | 1,932434574653378 |
| Perfiles-var-mediaDesv-media | 3,067172212250413 | 1,8860030341071892 |
| Perfiles-var-mediaDesv-ponderada | 2,844428240516137 | 1,8803211311400605 |
| Perfiles-var-pMedio-media | **2,5500702304635494** | 1,8829664648413698 |
| Perfiles-var-pMedio-ponderada | **2,5668111781025047** | 1,9129609020677627 |
| Perfiles-ecm-media-media | **2,5083138529417544** | 1,7738648175781848 |
| Perfiles-ecm-media-ponderada | **2,2016550159247745** | 1,7305011092828306 |
| Perfiles-ecm-mediaDesv-media | 3,055045604333233 | 1,9418794178763759 |
| Perfiles-ecm-mediaDesv-ponderada | **2,6040776456522297** | 1,7647328114366112 |
| Perfiles-ecm-pMedio-media | 2,9523029471259017 | 1,9838511367783176 |
| Perfiles-ecm-pMedio-ponderada | **2,491551015853626** | 1,7820757608709463 |
| Perfiles-sea-media-media | **2,1961127804376153** | 1,7209346959765397 |
| Perfiles-sea-media-ponderada | **2,016853999451857** | 1,6927792000613358 |
| Perfiles-sea-mediaDesv-media | 2,9816154536048254 | 1,8324512461713218 |
| Perfiles-sea-mediaDesv-ponderada | **2,5839267799659256** | 1,717407117138063 |
| Perfiles-sea-pMedio-media | **2,327866595356977** | 1,8085951541678322 |
| Perfiles-sea-pMedio-ponderada | **2,1413777831601797** | 1,7075601593042442 |

Por otro lado, con respecto a los resultados obtenidos de MAPE con el ensemble homogéneo, recogidos en la tabla 5.13, vemos que en los ensembles homogéneos dinámicos no consiguen mejorar el error obtenido por su equivalente estático. Si observamos la gráfica 5.8 vemos de manera más clara que todos los métodos dinámicos obtienen peores (más altos) resultados de error MAPE. Las razones de que no se gane al estático pueden ser dos: En el homogéneo, en general, hay menos diversidad y, como consecuencia, cuesta más detectar a los mejores regresores; y, por otro lado, puede afectar, sobre todo, el hecho de que se reduzca el conjunto de entrenamiento (al 80 %), ya que recordemos que no se está utilizando solapamiento en esta prueba, y el DSEL a solo el 20 %. Esto penaliza bastante buscar a los mejores regresores.

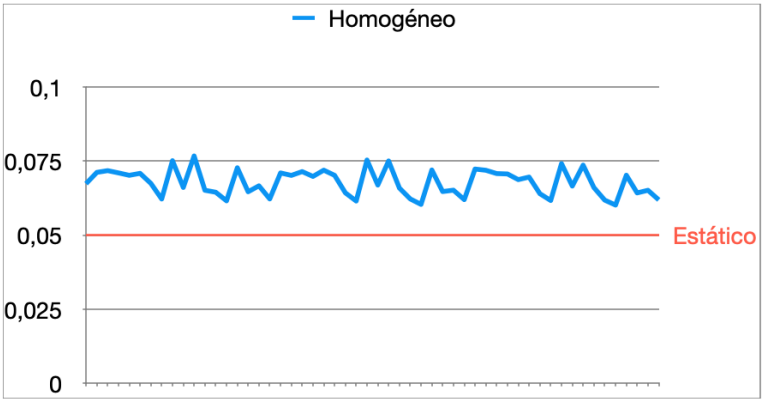


Figura 5.8: Gráfica MAPE con %DSEL = 20 HOMOGÉNEO Fuente: Elaboración Propia.

Por último, referente a los resultados obtenidos del error RMSE descritos en la tabla 5.14, se observa que para el ensemble dinámico heterogéneo, los tres mejores métodos en este caso en orden de menor a mayor error (de mejor a peor), son: ***Perfiles-sea-media-ponderada***, *Knn-sea-media-ponderada* y *Perfiles-sea-pMedio-ponderada*. Con este error también vemos que en más de la mitad los resultados de los métodos dinámicos se obtiene una mejora con respecto a la puntuación RMSE del estático heterogeneo.

Siguiendo con dicha tabla (5.14), para los ensembles homogéneos vemos que, con este error, tampoco se detectan mejoras en los resultados de los métodos dinámicos con respecto al estático.

**Prueba 2**

Para esta prueba se cambian los valores de los parámetros de los métodos dinámicos: El porcentaje DSEL se sube al 50 % y sí se permite solapamiento, como en la tabla 5.4 se

indica.

Los resultados obtenidos en esta prueba para los errores MAPE y RMSE por los métodos dinámicos que se han implementado en la librería son los correspondientes a la tablas 5.15 y 5.16, respectivamente.

En la tabla 5.13 encontramos los resultados de los métodos para el ensemble heterogéneo y el ensemble homogéneo fijados en el marco experimental (como todas las demás tablas de resultados, misma estructura para todas).

Con respecto a dicha tabla (5.15) que indica el error MAPE cometido por cada método dinámico, vemos que 43 de los métodos dinámicos implementados superan (más bajo error) al ensemble estático heterogéneo. Esto supone una mejora con respecto a la prueba anterior (en términos de MAPE obtenidos de los métodos dinámicos en un ensemble heterogéneo para los datos de Student Marks).

Específicamente, los tres mejores métodos han resultado ser: ***Cluster-sea-pMedio-ponderada***, *Cluster-sea-media-ponderada* y *knn-sea-media-pond*, siendo el primero el que mejor resultado obtiene. Cabe destacar que son exactamente los mismos tres mejores que en la prueba anterior, solo que cambian de orden.

Si miramos la gráfica 5.9, vemos que la varianza entre los métodos es bastante más grande respecto a la prueba anterior. Sin embargo, la mayor diferencia es que los picos que representan mínimos locales en la gráfica, son más bajos que en el caso anterio 5.7, por lo que los nuevos parámetros consiguen que los métodos obtengan mejores resultados.

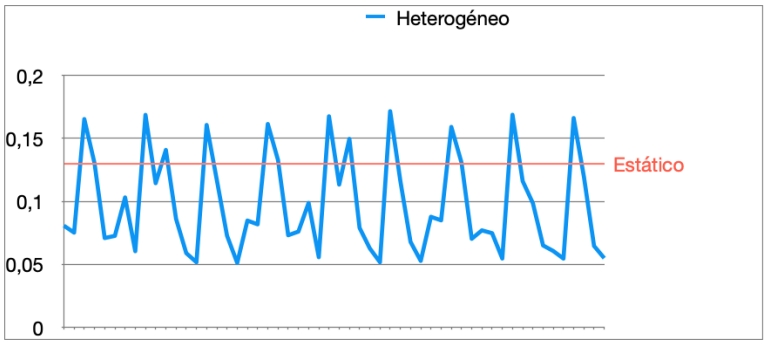


Figura 5.9: Gráfica MAPE con %DSEL = 50 HETEROGÉNEO Fuente: Elaboración Propia.

Por otro lado, siguiendo con la tabla 5.15, referente a los ensembles homogéneos encontramos que se mejora al estático en 34 ocasiones, o lo que es lo mismo, 24 métodos dinámicos ganan a su equivalente estático. Observando la gráfica 5.10 también vemos que ya los valores de los dinámico se encuentran más cercanos en general al error cometido por

Tabla. 5.15: SM-MAPE-dsel50

| | Dinámico | |
|---|---|---|
| **Método** | **Heterogéneo** | **Homogéneo** |
| Cluster-var-media-media | **0,08060570778778245** | **0,05548334599302106** |
| Cluster-var-media-ponderada | **0,07510684264661731** | 0,06456282458487882 |
| Cluster-var-mediaDesv-media | 0,16526755860386716 | **0,05586126716414023 5** |
| Cluster-var-mediaDesv-ponderada | **0,13030484280520552** | **0,05745433256929429** |
| Cluster-var-pMedio-media | **0,07078726659675683** | **0,057196310442667056** |
| Cluster-var-pMedio-ponderada | **0,07251463864411828** | 0,05950316280255861 |
| Cluster-ecm-media-media | **0,10303634560450241** | **0,057091672658714857** |
| Cluster-ecm-media-ponderada | **0,060348020949572576** | **0,053805832503379794** |
| Cluster-ecm-mediaDesv-media | 0,16858666244922907 | 0,060353573489797266 |
| Cluster-ecm-mediaDesv-ponderada | **0,11438270038948144** | **0,054566187022707116** |
| Cluster-ecm-pMedio-media | 0,14075892591622793 | 0,07047278790181126 |
| Cluster-ecm-pMedio-ponderada | **0,08601307656647644** | **0,0563909877153992** |
| Cluster-sea-media-media | **0,058816215274842196** | **0,05460804952268565** |
| Cluster-sea-media-ponderada | **0,05158097640191413** | **0,053156124** |
| Cluster-sea-mediaDesv-media | 0,1605965983919908 | 0,05893312922830814 |
| Cluster-sea-mediaDesv-ponderada | **0,11681702269719821** | **0,0548817756004523** |
| Cluster-sea-pMedio-media | **0,07265253953196424** | 0,06058580049400183 |
| Cluster-sea-pMedio-ponderada | **0,0510538273971094** | **0,055163695745949516** |
| Knn-var-media-media | **0,08466598900186886** | **0,056363007479678484** |
| Knn-var-media-ponderada | **0,08167970195730608** | **0,05635861655176803** |
| Knn-var-mediaDesv-media | 0,1613702208810528 | 0,057885666 |
| Knn-var-mediaDesv-ponderada | 0,13275424640488215 | **0,05730032953900381** |
| Knn-var-pMedio-media | **0,07301620534558065** | **0,05711727865798564** |
| Knn-var-pMedio-ponderada | **0,07591034648017306** | **0,05604885671108861** |
| Knn-ecm-media-media | **0,09864289937756192** | 0,05867995277152912 |
| Knn-ecm-media-ponderada | **0,05562165745133483** | **0,05435476455259584** |
| Knn-ecm-mediaDesv-media | 0,16747149563575314 | 0,0608209657251441 |
| Knn-ecm-mediaDesv-ponderada | **0,11327715899522606** | **0,05527965521695667** |
| Knn-ecm-pMedio-media | 0,149532439 | 0,06782276144795114 |
| Knn-ecm-pMedio-ponderada | **0,07886731159280873** | **0,05658858582352534** |
| Knn-sea-media-media | **0,06263135354577808** | **0,05760188019220095** |
| Knn-sea-media-ponderada | **0,051593073767512376** | **0,0545198220914638** |
| Knn-sea-mediaDesv-media | 0,17152703 | **0,05761613180420948** |
| Knn-sea-mediaDesv-ponderada | **0,11617980026435903** | **0,05441061923423628** |
| Knn-sea-pMedio-media | **0,06772559649337448** | 0,05889848721050754 |
| Knn-sea-pMedio-ponderada | **0,052645942465784454** | **0,05496511318105487** |
| Perfiles-var-media-media | **0,08765645828687722** | 0,057903494504479305 |
| Perfiles-var-media-ponderada | **0,08484621510360266** | 0,05980210605312712 |
| Perfiles-var-mediaDesv-media | 0,1590328714518982 | 0,05812190591725368 |
| Perfiles-var-mediaDesv-ponderada | 0,13067330720531625 | 0,05897553439739791 |
| Perfiles-var-pMedio-media | **0,0701836306040616** | **0,057576075542547026** |
| Perfiles-var-pMedio-ponderada | **0,07688773297381032** | 0,05930250043035058 |
| Perfiles-ecm-media-media | **0,0746443630976796** | **0,053787930926675574** |
| Perfiles-ecm-media-ponderada | **0,05452491039052485** | **0,0530780367596929** |
| Perfiles-ecm-mediaDesv-media | 0,16869481567494904 | 0,05892974936351649 |
| Perfiles-ecm-mediaDesv-ponderada | **0,11597314615268986** | **0,05540676565722128** |
| Perfiles-ecm-pMedio-media | **0,09848783451995018** | 0,0644317104431663 |
| Perfiles-ecm-pMedio-ponderada | **0,06489709573007874** | **0,05640825540900211 4** |
| Perfiles-sea-media-media | **0,06047832459711525** | **0,05541587777001551** |
| Perfiles-sea-media-ponderada | **0,054466968475337316** | **0,053852465706791586** |
| Perfiles-sea-mediaDesv-media | 0,16609735594783612 | 0,058173903835706066 |
| Perfiles-sea-mediaDesv-ponderada | **0,11973396865627342** | **0,05502869090487966** |
| Perfiles-sea-pMedio-media | **0,06449616871829524** | 0,057959355619196164 |
| Perfiles-sea-pMedio-ponderada | **0,05480315564286738** | **0,05504949147655267** |

