PROYECTO TESLA

1. INTRODUCCIÓN

En los últimos años, la industria automotriz ha experimentado una profunda transformación gracias a la integración de tecnologías digitales avanzadas y técnicas de análisis de datos. Tesla, como líder en innovación y sostenibilidad, ha marcado un punto de inflexión en el desarrollo de vehículos eléctricos y en la adopción de herramientas de análisis predictivo para comprender las dinámicas del mercado. Modelos como el Tesla Model 3, Model S, Model X y Model Y han revolucionado la movilidad eléctrica. Esta evolución refleja cómo las empresas están aprovechando el poder de los datos para optimizar procesos y mejorar la toma de decisiones estratégicas.

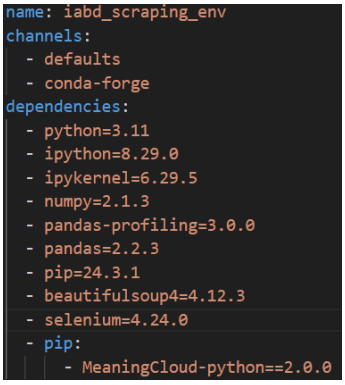
El scraping web, una técnica esencial en la recolección de información de fuentes digitales, ha ganado relevancia como método para obtener datos estructurados en tiempo real. En el contexto automotriz, permite capturar información sobre precios, características y tendencias del mercado desde múltiples plataformas en línea. Esta práctica se ha convertido en una pieza clave para las empresas que buscan mantenerse competitivas en un entorno dinámico y globalizado.

Paralelamente, el desarrollo de modelos de análisis basados en datos extraídos a través de scraping ha impulsado avances significativos en la comprensión de patrones de consumo y la anticipación de necesidades del mercado. Además, el uso de estas herramientas fomenta la diversificación de estrategias empresariales y la personalización de ofertas, mejorando así la experiencia del cliente y fortaleciendo la competitividad del sector.

El análisis de precios y tendencias de vehículos Tesla, utilizando datos recopilados de diversas fuentes, permite explorar el potencial del scraping web y de los modelos analíticos como motores de innovación en la industria.

1.1 ENVIRONMENT

Para la realización de los distintos webs scraping, los procesos de manipulación de datos y aplicación de modelos se ha utilizado un environment común con las siguientes dependencias:



```yaml
name: iabd_scraping_env
channels:
  - defaults
  - conda-forge
dependencies:
  - python=3.11
  - ipython=8.29.0
  - ipykernel=6.29.5
  - numpy=2.1.3
  - pandas-profiling=3.0.0
  - pandas=2.2.3
  - pip=24.3.1
  - beautifulsoup4=4.12.3
  - selenium=4.24.0
  - pip:
      - MeaningCloud-python==2.0.0
```

2. BUSINESS UNDERSTANDING

2.1 OBJETIVOS

A. OBJETIVO GENERAL

Predecir los precios de vehículos Tesla mediante un modelo analítico basado en datos obtenidos a través de scraping web de Teslahunt.io, Tesla y Autocasion. aprovechando técnicas de análisis de datos y aprendizaje automático.

B. OBJETIVOS ESPECÍFICOS

- Extraer información relevante de las plataformas seleccionadas, mediante técnicas de scraping web.
- Estandarizar y depurar los datos recopilados para asegurar su calidad y consistencia, preparándolos para el análisis posterior.
- Consolidar la información de las distintas fuentes en una base de datos única y completa que refleje el mercado de vehículos Tesla.
- Diseñar y entrenar un modelo analítico o de aprendizaje automático capaz de predecir los precios de los vehículos

Tesla con alta precisión y validar y optimizar el modelo desarrollado mediante técnicas de evaluación, asegurando su fiabilidad en escenarios reales.

2.2 BUSSINES CRITERIA

Para evaluar si está funcionando correctamente, se podrá comparar los precios predichos con los precios reales de los vehículos de Tesla disponibles en las plataformas seleccionadas.

3. DATA UNDERSTANDING

Proporciona la base para tomar decisiones informadas sobre cómo procesar, transformar y analizar los datos en las etapas posteriores del proyecto. Una comprensión sólida de los datos asegura un análisis más efectivo y resultados de mayor calidad.

En este proyecto se han recolectado datos de múltiples páginas web (teslahunt.io, Tesla y autocasion.com) con el objetivo de obtener información sobre los diferentes modelos de coches Tesla para su posterior uso en la predicción de precios.

En nuestro caso, hemos recopilado información mediante web scraping de anuncios de venta de coches Tesla. Entre los datos obtenidos se incluyen el modelo, el año de matriculación, el precio, el kilometraje, el color y el país. Para ello, utilizamos la librería Selenium, ya que nos resultó útil para navegar por las distintas páginas dentro de los sitios web mediante selectores CSS.

En el caso de la página oficial de Tesla, logramos extraer información adicional sobre los vehículos, como la descripción del modelo, la autonomía, la velocidad máxima y el tiempo necesario para alcanzar los 100 km/h.

Aunque toda la información obtenida es útil para predecir los precios de los coches, será necesario prescindir de algunos detalles, ya que no fue posible obtener la misma información de las tres fuentes.

4. DATA PREPARATION

La preparación de datos consiste en la limpieza y manipulación de los datos en bruto para facilitar su procesamiento y análisis.

Una vez obtenidos los datos de las tres plataformas, el primer paso es seleccionar la información relevante para el objetivo final. Como se mencionó anteriormente, no se ha podido recopilar la misma información de las tres webs. Por esta razón, hemos optado por seleccionar únicamente los datos coincidentes y de valor, tales como el modelo, el kilometraje, el precio, el color, el año de matriculación y el país.

