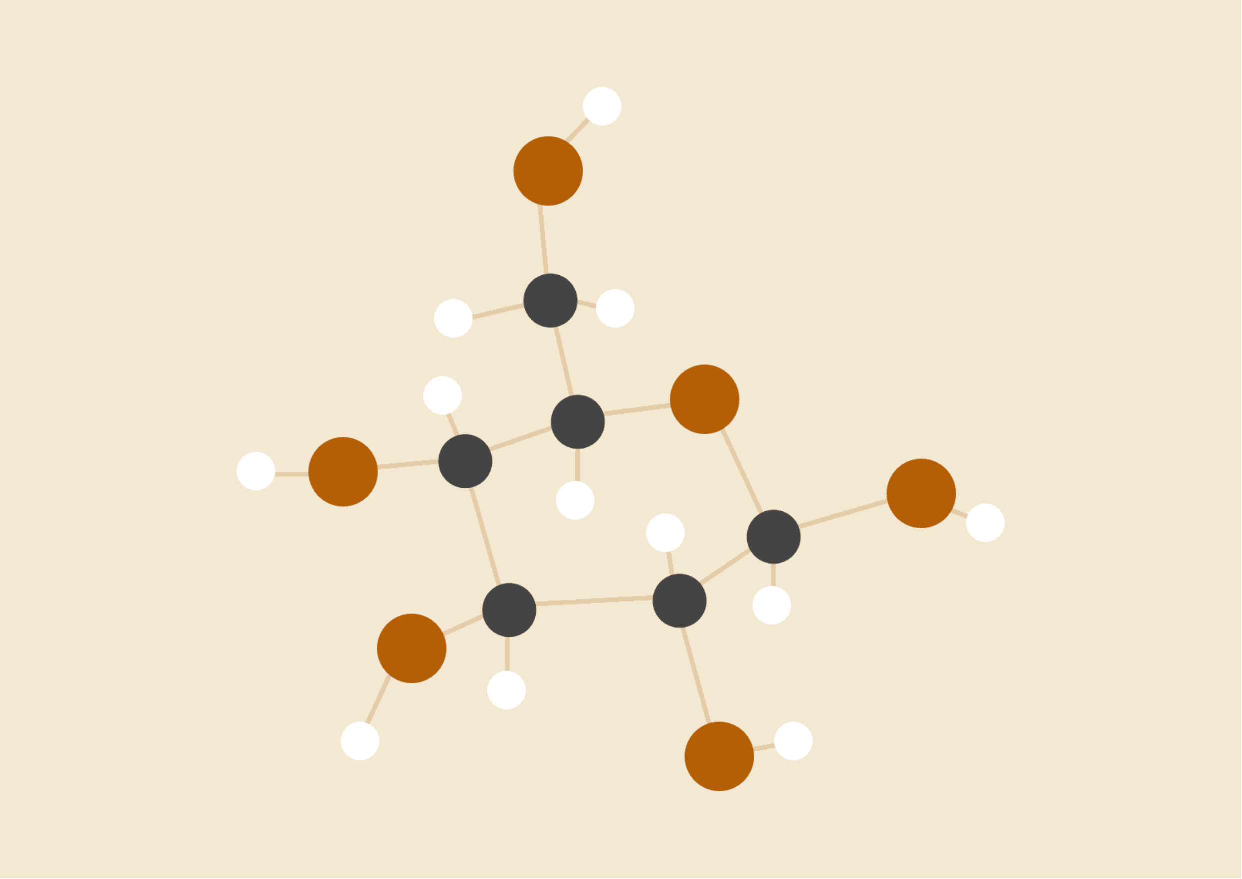
PROJECT REPORT

COSC 4426



Created By

Emma Ude - 199697810
Maria DeMelo- 199695520

TABLE OF CONTENTS

PROJECT PROPOSAL

Our project will focus on the price prediction of vehicles.

