Indian Institute of Information Technology and Management

Department of Computer Science

Mini Project Report

Masters of Computer Science
Data Analytics

Sentiment Analysis of Consumer Review

Project Guide: Dr. Asharaf S
Project Mentor: Nikhil V Chandran

Submitted by:

Palle Lakshmi Parimala
E-Mail: palle.da20@iiitmk.ac.in

Kevin S Varghese
E-Mail: kevin.da20@iiitmk.ac.in

July 1, 2021

Contents

ACKNOWLEDGEMENT

There is no greater joy than that of a job well done. Any success falls short of its purpose when due acknowledgement is not given to those who have been a part of it. We are ever grateful to the Almighty for guiding us through this time. We would like to express our sincere appreciation and gratitude to IIITMK for having given us the opportunity to be a part of this project.

We would like to thank and extend our heartfelt gratitude to our project supervisor, Dr. Asharaf S. and our mentor Prof. Nikhil V Chandran, who have always been more than willing to help us at any stage of the project and for helping with any queries and suggestions that aims to take the project to greater heights.

Let us also express our obligation to all the people who have indirectly contributed to this work.

Abstract

Since time immemorial customer sentiment have driven the commercial market and industry including the automobile industry. The automotive industry is an extremely dynamic and competitive industry with a lot of monetary potential. This massive level of competition among corporations has led to release of a substantial number of passenger vehicles-flooding the car market, which has made the process of selecting the 'right' car based on various attributes a mammoth task for a consumer. For the automobile corporations, understanding customers' sentiment and expectations are fundamental for attracting and retaining consumers. We will analyze consumer sentiment and point of view for various Passenger vehicle segments by performing Fine grain analysis on data extracted from Indian Autoportal and then Create a Multinomial Naive Bayes model to classify any given reviews.

Chapter 1

Introduction

In this project we intend to measure the sentiment of any given user review through Natural Language Processing and create a machine learning classification model. Throughout this project we've hit myriads of dead ends and through various chapters we'll touch upon all the relevant points. We have a literature review chapter which contains the theoretical nuances which was important for our understanding of concepts related to Web technologies, NLP and Machine learning algorithm.
The third chapter is a detailed explanation of the code and the related output and the methods used to extract relevant information from the python script.

1.1 Outline of the Method

Through this project we intend to better understand the vehicle attributes influencing customers' purchase decision, by undergoing four stages:

The first stage will consist of web scraping of review sources, this step would entail collecting key vehicle attributes from different automotive review websites such as Carwale and organizing this data.

In the second stage we have used the Regular expression library to preprocess text data to mitigate any error due to inconsistencies or inaccuracy.

The third stage will aim to perform the sentiment analysis for each review using two different libraries, namely, VADER (Valence Aware Dictionary and sEntiment Reasoner) and TextBlob library.

In the end, we will use our findings to build a classification model and check the accuracy of the model

Chapter 2

Literature Review

2.1 Web Scraping with Selenium

Web scraping refers to the extraction of data from a website. This information is collected and then exported into a format that is more useful for the user.
The web scraper, which can be proprietary software or self-built, will be given one or more URLs to load before scraping. The scraper then loads the entire HTML code for the page in question. More advanced scrapers will render the entire website, including CSS and Javascript elements.

Selenium is one of the most renowned open-source test automation frameworks. Selenium allows test automation of web-apps or websites across different browsers & operating systems. Selenium offers compatibility with multiple programming languages such as Java, JavaScript, and Python in addition Selenium supports multiple operating systems, this versatility made us choose selenium for our project over other available tools.

WebDriver Architecture is made up of four major components:

1. Selenium Client library
2. JSON wire protocol
3. Browser Drivers
4. Web Browsers

Let's understand how the components work to facilitate our code

H

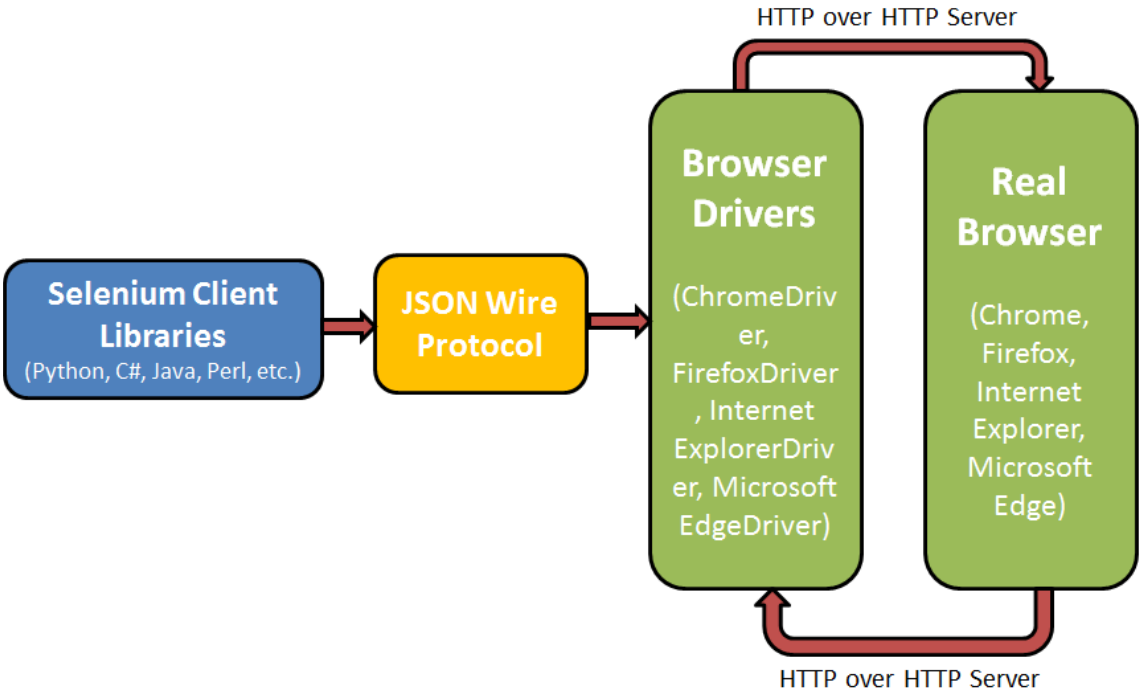


Figure 2.1: Selenium Webdriver Architecture

2.1.1 Selenium Client library

Selenium client library or Selenium language binding is the initial step to utilize the full potential of Selenium, this aspect of selenium facilitates multi-language support. To interact with the Selenium Server (Remote WebDriver) or create local Selenium WebDriver scripts, since our project is written in Python, we have used language-specific client drivers - Python binding.

2.1.2 JavaScript Object Notation

JSON (JavaScript Object Notation) Wire Protocol facilitates the capability of transferring the data between the Client and Server on the web. In simpler words, JSON Wire protocol is used when a client has an object that has to be sent to a server. The client converts this object into a JSON object and sends it to the server. The server parses the JSON object and converts it back to object for use. The server converts the response object into a JSON object and sends it back to the client. The client receives input from sever in JSON format and then uses it to extract relevant information.

For our code, JSON wire protocol will act as an abstract specification of how automation behaviors like clicking or typing will be mapped to selenium.

2.1.3 Browser Driver

Browser Drivers are used for interacting with browsers and relaying automation script instructions to the browsers. Browser drivers take care of loss of any internal logic of browser functionalities. Each browser has its specific Browser Webdriver. WebDriver's aim is to emulate a real user's interaction with the browser as closely as possible.

When a test script is executed with the help of WebDriver, the following tasks are performed in the background:

- An HTTP request is generated and it is delivered to the browser driver for every Selenium Command
- The HTTP request is received by the driver through an HTTP server
- All the steps/instructions to be executed on the browser is decided by an HTTP server
- The HTTP server then receives the execution status and in turn sends it back to the automation scripts

2.2 Sentiment Analysis

Sentiment Analysis is a predominantly classification algorithm aimed at finding an opinionated point of view and its disposition and highlighting the information of particular interest in the process. It monitors conversations and evaluates language to quantify attitudes, opinions, and emotions related to a business, product or service, or topic. Sentiment analysis is sometimes also referred to as opinion mining.

One of the major sentiment analysis algorithms is a rule-based approach.

2.2.1 Rule-Based Approach

Rule based approach is used by defining various rules for getting the opinion or it uses lexical rules, created by tokenizing each sentence in every document and then testing each token, or word, for its presence. If the word is there and has a positive sentiment, a +1 rating was applied to it. Each post starts with a neutral score of zero, and was considered positive if the final polarity score was greater than zero, or negative if the overall score was less than zero. In this, certain rules are to be formed and then the sentiments should be analyzed depending on it . The rule based approach result creates the rules by taking: Affecting words, Inverted words, and Negation words.

1. After the output of the rule based approach it will check or ask whether the output is correct or not. If the input sentence contains any word which is not present in the database which may help in the analysis of movie review, then such words are to be added to the database.
2. This is supervised learning in which the system is trained to learn if any new input is given.
3. This approach will always increase the efficiency of the system.

Here we will use two different libraries to perform sentiment analysis , to compare labels data to better fit the model.

2.2.1.1 NLTK's Pre-Trained Sentiment Analyzer

NLTK already has a built-in, pretrained sentiment analyzer called VADER (Valence Aware Dictionary and sEntiment Reasoner).
VADER Sentiment Analyzer was applied to the dataset. VADER is a rule-based sentiment analysis tool and a lexicon that is used to express sentiments. First, we created a sentiment intensity analyzer to categorize our dataset. The VADER Sentiment Analyzer was used to classify the preprocessed reviews. The polarity_scores method is used to determine the sentiment.

2.2.1.2 Sentiment Analysis using TextBlob

TextBlob is a python library for Natural Language Processing (NLP).TextBlob actively uses Natural Language ToolKit (NLTK) to achieve its tasks. TextBlob is a simple library which supports complex analysis and operations on textual data.The TextBlob Naive-BayesAnalyzer is apparently based on the Stanford NLTK. TextBlob returns **polarity** and **subjectivity** of a sentence. Polarity lies between [-1,1], -1 defines a negative sentiment and 1 defines a positive sentiment. Negation words reverse the polarity. TextBlob has semantic labels that help with fine-grained analysis. For example — emoticons, exclamation marks, emojis, etc. Subjectivity lies between [0,1]. Subjectivity quantifies the amount of personal opinion and factual information contained in the text. The higher subjectivity means that the text contains personal opinion rather than factual information. TextBlob has one more parameter — intensity. TextBlob calculates subjectivity by looking at the **'intensity'**. Intensity determines if a word modifies the next word. For English, adverbs are used as modifiers ('very good').