Tabla. 5.16: SM-RMSE-dsel50

| | Dinámico | |
|---|---|---|
| **Método** | **Heterogéneo** | **Homogéneo** |
| Cluster-var-media-media | **2,501007983693694** | **1,529494548502857** |
| Cluster-var-media-ponderada | **2,442652849669329** | 1,6917028427695247 |
| Cluster-var-mediaDesv-media | 3,1178144248872903 | **1,5321231938071616** |
| Cluster-var-mediaDesv-ponderada | 2,7799520699525777 | **1,5432490373363383** |
| Cluster-var-pMedio-media | **2,1959863891302964** | 1,593692040709552 |
| Cluster-var-pMedio-ponderada | **2,356707254348953** | **1,5674095382728424** |
| Cluster-ecm-media-media | **2,286061272924572** | **1,5167813506515748** |
| Cluster-ecm-media-ponderada | **1,7386747251228862** | **1,476637680862751** |
| Cluster-ecm-mediaDesv-media | 3,261382435781011 | 1,726318216689847 |
| Cluster-ecm-mediaDesv-ponderada | **2,431726277018647** | **1,5012427853417987** |
| Cluster-ecm-pMedio-media | 2,8333750772805693 | 1,9314613520386978 |
| Cluster-ecm-pMedio-ponderada | **2,0419679901709413** | **1,526257699345558** |
| Cluster-sea-media-media | **1,8153149381369615** | **1,496619799665978** |
| Cluster-sea-media-ponderada | **1,6215167636305936** | **1,46412885** |
| Cluster-sea-mediaDesv-media | 3,1395040160541616 | 1,6414394198774471 |
| Cluster-sea-mediaDesv-ponderada | **2,480490927810366** | **1,5033842642849087** |
| Cluster-sea-pMedio-media | **1,848330239587657** | 1,653596113956127 |
| Cluster-sea-pMedio-ponderada | **1,6342984103559768** | **1,5092248277122997** |
| Knn-var-media-media | **2,6364361412203854** | **1,562613475009434** |
| Knn-var-media-ponderada | **2,636624596198489** | **1,5577102261273819** |
| Knn-var-mediaDesv-media | 3,0631629794118105 | 1,5866233435937915 |
| Knn-var-mediaDesv-ponderada | 2,841602476270169 | **1,5612325609405708** |
| Knn-var-pMedio-media | **2,3215404976515517** | **1,5608284371162369** |
| Knn-var-pMedio-ponderada | **2,4528266348602834** | **1,5329215663506421** |
| Knn-ecm-media-media | **2,2029325711763703** | 1,5936985832965491 |
| Knn-ecm-media-ponderada | **1,7031254837711267** | **1,4587108763234022** |
| Knn-ecm-mediaDesv-media | 3,247049784143132 | 1,729643879926734 |
| Knn-ecm-mediaDesv-ponderada | **2,4260612411635725** | **1,5140066894055524** |
| Knn-ecm-pMedio-media | 2,909305039 | 1,8620130268351094 |
| Knn-ecm-pMedio-ponderada | **1,9791982351135105** | **1,5336320928442913** |
| Knn-sea-media-media | **1,8263192402366968** | **1,546732697826784** |
| Knn-sea-media-ponderada | **1,6609114809052001** | **1,4632161743795928** |
| Knn-sea-mediaDesv-media | 3,46466204 | 1,6034129979405851 |
| Knn-sea-mediaDesv-ponderada | **2,4728084709431393** | **1,4796791173437303** |
| Knn-sea-pMedio-media | **1,811061195412171** | 1,5904013338791967 |
| Knn-sea-pMedio-ponderada | **1,633550326091525** | **1,474080541651703** |
| Perfiles-var-media-media | **2,6434086698204347** | **1,5653974748204542** |
| Perfiles-var-media-ponderada | **2,6781688720413497** | 1,587902622914823 |
| Perfiles-var-mediaDesv-media | 3,034061132876131 | 1,608801523208991 |
| Perfiles-var-mediaDesv-ponderada | 2,84073576075449 | 1,6065378400556 |
| Perfiles-var-pMedio-media | **2,3304805351436464** | **1,5483432964182549** |
| Perfiles-var-pMedio-ponderada | **2,5042365146344605** | 1,578448130739178 |
| Perfiles-ecm-media-media | **2,008505319010421** | **1,4842646360205294** |
| Perfiles-ecm-media-ponderada | **1,704973149048294** | **1,4425400912771593** |
| Perfiles-ecm-mediaDesv-media | 3,2655830398875216 | 1,6312345921604017 |
| Perfiles-ecm-mediaDesv-ponderada | **2,457147981498039** | **1,5027466917519863** |
| Perfiles-ecm-pMedio-media | **2,3858322890059784** | 1,785132583204987 |
| Perfiles-ecm-pMedio-ponderada | **1,8936611884334482** | **1,5229599456006686** |
| Perfiles-sea-media-media | **1,8320859601895925** | **1,4861535247643363** |
| Perfiles-sea-media-ponderada | **1,6973918232591862** | **1,4494849019326295** |
| Perfiles-sea-mediaDesv-media | 3,225381528838004 | 1,5947667870586921 |
| Perfiles-sea-mediaDesv-ponderada | **2,5146189896729183** | **1,4894369981182476** |
| Perfiles-sea-pMedio-media | **1,8260004127047202** | **1,5523286406475663** |
| Perfiles-sea-pMedio-ponderada | **1,677780557644668** | **1,4978118400684586** |

el estático. Por lo tanto, con los cambios realizados en los parámetros (porcentaje de DSEL y solapamiento con el entrenamiento) mejora mucho la situación del ensemble homogéneo en los métodos dinámicos de la prueba 1 vista en la gráfica 5.8.

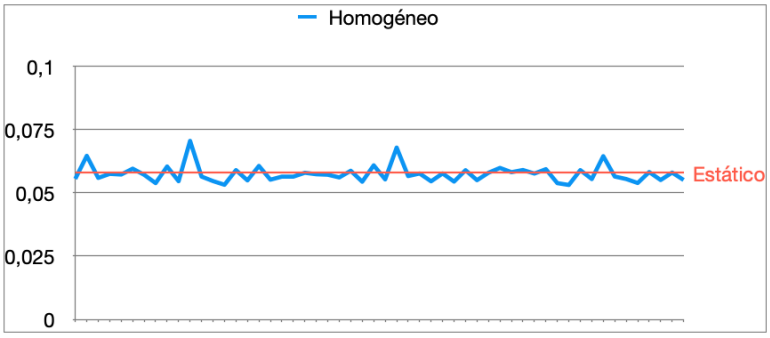


Figura 5.10: Gráfica MAPE con %DSEL = 50 HOMOGÉNEO Fuente: Elaboración Propia.

Los mejores métodos han resultados ser, de mejor a peor: ***Cluster-sea-media-ponderada***, *Knn-sea-pMedio-ponderada*, *Cluster-sea-pMedio-ponderada*.

Dejando a un lado el error MAPE, nos centramos en los errores RMSE (tabla 5.16). Para los ensembles heterogéneos, los mejores métodos para este error han sido, de mejor a peor: ***Cluster-sea-media-ponderada***, *Knn-sea-pMedio-ponderada* y *Cluster-sea-pMedio-ponderada*, coincidiendo en el primero y tercero con los mejores del error MAPE para ensembles heterogéneos (tabla 5.15).

Por último, siguiendo con el error RMSE, cabe destacar que los métodos dinámicos para el ensemble homogéneoque han dado los mejores resultados han sido ***Perfiles-ecm-media-ponderada***, *Perfiles-sea-media-ponderada* y *Knn-ecm-media-ponderada*, siendo el primero el mejor.

**Prueba 3**

La última modificación de los parámetros del porcentaje de DSEL (95 %) y solapamiento (permitido) (tabla 5.4), dan lugar a la tercera y última prueba con el dataset Student-Marks.

A continuación, las tablas 5.17 y 5.18 presentan los errores MAPE y RMSE obtenidos por los distintos métodos dinámicos, respectivamente.

Primero, nos centramos en el error MAPE. Para los ensembles heterogéneos, los mejores resultados, ordenados de mejor a peor, han sido los obtenidos por los métodos: ***Knn-sea-media-ponderada***, *Cluster-sea-media-ponderada* y *Cluster-sea-pMedio-ponderada*. Por

Tabla. 5.17: SM-MAPE-dsel95

| | Dinámico | |
|---|---|---|
| **Método** | **Heterogéneo** | **Homogéneo** |
| Cluster-var-media-media | **0,08026940658834603** | **0,0567805611471285** |
| Cluster-var-media-ponderada | **0,07611919281862739** | 0,0597142201422314 |
| Cluster-var-mediaDesv-media | 0,1661248696977013 | **0,05635552861917668** |
| Cluster-var-mediaDesv-ponderada | 0,1317530043418309 | **0,05772149902406691** |
| Cluster-var-pMedio-media | **0,08048730680307323** | **0,05711884715888897** |
| Cluster-var-pMedio-ponderada | **0,07587040316572573** | 0,0594834324167949 2 |
| Cluster-ecm-media-media | **0,10673089538612021** | 0,0593737569255578 9 |
| Cluster-ecm-media-ponderada | **0,05677618716246182** | **0,05318555742026683** |
| Cluster-ecm-mediaDesv-media | 0,16850193049909218 | 0,0627177798297077 9 |
| Cluster-ecm-mediaDesv-ponderada | **0,113183714682591** | **0,05447789868339441** |
| Cluster-ecm-pMedio-media | 0,14578038860866446 | 0,0672836330781144 7 |
| Cluster-ecm-pMedio-ponderada | **0,08813712210291594** | **0,05605579307878704** |
| Cluster-sea-media-media | **0,06068420814707672** | **0,05389507843619947** |
| Cluster-sea-media-ponderada | **0,050118883399451586** | **0,05385427742032959** |
| Cluster-sea-mediaDesv-media | 0,16046181770272638 | 0,0596100245526475 6 |
| Cluster-sea-mediaDesv-ponderada | **0,114069948569411 37** | **0,05436593759721794** |
| Cluster-sea-pMedio-media | **0,06616957484879776** | 0,0611140837245519 2 |
| Cluster-sea-pMedio-ponderada | **0,05112534355691354** | **0,0543908394785778** |
| Knn-var-media-media | **0,08484441479098552** | **0,054576502169444596** |
| Knn-var-media-ponderada | **0,08117860663723599** | **0,054841312730738626** |
| Knn-var-mediaDesv-media | 0,16417796091216205 | 0,0590392632425741 6 |
| Knn-var-mediaDesv-ponderada | 0,13261440993794243 | **0,056985847330156415** |
| Knn-var-pMedio-media | **0,07502291065264735** | **0,05637334991511731** |
| Knn-var-pMedio-ponderada | **0,07544170206665271** | **0,055794376286539306** |
| Knn-ecm-media-media | **0,10370346313645343** | 0,0594119751477764 5 |
| Knn-ecm-media-ponderada | **0,0551320860413901** | **0,05409823561470073** |
| Knn-ecm-mediaDesv-media | 0,1675033952599299 | 0,062817689283756 2 |
| Knn-ecm-mediaDesv-ponderada | **0,11137975802888433** | **0,05401952758430484** |
| Knn-ecm-pMedio-media | 0,14553366415859842 | 0,0673320582249506 3 |
| Knn-ecm-pMedio-ponderada | **0,0875925668801555** | **0,05474830774802766** |
| Knn-sea-media-media | **0,06095830790571803** | **0,05576984588207394** |
| Knn-sea-media-ponderada | **0,04761710084219227** | **0,05275116169002877** |
| Knn-sea-mediaDesv-media | 0,15869782208654687 | **0,057754173752860974** |
| Knn-sea-mediaDesv-ponderada | **0,11337621931407864** | **0,05352925485757869** |
| Knn-sea-pMedio-media | **0,07204239571288673** | 0,0602417060416073 9 |
| Knn-sea-pMedio-ponderada | **0,05151916466993295** | **0,05426685090533482** |
| Perfiles-var-media-media | **0,08099587967527265** | 0,058674969918088196 |
| Perfiles-var-media-ponderada | **0,08011632511656154** | 0,0604037361652881 8 |
| Perfiles-var-mediaDesv-media | 0,16350395216570798 | **0,05780873869765964** |
| Perfiles-var-mediaDesv-ponderada | 0,1306789750258349 | 0,0583394519744014 5 |
| Perfiles-var-pMedio-media | **0,06990525378621917** | 0,0591212268778598 1 |
| Perfiles-var-pMedio-ponderada | **0,075522627823685** | 0,0602426870111913 4 |
| Perfiles-ecm-media-media | **0,08423024116193642** | **0,05511961622014513** |
| Perfiles-ecm-media-ponderada | **0,05329239589275849** | **0,05306515721603962** |
| Perfiles-ecm-mediaDesv-media | 0,16910824892554585 | 0,0586430585302988 1 |
| Perfiles-ecm-mediaDesv-ponderada | **0,11428315370489435** | **0,054361291496757815** |
| Perfiles-ecm-pMedio-media | **0,11830344181036531** | 0,0626854969901938 6 |
| Perfiles-ecm-pMedio-ponderada | **0,06909847686084228** | **0,05537214335834636** |
| Perfiles-sea-media-media | **0,06273220178786085** | **0,05455879012929462** |
| Perfiles-sea-media-ponderada | **0,05324269260538207** | **0,05333114255899514** |
| Perfiles-sea-mediaDesv-media | 0,16584306231402418 | **0,05754012801412889** |
| Perfiles-sea-mediaDesv-ponderada | **0,11848373763896632** | **0,054160723914016475** |
| Perfiles-sea-pMedio-media | **0,06495182911626865** | **0,056418042990930214** |
| Perfiles-sea-pMedio-ponderada | **0,053123722229884526** | **0,05378955952275514** |

