Lab Assignment 8

Chaudhary Hamdan

1905387

Date: 23-03-2022

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn import preprocessing

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import accuracy\_score

from sklearn.metrics import recall\_score

from sklearn.metrics import precision\_score

from sklearn.metrics import f1\_score

df = pd.read\_csv('car\_sample.csv', encoding = "ISO-8859-1")

1.Find out if following variables are significant or insignificant and need to be dropped.

1. Seller-insignificant
2. offerType-insignificant
3. **abtest-**insignificant

Iv)vehicleType-significant

V)gearbox,

Vi)Model

Vii)Kilometer

Viii)Fueltype

Ix)Brand

X)notRepairedDamage

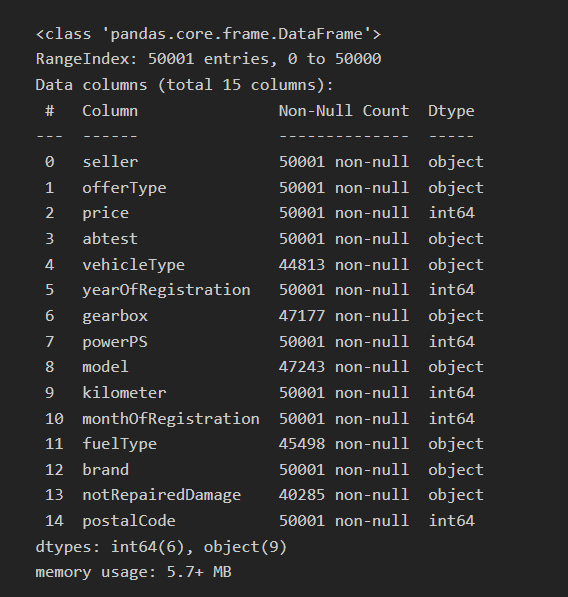
cols = ['dateCrawled', 'name', 'dateCreated', 'lastSeen']

df.drop(columns=cols, inplace=True)

df.head()



df.info()

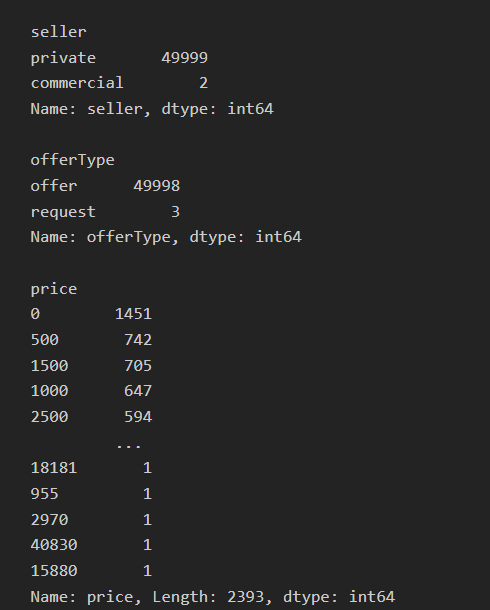


for col in df.columns:

print(col)

print(df[col].value\_counts())

print()



cols = ['seller', 'offerType', 'notRepairedDamage']

df.drop(columns=cols, inplace=True)

df.head()

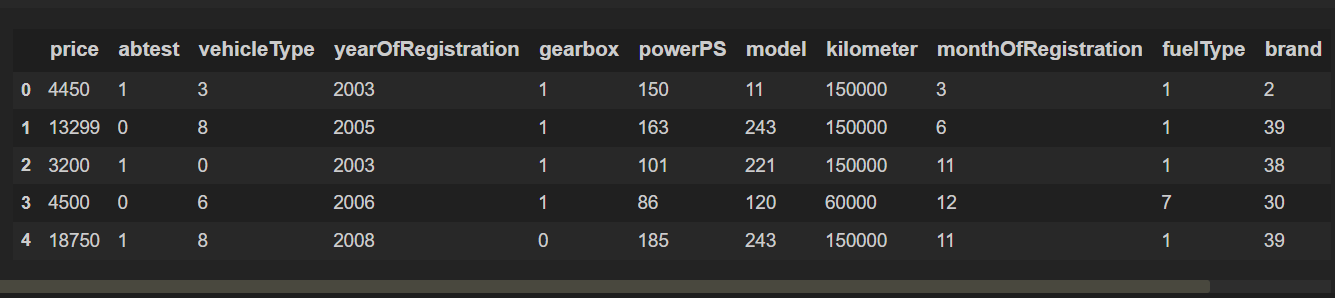
cols = ['abtest', 'vehicleType', 'gearbox', 'model', 'fuelType', 'brand']

for col in cols:

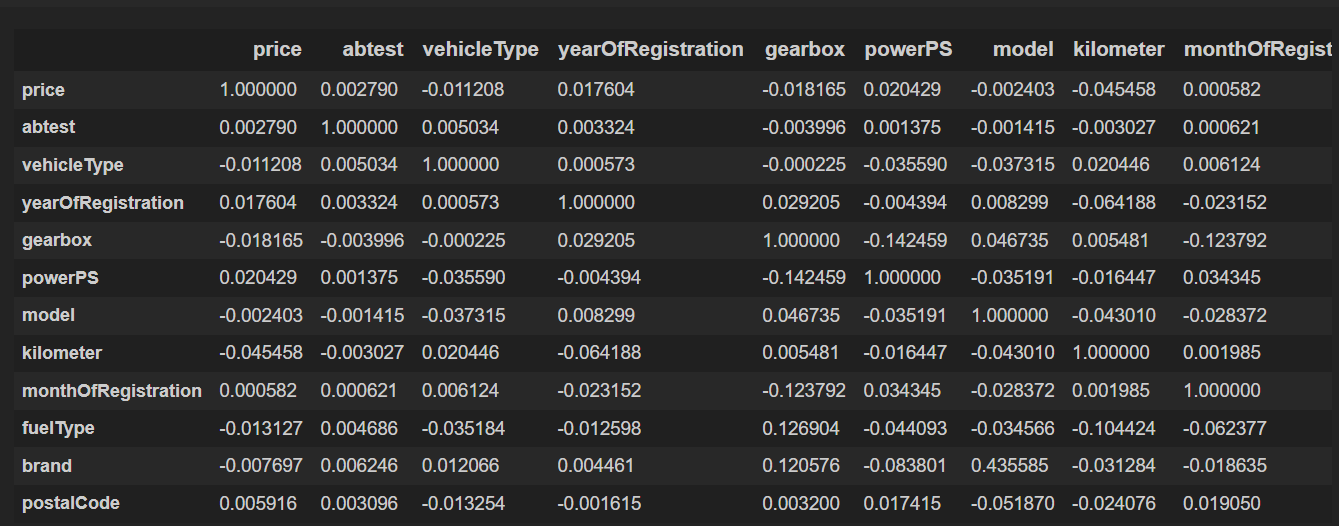
le = preprocessing.LabelEncoder()

df[col] = le.fit\_transform(df[col].astype(str))

df.head()



df.corr()

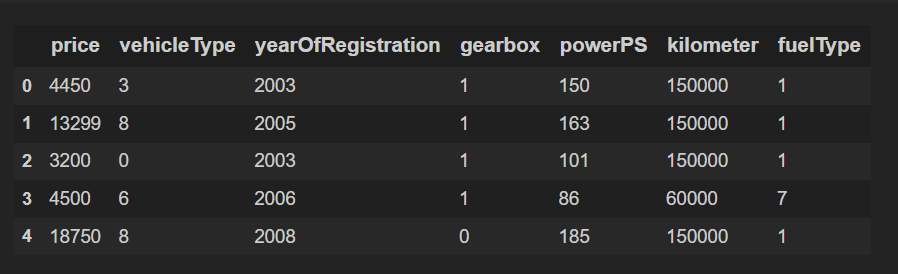


1. Drop insignificant variables from dataframe ‘cars’

cols = ['abtest', 'model', 'monthOfRegistration', 'brand', 'postalCode']

df.drop(columns=cols, inplace=True)

df.head()

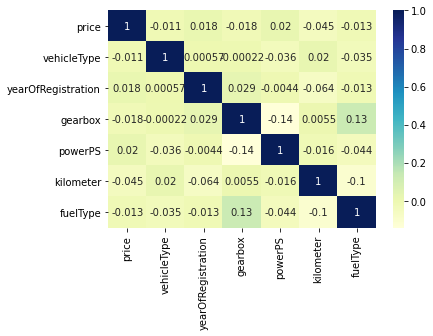


1. Find correlation between all numerical variables and find which variable has the highest correlation with price

cor = df.corr()

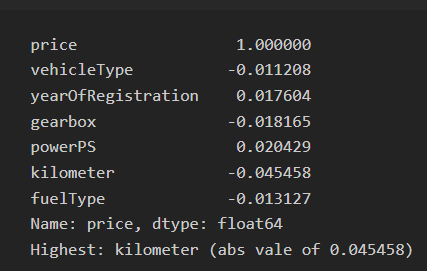
sns.heatmap(cor, cmap="YlGnBu", annot=True)

plt.show()



print(cor['price'])

print('Highest: kilometer (abs vale of 0.045458)')



1. Calculate the training data and testing data score using a linear regression model.

x\_train, x\_test, y\_train, y\_test = train\_test\_split(df.drop(columns = ['price']), df['price'], test\_size = 0.2)

x\_train.shape, y\_train.shape, x\_test.shape, y\_test.shape

algo = "Linear Regression\n"

model = LinearRegression()

model.fit(x\_train, y\_train)

print(algo)

print('Training error')

y\_pred = model.predict(x\_train)

e = (y\_pred - y\_train)

e = e.dot(e)

e /= y\_test.shape[0]

e = e\*\*0.5

print(e)

print('Testing error')

y\_pred = model.predict(x\_test)

e = (y\_pred - y\_test)

e = e.dot(e)

e /= y\_test.shape[0]

e = e\*\*0.5

print(e)