2.3 Multinomial Naïve Bayes

Multinomial Naïve Bayes is a Naïve Bayes algorithm which administers Multinomial data in text classification. Data in Multinomial Naïve Bayes is represented as the total of data vector, thus Multinomial Naïve Bayes is appropriate for estimating term frequency in a document. In Multinomial Naïve Bayes, firstly the probability of words in a class (prior) was computed as Equation 2.1 :

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c) \tag{2.1}$$

where $P(t_k|c)$ is the conditional probability of the term t_k occurring in a document d of a class c. P(c) is the prior probability of a document occurring in class c. The probability of document d in c was performed using Equation 2.2 :

$$\hat{P}(c) = \frac{N_c}{N} \tag{2.2}$$

N_C is the number of documents in class c and N is the total number of documents. The conditional probability P(t — c) as the relative frequency of term t in documents belonging to class c, as in Equation 2.3 :

$$\hat{P}(t|c) = \frac{T_{ct}}{\sum_{t' \epsilon V} T_{ct'}} \tag{2.3}$$

T_{ct} is the number of occurrences of term in training documents from class c . $\sum_{t' \epsilon V} T_{ct'}$ is the number of all terms in the whole document in class c including redundant term in the same document. The sparseness of term in the document resulted the estimation of frequency $P(t_k|c)$ w as zero (0),thus we added one or laplace smoothing as Equation 2.4:

$$\hat{P}(t|c) = \frac{T_{ct} + 1}{\sum_{t' \epsilon V} T_{ct'} + 1} = \frac{T_{ct} + 1}{(\sum_{t' \epsilon V} T_{ct'}) + B'} \tag{2.4}$$

Where $B = |V|$ is the number of terms in the vocabulary in training data. The algorithm of Multinomial Naïve Bayes for training step is as follows :

Multinomial NB *Training* (C,D)
V ←extract vocabulary (D)
N ←count documents (D)
for each*c* ∈C
do N_c ←count documents in class (D, *c*)
prior [*c*] ← N_c/ *N*
$text_c$ ← concatenate text of all documents in class (D, *c*)
for each*t* ∈ *V*
doT_{ct} ←count tokens of term ($text_c$, *t*)
for each *t* ∈ *V*
do *condprob*[*t*][*c*] ← $\frac{T_{ct}+1}{\sum_{t'}(T_{ct'}+1)}$
return*V*, *prior*, *condprob*

Training process in Multinomial Naïve Bayes algorithm is the step of the training the stage of data whereas the training sessions to the data describing that V is the number of terms in the vocabulary in the training data then N is the number of times of data that sentences Twitter comments. For each class of data is calculated on the amount of data in each class(NC) divided by the large number of data (NC—N) as listed in the Equation 2.2 and then for each text feature (t) in V calculated the odds on each class using the Equation 2.3. If the conditions are not zero value then do laplace smoothing salty Equation 2.4. the next step is to apply the testing stages as follows:

Apply Multinomial NB (*C, V, prior, condprob, d*)
W ←extract all tokens from document (*V, d*)
for each *c* ∈C
do *score* [*c*] ← log *prior* [*c*]
for each*t* ∈ *W*
do *score* [*c*] + = log *condprob* [*t*][*c*]
return $\text{argmax}_{c \in C}$*score*[*c*]

At the stage of testing applied to the testing data for each class by calculating a score using Equation 2, then for each text features were calculated using Equation 3. If there are features worth zero then applied to the Equation 4. Application testing phase resulted in the probability of each class so the highest probability value is the winner of the class that is the biggest opportunity in each document.

Chapter 3

Working

3.1 Webscraping

We discussed the architecture of Selenium Webdriver, now let's understand how we scraped Carwale.com

1. Libraries imported:

```
import selenium
from selenium import webdriver
```

   The **selenium package** is used to automate web browser interaction from Python. Selenium requires a driver to interface with the chosen browser.

```
from tqdm import tqdm
```

   Through this project we explored the use of **tqdm library**, tqdm was especially useful since the dataset which was scraped from website were of significant size. tqdm instantly make the loops show a smart progress meter this helped us to track the progress of the progress which were made while downloading the dataset. tqdm takes the iterable and wraps it in an object that returns the same values, and also keeps track of the progress.

```
import pandas as pd
```

   We have also employed **Pandas** to convert the extracted data into data frame which would then be used for sentiment analysis.

2. To utilize Selenium, we are required to download webdriver for the involved web browser, as we have discussed WebDriver is usually an open source tool for automated testing of webapps across many browsers. It provides capabilities for navigating to web pages, user input, JavaScript execution, and more. For this code we'll be sticking to Google chrome driver, ChromeDriver which is a standalone server that implements the W3C WebDriver standard. Every time we run selenium, we will have to reference the location where we have stored the chromedriver.