Tabla. 5.18: SM-RMSE-dsel95

| | Dinámico | |
|---|---|---|
| **Método** | **Heterogéneo** | **Homogéneo** |
| Cluster-var-media-media | **2,443846919385592** | **1,5306264605781172** |
| Cluster-var-media-ponderada | **2,454474670434746** | 1,5877277106788283 |
| Cluster-var-mediaDesv-media | 3,1335744151867755 | **1,538779412137577** |
| Cluster-var-mediaDesv-ponderada | 2,7977141846269205 | **1,5513475154291456** |
| Cluster-var-pMedio-media | **2,3719652383660645** | **1,5336546094939134** |
| Cluster-var-pMedio-ponderada | **2,3476797122025963** | 1,5988027626477705 |
| Cluster-ecm-media-media | **2,305809245442325** | 1,5874577634713702 |
| Cluster-ecm-media-ponderada | **1,6892316209629303** | **1,4438695071098724** |
| Cluster-ecm-mediaDesv-media | 3,264094844915083 | 1,7187279664332844 |
| Cluster-ecm-mediaDesv-ponderada | **2,3951610114547828** | **1,4888007949299271** |
| Cluster-ecm-pMedio-media | 2,8909393085636217 | 1,8373830918767617 |
| Cluster-ecm-pMedio-ponderada | **2,0464593424187023** | **1,5079939238140994** |
| Cluster-sea-media-media | **1,7863430705523675** | **1,449547064655767** |
| Cluster-sea-media-ponderada | **1,568523985382961** | **1,442457108381292** |
| Cluster-sea-mediaDesv-media | 3,1255581271233503 | 1,6065788925575675 |
| Cluster-sea-mediaDesv-ponderada | **2,4187510184834045** | **1,4868946696579353** |
| Cluster-sea-pMedio-media | **1,8641550937028164** | 1,6565897849606965 |
| Cluster-sea-pMedio-ponderada | **1,5939565746787367** | **1,470511308824705** |
| Knn-var-media-media | **2,5690524946550513** | **1,5550285085503865** |
| Knn-var-media-ponderada | **2,532861896796733** | **1,5649064715400653** |
| Knn-var-mediaDesv-media | 3,147377635764047 | **1,560386115431578** |
| Knn-var-mediaDesv-ponderada | 2,8263160861030845 | **1,5504150347247105** |
| Knn-var-pMedio-media | **2,350200328983286** | **1,531085842020752** |
| Knn-var-pMedio-ponderada | **2,404151974384473** | **1,5295618081176587** |
| Knn-ecm-media-media | **2,265371922202587** | 1,6362982392585246 |
| Knn-ecm-media-ponderada | **1,6568404672140251** | **1,4654708590277046** |
| Knn-ecm-mediaDesv-media | 3,2650799713569127 | 1,7517287810694462 |
| Knn-ecm-mediaDesv-ponderada | **2,3886121580098694** | **1,4936766705534734** |
| Knn-ecm-pMedio-media | 2,893305434554501 | 1,8310787519204623 |
| Knn-ecm-pMedio-ponderada | **2,0196763224459873** | **1,5036510327257444** |
| Knn-sea-media-media | **1,7755184746520172** | **1,5119928510222962** |
| Knn-sea-media-ponderada | **1,5611397725087373** | **1,427672700835413** |
| Knn-sea-mediaDesv-media | 3,1197901239752994 | 1,6132223108517252 |
| Knn-sea-mediaDesv-ponderada | **2,416688997758217** | **1,4702242381241177** |
| Knn-sea-pMedio-media | **1,86020390961743** | 1,657611429485088 |
| Knn-sea-pMedio-ponderada | **1,6252426022876727** | **1,4825530477593039** |
| Perfiles-var-media-media | **2,573676504778503** | 1,602122148886433 |
| Perfiles-var-media-ponderada | **2,605156785639221** | 1,6214223185650483 |
| Perfiles-var-mediaDesv-media | 3,0918068814832855 | 1,5950511337387034 |
| Perfiles-var-mediaDesv-ponderada | 2,8450584679548334 | 1,5976506519411662 |
| Perfiles-var-pMedio-media | **2,355564490135822** | **1,5656156228661875** |
| Perfiles-var-pMedio-ponderada | **2,497713990221891** | 1,5897867938863548 |
| Perfiles-ecm-media-media | **2,0997830514427873** | **1,5231660407462597** |
| Perfiles-ecm-media-ponderada | **1,656057705032849** | **1,452296757036819** |
| Perfiles-ecm-mediaDesv-media | 3,2582489369005954 | 1,6302052685148969 |
| Perfiles-ecm-mediaDesv-ponderada | **2,4061634351610293** | **1,494743113506075** |
| Perfiles-ecm-pMedio-media | **2,6301969428977365** | 1,6928208580313542 |
| Perfiles-ecm-pMedio-ponderada | **1,9069611993836357** | **1,5009407083269672** |
| Perfiles-sea-media-media | **1,785152658737355** | **1,4941468477845246** |
| Perfiles-sea-media-ponderada | **1,5964124986705177** | **1,4527273938168679** |
| Perfiles-sea-mediaDesv-media | 3,197580705809288 | **1,5653086987537592** |
| Perfiles-sea-mediaDesv-ponderada | **2,464536686473286** | **1,4769932441601454** |
| Perfiles-sea-pMedio-media | **1,7752763108889784** | 1,5854710259367093 |
| Perfiles-sea-pMedio-ponderada | **1,56838419499479** | **1,480196024922875** |

otro lado, si no fijamos en la gráfica 5.11 que muestra el comportamiento de los métodos implementados (dinámicos) en ensembles heterogéneos con respecto a su equivalente estático, vemos que en esta prueba se obtienen resultados muy parecidos a la prueba anterior en ensembles heterogéneos (5.9), por lo que este último cambio no genera mucha mejoría general en los métodos dinámicos. Sin embargo, es con esta configuración de parámetros donde se obtiene, para el ensemble heterogéneo, el valor más bajo de error de entre las tres pruebas de este experimento. Concretamente, el obtenido por el método dinámico ***Knn-sea-media-ponderada*** , seguido de *Cluster-sea-media-ponderada* y *Cluster-sea-pMedio-ponderada*. Destaca que en todos se calculan los niveles de competencia a partir de la suma de errores absolutos y la agregaación de salidas mediante la media ponderada por los niveles de competencia de los regresores escogidos.

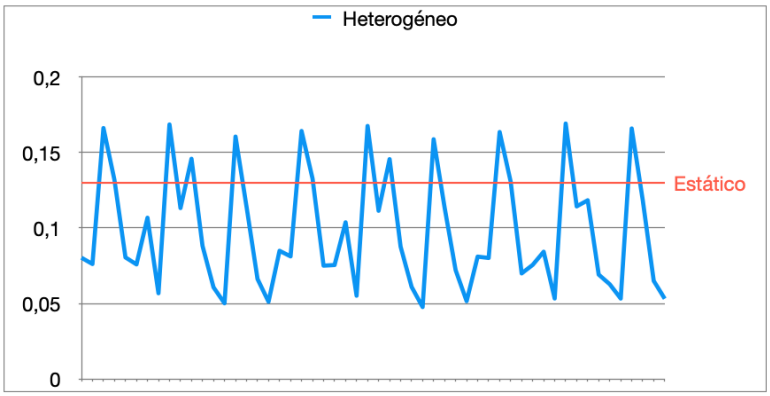


Figura 5.11: Gráfica MAPE con %DSEL = 95 HETEROGÉNEO Fuente: Elaboración Propia.

Siguiendo con la tabla 5.17, y orientándonos con la gráfica 5.12 los resultados del error MAPE para los ensembles homogéneos obtenidos por los métodos implementados (dinámicos) experimentan una mejora con respecto a la prueba anterior en los ensembles homogéneos (los que se presentan en la gráfica anterior 5.10).

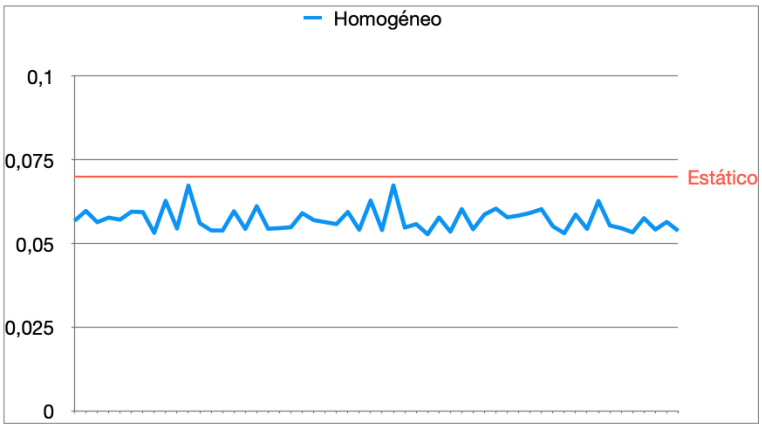


Figura 5.12: Gráfica MAPE con %DSEL = 95 HOMOGÉNEO Fuente: Elaboración Propia.

Cambiando a la tabla que representa el error RMSE (tabla 5.18), notamos que para el ensemble heterogéneo, los mejores resultados son obtenidos por los métodos: ***Knn-sea-***

***media-ponderada***, *Perfiles-sea-pMedio-ponderada* y *Cluster-sea-media-ponderada*, coincidiendo con los mejores métodos según la medida de error MAPE en el primero y tercero. Por otro lado, para los ensembles homogéneos, los mejores métodos dinámicos resultan ser: ***Knn-sea-media-ponderada***, *Cluster-sea-media-ponderada*, *Cluster-ecm-media-ponderada*, coincidiendo en el primero y el tercero mejores del MAPE para ensembles homogéneos.

**Conclusiones del experimeto 2**

En este experimento, el ensemble homogéneo dinámico ha pasado de obtener en todos los casos resultados peores (en la prueba 1), a conseguir que todos superen al estático. Por otro lado, el ensemble heterogéneo no ha variado tanto entre prueba y prueba, pero sí que ha obtenido en ciertos métodos mejores resultados a medida que el porcentaje del DSEL ha subido.

Los mejores resultados obtenidos por los ensemble dinámicos han sido: para el heterogéneo, el menor MAPE es 0,047617 (frente al MAPE obtenido para su equivalente estatico: 0,130087382 ) y para el homogéneo, el mejor MAPE obtenido es 0,05275 (frente a 0,057850 que obtiene el estático homogéneo). Ambos resultados han sido obtenido duraante la prueba 3.

Podemos ver que el heterogéneo consigue disminuir en gran medida el error con respecto al estático. Por lo que, aunque haya menos veces que se gane al estático en el heterogéneo que en el homogéneo, el heterogéneo consigue mejoría mucho mejores que el homogéneo. Sin embargo, cabe destacar que el homogéneo lo tiene dificil, ya que el error que se comete el ensemble estático homogéneo ya es muy pequeño, por lo que mejorarlo rasulta más complicado.

Para cerrar la conclusión, cabe destacar que, de nuevo, en este experimento ha funcionado mejor la última configuración de parámetros ( %Dsel igual a 95 y permitido el solapamiento).

### 5.2.3. Experimento 3

En este último experimento se van a realizar las distintas pruebas a partir del conjunto de datos **Real Estate Price**.

El ensemble estático homogéneo y heterogéneo con estos datos han cometido los siguientes errores expuestos en la tabla 5.19

Tabla. 5.19: Estáticos RE

| | **Estático** | |
|---|---|---|
| | **Heterogéneo** | **Homogéneo** |
| **MAPE** | 0,170130613 | 0,139241542 |
| **RMSE** | 8,056226655 | 7,534420414 |

**Prueba 1**

Para la primera prueba, como ya sabemos, el porcentaje de DSEL utilizado es el 20 % y sin solapar con el conjunto de entrenamiento.

Los resultados obtenidos en esta prueba se presentan en las tablas 5.24 y 5.25, que representan el error MAPE y RMSE, respectivamente, obtenido por los métodos implementados en la librería para el caso de ensemble homogéneo y heterogéneo.

Para interpretar mejor el error MAPE mostrado en la tabla 5.24, se ha creado la gráfica asociada (como en los experimentos anteriores) para los ensembles homogéneos y otra para los ensembles heterogéneos.

Por lo tanto, observando esta gráfica 5.13 que hace referencia a los métodos dinámicos con ensemble heterogéneo, nos damos cuenta de que la mayoría de métodos consiguen un resultado menor de error que el estático. Precisamente, es en 40 métodos los que se mejora la marca del ensemble estático equivalente.

Los mejores métodos para los ensembles heterogéneos hasn resultado ser: ***Perfiles-sea-media-ponderada***,*Perfiles-sea-pMedio-ponderada*, *Cluster-sea-media-ponderada*, siendo el primero el mejor.

