UNIVERSIDAD EUROPEA DE MADRID

ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

GRADO EN MATEMATICAS APLICADAS AL ANÁLISIS DE DATOS

PROYECTO DE BIG DATA 1

EPANET

ANDREA BRAOJOS COGOLLUDO
LAURA POVEDA NICOLÁS
VÍCTOR RODRÍGUEZ ORTIZ
IGNACIO TORRES PRIEGO

CURSO 2024-2025

Índice

Capítulo 1. Descripción del proyecto

Nuestro proyecto consiste en predecir anomalías en la red eléctrica española, teniendo en cuenta el estado meteorológico, además de distintas características de la fecha (si es festivo, día de la semana, mes del año, etcétera). De esta manera podemos predecir en qué momento y en qué condiciones la previsión y la demanda real no son acordes y se clasifica como anomalía. Depende de la diferencia entre la previsión y la demanda real, se clasificará por niveles del 0 al 2, siendo el 0 que no hay anomalía, y el 2 siendo una anomalía significativa.

1.1 Contexto y justificación

Nos ubicamos en un contexto donde la eficiencia energética está a la orden del día en España. Ana Barillas, Head of Iberia, nos dice textualmente: “La congestión y otras limitaciones operativas en la red eléctrica representan un riesgo importante para el sector de energías renovables en España” (Barillas, 2023).

Esto saca a la luz un problema real en la operativa de la red eléctrica española a la hora de calcular la previsión demanda, generando anomalías a la hora de proveer la suficiente energía a la sociedad española.

Es de vital importancia poder predecir con mayor exactitud cuándo y dónde se van a producir dichas anomalías, con el fin de obtener una mejor calidad en el uso de la red.

1.2 Objetivos del proyecto

El objetivo principal de nuestro proyecto es utilizar técnicas avanzadas de machine learning para estimar la inestabilidad en la demanda de electricidad, detectando anomalías en la demanda en la red, clasificando por niveles en función del desajuste entre la previsión y la demanda real.

Además, identificar los factores externos clave que influyen en la volatilidad de la demanda eléctrica, como el estado meteorológico o características claves de la fecha.

Validar el modelo con datos históricos para evaluar su precisión en la predicción de desviaciones de la demanda.

Implementar un sistema de alerta temprana que avise sobre posibles desviaciones de la demanda en tiempo real.

Capítulo 2. Tecnologías

- API de Esios de Red Eléctrica Española
- Docker
- ElasticSearch (Postman)
- Lenguajes de desarrollo de página web (HTML, CSS y JavaScript)
- Leaflet para el desarrollo del mapa
- Código abierto de OpenStreetMap
- Python
    - Sklearn para la creación del modelo (ExtraTree)
    - FastAPI
    - Seaborn
    - Matplotlib
    - Pandas
    - Numpy

Capítulo 3. Desarrollo del proyecto

El desarrollo de nuestro proyecto se ha dividido en varias fases clave que abarcan desde la recopilación de datos hasta la implementación de un sistema de predicción accesible a través de una página web, que explicaremos a continuación en este capítulo.

3.1 Datos

En primer lugar, hemos extraído datos históricos de dos fuentes principales: la API de Red Eléctrica de España (Esios), que proporciona información sobre la demanda eléctrica y previsiones, y la API de la Agencia Estatal de Meteorología (AEMET), para obtener datos meteorológicos relevantes.

Para el caso de los datos extraídos de la API de Red Eléctrica Española, mediante el portal de Esios, hemos extraído diversos conjuntos de datos. Entre estos encontramos la generación energética, tanto renovable como no renovable a nivel peninsular. Por otro lado, extrajimos también datos sobre demanda, tanto real, como prevista, como programada a nivel peninsular. Sobre todos estos datos, tenemos los registros del valor, y en el momento en el que se registra, con un intervalo de 5 minutos. Los conjuntos de datos tienen registros desde 2019 hasta finales de 2024.

Por otro lado, tenemos los datos meteorológicos extraídos de la AEMET. En este caso, hemos extraído en forma de CSV todos los registros meteorológicos desde 2013 hasta 2024 a nivel diario a nivel de estación meteorológica. De cada registro podemos obtener información sobre la temperatura (media, máxima y mínima), la provincia en donde se ubica, el nivel de viento, nivel de precipitaciones, entre otra información relevante.