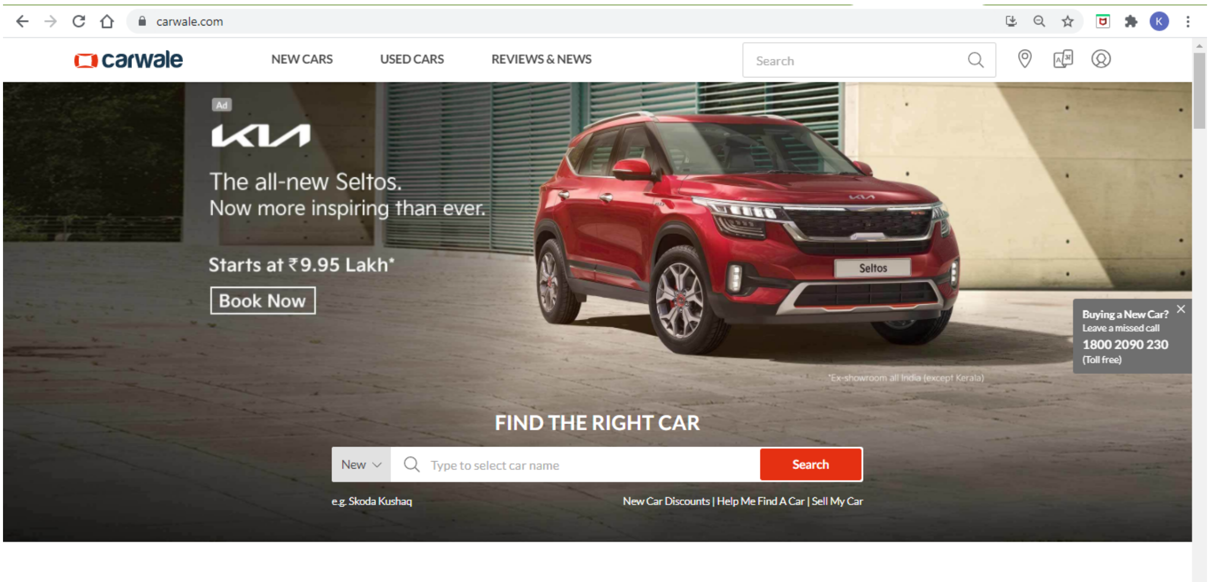
Post this, using the get method in selenium we'll access the website which we want to access.

```
path='C:\Program Files (x86)\chromedriver.exe'

driver=webdriver.Chrome(path)

driver.get('https://www.carwale.com/')
```

Implementing the above mentioned code, the system will open a new chrome browser with the required URL, shown as below