Sin embargo, para los ensembles homogéneos, los métodos dinámicos configuraados, con los parámetros especificados en esta prueba, no parecen funcionar de manera tan excepcional como con los ensembles heterogéneos. No obstante, sí que se consigue superar al ensemble estático en 10 ocasiones, obteniendo la mejor marca con el método implementado ***Perfiles-sea-media-ponderada***, seguido de *Perfiles-sea-pMedio-ponderada* y

Tabla. 5.20: RE-MAPE-dsel95

| | Dinámico | |
|---|---|---|
| **Método** | **Heterogéneo** | **Homogéneo** |
| Cluster-var-media-media | 0,17244359564105735 | **0,13702102489477574** |
| Cluster-var-media-ponderada | **0,16917606582671468** | **0,13747710417575507** |
| Cluster-var-mediaDesv-media | **0,16403801770783627** | **0,13800303688946186** |
| Cluster-var-mediaDesv-ponderada | **0,1673468317199768** | **0,1363199118273542** |
| Cluster-var-pMedio-media | **0,1642361806652522** | **0,13856621125087767** |
| Cluster-var-pMedio-ponderada | **0,16513072069138107** | **0,13651825877351628** |
| Cluster-ecm-media-media | **0,15571267024483407** | 0,1407414740957329 |
| Cluster-ecm-media-ponderada | **0,14994572701027462** | 0,14027483674664049 |
| Cluster-ecm-mediaDesv-media | 0,1767638773880785 | 0,140355117003562 |
| Cluster-ecm-mediaDesv-ponderada | **0,15894773063812376** | **0,1382556473684872** |
| Cluster-ecm-pMedio-media | **0,16618784165190925** | 0,1414454517796998 |
| Cluster-ecm-pMedio-ponderada | **0,1576670896869746** | **0,1386700536517298** |
| Cluster-sea-media-media | **0,14895972104246513** | 0,1399427459928612 |
| Cluster-sea-media-ponderada | **0,14605590316284042** | 0,13993334427386223 |
| Cluster-sea-mediaDesv-media | 0,17531550900881682 | **0,13871807848843062** |
| Cluster-sea-mediaDesv-ponderada | **0,15862296859684738** | **0,13849961947542108** |
| Cluster-sea-pMedio-media | **0,1517571978694084** | 0,14055164552834915 |
| Cluster-sea-pMedio-ponderada | **0,14817614836137147** | **0,13894088292974818** |
| Knn-var-media-media | 0,17138359425914168 | **0,13612113140467302** |
| Knn-var-media-ponderada | **0,16953001553119712** | **0,1373126240043765** |
| Knn-var-mediaDesv-media | **0,16906907033418708** | **0,1363690294958163** |
| Knn-var-mediaDesv-ponderada | **0,16865311503539118** | **0,13621919708262012** |
| Knn-var-pMedio-media | **0,16640777871310986** | **0,13724700330304646** |
| Knn-var-pMedio-ponderada | **0,16522973793910062** | **0,1374387510005574** |
| Knn-ecm-media-media | **0,1589991159750268** | 0,14197626751108983 |
| Knn-ecm-media-ponderada | **0,1501358246870639** | 0,13925033237849221 |
| Knn-ecm-mediaDesv-media | 0,17712919935095078 | 0,13972300634626406 |
| Knn-ecm-mediaDesv-ponderada | **0,15788966575986924** | **0,13680971338118125** |
| Knn-ecm-pMedio-media | 0,17150144364392947 | 0,14250404359559737 |
| Knn-ecm-pMedio-ponderada | **0,15859777807332343** | **0,13701908430191784** |
| Knn-sea-media-media | **0,15019606235266603** | 0,14359293246018584 |
| Knn-sea-media-ponderada | **0,144863364352875** | 0,14057256330867954 |
| Knn-sea-mediaDesv-media | 0,17439248130760682 | **0,13793526817947807** |
| Knn-sea-mediaDesv-ponderada | **0,1582055095312944** | **0,13754318094807183** |
| Knn-sea-pMedio-media | **0,15402850949499217** | 0,13948535397089518 |
| Knn-sea-pMedio-ponderada | **0,15070489354328503** | **0,1381297792232184** |
| Perfiles-var-media-media | 0,1782734274565551 | **0,13907287793459303** |
| Perfiles-var-media-ponderada | 0,17883580800196747 | **0,13764106238388538** |
| Perfiles-var-mediaDesv-media | **0,1643725274123613** | **0,1367341404378562** |
| Perfiles-var-mediaDesv-ponderada | 0,17301440109359725 | **0,13651578140350512** |
| Perfiles-var-pMedio-media | 0,17545390620285106 | **0,13786636574857064** |
| Perfiles-var-pMedio-ponderada | 0,1775267708961008 | **0,1371144989762741** |
| Perfiles-ecm-media-media | **0,15415684074840214** | **0,13826217321124218** |
| Perfiles-ecm-media-ponderada | **0,15099243219719968** | **0,1389677160808932** |
| Perfiles-ecm-mediaDesv-media | 0,1771610093121221 | **0,13691292327537163** |
| Perfiles-ecm-mediaDesv-ponderada | **0,16036180464269925** | **0,13597825801081684** |
| Perfiles-ecm-pMedio-media | **0,1562064053063478** | **0,13777869287298328** |
| Perfiles-ecm-pMedio-ponderada | **0,1520730599619251** | **0,13657948617372961** |
| Perfiles-sea-media-media | **0,1499256445189342** | 0,1411919104699127 |
| Perfiles-sea-media-ponderada | **0,14729177282739783** | 0,14025907310061705 |
| Perfiles-sea-mediaDesv-media | 0,17690645472318756 | **0,13726231782344783** |
| Perfiles-sea-mediaDesv-ponderada | **0,160291863764826** | **0,13746360852695988** |
| Perfiles-sea-pMedio-media | **0,15142932853525018** | 0,14108707105131524 |
| Perfiles-sea-pMedio-ponderada | **0,1490404935766067** | 0,13975082465830485 |

Tabla. 5.21: RE-RMSE-dsel95

| | Dinámico | |
|---|---|---|
| **Método** | **Heterogéneo** | **Homogéneo** |
| Cluster-var-media-media | 8,371686151635842 | **7,515764230810474** |
| Cluster-var-media-ponderada | 8,346773007587542 | **7,510376444297593** |
| Cluster-var-mediaDesv-media | 8,059913881005963 | **7,487506624569171** |
| Cluster-var-mediaDesv-ponderada | 8,19830939809258 | **7,433186142334394** |
| Cluster-var-pMedio-media | 8,113447979594781 | 7,5436793791774575 |
| Cluster-var-pMedio-ponderada | 8,155870775581757 | **7,432663046176269** |
| Cluster-ecm-media-media | **7,89818763519699** | 7,563492868941344 |
| Cluster-ecm-media-ponderada | **7,8506120553383** | 7,565990065579539 |
| Cluster-ecm-mediaDesv-media | 8,366742605258173 | 7,539549831889621 |
| Cluster-ecm-mediaDesv-ponderada | **7,934767458273153** | **7,527262449749934** |
| Cluster-ecm-pMedio-media | 8,174689401487647 | 7,983739147197511 |
| Cluster-ecm-pMedio-ponderada | 8,14069439657329 | 7,56821839691186 |
| Cluster-sea-media-media | **7,720662608972138** | 7,573544042982074 |
| Cluster-sea-media-ponderada | **7,717060940326876** | 7,597944440903736 |
| Cluster-sea-mediaDesv-media | 8,342995771197517 | 7,53796716429409 |
| Cluster-sea-mediaDesv-ponderada | **7,954597022155225** | 7,562325572517047 |
| Cluster-sea-pMedio-media | **7,799772915914341** | 7,76155408137429 |
| Cluster-sea-pMedio-ponderada | **7,886685209277236** | 7,586970666393343 |
| Knn-var-media-media | 8,24352567180183 | **7,382092572505449** |
| Knn-var-media-ponderada | 8,317973026538153 | **7,410220625362553** |
| Knn-var-mediaDesv-media | 8,159430020407543 | **7,487198413640594** |
| Knn-var-mediaDesv-ponderada | 8,221104008413622 | **7,382778636391312** |
| Knn-var-pMedio-media | 8,230992282775762 | **7,463476460654706** |
| Knn-var-pMedio-ponderada | 8,133424090403924 | **7,421011739015675** |
| Knn-ecm-media-media | **7,972338505560612** | 7,581716393574015 |
| Knn-ecm-media-ponderada | **7,8603675787271925** | 7,573196720421743 |
| Knn-ecm-mediaDesv-media | 8,380916562563826 | 7,989741201643689 |
| Knn-ecm-mediaDesv-ponderada | **7,920368944097231** | **7,5328577103260015** |
| Knn-ecm-pMedio-media | 8,349814583974277 | 8,076933078976387 |
| Knn-ecm-pMedio-ponderada | 8,177793341354175 | 7,535749614113817 |
| Knn-sea-media-media | **7,717454738008735** | 7,884367714202656 |
| Knn-sea-media-ponderada | **7,669007684962773** | 7,717842471426172 |
| Knn-sea-mediaDesv-media | 8,32501049991898 | 7,591010256465208 |
| Knn-sea-mediaDesv-ponderada | **7,940311785039147** | 7,542423401778308 |
| Knn-sea-pMedio-media | **7,885321097820371** | 7,745705826378304 |
| Knn-sea-pMedio-ponderada | 8,119024967761328 | 7,595971029019753 |
| Perfiles-var-media-media | 8,527010967650156 | 7,6058569409876124 |
| Perfiles-var-media-ponderada | 8,532298828385846 | 7,5821249729459455 |
| Perfiles-var-mediaDesv-media | **8,050557520361243** | **7,303859939188477** |
| Perfiles-var-mediaDesv-ponderada | 8,328479786114382 | **7,396690570347208** |
| Perfiles-var-pMedio-media | 8,300662159177934 | **7,344221294140927** |
| Perfiles-var-pMedio-ponderada | 8,436398005606604 | **7,414770923134019** |
| Perfiles-ecm-media-media | **7,949139111861821** | 7,582254087801682 |
| Perfiles-ecm-media-ponderada | **7,892532479455072** | 7,6449217548753365 |
| Perfiles-ecm-mediaDesv-media | 8,37005247784651 | **7,303354079628784** |
| Perfiles-ecm-mediaDesv-ponderada | **7,968373175820015** | **7,420566053420522** |
| Perfiles-ecm-pMedio-media | 8,09165484186587 | **7,380460290023374** |
| Perfiles-ecm-pMedio-ponderada | **8,048571593945763** | **7,43504197507372** |
| Perfiles-sea-media-media | **7,758876537631615** | 7,658162050055485 |
| Perfiles-sea-media-ponderada | **7,731583510428953** | 7,646649330526773 |
| Perfiles-sea-mediaDesv-media | 8,375702034166178 | **7,455152092361803** |
| Perfiles-sea-mediaDesv-ponderada | **7,981733346422712** | **7,502616677444256** |
| Perfiles-sea-pMedio-media | **7,833099520706268** | 7,6323344987243775 |
| Perfiles-sea-pMedio-ponderada | **7,841859642255751** | 7,612687126876994 |

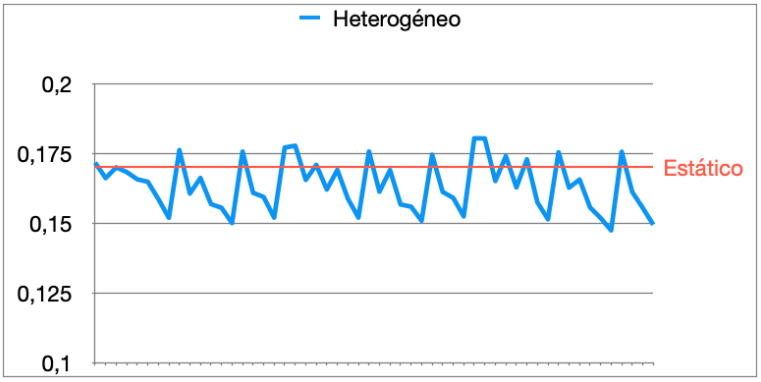


Figura 5.13: Gráfica MAPE con %DSEL = 20 HETEROGÉNEO Fuente: Elaboración Propia.

*Perfiles-ecm-media-ponderada*.

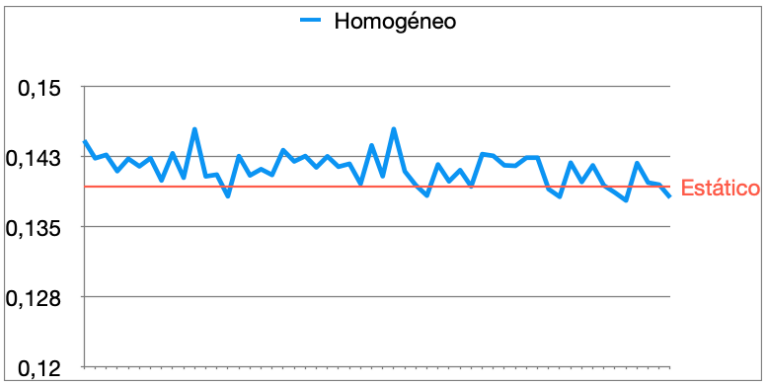


Figura 5.14: Gráfica MAPE con %DSEL = 20 HOMOGÉNEO Fuente: Elaboración Propia.

Por último, recpecto a los resultados de RMSE presentados en la tabla 5.25, los mejores métodos que resultan para el ensemble heterogéneo son: ***Cluster-sea-pMedio-ponderada***, *Cluster-sea-media-ponderada* y *Knn-sea-media-ponderada*, coincidiendo solo en el segundo con uno de los que también han resultado mejores con el error MAPE (tabla 5.24.

Por otro lado, los mejores resultados de RMSE para el ensemble homogéneo han resultado ser: ***Perfiles-sea-media-media*** (el mejor resultado), *Perfiles-sea-media-ponderada* y *Perfiles-sea-pMedio-ponderada*. Con esta medida de error, el ensemble homogéneo con método dinámico consigue ganar en la mayoría de ocasiones al ensemble estático equivalente.

**Prueba 2**

Para comenzar la segunda prueba, ya sabemos que hay que hacer un cambio en los parámetros de los métodos dinámicos implementados. En ete caso, se cambia el porcentaje

de DSEL a 50 % y solapamiento con el conjunto de entrenamiento.

De igual manera que en las pruebas anteriores, los resultados de errores MAPE y RMSE cometidos por los métodos se exponen en las tablas 5.22 y 5.23, respectivamente.

Con respecto al MAPE en el ensemble heterogéneo, con ayuda de la gráfica 5.15 vemos que para esta nueva configuración de parámetros, los métodos dinámicos siguen superando en numerosas ocasiones a el error MAPE del estático. Además, como se puede ver en la tabla 5.22, se obtiene un error mínimo más bajo que en la prueba anterior. Por lo que se deduce, que estos nuevos parámetros de los métodos dinámicos funcionan mejor en ensembles heterogéneo y para este conjunto de datos en concreto.

