Introducción

El objetivo de este trabajo de fin de máster es realizar una Aplicación Web usando Streamlit para visualizar la información presente en un Dataset, así como aspectos relacionados con posibles predicciones (mediante el uso de Inteligencia Artificial).

He utilizado un conjunto de datos obtenidos en https://datamarket.es/, que contiene información detallada sobre diversos vehículos, incluyendo características como marca, modelo, año de fabricación, kilometraje, tipo de combustible y más.

Pasos para la realización del proyecto:

Para la realización del análisis exploratorio de datos así como el modelo de machine learning he utilizado Jupyter Notebook.

EXPLORATORY DATA ANALYSIS (EDA).(archivo TFM.ipynb)

El dataset utilizado en este proyecto consta de 100.000 registros y 28 columnas, lo cual es bastante grande. Durante el análisis exploratorio de datos, se detectó que muchos de estos registros estaban duplicados, siendo la única diferencia la ubicación. Esto indica que se han duplicado registros para rellenar el dataset. Para mantener la integridad y calidad de los datos, se decidió eliminar estos registros duplicados

Nos encontramos con columnas que no son relevantes para el objetivo del proyecto. En este caso, se decidió eliminar columnas que no se consideraron pertinentes para el proyecto, reduciendo así el número de columnas de 28 a 12.

Durante el análisis, se encontró que la columna "versión del modelo" tenía un número elevado de registros únicos, lo que dificulta su análisis y modelado. Debido a esto, se tomó la decisión de eliminar esta columna del dataset. Esto ayudará a simplificar el modelo y centrar el enfoque en variables más significativas para la predicción precisa del precio de los vehículos.

Para comprender la relación entre las variables y su impacto en la predicción del precio de los vehículos, se realizó un análisis de correlación. Esto permitió identificar las variables que estaban más fuertemente relacionadas con el precio

La variable más relevante para predecir el precio de un vehículo en este dataset es el año del vehículo, con una correlación positiva moderada. El kilometraje también muestra una correlación negativa moderada con el precio, mientras que la potencia tener una menor relación con el precio en comparación con las otras dos variables.

Se detectaron valores vacíos , con datos "no disponibles" en las columnas de Kilómetros, potencia,tipo de combustible y tipo de transmisión .

Para que el modelo sea lo más preciso posible, se ha rellenado los km , teniendo en cuenta la media de kilómetros según el año del vehículo,ya que son dos variables íntimamente relacionadas.

Para el resto de variables hemos utilizado la moda(el valor más repetido).

Antes de comenzar con la transformación de datos guardamos este dataset(df_streamlit.csv), ya que será este dataset el que utilicemos para nuestra aplicación de streamlit.

TRANSFORMACIÓN DE DATOS PARA APLICAR MODELOS DE MACHINE LEARNING:

Durante la etapa de transformación de datos, se realizaron diversas tareas para preparar los datos antes de aplicar los modelos de machine learning. A continuación, se desarrolla cada punto mencionado:

Codificación de variables categóricas

En este proyecto, se identificaron cuatro variables categóricas: "make" (marca), "model" (modelo), "fuel" (tipo de combustible) y "shift" (tipo de transmisión). Para poder utilizar estas variables en los modelos de machine learning, se llevó a cabo la codificación de las mismas.

Específicamente, se realizó un mapeo de datos utilizando la función map() para las columnas "fuel" y "shift". Este proceso consiste en asignar valores numéricos a las categorías correspondientes. las variables categóricas se convierten en variables numéricas.

Unificación de la columna "make" y "model":

La marca y el modelo son dos atributos clave que influyen significativamente en el precio de un vehículo. Para capturar mejor esta información relevante, se decidió unificar las columnas "make" y "model" en un solo campo. Esta combinación permite reducir la dimensionalidad del dataset y simplificar el procesamiento de los datos.