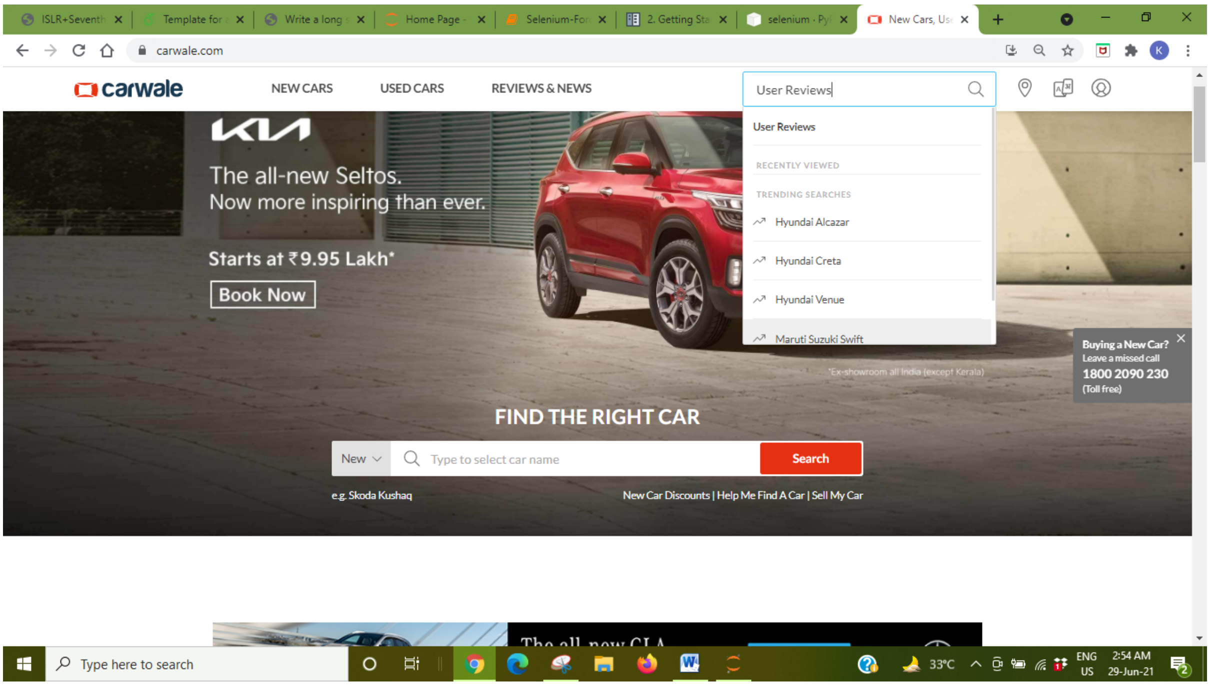


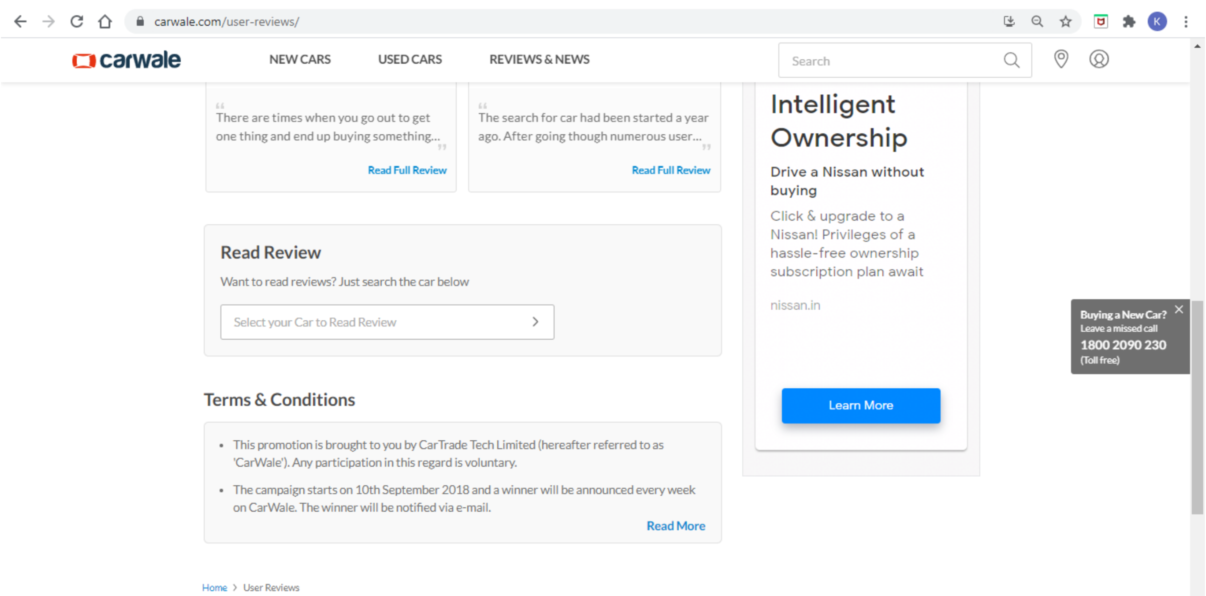
3. After arriving at the home page, we'll navigate our way to the user reviews page, to achieve that we'll manipulate the HTML structure of the page. This could be done by using XPath, XPath is a technique in Selenium to navigate through the HTML structure of a page and to handle dynamic element of the page. The simplest XPath locator in Selenium is to provide the absolute path of an element in the DOM structure.
   Here, we will create the Xpath using attribute, the method to form the XPath is:
   \\tagname[@class_name='attribute_name']
   We will use this method to interact with various aspects of the webpage

```
action1=driver.find_element_by_xpath('//input[@class="o-dqBOwT o-cKuOoN o-eZCpVk o-bfyaNx o-eZTujG o-jjpuv o-fzptVd o-fzptYr \
                                      o-fEEqXL o-eKWNKE o-bIMsfE o-ItVGT o-fznJDS  E6NC89 _2XUNC9  _32ibOJ E6NC89 _1VxuJc"]')
action2=action1.send_keys('User Reviews')

a=driver.find_element_by_xpath('//li[@class="o-brXWGL o-fzptZB o-fzptOP  o-bCRRBE o-frwuxB _38PfqU _3wQ1D6 "]')
a.click()
```

In this set of code, we are accessing the 'Search Box' and then using 'send_key' function enter User Reviews to access the reviews page. The below mentioned image shows the progress to reviews page.

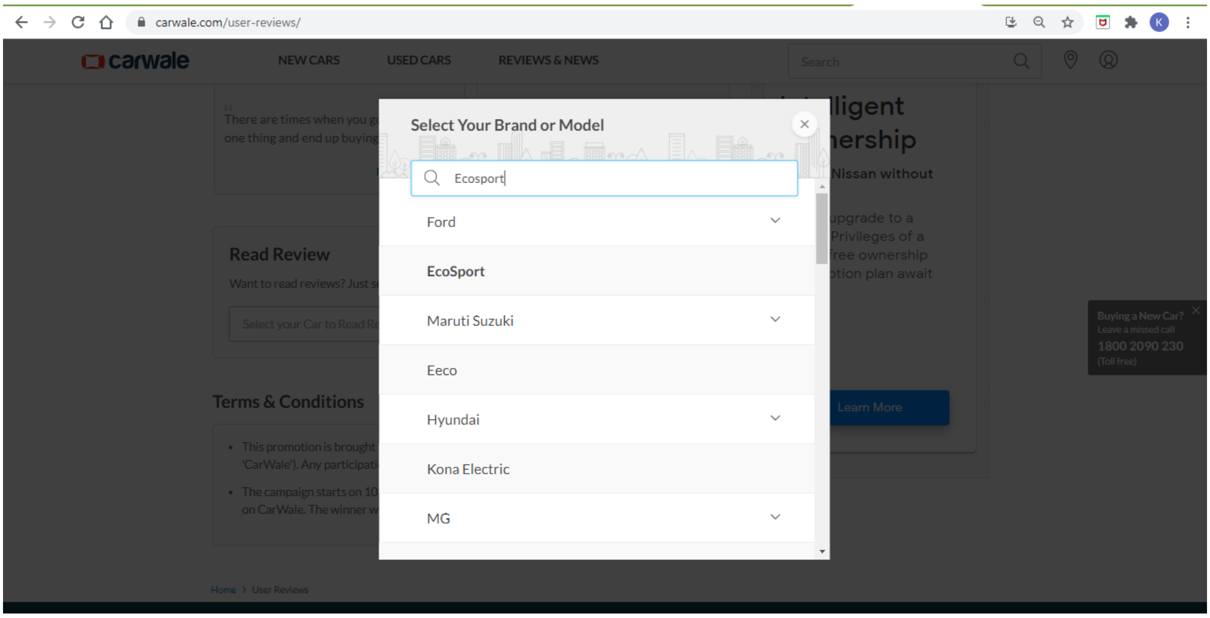




4. Now, we need to enter the name of the car in the 'Read Review' search bar, we achieve this using this Xpath, and then enter the Name of the car, here 'Ecosport'.