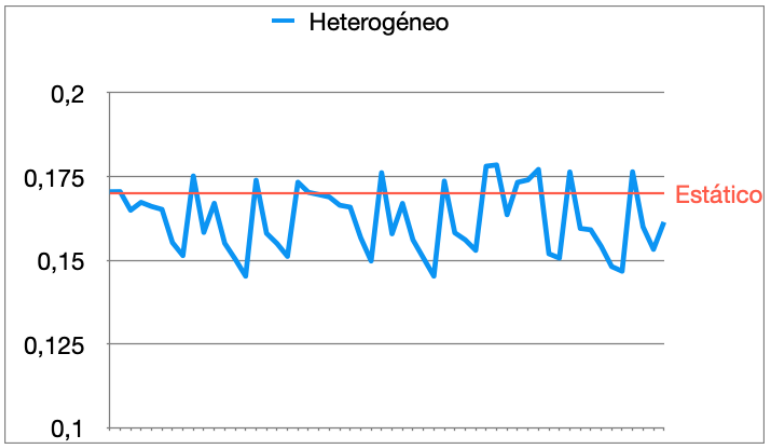


Figura 5.15: Gráfica MAPE con %DSEL = 50 HETERGÉNEO Fuente: Elaboración Propia.

Los mejores métodos dinámicos con respecto al error MAPE de ensemble heterogéneo (tabla 5.22) son: ***Cluster-sea-media-ponderada***, *Knn-sea-media-ponderada* y *Perfiles-sea-media-ponderada*, siendo la primera la que menor error obtiene. Además, cabe destacar que 39 métodos dinámicos ganan al estático.

Con respecto a los ensembles homogéneos, como podemos observar en los resultados obtenidos en la tabla 5.22, vemos que casi todos los métodos dinámicos implementados ganan al estático. Concretamente, en 50 de ellos se obtienen errores MAPE menores.

Observando la gráfica 5.16, donde se representan los errores MAPE cometidos por los métodos dinámicos en contraste con el obtenido por el estático, nos damos cuenta de que la línea azul se encuentra definida prácticamente por debajo de la marca del error estático.

Tabla. 5.22: RE-MAPE-dsel50

| | Dinámico | |
|---|---|---|
| **Método** | **Heterogéneo** | **Homogéneo** |
| Cluster-var-media-media | 0,1704282070563099 | **0,13759678126860445** |
| Cluster-var-media-ponderada | 0,17046603160080848 | **0,1363643476825252** |
| Cluster-var-mediaDesv-media | **0,1649233575865812** | **0,1364304093684033** |
| Cluster-var-mediaDesv-ponderada | **0,16727066536286583** | **0,13545636876014847** |
| Cluster-var-pMedio-media | **0,16605466667200033** | **0,13619555663262264** |
| Cluster-var-pMedio-ponderada | **0,16516617664580355** | **0,1356638106052675** |
| Cluster-ecm-media-media | **0,15527485256001086** | **0,13754754700539912** |
| Cluster-ecm-media-ponderada | **0,15139034287135647** | **0,13818298423232334** |
| Cluster-ecm-mediaDesv-media | 0,17513590205337243 | **0,13718484540675308** |
| Cluster-ecm-mediaDesv-ponderada | **0,15830837911955323** | **0,1366390785480548** |
| Cluster-ecm-pMedio-media | **0,1669764955957253** | **0,13845408223862116** |
| Cluster-ecm-pMedio-ponderada | **0,1550731557878861** | **0,13776797629871643** |
| Cluster-sea-media-media | **0,15033849022238516** | 0,14051220308837395 |
| Cluster-sea-media-ponderada | **0,1452380110444363** | 0,13925173711430658 |
| Cluster-sea-mediaDesv-media | 0,1738577816849288 | **0,13551894270802492** |
| Cluster-sea-mediaDesv-ponderada | **0,15803234342895167** | **0,13699728171547584** |
| Cluster-sea-pMedio-media | **0,1549924266647007** | **0,13798128018345582** |
| Cluster-sea-pMedio-ponderada | **0,15114789913707388** | **0,13764650596157013** |
| Knn-var-media-media | 0,17327507043560325 | **0,13745364589291306** |
| Knn-var-media-ponderada | 0,17026440684428396 | **0,1374854480596011** |
| Knn-var-mediaDesv-media | **0,16958784294325663** | **0,1362449667677666** |
| Knn-var-mediaDesv-ponderada | **0,16887999715820648** | **0,1358819434739258** |
| Knn-var-pMedio-media | **0,16640744325956125** | **0,1372978915650805** |
| Knn-var-pMedio-ponderada | **0,1658313472298624** | **0,13637190788496392** |
| Knn-ecm-media-media | **0,15674329591877492** | **0,13832604507945273** |
| Knn-ecm-media-ponderada | **0,1497803218527015** | **0,13755013117282555** |
| Knn-ecm-mediaDesv-media | 0,17610856835864128 | 0,14079805780130428 |
| Knn-ecm-mediaDesv-ponderada | **0,1578933218077026** | **0,13674820538254415** |
| Knn-ecm-pMedio-media | **0,16694506654653224** | 0,14126222717242376 |
| Knn-ecm-pMedio-ponderada | **0,15607699986521847** | **0,13626088702721237** |
| Knn-sea-media-media | **0,150657424** | **0,13896456066965152** |
| Knn-sea-media-ponderada | **0,14525581795960676** | **0,13761826748738035** |
| Knn-sea-mediaDesv-media | 0,17360814941928532 | **0,13738843050753025** |
| Knn-sea-mediaDesv-ponderada | **0,1581927151604137** | **0,1370399442221538** |
| Knn-sea-pMedio-media | **0,15608253259964514** | **0,13831884550301315** |
| Knn-sea-pMedio-ponderada | **0,15295256041859337** | **0,1370706350161811** |
| Perfiles-var-media-media | 0,1780501368251803 | **0,1361474465596695** |
| Perfiles-var-media-ponderada | 0,178442674321835 | **0,13639143573729917** |
| Perfiles-var-mediaDesv-media | **0,16358311684539703** | **0,13579073453735604** |
| Perfiles-var-mediaDesv-ponderada | 0,17325098178476997 | **0,1360445206345922** |
| Perfiles-var-pMedio-media | 0,17397306465259685 | **0,1361909413580915** |
| Perfiles-var-pMedio-ponderada | 0,17708977046615534 | **0,13617881381342617** |
| Perfiles-ecm-media-media | **0,151931275379031** | **0,13578054957962596** |
| Perfiles-ecm-media-ponderada | **0,1506376554179856** | **0,13722278969908136** |
| Perfiles-ecm-mediaDesv-media | 0,17633233687227015 | **0,1358028258513129** |
| Perfiles-ecm-mediaDesv-ponderada | **0,1594909205810889** | **0,13528327394884226** |
| Perfiles-ecm-pMedio-media | **0,15907722770846844** | **0,13567966858691766** |
| Perfiles-ecm-pMedio-ponderada | **0,15415002874781422** | **0,13528916857549045** |
| Perfiles-sea-media-media | **0,14811934787223843** | **0,1376841116081967** |
| Perfiles-sea-media-ponderada | **0,14672963516505022** | **0,13774869272086326** |
| Perfiles-sea-mediaDesv-media | 0,1763797450996525 | **0,13480080526650964** |
| Perfiles-sea-mediaDesv-ponderada | **0,15995311666083734** | **0,13609663683224318** |
| Perfiles-sea-pMedio-media | **0,15325341821496455** | **0,13621450241344446** |
| Perfiles-sea-pMedio-ponderada | **0,161386319** | **0,1365763810394599** |

Tabla. 5.23: RE-RMSE-dsel50

| | Dinámico | |
|---|---|---|
| **Método** | **Heterogéneo** | **Homogéneo** |
| Cluster-var-media-media | 8,357120187118579 | **7,4481848766052305** |
| Cluster-var-media-ponderada | 8,373137525041091 | **7,42640955622217** |
| Cluster-var-mediaDesv-media | **8,05375288710329** | **7,3415464322527955** |
| Cluster-var-mediaDesv-ponderada | 8,215548534428413 | **7,361923895274361** |
| Cluster-var-pMedio-media | 8,260885797185223 | **7,280844876322723** |
| Cluster-var-pMedio-ponderada | 8,197846934686115 | **7,386980137115867** |
| Cluster-ecm-media-media | **7,828232694888586** | **7,391692529151875** |
| Cluster-ecm-media-ponderada | **7,880139982825296** | **7,43525660629137** |
| Cluster-ecm-mediaDesv-media | 8,327427776423743 | **7,355453972293914** |
| Cluster-ecm-mediaDesv-ponderada | **7,918624467735858** | **7,392670896318668** |
| Cluster-ecm-pMedio-media | 8,27207568949136 | 7,880777242286628 |
| Cluster-ecm-pMedio-ponderada | **7,98876915691596** | **7,467544296927484** |
| Cluster-sea-media-media | **7,75963326524081** | 7,557515715451471 |
| Cluster-sea-media-ponderada | **7,667909470111373** | 7,548912443462316 |
| Cluster-sea-mediaDesv-media | 8,32128796126296 | **7,3487094147019745** |
| Cluster-sea-mediaDesv-ponderada | **7,947877767828044** | **7,406467400529939** |
| Cluster-sea-pMedio-media | 8,124771277608506 | 7,6564135979687595 |
| Cluster-sea-pMedio-ponderada | **7,997887403737255** | **7,469704167848098** |
| Knn-var-media-media | 8,295016515025361 | **7,424839964362926** |
| Knn-var-media-ponderada | 8,32626595044105 | **7,434644738561696** |
| Knn-var-mediaDesv-media | 8,167540291821995 | **7,35070556557568** |
| Knn-var-mediaDesv-ponderada | 8,225747072085557 | **7,380797400838736** |
| Knn-var-pMedio-media | 8,197846446016342 | **7,361196385573979** |
| Knn-var-pMedio-ponderada | 8,135961824690835 | **7,400040837810735** |
| Knn-ecm-media-media | **7,979832478147256** | **7,361776644479191** |
| Knn-ecm-media-ponderada | **7,863469197637423** | **7,41593440477786** |
| Knn-ecm-mediaDesv-media | 8,352184597212492 | 7,9976349197675445 |
| Knn-ecm-mediaDesv-ponderada | **7,9183628105194375** | **7,45385360707081** |
| Knn-ecm-pMedio-media | 8,261084940937817 | 7,980689772974664 |
| Knn-ecm-pMedio-ponderada | 8,098523756867682 | **7,421819666926275** |
| Knn-sea-media-media | **7,919680408** | 7,616036611503229 |
| Knn-sea-media-ponderada | **7,6830225229402345** | **7,51448416025585** |
| Knn-sea-mediaDesv-media | 8,307745941315993 | 7,567012712535724 |
| Knn-sea-mediaDesv-ponderada | **7,944906936754748** | **7,448886234182747** |
| Knn-sea-pMedio-media | 8,240978076997171 | 7,61083106959758 |
| Knn-sea-pMedio-ponderada | 8,409646642347798 | **7,452690125140899** |
| Perfiles-var-media-media | 8,516981821175154 | **7,40420346709697** |
| Perfiles-var-media-ponderada | 8,523645221560107 | **7,456678369561352** |
| Perfiles-var-mediaDesv-media | **8,035583019471561** | **7,30975708299075** |
| Perfiles-var-mediaDesv-ponderada | 8,333551474549122 | **7,376559470389795** |
| Perfiles-var-pMedio-media | 8,301882786560704 | **7,3471747477448** |
| Perfiles-var-pMedio-ponderada | 8,441520011402192 | **7,3929605901937325** |
| Perfiles-ecm-media-media | **7,835544123366221** | **7,333075886844486** |
| Perfiles-ecm-media-ponderada | **7,847361958398556** | **7,436031003611329** |
| Perfiles-ecm-mediaDesv-media | 8,352525135263662 | **7,318244270150939** |
| Perfiles-ecm-mediaDesv-ponderada | **7,960815307774352** | **7,353844371812556** |
| Perfiles-ecm-pMedio-media | 8,158414666402482 | **7,348250288454723** |
| Perfiles-ecm-pMedio-ponderada | 8,112862109047345 | **7,347893602355785** |
| Perfiles-sea-media-media | **7,657383086963963** | **7,48670843956256** |
| Perfiles-sea-media-ponderada | **7,673955653781883** | **7,5236498122650115** |
| Perfiles-sea-mediaDesv-media | 8,368802296918787 | **7,284760676084693** |
| Perfiles-sea-mediaDesv-ponderada | **7,989652456060542** | **7,381495075437391** |
| Perfiles-sea-pMedio-media | **7,929931537781511** | **7,4121919628257285** |
| Perfiles-sea-pMedio-ponderada | 8,093863227 | **7,449379107306649** |

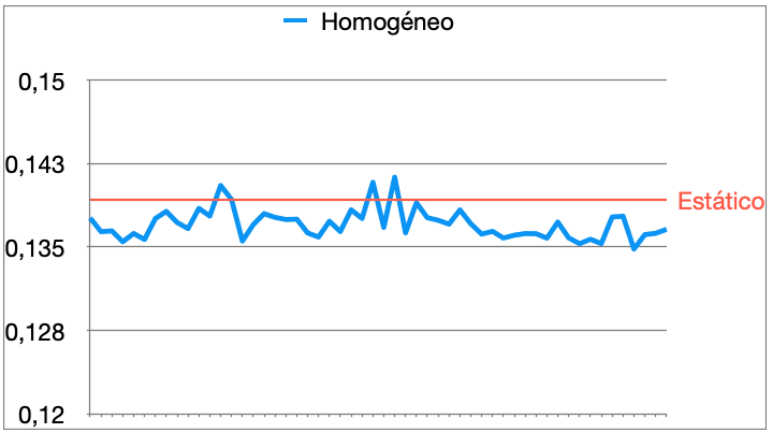


Figura 5.16: Gráfica MAPE con %DSEL = 50 HOMOGÉNEO Fuente: Elaboración Propia.

Los mejores métodos en este caso han resultado ser: ***Perfiles-sea-mediaDesv-media***, *Perfiles-ecm-mediaDesv-ponderada* y *Perfiles-ecm-pMedio-ponderada*. Destaca que en todos se calcula la región de competencia mediante el uso de perfiles de salida.

