#importing necessary libraries

import pandas as pd

import numpy as np

import seaborn as sns

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import train\_test\_split

# from google.colab import drive

# drive.mount("/content/drive")

df= pd.read\_csv("/content/drive/My Drive/Ford Car Price Prediction.csv")

df.head()

df.shape

df.isnull().sum()

#There are 9 missing values in tax column.

print("Number of rows with null values in tax column:", df["tax"].isnull().sum())

subset = df[df["tax"].notnull()]

# Print out the shape of the subset

print("Shape after removing null values:", subset.shape)

print("Shape of dataframe before dropping:", df.shape)

df = df.dropna(axis = 0, subset = ["tax"]) #Removing rows containing null values

print("Shape after dropping:", df.shape)

dup = df[df.duplicated()]

print("Duplicate rows:\n",dup)

print("Shape of dataframe before dropping:", df.shape)

df.drop\_duplicates(inplace=True) #Removing duplicate rows

print("Shape after dropping:", df.shape)

df.info()

#Categorical Data

display(df["model"].unique())

display(df["transmission"].unique())

display(df["fuelType"].unique())

#Categorical Encoding

df["model"] = df["model"].map({" Focus": 0," Fiesta": 1, " EcoSport": 2, " Kuga": 3, " Mondeo": 4, " Ka+": 5, " C-MAX": 6, " S-MAX": 7,

" B-MAX": 8, " Edge": 9, " Tourneo Custom": 10, " Grand C-MAX": 11, " Tourneo Connect": 12,

" Mustang": 13, " Grand Tourneo Connect": 14, " Galaxy": 15, " Ranger": 16, " Streetka": 17,

" Escort": 18, " Fusion": 19, " Puma": 20, " KA": 21, " Transit Tourneo": 22, "Focus": 23 })

df["transmission"] = df["transmission"].map({"Manual": 0,"Semi-Auto": 1, "EcoSport": 2, "Automatic": 3})

df["fuelType"] = df["fuelType"].map({"Petrol": 0,"Diesel": 1, "Hybrid": 2, "Kuga": 3, "Electric": 4, "Other": 5})

df.head()

#feature scaling

X = df.drop("price", axis=1)

y = df["price"]

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

X\_train

#feature selection

df = df.drop("tax", axis=1) #tax is redundant here because it's less important and can be calculated using other features.

df.head()

df.corr()

sns.heatmap(df.corr(), cmap = "YlGnBu")

#dropping year column because the correlation value is greater than 0.75 wrt price

df = df.drop("year", axis=1)

df