- Why is This Necessary: Car price predictions are beneficial for both clients and dealerships. It benefits the client by giving an estimate for how much their used car will sell for, or how much they should expect to pay if they are purchasing a used car. Further, the dealership may use the program for transparency when presenting estimates to clients.
-
- Objectives: Our goal is to create models and present the data in a way that allows us to predict the expected value of a given vehicle based on the dataset.
- Languages and Libraries:
  - Python
  - Scikit-Learn
  - Tensor Flow
  - Keras
- Timeline:

PHASE	TASK	START & END DATES
Week 1	Visualization of data (Get the data and Visualize it to gain insights on what model to use)	Nov 1- Nov 8
Week 2	Prepare the data, choose the model and train it	Nov 8 - Nov 15
Week 3	Fine Tune the model and check solutions	Nov 15- Nov 22

INTRODUCTION

We predicted the sales price of vehicles based on year, brand, model, colour, and state (USA based). Car dealerships can use the car's attributes to estimate pricing trends and maximize profit. In order to accomplish this, we analyzed the data and then chose a model to use on the dataset. We then trained the model and created a frontend application to test the prediction. This project will outline the process of creating this tool.

DATASET

Description

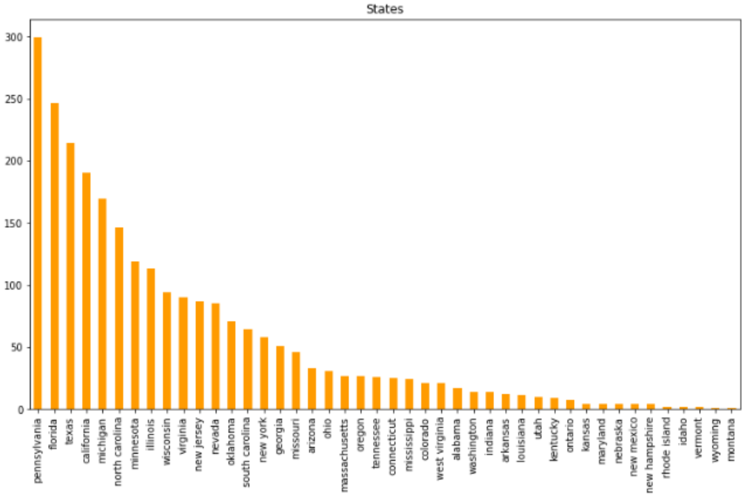
The dataset was created for the purpose of predicting the prices of vehicles using factors like year, brand, model, mileage, etc. An ideal model would be able to accurately predict the price of vehicles when it ingests the aforementioned attributes. US Cars' data was scraped from AUCTION EXPORT.com. This dataset included information about 28 brands of clean and used vehicles for sale in the US. Twelve features were assembled for each car in the dataset.