Con los datos seleccionados, el siguiente paso es limpiar las columnas correspondientes. Esto incluye revisar si existen duplicados, eliminar valores nulos, ajustar el tipo de dato en las columnas que deberían ser numéricas (precio, kilometraje y año) y verificar la presencia de caracteres en blanco en las columnas de texto.

5. MODELLING

Con el paso anterior realizado, procedemos a juntas todos los datos en un archivo csv conjunto cuya estructura es la siguiente:

modelo	kilometraje	precio	color	año	pais
Model S	6197	88890	azul	2024	España
Model S	997	90270	azul	2024	España
Model S	577	90520	negro	2024	España
Model S	1151	91080	gris	2024	España
Model S	27	91790	gris	2024	España
Model S	16	97500	plata	2024	España
Model S	503	99600	blanco	2024	España
Model S	3652	101270	negro	2024	España
Model S	98	102270	blanco	2024	España
Model S	0	102270	blanco	2024	España
Model S	1700	102320	negro	2024	España
Model S	0	102330	gris	2024	España

Este archivo consta de 971 registros que se utilizarán para la predicción de precios.

Para la predicción de precios, se ha optado por utilizar el algoritmo **Random Forest Regressor**, el cual combina los resultados de múltiples árboles de decisión para generar un único resultado.

Para aplicar este algoritmo, es necesario transformar los datos categóricos en valores numéricos. Para ello, se procederá a factorizar dichas columnas. Por ejemplo, si existen cuatro modelos de coches Tesla, en lugar de conservar sus nombres, se les asignará un número correspondiente, como se ilustra en la siguiente imagen.

modelo	kilometraje	precio	color	año	pais
0	6197	88890	0	2024	0
0	997	90270	0	2024	0
0	577	90520	1	2024	0
0	1151	91080	2	2024	0
0	27	91790	2	2024	0
...	...	...	...	...	...
0	133500	49900	1	2016	0
0	54500	59900	1	2019	0
0	72000	28475	1	2021	0
0	109000	26475	1	2017	0
0	48000	33850	1	2022	0

Ya resuelto nuestro problema se procederá a aplicar el algoritmo a nuestros datos.

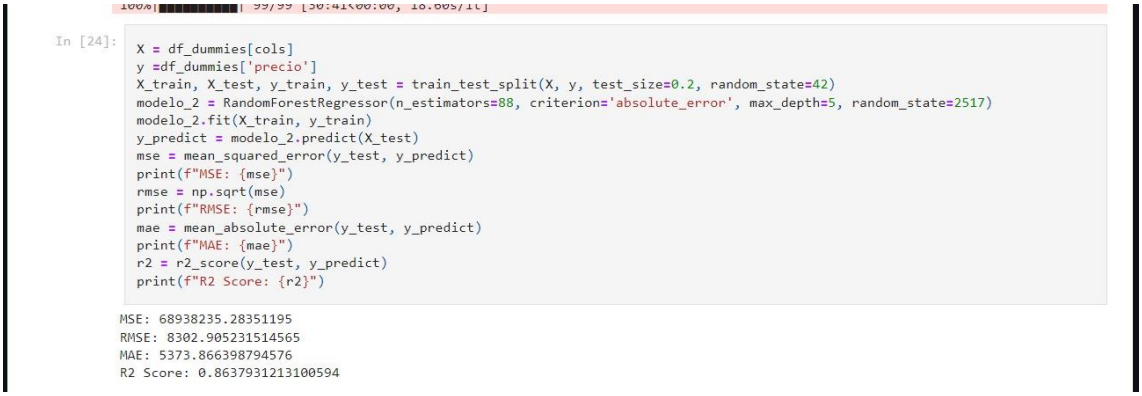
6. EVALUATION

Para proceder a aplicar nuestro algoritmo primero hay que dividir los datos en entrenamiento y test. Los datos de entrenamiento serán el 80% de los datos y el 20% para hacer test del algoritmo.

Con el fin de obtener el mejor resultado posible se han probado distintas combinatorias de parámetros y distintas formas de modelado como factorizar o get_dummies.

Para saber si el resultado es óptimo se han obtenido el MAE (mean absolute error), RMSE (root mean square error) y el índice de determinación que determina la calidad del modelo para predecir resultados.

Al aplicar las técnicas necesarias se han obtenido los siguientes resultados:



In [24]:

```
X = df_dummies[cols]
y =df_dummies['precio']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
modelo_2 = RandomForestRegressor(n_estimators=88, criterion='absolute_error', max_depth=5, random_state=2517)
modelo_2.fit(X_train, y_train)
y_predict = modelo_2.predict(X_test)
mse = mean_squared_error(y_test, y_predict)
print(f"MSE: {mse}")
rmse = np.sqrt(mse)
print(f"RMSE: {rmse}")
mae = mean_absolute_error(y_test, y_predict)
print(f"MAE: {mae}")
r2 = r2_score(y_test, y_predict)
print(f"R2 Score: {r2}")
```

```
MSE: 68938235.28351195
RMSE: 8302.905231514565
MAE: 5373.866398794576
R2 Score: 0.8637931213100594
```

Como se puede observar, el error promedio absoluto (MAE) es de 5373, lo que indica que las predicciones de media se alejan dicho valor de la realidad y además el índice de determinación es del 0.86 lo cual nos indica que el modelo es bastante bueno prediciendo ya que 1 es la predicción perfecta y 0 nos indica que el modelo es tan bueno como utilizar la media.

A pesar de contar con un conjunto de datos relativamente limitado, los resultados obtenidos han sido satisfactorios, demostrando la eficacia del modelo en la predicción de precios. Sin embargo, es importante destacar que la disponibilidad de una mayor cantidad de datos podría mejorar significativamente la precisión y robustez de las predicciones.