Por último, con respecto a esta prueba, cabe destacar que para el error RMSE (tabla 5.23), en ensembles heterogéneos los mejores resultados han sido obtenido por los métodos ***Perfiles-sea-media-media***, *Cluster-sea-media-ponderada* y *Perfiles-sea-media-ponderada*, coincidiendo estos dos últimos con dos de los que también han resultado mejores en ensembles heterogéneos respecto al error MAPE (en esta misma prueba), y para los ensembles homogéneos, los mejores resultaodos de RMSE los han obtenido los métodos ***Cluster-var-pMedio-media***, *Perfiles-sea-mediaDesv-media* y *Perfiles-var-mediaDesv-media*, de los cuales solo el último coincide con uno de los mejores en MAPE de ensembles homogéneos (dentro de esta prueba).

**Prueba 3**

La último prueba dentro de este experimento, consiste en cambiar los valores de los parámetros que controlan el porcentaje de DSEL y el solapamiento. En este caso, el porcentaje de DSEL será 95 % y sí se permite el solapamiento (como indica la tabla 5.4).

Los resultados obtenidos en esta prueba se exponen en las tablas 5.20 y 5.21, representando el error MAPE y RMSE, respectivamente, cometido por los métodos dinámicos implementados.

Haciendo referencia al error MAPE de la tabla 5.20 para los ensembles heterogéneos, se ha creado la gráfica 5.17. En esta, que se usa a modo orientativo, vemos que con respecto a la gráfica de la prueba anterior (gráfica 5.15) casi no cambia. Ambas gráficas representan

Tabla. 5.24: RE-MAPE-dsel20

| | Dinámico | |
|---|---|---|
| **Método** | **Heterogéneo** | **Homogéneo** |
| Cluster-var-media-media | 0,171645186754041 | 0,14421080624882185 |
| Cluster-var-media-ponderada | **0,16612143983957442** | 0,1423021996303831 |
| Cluster-var-mediaDesv-media | **0,16995370826570758** | 0,14266293927121765 |
| Cluster-var-mediaDesv-ponderada | **0,16828029317321352** | 0,1409262179675638 |
| Cluster-var-pMedio-media | **0,16568095161875807** | 0,1422476426177548 |
| Cluster-var-pMedio-ponderada | **0,16479987786880979** | 0,1414393762312211 |
| Cluster-ecm-media-media | **0,15862552134924848** | 0,14230782416736093 |
| Cluster-ecm-media-ponderada | **0,15190605136952923** | 0,1399294377561835 |
| Cluster-ecm-mediaDesv-media | 0,17616559148875519 | 0,14282417591567084 |
| Cluster-ecm-mediaDesv-ponderada | **0,16061007088098747** | 0,14022447782798134 |
| Cluster-ecm-pMedio-media | **0,16616517810182818** | 0,14540177761552778 |
| Cluster-ecm-pMedio-ponderada | **0,15681196349996926** | 0,14036439828705252 |
| Cluster-sea-media-media | **0,15548857798704085** | 0,14054781064929636 |
| Cluster-sea-media-ponderada | **0,1500070217507776** | 0,13823345176184296 |
| Cluster-sea-mediaDesv-media | 0,17562792044689024 | 0,14252588490517798 |
| Cluster-sea-mediaDesv-ponderada | **0,16085534564352325** | 0,1404681805475731 |
| Cluster-sea-pMedio-media | **0,15935866490007183** | 0,1411299217790964 |
| Cluster-sea-pMedio-ponderada | **0,15197469592450236** | 0,1405176398556732 |
| Knn-var-media-media | 0,1770674937940494 | 0,14316624967638605 |
| Knn-var-media-ponderada | 0,17772220871089048 | 0,14196356793254292 |
| Knn-var-mediaDesv-media | **0,16547747296494375** | 0,14253414425022304 |
| Knn-var-mediaDesv-ponderada | 0,1708674565065706 | 0,14132095823471053 |
| Knn-var-pMedio-media | **0,1620430663707265** | 0,14250609844796533 |
| Knn-var-pMedio-ponderada | **0,16906895906454594** | 0,14139467663765898 |
| Knn-ecm-media-media | **0,15876090182681177** | 0,14170504019548705 |
| Knn-ecm-media-ponderada | **0,15198901801773307** | 0,13956831945929923 |
| Knn-ecm-mediaDesv-media | 0,17564677718933605 | 0,1436749487948698 |
| Knn-ecm-mediaDesv-ponderada | **0,1612427237420794** | 0,1403824496414526 |
| Knn-ecm-pMedio-media | **0,16899587050536885** | 0,14542263349615062 |
| Knn-ecm-pMedio-ponderada | **0,15668563925302528** | 0,14089732375047428 |
| Knn-sea-media-media | **0,15588349917866923** | **0,13940710682035423** |
| Knn-sea-media-ponderada | **0,15084303023774465** | **0,13830810466263846** |
| Knn-sea-mediaDesv-media | 0,17449114508012362 | 0,14163023464296867 |
| Knn-sea-mediaDesv-ponderada | **0,16118559699035642** | 0,1398312476852836 |
| Knn-sea-pMedio-media | **0,15904635996665734** | 0,14102416283198155 |
| Knn-sea-pMedio-ponderada | **0,1524234565062443** | 0,13928637181825257 |
| Perfiles-var-media-media | 0,18038304161379548 | 0,14274412204819747 |
| Perfiles-var-media-ponderada | 0,18033379225893578 | 0,14255892886793525 |
| Perfiles-var-mediaDesv-media | **0,1650840869187662** | 0,1415471678694052 |
| Perfiles-var-mediaDesv-ponderada | 0,1739763164581405 | 0,14148816306722495 |
| Perfiles-var-pMedio-media | **0,16280283067792342** | 0,14236057770234542 |
| Perfiles-var-pMedio-ponderada | 0,1728426396086092 | 0,14236053224570314 |
| Perfiles-ecm-media-media | **0,15738686351783818** | **0,13900109340274683** |
| Perfiles-ecm-media-ponderada | **0,15137936321860027** | **0,13818314410209004** |
| Perfiles-ecm-mediaDesv-media | 0,17536775714744826 | 0,14181886608271976 |
| Perfiles-ecm-mediaDesv-ponderada | **0,1626907340036464** | 0,13976782511777433 |
| Perfiles-ecm-pMedio-media | **0,1655953698779796** | 0,14153375397701048 |
| Perfiles-ecm-pMedio-ponderada | **0,15559043626742125** | 0,13939114311630232 |
| Perfiles-sea-media-media | **0,15181302484044326** | **0,13862601093908394** |
| Perfiles-sea-media-ponderada | **0,14739254230307386** | **0,13778220903694083** |
| Perfiles-sea-mediaDesv-media | 0,17554526695590897 | 0,14177702556053834 |
| Perfiles-sea-mediaDesv-ponderada | **0,16115637754013007** | 0,13969463847797375 |
| Perfiles-sea-pMedio-media | **0,15543201529985337** | 0,13947353443775384 |
| Perfiles-sea-pMedio-ponderada | **0,14941154314796698** | **0,13809004162995947** |

Tabla. 5.25: RE-RMSE-dsel20

| | Dinámico | |
|---|---|---|
| **Método** | **Heterogéneo** | **Homogéneo** |
| Cluster-var-media-media | 8,30649700848241 | 7,704163886798723 |
| Cluster-var-media-ponderada | 8,265117228068377 | **7,52967046109841** |
| Cluster-var-mediaDesv-media | 8,070847884408249 | 7,566888232047338 |
| Cluster-var-mediaDesv-ponderada | 8,207729388494315 | **7,517428413367071** |
| Cluster-var-pMedio-media | **8,027462071796002** | **7,494884989521383** |
| Cluster-var-pMedio-ponderada | 8,169830869158016 | **7,435292575238651** |
| Cluster-ecm-media-media | **7,968293636385006** | 7,6562369375740955 |
| Cluster-ecm-media-ponderada | **7,786290960921946** | **7,4723518796970625** |
| Cluster-ecm-mediaDesv-media | 8,341131960647427 | 7,551856513088746 |
| Cluster-ecm-mediaDesv-ponderada | **7,976089672265682** | **7,450067166149376** |
| Cluster-ecm-pMedio-media | 8,099249238550865 | **7,500921043989199** |
| Cluster-ecm-pMedio-ponderada | **7,870111595383638** | **7,395336864620814** |
| Cluster-sea-media-media | **7,850610923531749** | 7,537604533514927 |
| Cluster-sea-media-ponderada | **7,717684168786123** | **7,409001539461988** |
| Cluster-sea-mediaDesv-media | 8,307200716110176 | **7,487044265208117** |
| Cluster-sea-mediaDesv-ponderada | **7,984201650502355** | **7,467552498186981** |
| Cluster-sea-pMedio-media | **7,881949050703628** | **7,425615332321007** |
| Cluster-sea-pMedio-ponderada | **7,705484385972855** | **7,413262695955325** |
| Knn-var-media-media | 8,334946833363704 | 7,54910735173443 |
| Knn-var-media-ponderada | 8,391687460362554 | **7,458468291297396** |
| Knn-var-mediaDesv-media | **8,04815340109169** | 7,580783607142675 |
| Knn-var-mediaDesv-ponderada | 8,23404955677265 | **7,475315398805025** |
| Knn-var-pMedio-media | **7,9241398369180045** | 7,627054809397206 |
| Knn-var-pMedio-ponderada | 8,121420311896037 | **7,5020193567459525** |
| Knn-ecm-media-media | **7,962246145739435** | **7,4170047219429165** |
| Knn-ecm-media-ponderada | **7,777064701598951** | **7,390285734546914** |
| Knn-ecm-mediaDesv-media | 8,321018959054898 | **7,531848738866666** |
| Knn-ecm-mediaDesv-ponderada | **7,988272309281773** | **7,431906683608588** |
| Knn-ecm-pMedio-media | 8,153185072531947 | 7,632439920411085 |
| Knn-ecm-pMedio-ponderada | **7,85467637767304** | **7,4612352201807415** |
| Knn-sea-media-media | **7,88227970184446** | **7,395391607900696** |
| Knn-sea-media-ponderada | **7,745860834744882** | **7,3750807539156416** |
| Knn-sea-mediaDesv-media | 8,303011088451568 | **7,509235311176473** |
| Knn-sea-mediaDesv-ponderada | **7,995049952376425** | **7,432643170092878** |
| Knn-sea-pMedio-media | **7,9157618713395435** | **7,488294965815109 4** |
| Knn-sea-pMedio-ponderada | **7,746571672064808** | **7,409520621213366** |
| Perfiles-var-media-media | 8,534697575520362 | 7,6010300102456325 |
| Perfiles-var-media-ponderada | 8,543740456181851 | 7,548891711590265 |
| Perfiles-var-mediaDesv-media | 8,073508838482178 | **7,479866946417543** |
| Perfiles-var-mediaDesv-ponderada | 8,343387281072276 | **7,465323479227479** |
| Perfiles-var-pMedio-media | 8,080835895733525 | **7,472122630235601** |
| Perfiles-var-pMedio-ponderada | 8,337123962301854 | **7,501807198976266** |
| Perfiles-ecm-media-media | **8,01077513338889** | **7,331911495306407** |
| Perfiles-ecm-media-ponderada | **7,844964634650535** | **7,35582339571845** |
| Perfiles-ecm-mediaDesv-media | 8,314322742289864 | **7,526381860626158** |
| Perfiles-ecm-mediaDesv-ponderada | 8,05953578652365 | **7,435013450253078** |
| Perfiles-ecm-pMedio-media | 8,15696542193161 | **7,443389027295543** |
| Perfiles-ecm-pMedio-ponderada | **7,931538377202673** | **7,397046123703156** |
| Perfiles-sea-media-media | **7,897684836525539** | **7,273478064664284** |
| Perfiles-sea-media-ponderada | **7,746734931150263** | **7,309410235420893** |
| Perfiles-sea-mediaDesv-media | 8,294875450240829 | **7,440873218629797** |
| Perfiles-sea-mediaDesv-ponderada | **8,011057518207405** | **7,381235413535056** |
| Perfiles-sea-pMedio-media | **7,890139185724432** | **7,329735275046646** |
| Perfiles-sea-pMedio-ponderada | **7,751152091749013** | **7,322176079150173** |

buenos resultados, mejoran en gran medida el error MAPE del estático, pero entre ellas apenas encontramos diferencia.

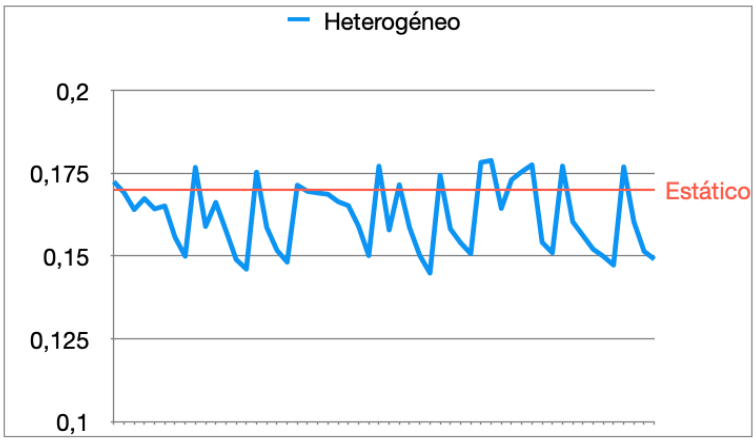


Figura 5.17: Gráfica MAPE con %DSEL = 95 HETEROGÉNEO Fuente: Elaboración Propia.