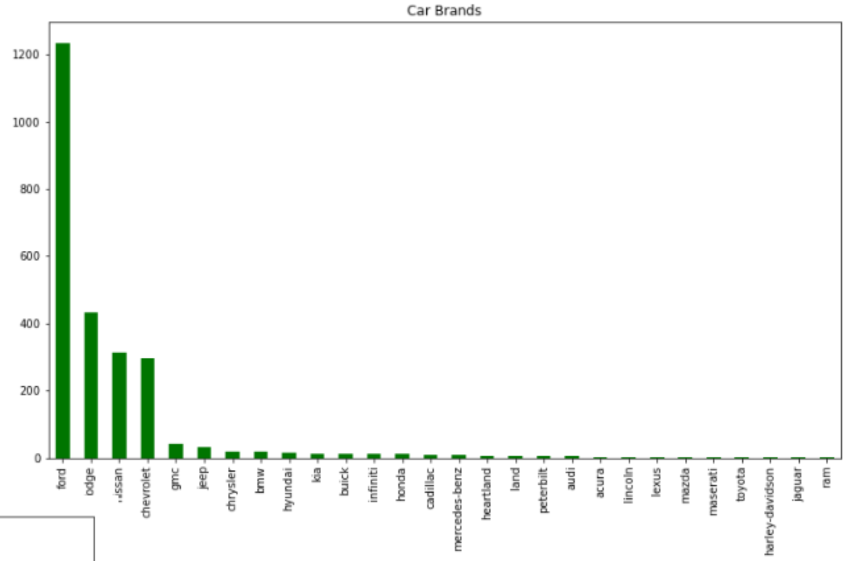
Attributes

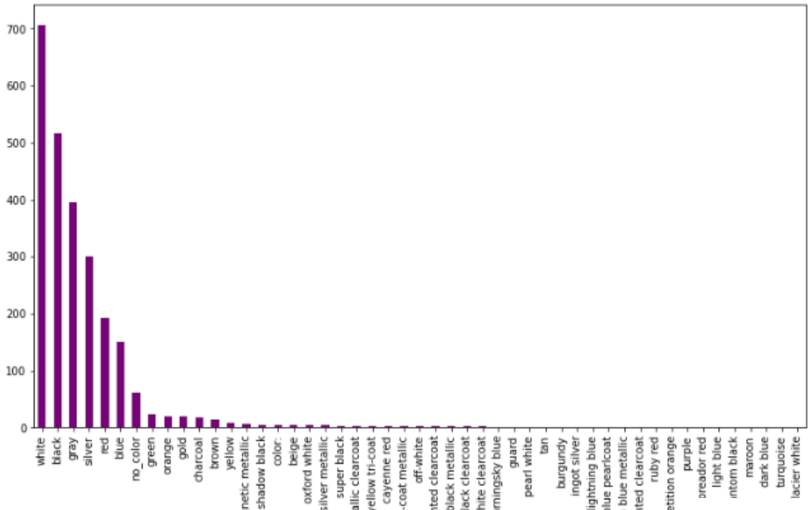
1. Price - The sale price of the vehicle in the ad
2. Years - The vehicle's registration year
3. Brand - The car brand
4. Model - The model of the vehicle
5. Color - The color of the car
6. State - The location where the car is available for purchase
7. Mileage - Miles traveled by the vehicle
8. VIN- Vehicle Identification Number (17 characters of digits and capital letters)
9. Title Status - Clean Title Vehicles or Salvage Insurance vehicles
10. Lot - Identification number assigned to a particular quantity or lot of material from a single manufacturer.For cars, a lot number is combined with a serial number to form the Vehicle Identification Number
11. Condition - time
12. Country - Country where the car is available for purchase