3.2 Descripción de la solución, metodologías y herramientas utilizadas

Nuestra solución consiste en ofrecer un sistema de predicción de anomalías en la demanda eléctrica. El objetivo principal es detectar, de manera anticipada, desviaciones significativas entre la demanda real de electricidad y las previsiones oficiales. De esta manera, podemos prever en qué momentos es más probable que se generen anomalías en la red, y estar preparado para cuando esto ocurra.

Para la resolución del proyecto, se han seguido una serie de etapas diferenciadas.

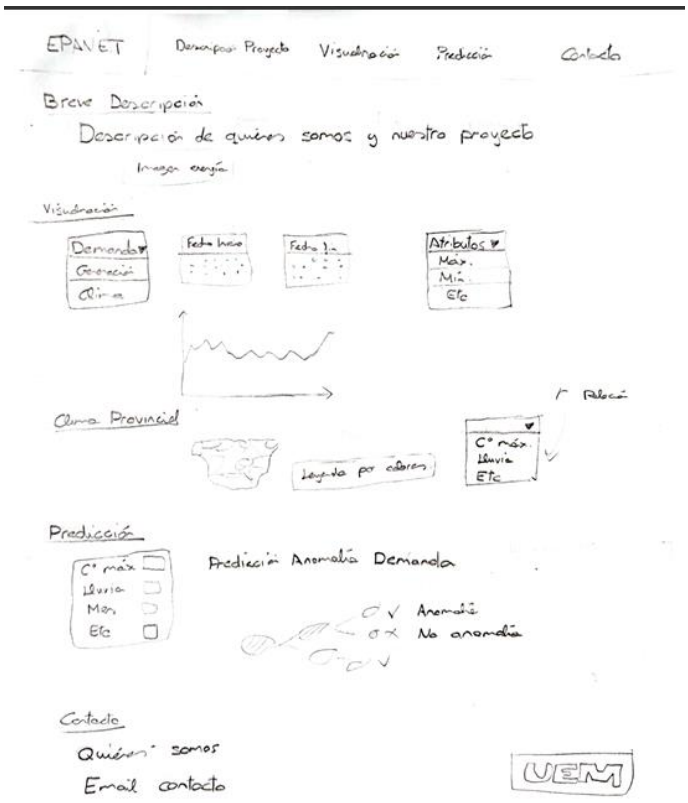
Primero de todo, la definición de nuestro proyecto y el saber exactamente qué íbamos a desarrollar. Para este punto, revisamos qué datos estaban disponibles en la página de REE, e hicimos una lluvia de ideas. Finalmente, decidimos centrarnos en la demanda

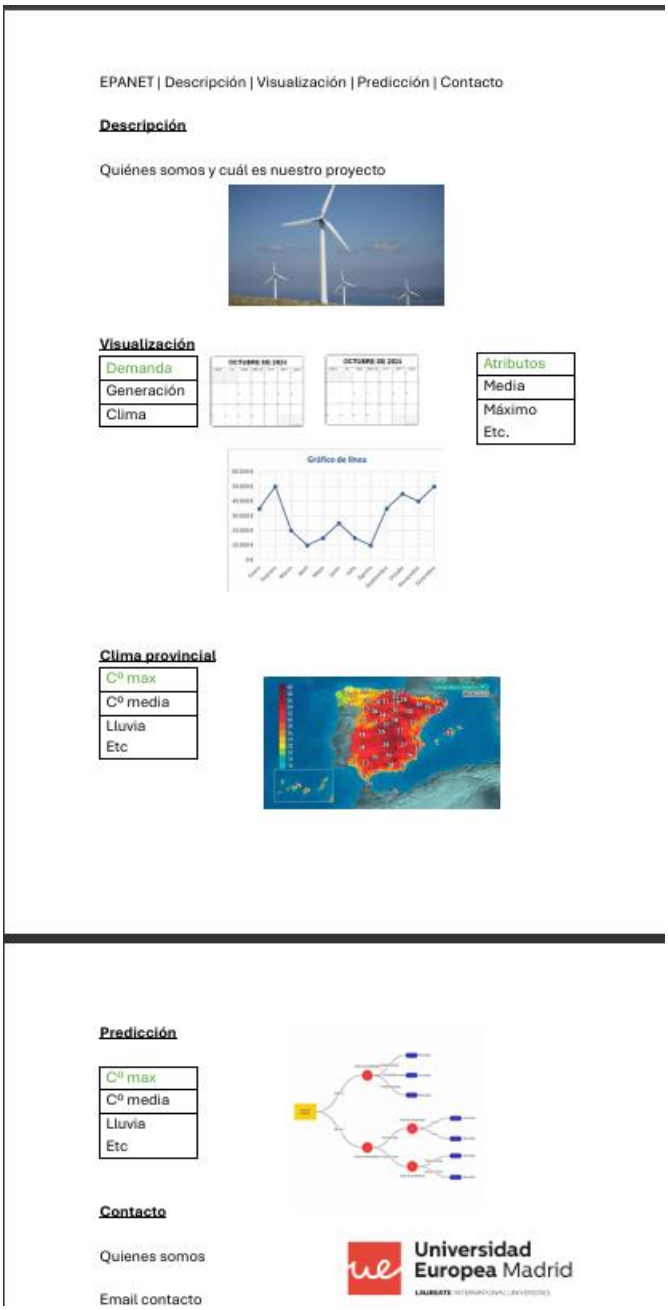
energética a nivel peninsular, y la posibilidad de realizar un modelo de detección de anomalías en la red.