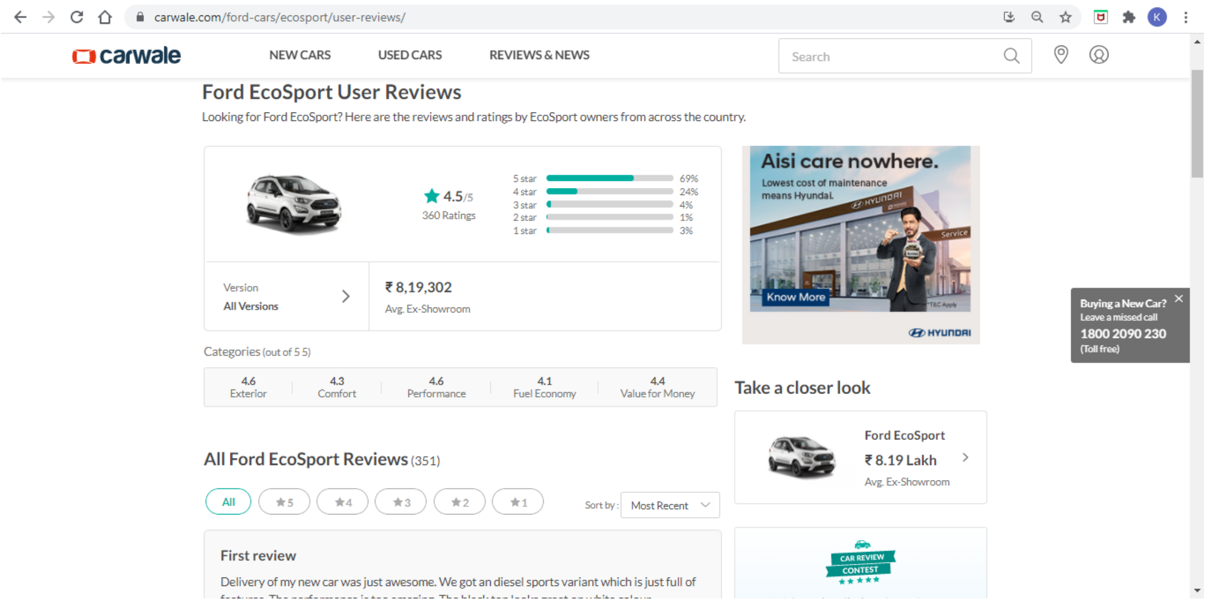
```
car_select=driver.find_element_by_xpath('//input[@data-testing-id="search-box-mmv-popup-input"]')
car_select.send_keys('EcoSport')
```

After this we will get a drop-down list, from which we will select the required car.



5. After clicking we arrive at the Page where the reviews for Ford Ecosport are present. This is the page from which we will extract the data.

```
select1=driver.find_element_by_xpath('//mark[@class="o-eqqVmt o-eZTujG o-foCBAZ"]')
select1.click()
```



We will now extract the URL of this page, we will use this to traverse through all the pages for a particular car

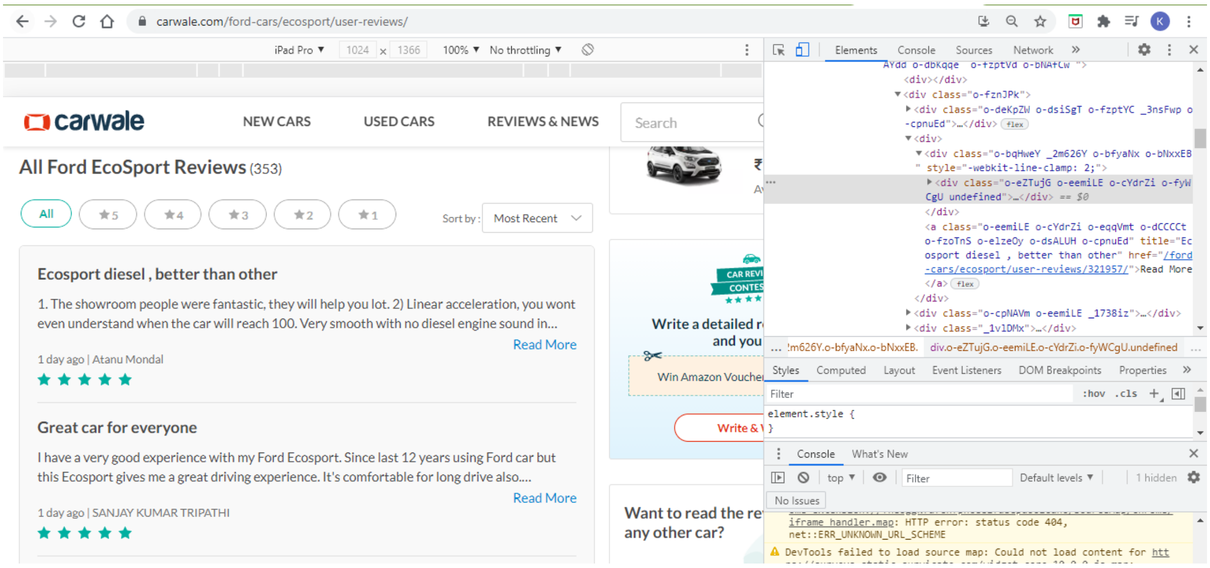
```
url=(driver.current_url)
url=url[:-1]
url
```