En particular, siguiendo los resultados de la tabla 5.20, los mejores métodos para el ensemble heterogéneo han resultado ser: ***Knn-sea-media-ponderada***, *Cluster-sea-media-ponderada* y *Perfiles-sea-media-ponderada*, siendo el primero el que menor error obtiene. Cabe destacar que ninguno de estos tres métodos ha conseguido disminuir el error mínimo que ya se conseguía con el mejor método de la prueba anterior (con los parámetros configurados para la prueba 2).

Por otro lado, si observamos los resultados de la tabla 5.20 para los ensembles homogéneos, y ayudándonos de la gráfica 5.18 creada a partir de ellos, vemos que en este caso los resultados tampoco varían demasiado con respecto a la prueba anterior (referida a los ensembles homogéneos), como ha ocurrido con el ensemble heterogéneo.

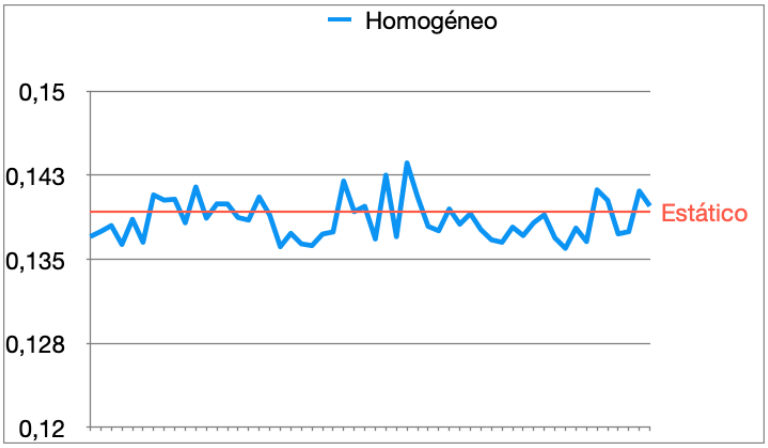


Figura 5.18: Gráfica MAPE con %DSEL = 95 HOMOGÉNEO Fuente: Elaboración Propia.

Los tres mejores métodos dinámicos en este caso han sido: ***Perfiles-ecm-mediaDesv-ponderada***, *Knn-var-media-media* y *Knn-var-mediaDesv-ponderada*.

Para concluir, revisando la tabla referente al error RMSE, vemos que para el ensemble heterogéneo, los tres mejores métodos dinámicos son ***Perfiles-ecm-mediaDesv-media***, *Perfiles-var-mediaDesv-media*, *Perfiles-var-pMedio-media* coincidiendo con dos de los mejores en MAPE, mientras que para el homogéneo, los mejores han resultado ser ***Perfiles-ecm-mediaDesv-media***, *Perfiles-var-mediaDesv-media* y *Perfiles-var-pMedio-media*, sin coincidir ninguno con los mejores del MAPE.

**Conclusiones del experimento 3**

Durante este experimento, en los ensembles heterogéneos dinámicos, se observa que los resultados no presentan mucha variación de una prueba a otra. Sin embargo, cabe destacar que ya se obtienen buenos resultados desde la primera prueba. Con respecto al homogéneo dinámico, ocurre de manera similar, los resultados no varían demasiado con respecto a la configuración de los parámetros de las pruebas. No obstante, para el homogéneo sí que se nota una mejoría más notoria que en el heterogéneo (en términos generales).

Se observa en esta experimentación que, tanto para ensembles heterogéneos como para homogéneos (se ve más claro para estos últimos), ha resultado mejorar más los resultados la configuración de la prueba 2, con el porcentaje de DSEL al 50 % y solapamiento del DSEL con entrenamiento permitido.

Los mejores resultados obtenidos por ensembles dinámicos durante la experimentación han sido: 0,145238 de MAPE para el ensemble heterogéneo (frente a 0,1701306 obtenido por el ensemble estático equivalente) y 0,13480 de MAPE para el ensemble homogéneo dinámico (frente a 0,139241542). Con respecto a estos resultados, el heterogéneo consigue mejorar en mayor medida a su equivalente estático. Los mejores resultados de ambos pertenecen a los resultados obtenidos en la prueba 2.

Por último, destaca que para este dataset los mejores resultados son los obtenidos en la prueba 2,con el DSEL al 50 % y con solapamiento permitido. Por lo que se deduce que para este conjunto de datos esta sería la mejor configuración encontrada.

## 5.3. Análisis general

Con el comentario final, se pretende dar una conclusión más generalizada del rendimiento de los métodos dinámicos que se han implementado en la librería.

Para ello, se ha creado un *ranking* de dichos métodos, teniendo también en cuenta el estático. Dicho *ranking* se ha obtenido a partir de las posiciones en las que se han quedado los distintos métodos durante las diversas pruebas y experimentos, con respecto al error MAPE. Se utiliza este, ya que los resultados son más manejables y, al representar un error porcentual, son más interpretables.

A este ranking también se añade la desviación típica para tener más información sobre el comportamiento que han tenido los distintos métodos durante la experimentación.

Como es evidente, el *ranking* estará ordenado de forma ascendente, pues los que consigan una puntuación menor de *ranking* serán los que han conseguido durante la experimentación puestos menores (1, 2, 3, 4,...), es decir, mejores.

El *ranking* resultado es el siguiente 5.26:

El *ranking* se ha ordenado primero por la media de las posiciones obtenida por los métodos en los experimentos y, en segundo lugar, por la desvicación típica de manera ascendente. Se entiende que una menor desviación típica refleja que ese método no se ha movido por posiciones muy dispares en los experimentos (es decir, que en una prueba consiga ser el segundo mejor, y en otra prueba se quede en el puesto cincuenta), es decir, se ha mantenido en un rango. Esto implica que el método es más estable independientemente del conjunto de datos que se está evaluando y los parámetros del porcentaje de DSEL y solapamineto de los métodos dinámicos.

Como se puede observar en la tabla 5.26, los tres mejores métodos obtienen la región de competencia mediante el uso de perfiles de salida. Esto nos hace llegar a la conclusión de que en general, los perfiles de salida funcionan mejor porque utilizan todo el DSEL para obtener la región de competencia. Como ya se ha explicado, la región de competencia es problema-dependiente, por lo que, tiene sentido que en este *ranking* general en el que se están teniendo en cuenta distintos datasets, ocupen los primero puestos aquellos métodos que determinan la región de competencia mediante perfiles de salida, pues es el único método que se basan directamente en los datos (los umbrales de similitud que se definen son dinámicos).

Por otro lado, cabe destacar que en los 22 primeros puestos, encontramos que los niveles de competencia son calculados mediante los métodos que se basan en algún tipo de error, concretamente, en el error cuadrático medio (ECM en español o MSE en inglés)

Tabla. 5.26: Ranking métodos dinámicos implementados.

| Método | Ranking | Desviación típica |
|---|---|---|
| Perfiles-ecm-pMedio-ponderada | **8,222** | 6,844 |
| Perfiles-ecm-media-ponderada | **8,222** | 13,096 |
| Perfiles-sea-media-ponderada | **9,278** | 19,839 |
| Cluster-sea-media-media | **9,333** | 16,956 |
| Knn-sea-media-ponderada | **9,833** | 19,166 |
| Perfiles-ecm-mediaDesv-ponderada | 10,056 | 11,532 |
| Cluster-sea-media-ponderada | 10,111 | 19,960 |
| Cluster-sea-pMedio-ponderada | 11,000 | 13,054 |
| Knn-sea-pMedio-ponderada | 11,389 | 15,876 |
| Perfiles-sea-pMedio-ponderada | 11,389 | 16,120 |
| Knn-ecm-pMedio-ponderada | 11,778 | 5,865 |
| Perfiles-sea-media-media | 11,833 | 18,921 |
| Knn-ecm-media-ponderada | 11,889 | 16,372 |
| Perfiles-sea-pMedio-media | 12,000 | 13,858 |
| Knn-ecm-mediaDesv-ponderada | 12,500 | 10,470 |
| Cluster-ecm-media-ponderada | 12,500 | 19,277 |
| Perfiles-ecm-media-media | 12,722 | 16,408 |
| Perfiles-sea-mediaDesv-ponderada | 12,889 | 11,084 |
| Knn-sea-mediaDesv-ponderada | 12,889 | 12,282 |
| Cluster-ecm-mediaDesv-ponderada | 13,611 | 8,960 |
| Knn-var-pMedio-media | 13,944 | 10,429 |
| Perfiles-var-mediaDesv-media | 13,944 | 14,700 |
| Knn-var-pMedio-ponderada | 14,056 | 8,460 |
| Cluster-var-pMedio-ponderada | 14,167 | 13,657 |
| Cluster-sea-mediaDesv-ponderada | 14,333 | 9,211 |
| Cluster-var-pMedio-media | 14,444 | 10,055 |
| Knn-var-mediaDesv-ponderada | 14,444 | 13,408 |
| Cluster-var-mediaDesv-ponderada | 14,444 | 13,776 |
| Cluster-ecm-pMedio-ponderada | 14,500 | 8,837 |
| Knn-sea-pMedio-media | 15,278 | 14,121 |
| Knn-var-media-media | 15,389 | 16,654 |
| Knn-sea-media-media | 15,667 | 17,546 |
| Knn-var-mediaDesv-media | 16,056 | 15,198 |
| Cluster-sea-pMedio-media | 16,056 | 16,656 |
| Perfiles-var-pMedio-media | 16,556 | 13,823 |
| Perfiles-ecm-pMedio-media | 16,667 | 15,011 |
| Cluster-var-mediaDesv-media | 17,056 | 11,945 |
| Perfiles-var-mediaDesv-ponderada | 17,444 | 15,094 |
| Perfiles-var-media-media | 18,611 | 16,854 |
| Cluster-var-media-ponderada | 18,722 | 11,255 |
| Cluster-var-media-media | 18,778 | 14,164 |
| Cluster-ecm-media-media | 19,222 | 11,433 |
| Knn-var-media-ponderada | 19,278 | 12,781 |
| Perfiles-var-pMedio-ponderada | 19,722 | 15,558 |
| Perfiles-ecm-mediaDesv-media | 20,333 | 15,086 |
| Perfiles-sea-mediaDesv-media | 20,444 | 16,732 |
| Perfiles-var-media-ponderada | 20,444 | 17,819 |
| Cluster-sea-mediaDesv-media | 20,500 | 14,533 |
| Knn-ecm-media-media | 21,389 | 10,794 |
| Cluster-ecm-pMedio-media | 21,500 | 18,732 |
| Knn-sea-mediaDesv-media | 21,778 | 13,779 |
| Knn-ecm-mediaDesv-media | 24,167 | 14,902 |
| Estático | 24,667 | 14,711 |
| Cluster-ecm-mediaDesv-media | 24,778 | 9,698 |
| Knn-ecm-pMedio-media | 24,944 | 18,115 |

y la suma de errores absolutos (SAE). Por lo que se puede abstraer que la razón de ello es porque las medidas de error verdaderamente ofrecen información sobre la “bondad“ del regresor en la región de competencia, frente al cálculo de los niveles de competencia a partir de la varianza, que solo nos ofrece cómo de dispersos están siendo los resultados que obtiene el regresor.

Con respecto a la selección de los regresores base, es decir, el umbral de selección, 4 de los 5 mejores métodos utilizan la media. Parece ser que, aun siendo una manera sencilla de calcular un umbral, este es bastante efectivo.

Por último, 12 de los 13 mejores métodos, utilizan la media ponderada para la agregación de salidas. Se concluye con esto que, se consiguen buenos resultados debido a que se está dando más peso a la predicción de aquellos regresores que han conseguido un nivel de competencia mayor. Por lo tanto, se corrobora la idea de la región de competencia. Parece ser que sí que se acierta en la hipótesis de que si un regresor es bueno en instancias parecidas a la de prueba, este regresor también lo será para dicha instancia de prueba.

# Capítulo 6

# CONCLUSIONES

Durante este Trabajo Fin de Grado, se ha desarrollado una librería que implementa 54 métodos distintos mediante los que se llevan a cabo la generación, selección y agregación de ensembles dinámicos de modelos *Machine Learning* para regresión. Esta idea surge debido a que se identifica una carencia de librerías que implementen este tipo de técnicas de ensembles dinámicos para problemas de regresión.

Para desarrollarla, se realiza una extensa revisión bibliográfica en la que se pone en antecendente el estado del arte del área de los ensembles dinámicos en general. Mediante el análisis de dicha revisión bibliográfica, van tomando forma las distintas decisiones tomadas en el diseño y desarrollo de la librería.

Finalmente, para comprobar la funcionalidad de la misma y confirmar que se trata de una técnica que obtiene buenos resultados, se llevan a cabo diversas pruebas.

Con dichas pruebas, se corrobora que la librería de métodos de ensembles dinámicos de modelos *Machine Learning* para regresión desarrollada, obtiene resultados muy competitivos frente a técnicas de ensembles tradiciones, es decir, estáticas.

Por último, destacar que durante la experimentación se descubre que el método que ha obtenido mejores resultados generales es *Perfiles-ecm-pMedio-pond*, donde se calcula la región de competencia mediante perfiles de salida, los niveles de competencia se obtienen mediante el cálculo del error cuadrático medio cometido por cada uno en la región de competencia, la selección se basa en el umbral obtenido por el punto medio y la agregación de salidas se realiza mediante la media ponderada.

# Capítulo 7

# Anexo I: Manual de usuario

Este anexo está dedicado al manual de usario de la librería implementada, con el fin de que el usuario le pueda dar un uso correcto a la misma.

## 7.1. Instalación

El primer paso para poder utilizar la librería es instalarla en nuestro equipo, concretamente en el directorio en que se va a trabajar.

El único requisito que se necesita para poder utilizara la librería es tener preinstalado Python en nuestro equipo. Si no es así, en el siguiente enlace se explica detalladamente cómo hacerlo https://tutorial.djangogirls.org/es/python_installation/.