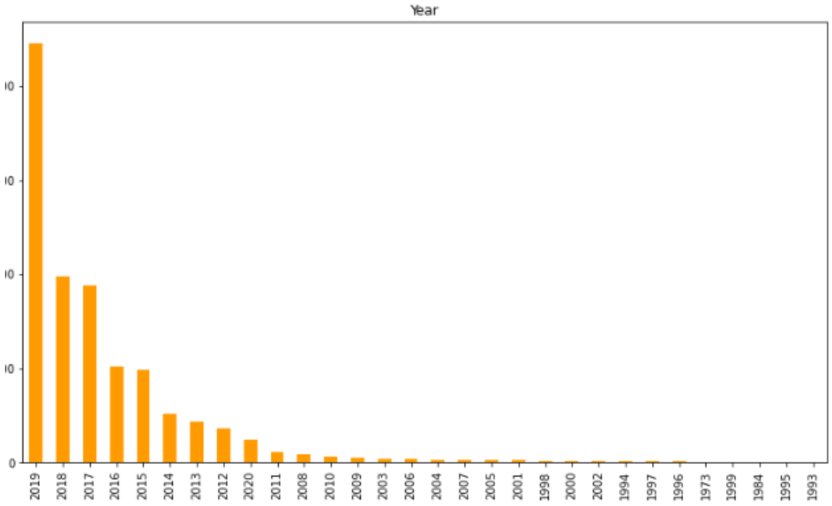
Bigger Picture

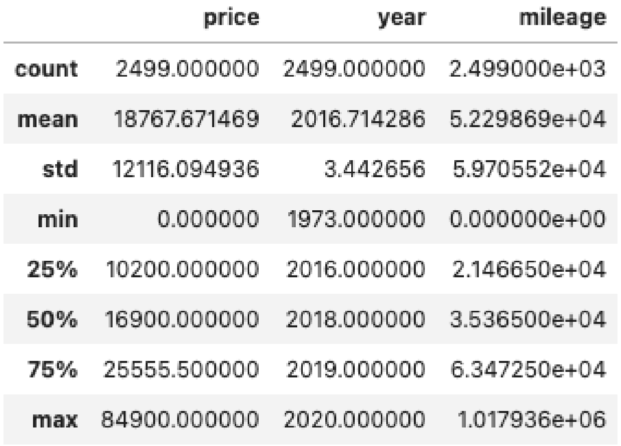
After loading the data we visualized it to get a better understanding of what it entails. From plotting the attributes we noticed vehicles were also manufactured between the years 2008-2020.The most popular car company is Ford, the most common color is white and the state is pennsylvania.


States
300
250
200
150
100
50
0
pennsylvania
florida
texas
california
michigan
north carolina
minnesota
illinois
wisconsin
virginia
new jersey
nevada
oklahoma
south carolina
new york
georgia
missouri
arizona
ohio
massachusetts
oregon
tennessee
connecticut
mississippi
colorado
west virginia
alabama
washington
indiana
arkansas
louisiana
utah
kentucky
ontario
kansas
maryland
nebraska
new mexico
new hampshire
rhode island
idaho
vermont
wyoming
montana


Car Brands
1200
1000
800
600
400
200
0
ford
odge
issan
chevrolet
gmc
jeep
chrysler
bmw
hyundai
kia
buick
infiniti
honda
cadillac
mercedes-benz
heartland
land
peterbilt
audi
acura
lincoln
lexus
mazda
maserati
toyota
harley-davidson
jaguar
ram

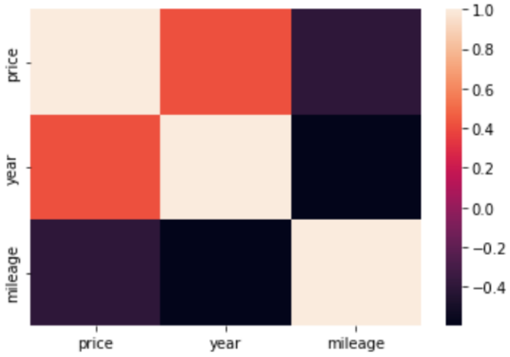

Car Colours
700
600
500
400
300
200
100
0
white
black
gray
silver
red
blue
no_color
green
orange
gold
charcoal
brown
yellow
magnetic metallic
shadow black
color:
beige
oxford white
ingot silver metallic
super black
billet silver metallic clearcoat
triple yellow tri-coat
cayenne red
white platinum tri-coat metallic
off-white
ruby red metallic tinted clearcoat
tuxedo black metallic
black clearcoat
bright white clearcoat
morningsky blue
guard
pearl white
tan
burgundy
ingot silver
lightning blue
jazz blue pearlcoat
kona blue metallic
royal crimson metallic tinted clearcoat
ruby red
competition orange
purple
oreador red
light blue
intom black
maroon
dark blue
turquoise
lacier white


Year
0
2019
2018
2017
2016
2015
2014
2013
2012
2020
2011
2008
2010
2009
2003
2006
2004
2007
2005
2001
1998
2000
2002
1994
1997
1996
1973
1999
1984
1995
1993



	price	year	mileage
count	2499.000000	2499.000000	2.499000e+03
mean	18767.671469	2016.714286	5.229869e+04
std	12116.094936	3.442656	5.970552e+04
min	0.000000	1973.000000	0.000000e+00
25%	10200.000000	2016.000000	2.146650e+04
50%	16900.000000	2018.000000	3.536500e+04
75%	25555.500000	2019.000000	6.347250e+04
max	84900.000000	2020.000000	1.017936e+06

As shown above, by printing a summary of numerical attributes, the average price is $18767. Also after checking for the correlation between the numerical attributes, we discovered that there was a strong positive correlation between the year and price. The newer a car is, the higher its price. There is also a strong negative correlation between mileage and price and year. The lower the mileage the higher the price, also newer cars seem to have lower mileage.