Una vez definido, hicimos la recolección de los datos explicado ya en el 3.1 utilizando Python. Una vez cargados todos los datos en Python, hicimos una comprobación de nulos. En los datos extraídos de la API no teníamos registros nulos, y como aún no teníamos del todo definido lo que queríamos realizar, hicimos directamente el volcado a Elastic con todos los datos que obtuvimos.

Para los datos de clima, sí había presencia de nulos, pero en proporción a la cantidad de datos que teníamos era muy pequeño, así que los eliminamos. Tuvimos un problema a la hora de subir el CSV en GitHub debido a que el archivo era demasiado grande. Por esto, particionamos el CSV en CSVs más pequeños para solucionar este problema. Una vez realizado, volcamos los datos a Elastic.

De forma paralela, empezamos a plantear ideas de cómo íbamos a terminar de enfocar el modelo y la página web. Para la web, realizamos algunos bocetos para plasmar de forma más clara lo que queríamos realizar. Primero hicimos un boceto a papel, y luego lo hicimos en Word para que quedara más claro cómo lo habíamos planteado:





Volviendo con los datos, cargamos los datos del Elastic, y ya con una idea más clara de cómo íbamos a proseguir con el proyecto, realizamos la limpieza de datos.

Para la limpieza de los datos, en los datos de REE creamos diversos atributos con relación a la fecha que pueden ser relevantes a la hora de predecir anomalías. De esta forma, obtuvimos información adicional como el día de la semana, el mes del año, si es festivo o no, el año, la hora y los minutos. De esta forma tenemos información suficiente en cuanto a lo temporal.

Eliminamos las columnas de datetime (ya que, con las generadas anteriormente, ya no son útiles), además de todo lo referente con la localización, ya que todos los registros se refieren a nivel peninsular, no aporta información.

Hicimos un mapping para aquellas variables que no fueran numéricas, como el mes o día de la semana, para conseguir un conjunto de datos solo numérico para facilitar el trabajo posterior a nuestro modelo.

Finalmente, pusimos una fecha de corte para que en todos los datos tuviéramos el mismo número de datos. Definimos el intervalo de tiempo desde el 1 de enero de 2019 hasta el 4 de noviembre de 2024. Para facilitar el entendimiento de los datos, hicimos un inner join de los conjuntos de datos de previsión, programación y demanda real para tener todo en un mismo conjunto de datos.

Hemos definido las anomalías en la demanda eléctrica basándonos en la diferencia porcentual entre la demanda programada y la demanda real. Esta diferencia refleja cuánto se desvía la demanda respecto a lo previsto, lo que nos permite clasificar las anomalías en tres niveles:

- **Nivel 0:** No hay anomalía si la diferencia es igual o inferior al 2%.
- **Nivel 1:** Anomalía leve si la diferencia está entre el 2% y el 5%.
- **Nivel 2:** Anomalía significativa si la diferencia supera el 5%.