Una vez tenemos instalada la librería en nuestro equipo, ya podemos disfrutar de toda su funcionalidad.

## 7.2. Manual de uso de la librería

Para poder utilizar los distintos métodos de ensembles dinámicos implementados en la librería, basta con crear un objeto a partir de la clase *DynamicEnsemble.py*.

Los parámetros del constructor de la clase DynamicEnsemble.py, junto con sus valores por defecto, son los siguientes :

- pool_regresores=None –>Parámetro en el que se indica el conjunto de regresores base en forma de vector. Los regresores base propuestos, si no se van a someter a proceso de Bagging interno (baagging_base=False, deben introducirse en este parámetro ya habiendo sido entrenados previamente cada uno de ellos.

  Si el valor de este parámetro es None, significa que no se le pasa ningún regresor base y por lo tanto, se implementa internamente un ensemble homogéneo mediante de un proceso *Bagging* a partir del algoritmo por defecto DecissionTreeRegressor() de *Scikit-Leanr*.

- k=7 –>Parámetro que, para métodos donde la región de competencia se define mediante KNN, hace referencia al número de vecinos, mientras que si el método utiliza *clustering*, este parámetro define el número de grupos. Por último, si para el cálculo de. la región de competencia se utilizan perfiles de salida, este parámetro es ignorado.

- random_state=None –>Indica la semilla para un aleatorio.

- DSEL_perc=0.50 –>Parámetro con el que se controla el porcentaje de DSEL a utilizar con respecto al conjunto de entrenamiento.

- DSEL_solap=False –>Parámetro que controla si hay solapamiento entre el conjunto DSEL y el de entrenamiento. El valor *True* indicará que el conjunto de entrenemiento se mantiene con el mismo porcentaje del muestras (100 %) y el se solapa con este.

- metodo_region = "knn->Parámetro con el que se indica qué método se utiliza para el cálculo de la región de competencia. Las posibles opciones son “knn", “clusterz “perfiles", que implementan el KNN, KMEANS y cálculo de similitud de perfiles de salida.

- metodo_nivel=0 –>Parámetro que define cómo se van a medir los niveles de competencia de los regresores base. Existen tres posibles valores: 0, para usar la varianza; 1, para utilizar el error cuadrático medio; y 2, para utilizar la suma de errores absolutos.

- metodo_sel=0 –>Parámetro con el que se elige cómo calcular el umbral de selección para los niveles de competencia. Este puede tomar tres valores: 0, que significaría obtener el umbral mediante la media aritmética de los niveles de competencia; 1, para conseguir el umbral a partir de la media más la desviación típica; y 2, si se quiere obtener el umbral mediante el punto medio definido por los niveles de competencia.

- metodo_agregacion=0–>Este parámetro es el que define el método por el que las salidas del ensemble van a ser agregadas. Solo tiene dos posibles valores: 0 y 1, que se refieren a la media ponderada y la media aritmética, respectivamente.

- bagging_base=False –>Este parámetro sirve para controlar si se realiza el *Bagging* interno a cada regresor propuesto en el parámetro *pool_regresores*. (CUIDADO: Si no se determina ningún regresor base en el parámetro *pool_regresores*, se implementa de manera automática un proceso *Bagging* para obtener un ensemble homogeneo. En ese caso, dejar este parámetro por defecto, es decir, igual a False)

Como se puede observar, los 54 métodos implementados se pueden elegir de manera fácil y cómoda a partir de todas las combinaciones de los parámetros *metodo_region*, *metodo_nivel*, *metodo_sel* y *metodo_agregacion*.

Teniendo ya un ensemble dinámico creado, el siguiente paso es entrenarlo.

Para entrenarlo hay que hacer una llamada a la función *.fit(X,y)*, donde *X* serán las entradas del conjunto de entrenamiento e *y* las salidas.

IMPORTANTE: Este método internamente realiza la partición de DSEL y training a partir de los parámetros de entrada $X$ e $y$. Sin embargo, si no se quiere utilizar el *Bagging* interno (que se indicaría mediante *bagging_base = False*) solo en el caso de haber indicado regresores base en el parámetro *pool_regresores*, los parámetros de entrada $X$ e $y$ ya deben ser el conjunto de DSEL, debido a que los regresores que se introducen en el parámetro *pool_regresores* ya deben estar entrenados previamente con el conjunto de entrenamiento que se obtiene de la partición en DSEL y conjunto de entrenamiento. Así pues, dicha partición, en este caso, debe ser realizada por el usuario previamente, al igual que el entrenamiento de los regresores.

Por lo tanto, en este caso solo se entrena el ensemble dinámico en sí y por ello solo se

necesita el conjunto de DSEL, pues los regresores base ya deben introducirse entrenados.

Una vez entrenado el ensemble dinámico, se pueden utilizar las siguientes funciones:

- *predecir_muestra(X_test)*: El parámetro de entrada *X_test* es la muestra que se quiere predecir. Como salida, este método devuelve la predicción realizada por el ensemble dinámico mediante el método que se haya definido con los parámetros *metodo_region*, *metodo_nivel*, *metodo_sel* y *metodo_agregacion*.
- *score_ensemble(X_test, y_test)*: Esta función tiene como parámetros de entrada las variables el conjunto de test (*X_test*) y las salidas del mismo. (*y_test*). En este método se calculan los errores MAPE y MSE cometidos por el ensemble dinámico.Por lo tanto, devuelve dos valores, el error MAPE y el MSE, respectivamente.

## 7.3. Ejemplo de uso

Supongamos que queremos realizar un ensemble dinámico en el que la región de competencia se calcule mediante perfiles de salida, que los niveles de competencia se estimen mediante el error cuadrático media, que el umbral de selección se defina mediante la media de los niveles de competencia y por último, las salidas sean agregadas mediante la media ponderada. Además, no se va a introducir ningún regresor base. Los demás parámetros, se quedan con sus valores por defecto.

Un. ejemplo de uso con este tipo de ensemble dinámico sería:

```
#Creación del ensembles
ensemble_dinamico_homogeneo = DynamicEnsemble()
#Elección del  método dinámico mediante los parámetros
ensemble_dinamico_homogeneo.set_metodo_region_competencia("perfiles")
ensemble_dinamico_homogeneo.set_metodo_nivel_competencia(1)
ensemble_dinamico_homogeneo.set_metodo_seleccion_regresores(0)
ensemble_dinamico_homogeneo.set_metodo_agregacion(0)

X_train=[[2,2],[5,2],[2,2],[5,2],[3,1],[2,2],[5,3],[1,2],[2,2],[5,3],[1,2]]
y_train=[0.2,3,15,1,2,5,1,3,8,3,2]

#Entrenamiento
ensemble_dinamico_homogeneo.fit(X_train, y_train)

#Predicción de una muestra
print(ensemble_dinamico_homogeneo.predecir_muestra([[3,1]],2))

#Cálculo de errores
X_test=[[2,1],[3,2],[4,2]]
y_test=[0,10,3]

print(ensemble_dinamico_homogeneo.score_ensemble(X_test,  y_test))
```

Figura 7.1: Ejemplo creación y entrenamiento de ensemble dinámico mediante la librería. Fuente: Elaboración Propia.

# Bibliografía


Dave Evans. Internet de las cosas. *Cómo la próxima evolución de Internet lo cambia todo. Cisco Internet Bussiness Solutions Group-IBSG*, 11(1):4–11, 2011.

Jesús García, J Molina, Antonio Berlanga, M Patricio, A Bustamante, and W Padilla. Ciencia de datos. *Técnicas Analíticas y Aprendizaje Estadístico. Bogotá, Colombia. Publicaciones Altaria, SL*, 2018.

Albert HR Ko, Robert Sabourin, and Alceu Souza Britto Jr. From dynamic classifier selection to dynamic ensemble selection. *Pattern recognition*, 41(5):1718–1731, 2008.

Vasant Dhar. Data science and prediction. *Communications of the ACM*, 56(12):64–73, 2013.

Tom M Mitchell and Tom M Mitchell. *Machine learning*, volume 1. McGraw-hill New York, 1997.

Mike A. Kaggle: todo lo que hay que saber sobre esta plataforma, 2021. URL https://datascientest.com/es/kaggle-todo-lo-que-hay-que-saber-sobre-esta-plataforma.

Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

Usama M Fayyad, David Haussler, and Paul E Stolorz. Kdd for science data analysis: Issues and examples. In *KDD*, pages 50–56, 1996a.

Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–37, 1996b.

Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

Francisco Charte Ojeda and David Charte Luque. *Machine Learning y Ciencia de datos con Python y R*. 2020.

What Is Data Mining. Data mining: Concepts and techniques. *Morgan Kaufinann*, 10:559–569, 2006.

Damián Jorge Matich. Redes neuronales: Conceptos básicos y aplicaciones. *Universidad Tecnológica Nacional, México*, 41:12–16, 2001.

Sonia Singh and Priyanka Gupta. Comparative study id3, cart and c4. 5 decision tree algorithm: a survey. *International Journal of Advanced Information Science and Technology (IJAIST)*, 27(27):97–103, 2014.

Preeti Arora, Shipra Varshney, et al. Analysis of k-means and k-medoids algorithm for big data. *Procedia Computer Science*, 78:507–512, 2016.

David W Aha. *Lazy learning*. Springer Science & Business Media, 2013.

Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. Knn model-based approach in classification. In *OTM Confederated International Conferences."On the Move to Meaningful Internet Systems"*, pages 986–996. Springer, 2003.

William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

José Cristóbal Riquelme Santos, Roberto Ruiz, and Karina Gilbert. Minería de datos: Conceptos y tendencias. *Inteligencia Artificial: Revista Iberoamericana de Inteligencia Artificial, 10 (29), 11-18.*, 2006.

Irene Moral Peláez. Modelos de regresión: lineal simple y regresión logística. *Revista Seden*, 14:195–214, 2016.

Lior Rokach. Ensemble-based classifiers. *Artificial intelligence review*, 33(1):1–39, 2010.

Rising Odegua. An empirical study of ensemble techniques (bagging boosting and stacking). In *Proc. Conf.: Deep Learn. IndabaXAt*, 2019.

Rafael MO Cruz, Robert Sabourin, and George DC Cavalcanti. Dynamic classifier selection: Recent advances and perspectives. *Information Fusion*, 41:195–216, 2018.

Gulshan Kumar and Krishan Kumar. The use of artificial-intelligence-based ensembles for intrusion detection: a review. *Applied Computational Intelligence and Soft Computing*, 2012, 2012.

R.P.W. Duin. The combining classifier: to train or not to train? In *2002 International Conference on Pattern Recognition*, volume 2, pages 765–770 vol.2, 2002. doi: 10.1109/ICPR.2002.1048415.

Omer Sagi and Lior Rokach. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1249, 2018.

Paweł Zyblewski, Robert Sabourin, and Michał Woźniak. Preprocessed dynamic classifier ensemble selection for highly imbalanced drifted data streams. *Information Fusion*, 66: 138–154, 2021.

Reza Mousavi, Mahdi Eftekhari, and Farhad Rahdari. Omni-ensemble learning (oel): utilizing over-bagging, static and dynamic ensemble selection approaches for software defect prediction. *International Journal on Artificial Intelligence Tools*, 27(06):1850024, 2018.

Jiachao Wu, Jiang Shen, Man Xu, and Minglai Shao. A novel combined dynamic ensemble selection model for imbalanced data to detect covid-19 from complete blood count. *Computer Methods and Programs in Biomedicine*, 211:106444, 2021.

Wen-hui Hou, Xiao-kang Wang, Hong-yu Zhang, Jian-qiang Wang, and Lin Li. A novel dynamic ensemble selection classifier for an imbalanced data set: An application for credit risk assessment. *Knowledge-Based Systems*, 208:106462, 2020.

Thiago JM Moura, George DC Cavalcanti, and Luiz S Oliveira. On the selection of the competence measure for dynamic regressor selection. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1630–1637. IEEE, 2020.

Eraylson G Silva, Paulo SG De Mattos Neto, and George DC Cavalcanti. A dynamic predictor selection method based on recent temporal windows for time series forecasting. *IEEE Access*, 9:108466–108479, 2021.

Zhong-Liang Zhang, Yu-Yu Chen, Jing Li, and Xing-Gang Luo. A distance-based weighting framework for boosting the performance of dynamic ensemble selection. *Information Processing & Management*, 56(4):1300–1316, 2019.

Maciej Kinal and Michał Woźniak. Data preprocessing for des-knn and its application to imbalanced medical data classification. In *Asian Conference on Intelligent Information and Database Systems*, pages 589–599. Springer, 2020.

Rafael MO Cruz, Luiz G Hafemann, Robert Sabourin, and George DC Cavalcanti. Deslib: A dynamic ensemble selection library in python. *J. Mach. Learn. Res.*, 21(8):1–5, 2020.

Joao Mendes-Moreira, Alipio Mario Jorge, Carlos Soares, and Jorge Freire de Sousa. Ensemble learning: A study on different variants of the dynamic selection approach. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 191–205. Springer, 2009.

Joao Mendes-Moreira, Alípio Mário Jorge, Jorge Freire de Sousa, and Carlos Soares. Improving the accuracy of long-term travel time prediction using heterogeneous ensembles. *Neurocomputing*, 150:428–439, 2015.

Thiago JM Moura, George DC Cavalcanti, and Luiz S Oliveira. Mine: A framework for dynamic regressor selection. *Information Sciences*, 543:157–179, 2021.

Salvador Pita Fernández and Sonia Pértega Díaz. Relación entre variables cuantitativas. *Cad Aten Primaria*, 4:141–4, 1997.

Javier Canales Luna. Top programming languages for data scientists in 2022, 2022. URL https://www.datacamp.com/blog/top-programming-languages-for-data-scientists-in-2022.