ANALYSIS

Task Analysis

This is a supervised task and the dataset we chose is labeled. Additionally, since we are required to predict a value, it is also a regression task. Further, because we used various factors such as year, model, and brand to predict a singular value (price), it is a multiple regression problem as well as a univariate regression problem. A plain batch learning is required as the dataset is not online and is not continuous.

Model

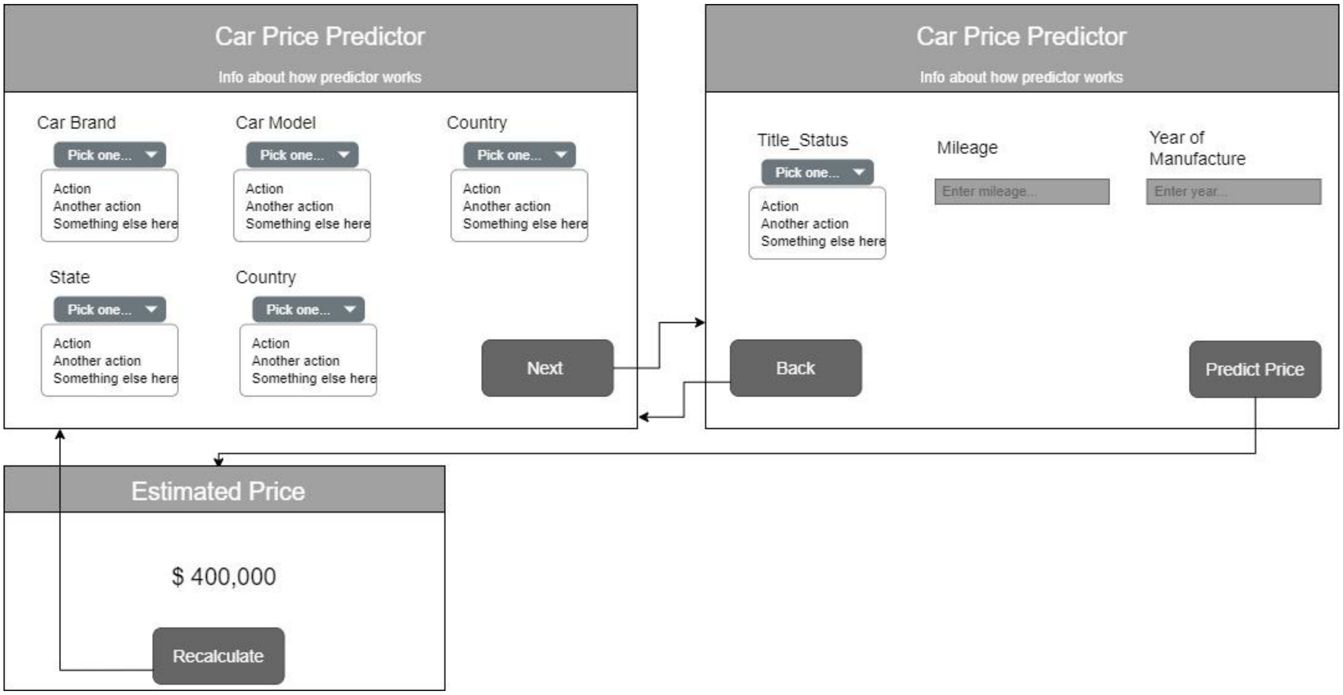
After extensive research and testing different models, we discovered that Linear Regression was the perfect model for our specific data. It is a model that is used to estimate the relationship between two or more quantitative variables. As shown in the data segment, there is a linear relationship between the numerical attributes in the dataset.