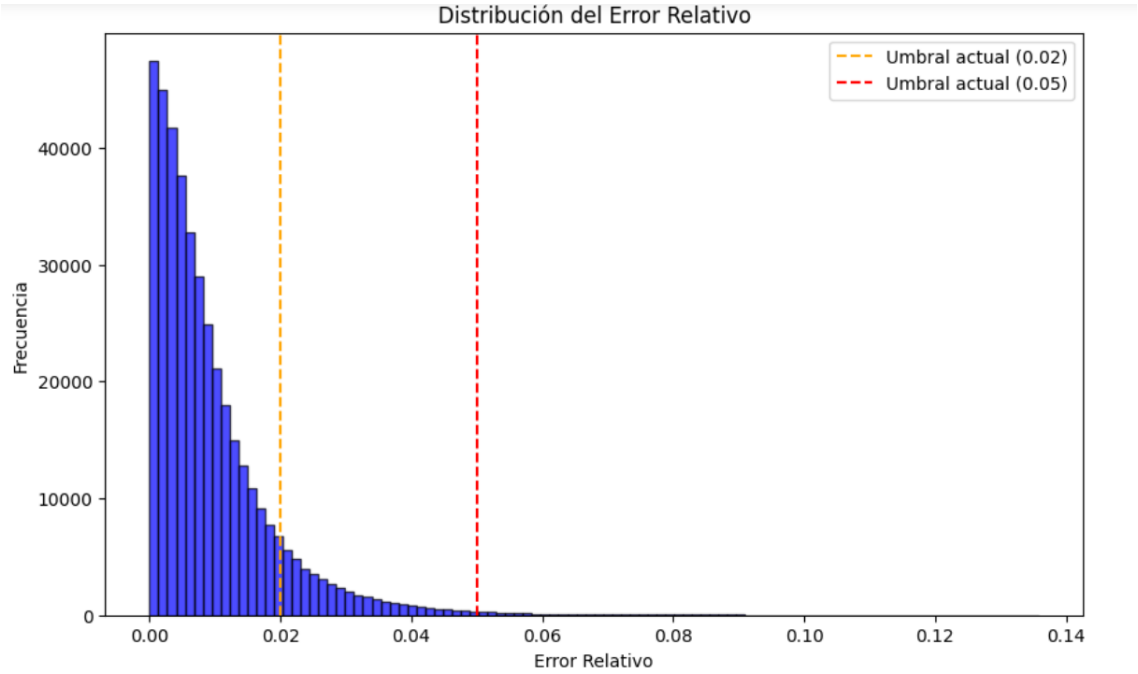
Esta clasificación nos permite identificar en qué situaciones la red eléctrica podría enfrentar un desajuste significativo entre lo que se esperaba consumir y lo que realmente se consumió.

Para los datos de clima, hicimos un agrupamiento por fecha, ya que, como la demanda la teníamos a nivel peninsular, el clima también se debe de ajustar de esta manera. De esta forma, los datos serán la media que ocurrió en ese día en todas las estaciones peninsulares.

Una vez ajustado, y eliminando alguna columna innecesaria, hemos hecho un merge con el conjunto de demanda. De esta forma, en cada registro tenemos información temporal (día, hora, etc), además de la demanda explicada anteriormente, el nivel de anomalía ya calculado e información sobre la meteorología en ese día.

Antes de la creación de modelo, dividimos el conjunto en train (80%) y test (20%) para poder evaluar posteriormente nuestro modelo. De esta forma, tenemos como conjunto de entrenamiento 322348 registros, y 80587 registros para evaluar los modelos.

En la siguiente imagen, podemos ver cómo se distribuyen los registros en función del error relativo, indicando los umbrales que hemos empleado para diferenciar entre anomalía leve y grave.



Para los modelos, primero de todo creamos un bosque aleatorio de clasificación. De esta manera, el modelo simplemente predecía el nivel de anomalía [0,1,2]. Para este caso, hicimos un bosque aleatorio conformado por 100 árboles. Y los resultados fueron los siguientes:

```
[[69722  1830    36]
 [ 5332  2933   140]
 [   93   239   262]]
Clasificación en datos de prueba:
              precision    recall  f1-score   support

           0       0.93      0.97      0.95     71588
           1       0.59      0.35      0.44      8405
           2       0.60      0.44      0.51       594

    accuracy                           0.90     80587
   macro avg       0.70      0.59      0.63     80587
weighted avg       0.89      0.90      0.89     80587
```

Como podemos observar, el modelo predice correctamente en un 93% de los casos que no hay anomalía, pero para detectar anomalías tiene una precisión de aproximadamente 60% para el nivel 1 y 2. Por lo general, cuando el nivel de anomalía es 2, es muy raro que lo detecte como que no hay anomalía y viceversa. El problema viene al intentar diferenciar el nivel 1 con los otros niveles.