Además, al tener una gran cantidad de marcas y modelos únicos en el conjunto de datos, la unificación de estas columnas ayuda a evitar combinaciones de marca y modelo que no existen. Si no se realiza esta unificación, el modelo de machine learning podría trabajar con combinaciones falsas de marca y modelo, lo que afectaría negativamente la precisión de las predicciones.

Para la codificación de la marca y modelo, se calculó el precio promedio para cada marca y modelo de vehículo y se ordenó de menor a mayor, asignándole el valor 1 a la marca y modelo con menor precio promedio y el valor 763 (nº de registros únicos de marca y modelo) a la marca y modelo con mayor precio medio.

Para asignar los valores a las marcas y modelo hemos utilizado un diccionario , que guardaremos para posteriormente hacer la reconversión de números a categoría. ('precio_numerico_dict2.pkl')

Creación de la columna antigüedad

para obtener una representación más adecuada de los datos,en lugar de utilizar el año directamente, convertirlo a años de antigüedad permitirá que el modelo de machine learning capture mejor la relación entre la antigüedad y el precio del vehículo

Identificación valores atípicos

Durante el análisis exploratorio de datos, se llevó a cabo la identificación y manejo de valores atípicos en el dataset. Utilizando gráficas y visualizaciones, se examinaron diferentes variables para detectar observaciones que estuvieran fuera de rangos coherentes o que pudieran distorsionar los resultados del modelo de machine learning.

Después de un análisis detallado, se establecieron intervalos coherentes para ciertas variables y se tomó la decisión de eliminar los registros que se encontraban fuera de dichos intervalos. En particular, se establecieron los siguientes criterios:

Potencia del vehículo: Se estableció un rango válido de 50 a 1.000 caballos. Cualquier registro que tuviera un valor de potencia fuera de este intervalo se consideró atípico y se eliminó del dataset.

Precio mínimo: Se estableció un valor mínimo de 1.000 € para el precio de los vehículos. Cualquier registro que tuviera un precio inferior a este umbral se consideró atípico y se eliminó del dataset.

Máxima antigüedad: Se estableció una antigüedad máxima de 27 años para los vehículos. Esto significa que se eliminaron los registros de vehículos cuya fecha de fabricación superara los 27 años. Estos registros se consideraron atípicos debido a su antigüedad y se excluyeron del análisis.

Estos criterios ayudaron a eliminar los valores atípicos que podrían haber influido en el análisis y la precisión del modelo de machine learning. Al definir intervalos coherentes y eliminar registros fuera de estos rangos, se buscó mejorar la calidad y la representatividad de los datos utilizados en el proyecto.

MODELOS DE MACHINE LEARNING

Una vez que tenemos todos los datos categorizados vamos a probar qué modelo de machine learning es el más adecuado para calcular la predicción del precio.

Establecemos "X" e "y" , y realizamos el entrenamiento dividiendo los datos en entrenamiento y ensayo (80%-20%) y probamos algunos modelos de regresión obteniendo los siguientes resultados:

- Regresión Lineal: Coeficiente de determinación R2: 0.629

- Ramdon Forest Regression:Coeficiente de determinación R2: 0.885

- K-Nearest Neighbors:Coeficiente de determinación R2: 0.6

- Decision Tree Regression:Coef, de determinación R2 (Árboles de Decisión): 0.830

Viendo los resultados , nos quedaremos con el modelo de Ramdom Forest Regresion y guardamos la predicción en un archivo pkl ('modelo.pkl')

APP STREAMLIT.

La creación de la web utilizando Streamlit se ha realizado desde Visual Studio Code y se han llevado a cabo los siguientes pasos:

Creación de un entorno virtual (ENV_TFM):

Se optó por crear un entorno virtual para el desarrollo del proyecto.