FRONT END

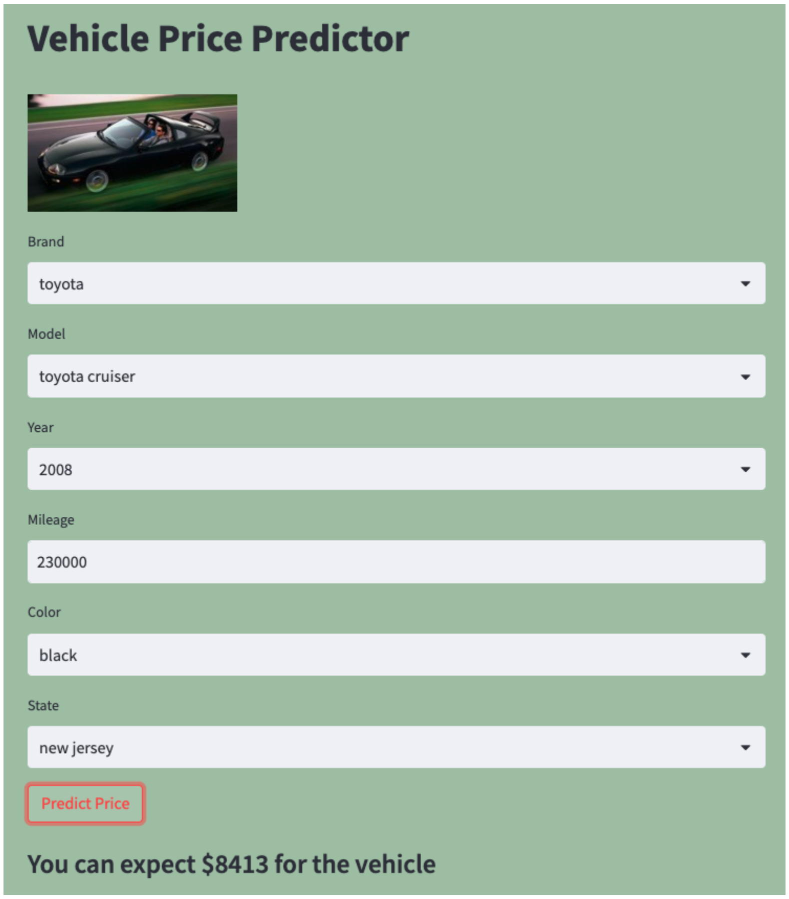
We decided to create a website that is linked to our program. This allows an average user to predict the price of a car and the website outputs the predicted price. In order to create a web app for machine learning in a short period of time, we used a framework called Streamlit.

Wireframe

Our plan was to create a website where users could select the various features describing the car and once they clicked Predict price, it would output the predicted price. The categorical Attributes would have drop down menus, whereas the numerical attributes would have a text box for the users to fill with their desired value.



End Product



Vehicle Price Predictor

Brand

toyota

Model

toyota cruiser

Year

2008

Mileage

230000

Color

black

State

new jersey

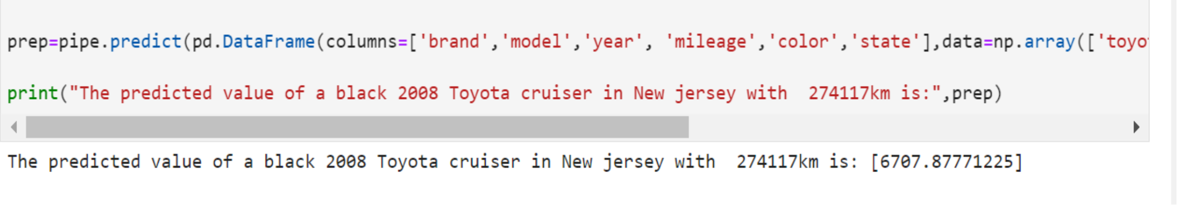
You can expect $8413 for the vehicle

Linking

In order to create an user interface we used Pickle to pass the model, serialize, and convert into a byte stream. Once saved as a .pkl file it could later be used to run the data without having to train the model again. After taking the input from the user, the app.py file we can then assign and run the model's predict function previously created in the .ipynb file.