Como no hemos obtenido unos buenos resultados, hemos probado con otro modelo. Hemos creado otro bosque aleatorio, pero en este caso, de regresión. En este caso, la

misión era predecir la demanda programada y la demanda real, y de esta manera poder calcular el nivel de anomalía. El modelo muestra un rendimiento muy sólido, especialmente en términos de R^2 Score, lo que indica que es altamente fiable para detectar anomalías en la demanda eléctrica. Sin embargo, el MAE y el MSE sugieren que todavía hay un margen de mejora, sobre todo en casos extremos donde el modelo puede cometer errores más grandes.

```
Random Forest - Mean Absolute Error (MAE): 112.6045692921649
Random Forest - Mean Squared Error (MSE): 29420.392196313165
Random Forest - R2 Score: 0.9984519937090592
```

Finalmente, hemos creado un último modelo ExtraTrees regresivo. Al igual que el anterior, es un modelo de regresión que busca predecir la demanda, para posteriormente calcular el nivel de anomalía. Al ver las métricas podemos comprobar que el R^2 Score es similar al segundo modelo, pero en términos de MAE y MSE hemos mejorado las métricas del modelo, por lo que finalmente nos quedaremos con este último modelo.

```
Extra Trees - Mean Absolute Error (MAE): 90.35759036817332
Extra Trees - Mean Squared Error (MSE): 20644.84160329359
Extra Trees - R2 Score: 0.9989130781682456
```

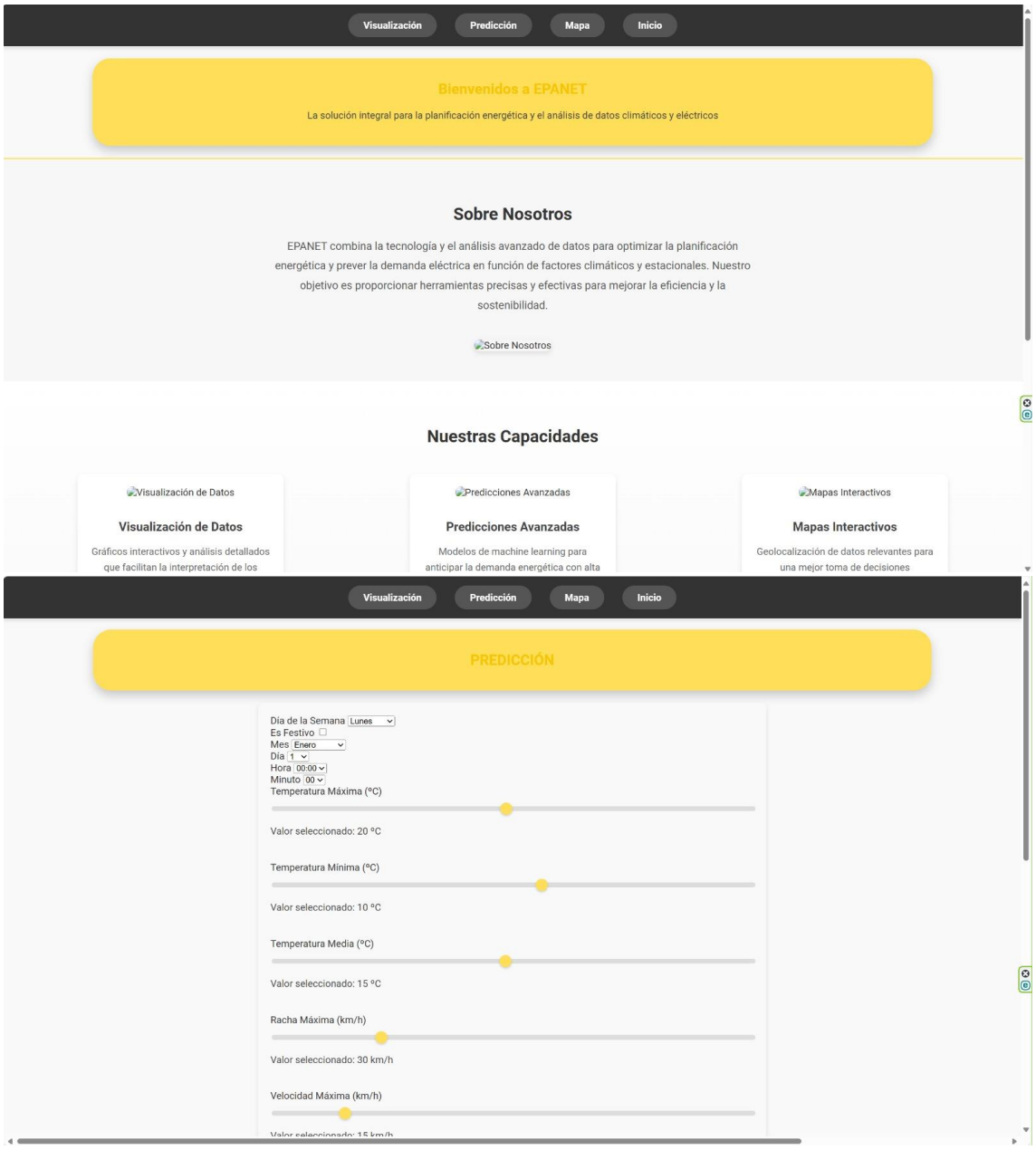
Con lo que ha predicho el modelo, calculamos el nivel de anomalía y comparamos para ver cómo se comporta el modelo a la hora de definir los niveles de anomalía. Los resultados para el conjunto de prueba son los siguientes:

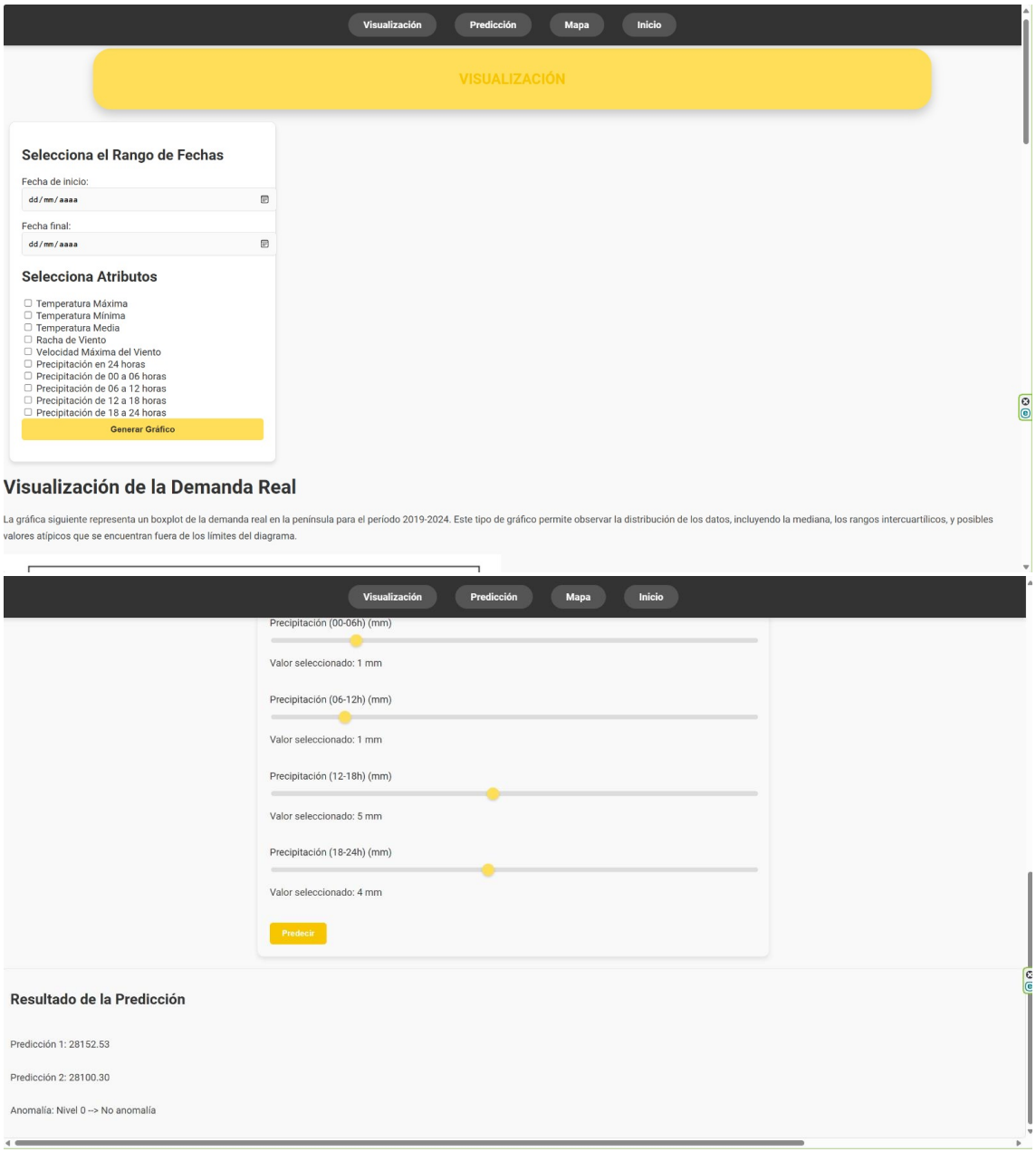
```
Coincidencias: 74909
No coincidencias: 5678
Porcentaje de coincidencias: 92.95419856800726 %
[[70469  1193     0]
 [ 4114  4201    24]
 [   10   337   239]]
                precision    recall  f1-score   support

    No Anomalía       0.94      0.98      0.96     71662
 Anomalía Leve        0.73      0.50      0.60      8339
Anomalía Grave        0.91      0.41      0.56       586

      accuracy                           0.93     80587
     macro avg        0.86      0.63      0.71     80587
  weighted avg        0.92      0.93      0.92     80587
```