Así se asegura que las bibliotecas y versiones de paquetes necesarios para el proyecto no entren en conflicto con otros proyectos o el entorno global del sistema. Esto proporciona una mayor organización, control y así poder crear un archivo de requeriment.txt específico para nuestro proyecto

Creación del archivo App.py:

Se ha creado un archivo llamado App.py para configurar la web utilizando Streamlit. En este archivo, se definieron las diferentes secciones, componentes y elementos de la interfaz de usuario de la web. También se realizaron las llamadas a las funciones necesarias para el funcionamiento de la aplicación.(funciones.py)

Creación del archivo funciones.py:

Se creó un archivo separado llamado funciones.py donde se implementaron las funciones específicas requeridas para la funcionalidad de la web. Estas funciones

pueden incluir desde el preprocesamiento de datos hasta la generación de predicciones de precios de vehículos. Al separar las funciones en un archivo aparte, se mejora la modularidad y la legibilidad del código, lo cual facilita su mantenimiento y reutilización.

Funciones creadas para la web de Streamlit:

- Calculo del precio de un vehículo:

para ello cargamos el archivo modelo.pkl que guardamos anteriormente, pedimos al usuario que introduzca unos datos utilizando st.selectbox,transformamos los datos a variables numéricas aplicamos el modelo y transformamos los datos a variables categóricas y lanzamos por pantalla la predicción del precio.

- mostrar dataframe:

mostramos el conjunto de vehículos de segunda mano disponibles y el usuario puede seleccionar marca y modelo.

- histograma precios:

permite al usuario seleccionar un rango de precios y la web muestra el nº de vehículos para ese rango. Se ha realizado con la función st.slider.

- gafico_evolucion_precio:

muestra el precio medio de un vehículo a lo largo de los años.

- mapa ubicación(FOLIUM):

se selecciona una marca y modelo , y te muestra un mapa de dónde se encuentran las unidades de tu modelo seleccionado. Para ello he utilizado la librería Folium.

Para usar la librería Folium era necesario tener las coordenadas de cada localización Por lo que se tuvieron que calcular con la biblioteca "geopy". Este proceso lo he realizado desde jupyter Notebook . Se muestra al final del archivo TFM.ipynb

- Análisis de datos:

Esta función la he incluido para ver el potencial de la librería streamlit_pandas_profiling. Esta biblioteca es una extensión de Streamlit para generar informes de perfil interactivos utilizando “pandas profiling”.

Con una línea de código nos muestra un análisis impresionante de nuestro Dataset.

CREACIÓN DE REPOSITORIO EN GITHUB:

Este repositorio contiene un archivo README.md donde se incluye la descripción del proyecto, breve descripción del contenido de cada archivo, docentes relacionados con el TFM asi como el nombre del autor del proyecto.

Comparto la URL de Guthub del proyecto para poder acceder :

https://github.com/jhidal82/Vehiculos-Segunda-mano-TFM-.git

NUBE DE STREAMLIT:

me parecía muy interesante la oportunidad que nos da Cloud Streamlit para subir proyectos a su nube y así poder compartirlos mediante una URL, por tanto he subido mi aplicación web a la nube de streamlit.

Para ello he tenido que vincular mi proyecto en Github con mi cuenta de usuario de Streamlit.

He tenido que modificar algunas lineas de mi codigo para que no creara conflicto,ya que algun archivo de Github lo tuve que subir comprimido, por tanto tuve que incluir en el codigo la biblioteca "Zipfile" para descomprimir el archivo pkl del modelo de predicción y que así funcionara la predicción del precio de un vehiculo.

URL para acceder a la aplicación:

https://jhidal82-vehiculos-segunda-mano-tfm--app-cwokgm.streamlit.app/

Referencias bibliográficas:

Python Software Foundation. Documentation for Python 3

https://docs.python.org/3/

Streamlit. (s.f.). Streamlit Documentation.

https://docs.streamlit.io/

Streamlit Community:
https://streamlit.io/community

Material de clase, proporcionado por ITTI High Tech Institute y sus docentes José Manuel Peña Castro e Isabel Maniega Cuadrado del Master de Data Science 2022-23