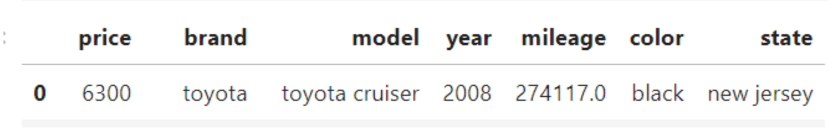
RESULTS

The predictions of car prices are quite close. The first image is the prediction of an existing row in the original dataset. The second image is a screenshot of the row belonging in the dataset. The actual value of the vehicle is $6300 while the predicted value is $6708.



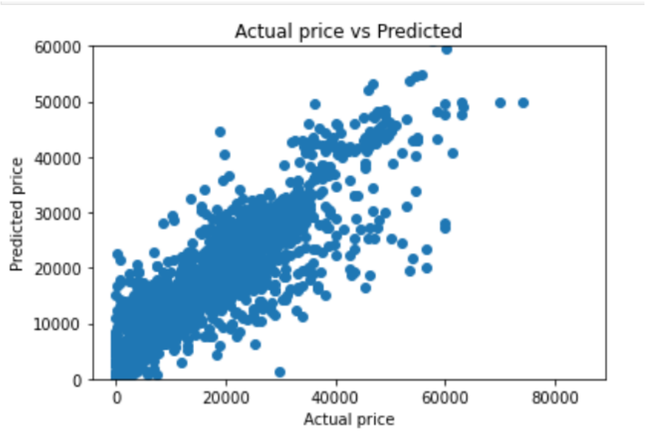
```
prep=pipe.predict(pd.DataFrame(columns=['brand','model','year', 'mileage','color','state'],data=np.array(['toyo

print("The predicted value of a black 2008 Toyota cruiser in New jersey with  274117km is:",prep)
```

```
The predicted value of a black 2008 Toyota cruiser in New jersey with  274117km is: [6707.87771225]
```



	price	brand	model	year	mileage	color	state
0	6300	toyota	toyota cruiser	2008	274117.0	black	new jersey

When the predicted value is plotted against the actual values it also shows that they are quite close. The higher the actual price, the higher the prediction price meaning it is an accurate estimate. The same applies for low prices.



DISCUSSION

Code Structure

First, we prepare the data. Because we were planning to link the code to a website, we realized that we would have to change a few things. While creating the website, we ran into the issue of encoding the string input submitted. To solve this issue, we seeked help from the internet.. We divided the pipeline into two steps.

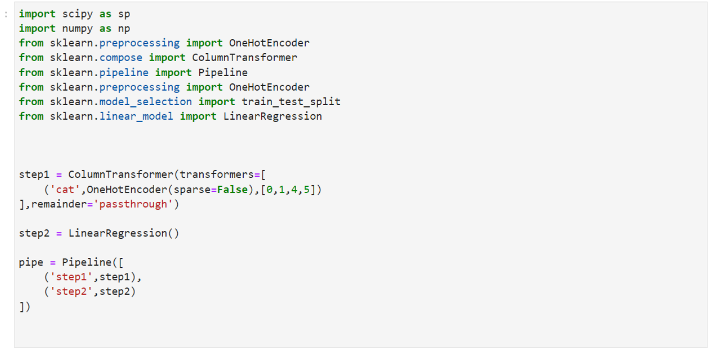
Step 1 - First we prepare the code using OneHotEncoder, to encode the non-numerical data. We did not need to use Simple Imputer because there were no attributes with missing values.