Como podemos ver, el modelo tiene mejores métricas con respecto al bosque aleatorio de clasificación. El modelo alcanzó un 92.95% de coincidencias entre las predicciones y los valores reales, mostrando un buen desempeño general. La precisión para detectar anomalías graves es alta (91%), pero el recall es bajo (41%), lo que indica que algunas de estas anomalías no se identifican correctamente. Para las anomalías leves, el modelo tiene una precisión moderada (73%) y un recall bajo (50%), lo que sugiere margen de mejora en la detección de estas. El f1-score global es de 0.92, lo que refleja un modelo equilibrado y eficaz, especialmente en la detección de situaciones sin anomalía.

Una vez realizado el modelo, lo guardamos mediante Pickle. De esta forma, podremos conectar la página web y el modelo levantando una API.

De forma paralela hemos realizado la web utilizando HTML, CSS y JavaScript, teniendo como inspiración los bocetos realizados en anteriores fases, aunque cambiando y reajustando en función de cómo hemos ido necesitando. Para conectar la web con el modelo, hemos levantado una API la cual, la web hace una llamada a esta, con los parámetros de entrada para el modelo. El modelo hace la predicción y calcula el nivel de anomalía en función de los parámetros insertados por el usuario. Una vez calculado, la API devuelve la respuesta a la web, y se visualiza por pantalla el nivel de anomalía calculado por nuestro modelo. A continuación, se muestran unas imágenes de la web operativa:


Visualización
Predicción
Mapa
Inicio
Bienvenidos a EPANET
La solución integral para la planificación energética y el análisis de datos climáticos y eléctricos
Sobre Nosotros
EPANET combina la tecnología y el análisis avanzado de datos para optimizar la planificación energética y prever la demanda eléctrica en función de factores climáticos y estacionales. Nuestro objetivo es proporcionar herramientas precisas y efectivas para mejorar la eficiencia y la sostenibilidad.
Sobre Nosotros
Nuestras Capacidades
Visualización de Datos
Visualización de Datos
Gráficos interactivos y análisis detallados que facilitan la interpretación de los
Predicciones Avanzadas
Predicciones Avanzadas
Modelos de machine learning para anticipar la demanda energética con alta
Mapas Interactivos
Mapas Interactivos
Geolocalización de datos relevantes para una mejor toma de decisiones
Visualización
Predicción
Mapa
Inicio
PREDICCIÓN
Día de la Semana Lunes
Es Festivo
Mes Enero
Día 1
Hora 00:00
Minuto 00
Temperatura Máxima (ºC)
Valor seleccionado: 20 ºC
Temperatura Mínima (ºC)
Valor seleccionado: 10 ºC
Temperatura Media (ºC)
Valor seleccionado: 15 ºC
Racha Máxima (km/h)
Valor seleccionado: 30 km/h
Velocidad Máxima (km/h)