'https://www.carwale.com/ford-cars/ecosport/user-reviews'

We use, 'execute_script' and 'window.scroll' to reach the end of the page to get total number of pages available

```
# To get the number of pages
driver.execute_script("window.scrollTo(0,document.body.scrollHeight)")
```

6. After a long iteration of finding the right XPath we were able to get access to the review content by each consumer, in the code below we have defined two lists to store Review title and review content where we have stored the corresponding inputs. In this portion of code, we are using two nested loops, inner loop will iterate over the reviews in one page and the outer loop will move through all the pages. Here tqdm has proved very helpful, as it has helped us determine the time required to complete the code.



```
Review_title=[]
Review=[]

for i in tqdm(range(1,36)):
    driver.get(url+'-p'+str(i))

    titles=driver.find_elements_by_xpath('//div[@class="o-bqHweY _2m626Y o-bfyaNx o-bNxxEB o-jjpuv o-dqVwuv o-byFsZJ \
                                         o-dsJzhG o-eZTujG"]')
    for title in (titles):
        Review_title.append(title.text)

    reviews=driver.find_elements_by_xpath('//div[@class="o-eZTujG o-eemiLE o-cYdrZi o-fyWCgU undefined"]')
    for review in (reviews):
        Review.append(review.text)
```

```
100%|██████████████████████████████████████████| 35/35 [03:52<00:00,  6.65s/it]
```

7. We then convert these lists into a CSV file using pandas, the dataframe has also been given below.

```
df = pd.DataFrame(list(zip(Review_title,Review)),
                 columns =['Review Title','Review'])
df['Car Name']='Ford EcoSport'
df.to_csv('Ford EcoSport.csv')
df
```

	Review Title	Review	Car Name
0	Gazab	Awesome car very comfortable very very good sm...	Ford EcoSport
1	Ford EcoSport review	The car was good looking big with excellent fe...	Ford EcoSport
2	It's good in your budget	Driving & performance is good, budget car whic...	Ford EcoSport
3	The drivers car	Dealer was very cordial and friendly, steering...	Ford EcoSport
4	Ecosport Thunder edition 202,	This is my first car. Driving comfort is so ni...	Ford EcoSport
...	...	...	...
341	The most fun to drive car!!!	It's always been a great experience for owning...	Ford EcoSport
342	Award to Ecosport as a customer	We bought a top end model as per my wish its w...	Ford EcoSport
343	Ford Ecosport	The Ford Ecosport vehicle is one of the best c...	Ford EcoSport
344	Grab this car and enjoy your ride	1. It is worth to putting your money to this c...	Ford EcoSport
345	Very much better than other cars in the segment	It's been a time since I had purchased the For...	Ford EcoSport

346 rows × 3 columns

8. We will repeat the above mentioned process with various other cars, by changing the car name at 'Step 4'. For this project we collected the reviews for:
    - Honda Amaze
    - Ford Ecosport
    - Kia Seltos
    - Maruti Suzuki Baleno
    - Hyundai Venue
    - Mahindra Scorpio

3.2 Text Preprocessing

In pre-processing of text meaningless words are to be removed which are not describing any sentiments. The words like this, it, is, am, are, who, or, etc does not give any idea for analysis of sentiments. So these words must be discarded from the input. Python libraries regular expression is used to preprocess the reviews. The preprocessing happens in the following steps :

- Contract the words such as won't , can't to will not and can not.
- Downloading the stop words from nltk.corpus library and removing the stopwords from the reviews.
- Removing all the special characters
- Clearing all the punctuation and changing all the reviews into lowercase.

3.3 Sentiment Analysis

NLTK's VADER method
This method takes a sentence as input and returns a dictionary with keys positive , negative , neutral and compound. Positive , negative and neutral scores are the probabilities of a sentence being any one of them and the compound score is the sum of the probabilities which are normalized. At any given dictionary the sum of the positive, negative and neutral scores is equal to 1. The compound value is used as a polarity score to understand whether a given review is positive or negative or neutral. It lies between -1 and 1.For our project we have tried to divide these compound scores into five different ranges in order to give it a more fine grained sentiment analysis.These structures are classified by the cut function in pandas.

The classes are :

- 1 for most positive review
- 2 for slightly positive review
- 3 for a neutral review
- 4 for slightly negative review
- 5 for a most negative review

	Review Title	Review	Car Name	preprocessed_reviews	vader_score	vader_rating
0	Desirable car	As I experienced the amaze is one the best car...	Honda Amaze	experienced amaze one best car price range com...	0.9590	5
1	Best car I recommend to all	I bought this car in 2018 .I have no problem w...	Honda Amaze	bought car 2018 problem car mileage awesome po...	0.2575	4
2	Amazing car for Nuclear Family	It's best sub compact sedan car , spacious ,mi...	Honda Amaze	best sub compact sedan car spacious mileage re...	0.9847	5
3	One word - practical.	Start from outside looks - you don't need my w...	Honda Amaze	start outside looks need word stunning look ug...	0.8074	5
4	Best in it's Segment.	Before buying this car I am not much sure abou...	Honda Amaze	buying car much sure performance using 2 years...	0.9637	5

TextBlob method
Following the similar approach as Vader sentiment analyser , the polarity score is considered as the sentiment score of the review. And the same classes are divided for textblob polarity scores as well.