Step 2- We use Linear Regression to predict the values.

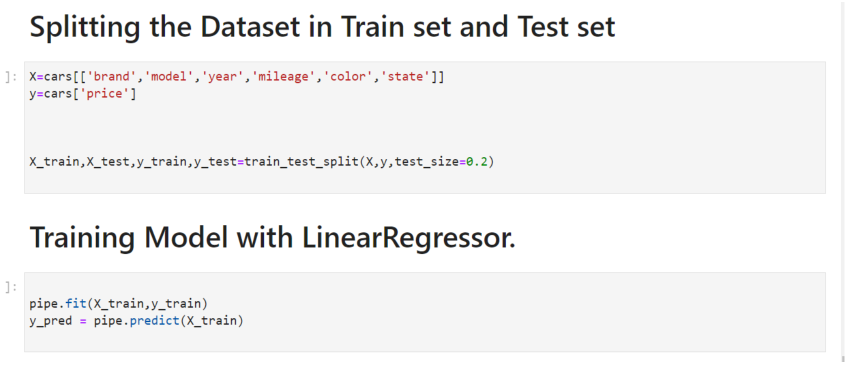
By combining these two into a single module, we are able to easily encode the values collected from the user before a prediction is made.

After that we split the dataset and then fit our model to the training dataset.

After that, using pickle, we export the module we created so it can be used in the website



```
import scipy as sp
import numpy as np
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression


step1 = ColumnTransformer(transformers=[
    ('cat',OneHotEncoder(sparse=False),[0,1,4,5])
],remainder='passthrough')

step2 = LinearRegression()

pipe = Pipeline([
    ('step1',step1),
    ('step2',step2)
])
```

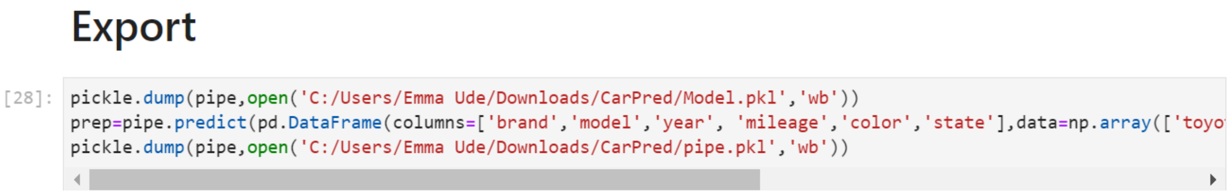


Splitting the Dataset in Train set and Test set

```
X=cars[['brand','model','year','mileage','color','state']]
y=cars['price']


X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)
```

Training Model with LinearRegressor.

```
pipe.fit(X_train,y_train)
y_pred = pipe.predict(X_train)
```



Export

```
[28]: pickle.dump(pipe,open('C:/Users/Emma Ude/Downloads/CarPred/Model.pkl','wb'))
      prep=pipe.predict(pd.DataFrame(columns=['brand','model','year', 'mileage','color','state'],data=np.array(['toyo
      pickle.dump(pipe,open('C:/Users/Emma Ude/Downloads/CarPred/pipe.pkl','wb'))
```

ACKNOWLEDGEMENTS

The dataset, description, and attribute information are all obtained from Kaggle.

REFERENCES

1. Bevans, R. (2022, November 15). *Simple linear regression: An easy introduction & examples*. Scribbr. Retrieved December 1, 2022, from https://www.scribbr.com/statistics/simple-linear-regression/
2. *Car price prediction using linear regression model*. Azure AI Gallery. (n.d.). Retrieved December 1, 2022, from https://gallery.azure.ai/Experiment/Car-Price-Prediction-Using-Linear-Regression-Model
3. Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow by Géron.
4. Youtube. (2021). *Build A Machine Learning Web App From Scratch*. Retrieved December 1, 2022, from https://www.youtube.com/watch?v=xl0N7tHiwlw.