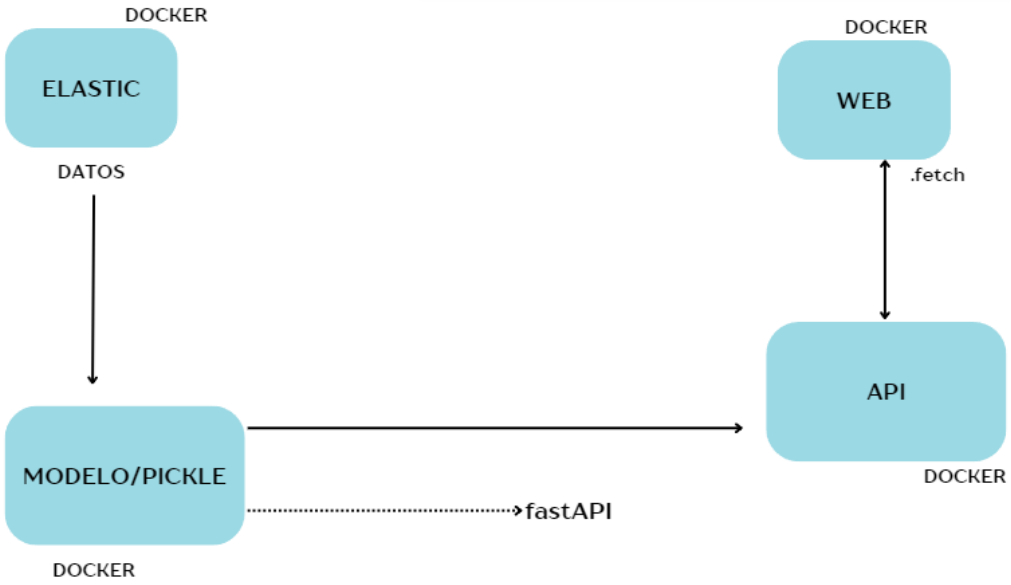
Finalmente, además de todo esto, hemos implementado Docker a nuestro proyecto, en donde podremos ver la infraestructura del proyecto en el siguiente capítulo.

Capítulo 4. Infraestructura real

La infraestructura real de nuestro proyecto se resume, a grandes rasgos, con los siguientes Docker:

- Docker para subir datos a Elastic
- Docker para el modelo (recogiendo datos desde Elastic y guardando el modelo como un archivo pickle)
- Docker para la página web
- Docker para la API (recibe peticiones desde la web, interactúa con el modelo, sirviendo como puente entre la web y el modelo entrenado, devuelve la predicción)

4.1 Diagrama de infraestructura



4.2 Flujo de datos

Tenemos un sistema compuesto por varios contenedores Docker que trabajan en conjunto para cumplir distintas funciones. Primero, hay un Docker encargado de subir los datos a Elastic. Por otro lado, existe un Docker dedicado al modelo, que se conecta nuevamente a Elastic para recoger los datos, entrenar el modelo con ellos y guardar dicho modelo en un archivo pickle dentro de otro contenedor Docker.

Además, tenemos un Docker que alberga el HTML de la página web, junto con otro Docker destinado a la API. El funcionamiento del Docker de la API es el siguiente: cuando la web está en ejecución y el usuario interactúa con un botón, como "Predecir," se realiza una petición a la API con los parámetros seleccionados en la web. La API utiliza estos parámetros para llamar al modelo, realizar la predicción y devolver el resultado. Finalmente, la predicción generada por el modelo se muestra en la interfaz web.

Capítulo 5. Conclusiones y trabajo futuro

5.1 Resultados obtenidos

A lo largo de este proyecto, se lograron consolidar varias fuentes de datos relevantes, incluyendo la demanda energética, información climatológica y datos de generación de energía. Estos datos fueron limpiados y almacenados eficientemente en Elasticsearch, permitiendo un acceso rápido y escalable. Además, se implementó un modelo de predicción de anomalías que alcanza un nivel de precisión de 92.95% para identificar desviaciones significativas respecto a las previsiones de demanda. La API desarrollada integró estos resultados en una plataforma web accesible y funcional.

5.2 Conclusiones del trabajo

El trabajo realizado destaca la importancia de la integración de diversas fuentes de datos para obtener una visión completa del sistema eléctrico. No hemos conseguido realizar el proyecto a nivel regional o provincial debido a la inexistencia de los datos, por lo que tuvimos que adaptarlo todo a nivel peninsular. El modelo de predicción implementado puede ser una herramienta útil para REE y empresas del sector energético, ayudándolos a anticipar posibles anomalías y a tomar decisiones informadas en tiempo real.

5.3 Trabajo futuro

Los siguientes pasos de nuestro proyecto son:

- Intentar mejorar la calidad del modelo, mejorando su nivel de precisión y que sea capaz de identificar con mayor claridad las anomalías
- Una mejor implementación web, añadiendo nuevas funcionalidades y mejorando las ya existentes
- Posibilidad de añadir nuevos parámetros a nuestro modelo para que trate de detectar las anomalías en la red en función de nuevas variables.
- Utilizar datos de demanda y meteorología en tiempo real para obtener mejores resultados.

Referencias bibliográficas

Barillas, Ana. 2023. *Aurora Energy Research.* [En línea] 12 de Abril de 2023. [Citado el: 8 de Enero de 2025.] https://auroraer.com/media/los-problemas-en-la-gestion-de-la-red-electrica-generan-un-coste-adicional-a-los-consumidores-en-espana/.