Entrega 2 Introducción a la inteligencia artificial

NICOLAS EDUARDO PEREZ VERGEL
NORBEY GARCIA ARBELAEZ
MANUEL ANDRES FURIO URBINA

INTRODUCCIÓN

A través de la inteligencia artificial y gracias a sus avances, al dia de hoy puede ser utilizada en diversos campos y aplicaciones, una de esta es la generación de modelos de predicción, los cuales nos ayudan a estimar de manera aproximada o incluso de manera exacta, cálculos o respuestas que queremos obtener, todo esto mediante unos datasets,que nos proporcionan miles de datos recopilados a los cuales se le puede hacer un análisis estadístico y generar una predicción de datos y a su vez esos datos obtenidos pueden ser parte de un nuevo datasets actualizado.

La finalidad de nuestro proyecto es llegar a estimar el precio de un auto, esto con la ayuda de un modelo de predicción, el cual se le ingresaran datos de las características del tipo de auto con la ayuda de nuestro dataset nos proporciona datos característicos que nos ayudarán a estimar el precio de un auto que se quiera comprar conociendo algunas características que el cliente desea que el carro las tenga, para ello se implementaremos un modelo

PROCEDIMIENTO

Lo primero que se hizo fue buscar un dataset con datos de precios y características de diferentes modelos de carros él cual fue obtenido de https://www.kaggle.com/datasets/austinreese/craigslist-carstrucks-data, se importo a nuestro modelo de prediccion. Se eligió este datasets porque contiene toda la información relevante que proporciona Craigslist sobre las ventas de automóviles, incluidas columnas como precio, cilindraje, fabricante, y otras 21 categorías. Luego con la ayuda de pandas observamos que se está trabajando con un conjunto de más de 400 000 entradas, lo cual es bastante grande, lo que hace que la importación sea un poco demorada.

```
[2]  auth.authenticate_user()
     gauth = GoogleAuth()
     gauth.credentials = GoogleCredentials.get_application_default()
     drive = GoogleDrive(gauth)

     file_id = '1Pt1akrURtchh4elqaWo4n_bskLI4nJme'
     download = drive.CreateFile({'id':file_id})
     download.GetContentFile('file.csv')

     import pandas as pd
     import numpy as np
     df = pd.read_csv("file.csv")
     pd.set_option('display.float_format', lambda x: '%.0f' % x)
     df.shape

     (426880, 26)
```

Fig 1. importación de datos y lectura de total de datos

Para analizar los datos faltantes y haciendo uso de Pandas, podemos ver en la fig 1 que hay columnas con demasiados datos faltantes, por lo que eliminamos esas columnas por completo. Como paso complementario, eliminamos todas las entradas duplicadas dentro del marco de datos. Finalmente, observamos los valores únicos que se encuentran dentro de algunas columnas y decidimos eliminar las filas con contenido cuestionable, como "solo piezas" en "title_status", ya que ya no tenemos información sobre las piezas que se venden. También eliminamos todos los autos anteriores a 1970 porque (después de inspeccionar el .csv), no había suficientes para crear un modelo confiable.

```
id                  0
url                 0
region              0
region_url          0
price               0
year             1205
manufacturer    17646
model            5277
condition      174104
cylinders      177678
fuel             3013
odometer         4400
title_status     8242
transmission     2556
VIN            161042
drive          130567
size           306361
type            92858
paint_color    130203
image_url          68
description        70
county         426880
state               0
lat              6549
long             6549
posting_date       68
dtype: int64
```

Fig 2. total de datos faltantes por columna

```
[9]  #Dropping certain rows according to our criteria
     df.drop(df[df['year'] < 1970].index, inplace=True)
     df.drop(df[df['fuel'] == "other"].index, inplace=True)
     df.drop(df[df['title_status'] == "parts only"].index, inplace=True)
     df.drop(df[df['title_status'] == "salvage"].index, inplace=True)
```

Fig 3. Eliminación de datos inservibles o no significativos

Otros, como la latitud o la longitud, probablemente no sean útiles para nuestros modelos. Adicionalmente, se mantendrán las columnas donde haya una cantidad más despreciable de datos faltantes, sin embargo, también se eliminarán las filas con los datos faltantes de estas columnas y por último, pero no menos importante, eliminamos todas las entradas duplicadas dentro del marco de datos.

Ya después de depurar y analizar los datos faltantes, echamos un vistazo a cómo se comporta nuestra variable objetivo. en donde se pudo apreciar varios valores que parecen fuera de lugar o excesivos. Para frenar este efecto, eliminaremos todas las filas cuyo valor de la variable de destino esté fuera de $1000 (mil) a $100000 (cien mil). Después de este proceso de limpieza, perdimos aproximadamente 120.000 entradas, o alrededor del 30 % de nuestro conjunto de datos; sin embargo, ahora tenemos un conjunto de datos único y completamente lleno.

```
[10] df.price.describe()

count       315296
mean         76518
std       12726399
min              1
25%           7500
50%          15990
75%          27990
max     3736928711
Name: price, dtype: float64

[11] df.drop(df[df['price'] > 100000].index, inplace=True)
     df.drop(df[df['price'] < 1000].index, inplace=True)
     df.price.describe()

count   307166
mean     19492
std      14165
min       1000
25%       7995
50%      16495
75%      28223
max     100000
Name: price, dtype: float64
```

Fig 4. Análisis de la variable objetivo

Matriz de correlación entre columnas numéricas.
Realizamos una pequeña prueba de correlación entre los valores numéricos de nuestro conjunto de datos y encontramos que el año de fabricación está fuertemente correlacionado con nuestra variable objetivo mientras que el odómetro no lo está. Sabiendo esto, sería una buena idea quitar la columna del odómetro.

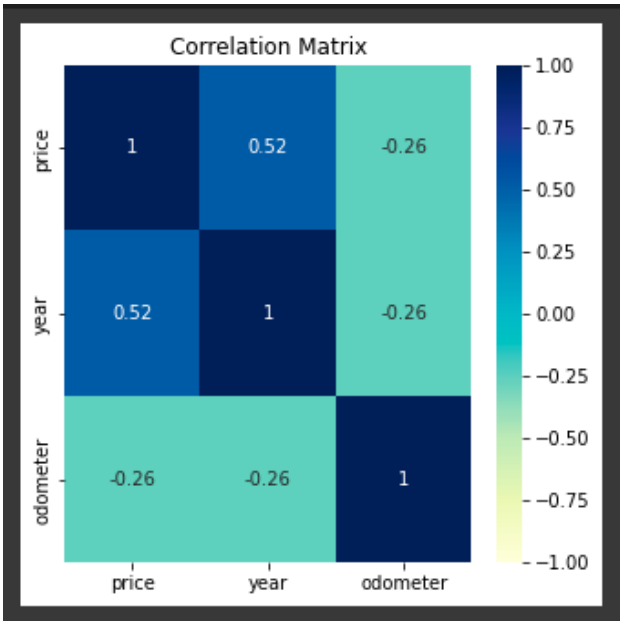

Fig 5. matriz de correlación

BIBLIOGRAFÍA

https://www.kaggle.com/datasets/austinreese/craigslist-carstrucks-data
https://rramosp.github.io/ai4eng.v1/content/M00_intro_udea.html