CAR PRICE PREDICTION

Submitted by:

RUPAMANANDA NANDI

ACKNOWLEDGMENT

I am over helmed in all humbleness and gratefulness to acknowledge my depth to all those who have helped me to put these ideas, well above the level of simplicity and into something concrete. I would like to express my special thanks of gratitude to my mentor who gave me the golden opportunity to do this wonderful project on the topic (Ratings Prediction), which also helped me in doing a lot of Research and I came to know about so many new things. I am really thankful to them. Any attempt at any level can ‘t be satisfactorily completed without the support and guidance of my parents and friends. I would like to thank Datatrained helped me a lot in gathering different information, collecting data and guiding me from time to time in making this project, despite of their busy schedules, they gave me different ideas in making this project unique.

INTRODUCTION

- ## Business Problem Framing

  In this section I need to scrape the data of used cars from websites .I need web scraping for this. I have to fetch data for different locations. The number of columns for data doesn't have limit. Generally, these columns are Brand, model, variant, manufacturing year, driven kilometers, fuel, number of owners, location and at last target variable Price of the car.

- ## Conceptual Background of the Domain Problem

  At first, we have to extract a data from a website and the ratings given by the customers and then form a data where we can check the suitable model.

- ## Review of Literature

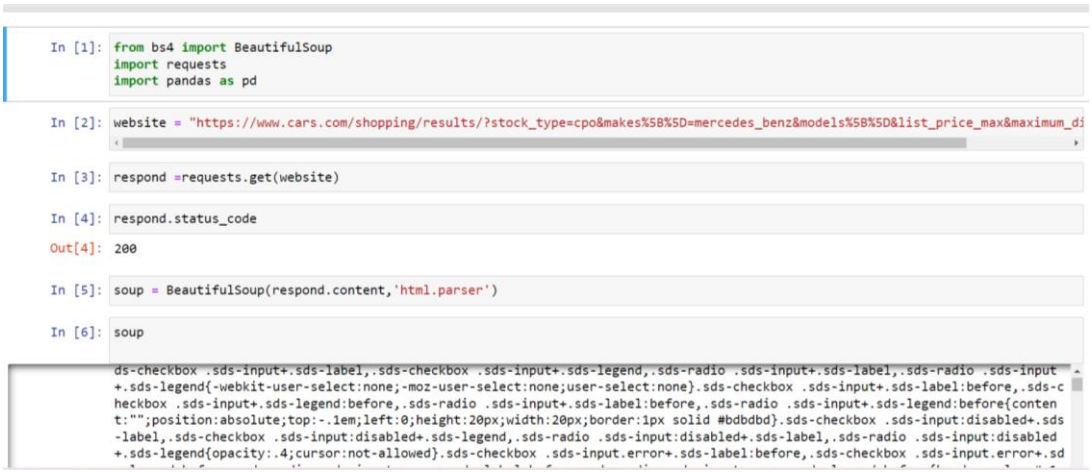
  Firstly I have to web scrap the website and then perform model prediction.

- ## Motivation for the Problem Undertaken

  Often it comes back to better goal setting, making it more compelling (to both the conscious and unconscious mind), being clearer about why you want to do something. NLP, is about the cause-and-effect equation

Analytical Problem Framing

- **Web Scrapping:**



```
In [1]: from bs4 import BeautifulSoup
        import requests
        import pandas as pd

In [2]: website = "https://www.cars.com/shopping/results/?stock_type=cpo&makes%5B%5D=mercedes_benz&models%5B%5D&list_price_max&maximum_di

In [3]: respond =requests.get(website)

In [4]: respond.status_code
Out[4]: 200

In [5]: soup = BeautifulSoup(respond.content,'html.parser')

In [6]: soup

ds-checkbox .sds-input+.sds-label,.sds-checkbox .sds-input+.sds-legend,.sds-radio .sds-input+.sds-label,.sds-radio .sds-input
+.sds-legend{-webkit-user-select:none;-moz-user-select:none;user-select:none}.sds-checkbox .sds-input+.sds-label:before,.sds-c
heckbox .sds-input+.sds-legend:before,.sds-radio .sds-input+.sds-label:before,.sds-radio .sds-input+.sds-legend:before{conten
t:"";position:absolute;top:-.1em;left:0;height:20px;width:20px;border:1px solid #bdbdbd}.sds-checkbox .sds-input:disabled+.sds
-label,.sds-checkbox .sds-input:disabled+.sds-legend,.sds-radio .sds-input:disabled+.sds-label,.sds-radio .sds-input:disabled
+.sds-legend{opacity:.4;cursor:not-allowed}.sds-checkbox .sds-input.error+.sds-label:before,.sds-checkbox .sds-input.error+.sd
```

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

Target necessary data

```
In [10]: # Name
         # Mileage
         # Rating
         # Rating count
         # Price

In [11]: results[0].find('h2').get_text()
Out[11]: '2018 Mercedes-Benz S-Class S 450 4MATIC'

In [12]: results[0].find('div',{'class':'mileage'}).get_text()
Out[12]: '33,834 mi.'

In [13]: results[0].find('span',{'class':'sds-rating__count'}).get_text()
Out[13]: '4.8'

In [14]: results[0].find('span',{'class':'sds-rating__link sds-button-link'}).get_text()
Out[14]: '(1,546 reviews)'

In [15]: results[0].find('span',{'class':'primary-price'}).get_text()
```

- Testing of Identified Approaches (Algorithms)

Put everything together inside a for-loop

```
In [17]: name = []
         mileage = []
         dealer_name = []
         rating = []
         rating_count = []
         price = []


         for result in results:
             #name
             try:
                 name.append(result.find('h2').get_text())
             except:
                 name.append('n/a')


              #mileage
             try:
                 mileage.append(result.find('div',{'class':'mileage'}).get_text())
             except:
                 mileage.append('n/a')

              #dealer_name
```

Create Pandas dataframe

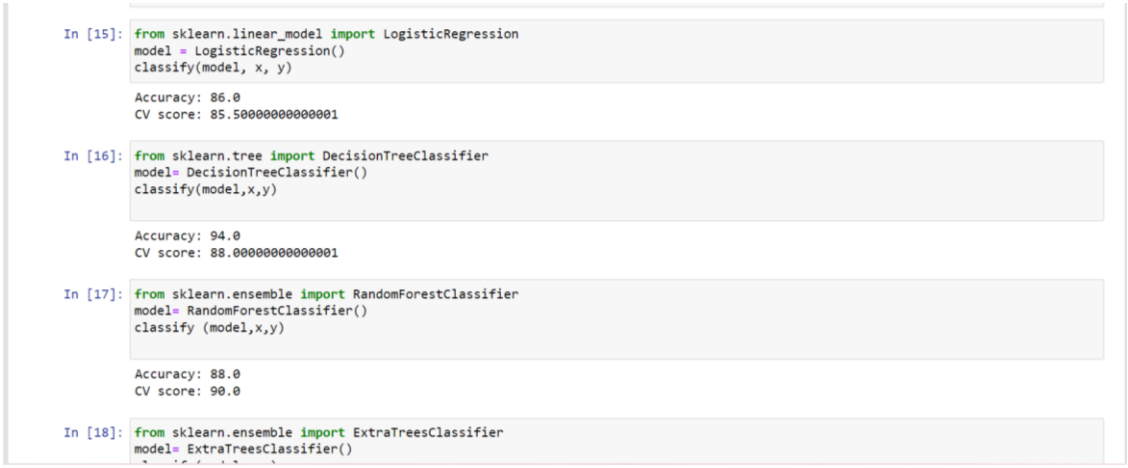
```
In [19]: car_dealer = pd.DataFrame({'Name':name, 'Mileage':mileage, 'Dealer Name': dealer_name,
                                    'Rating':rating, 'rating_count':rating_count , 'price':price})

In [20]: car_dealer
```

Out[20]:

	Name	Mileage	Dealer Name	Rating	rating_count	price
0	2018 Mercedes-Benz S-Class S 450 4MATIC	33,834 mi.	Mercedes-Benz of Massapequa	4.8	(1,546 reviews)	$69,995
1	2019 Mercedes-Benz AMG GLE 43 4MATIC Coupe	12,010 mi.	Mercedes-Benz Of Tampa	4.4	(104 reviews)	$80,990
2	2019 Mercedes-Benz C-Class C 300	22,289 mi.	Autobahn Motors	4.6	(890 reviews)	$32,998
3	2018 Mercedes-Benz E-Class E 300	26,095 mi.	Mercedes-Benz of Fayetteville	4.9	(597 reviews)	$43,998
4	2019 Mercedes-Benz C-Class C 300	11,673 mi.	Mercedes-Benz of South Charlotte	4.8	(161 reviews)	$41,954
5	2021 Mercedes-Benz C-Class C 300	9,971 mi.	Mercedes-Benz of Covington	4.8	(410 reviews)	$44,000
6	2018 Mercedes-Benz GLE 350 Base 4MATIC	32,274 mi.	Mercedes-Benz of San Francisco / Smart Center ...	3.6	(13 reviews)	$37,970
7	2017 Mercedes-Benz GLS 450 Base 4MATIC	25,119 mi.	Mercedes-Benz of Bellevue	4.7	(482 reviews)	$49,585
8	2019 Mercedes-Benz AMG E 53 4MATIC	20,741 mi.	Mercedes-Benz of Westwood	4.8	(124 reviews)	$71,298
9	2018 Mercedes-Benz E-Class E 300	33,285 mi.	Mercedes-Benz of Fayetteville	4.9	(597 reviews)	$42,500
10	2019 Mercedes-Benz C-Class C 300	20,505 mi.	Mercedes-Benz of South Charlotte	4.8	(161 reviews)	$57,724
11	2019 Mercedes-Benz GLC 300 Base	22,041 mi.	RBM of Atlanta	4.9	(1,001 reviews)	$39,968

- Run and Evaluate selected models



```
In [15]: from sklearn.linear_model import LogisticRegression
         model = LogisticRegression()
         classify(model, x, y)

         Accuracy: 86.0
         CV score: 85.50000000000001

In [16]: from sklearn.tree import DecisionTreeClassifier
         model= DecisionTreeClassifier()
         classify(model,x,y)

         Accuracy: 94.0
         CV score: 88.00000000000001

In [17]: from sklearn.ensemble import RandomForestClassifier
         model= RandomForestClassifier()
         classify (model,x,y)

         Accuracy: 88.0
         CV score: 90.0

In [18]: from sklearn.ensemble import ExtraTreesClassifier
         model= ExtraTreesClassifier()
```

CONCLUSION

- Key Findings and Conclusions of the Study

  Creating a Dataset and finding the best possible model.