After acquiring the vader compounds scores and the respective classes and also the textblob polarity scores and their respective classes for each review , each review is labelled now. There are a total of 2 columns for the labels where one column represents the Vader compound rating and other column represents the textblob popularity rating.

	Review Title	Review	Car Name	preprocessed_reviews	vader_score	vader_rating	textblob_score	textblob_rating
0	Desirable car	As I experienced the amaze is one the best car...	Honda Amaze	experienced amaze one best car price range com...	0.9590	5	0.680000	5
1	Best car I recommend to all	I bought this car in 2018 .I have no problem w...	Honda Amaze	bought car 2018 problem car mileage awesome po...	0.2575	4	0.167083	3
2	Amazing car for Nuclear Family	It's best sub compact sedan car , spacious ,mi...	Honda Amaze	best sub compact sedan car spacious mileage re...	0.9847	5	0.871429	5
3	One word - practical.	Start from outside looks - you don't need my w...	Honda Amaze	start outside looks need word stunning look ug...	0.8074	5	0.097470	3
4	Best in it's Segment.	Before buying this car I am not much sure abou...	Honda Amaze	buying car much sure performance using 2 years...	0.9637	5	0.625000	4
...	...	...	...	...	...	...	...	...

These are the count plots for vader rating and textblob rating.

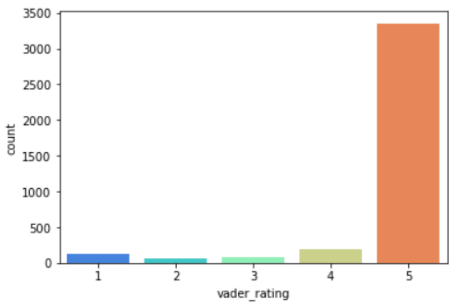


Figure 3.1: VADER's result

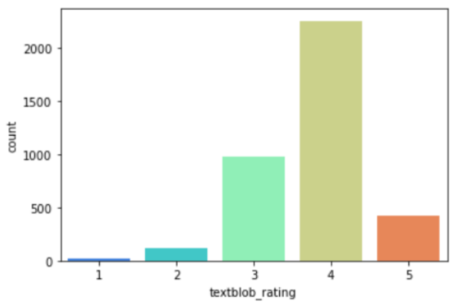


Figure 3.2: Textblob's result

3.4 Machine Learning Model

After the labelling of data in 1 to 5 classes , We know use this labelled data to create a classification model using Multinomial Naive Bayes Algorithm. One of the reasons to use this model is that it works well for text classification. Since we have labelled the data using two different sentiment analyzers , We are going to consider 2 target columns , Y_1 is the Vader rating and Y_2 is the textblob rating. The independent column X would be the preprocessed reviews. Before splitting the data , CountVectorizer() method is used to convert each preprocessed review into vectors for passing them into the model.

Training and splitting data : From scikit learn library , in model_selection module , train_test_split method is used to split the X and Y_1 into training and testing data and the same is done with X and Y_2.

Model fitting and evaluation : An object of MultinomialNB , from scikit learn , nb is created. The training data of X and Y1 is passed through fit() method and the same is done for training data of X and Y_2 . The predictions of the testing data are stored in `pred_1` for vader predictions and `pred_2` for textblob predictions.

Evaluation of vader predictions and target values using confusion matrix and classification report we get an accuracy of , 6%

```
VADER POLARITY EVALUATION


[[  0   0   7  33   0]
 [  0   0   2  25   0]
 [  0   0   2  26   0]
 [  0   0   0  61   0]
 [  0   0   1 987   0]]

              precision    recall  f1-score   support

           1       0.00      0.00      0.00        40
           2       0.00      0.00      0.00        27
           3       0.17      0.07      0.10        28
           4       0.05      1.00      0.10        61
           5       0.00      0.00      0.00       988

    accuracy                           0.06      1144
   macro avg       0.04      0.21      0.04      1144
weighted avg       0.01      0.06      0.01      1144
```

Evaluation of Textblob predictions and target values using confusion matrix and classification report we get an accuracy of , 59 %.

```
TEXTBLOB POLARITY EVALUATION


[[  0   0   1   8   0]
 [  0   0   5  38   0]
 [  0   0   6 280   0]
 [  0   0   0 668   0]
 [  0   0   0 138   0]]

              precision    recall  f1-score   support

           1       0.00      0.00      0.00         9
           2       0.00      0.00      0.00        43
           3       0.50      0.02      0.04       286
           4       0.59      1.00      0.74       668
           5       0.00      0.00      0.00       138

    accuracy                           0.59      1144
   macro avg       0.22      0.20      0.16      1144
weighted avg       0.47      0.59      0.44      1144
```

As we can see , we couldn't exactly predict classes 1,2,4,5 from Vader rating and the classes of 1,2,5 are not predicted well because the data is inconsistent.

Conclusion

In this project , we try to create a model that would predict emotion behind a car review. To build such a model the first and foremost necessity was to collect data. Selenium and beautiful soup are used to scrape data from carwale.com. package is used to create an interaction between an automated web browser and the python IDE. tqdm library is used to keep track of the progress made while downloading the data set. After the collection of data it is necessary to clean and preprocess the reviews , this is done using regular expression library from python. As this is a classification model it is important to me label the pre process data and then fit the model with labelled data.
For labelling of the data we have used two different sentiment analysis libraries.

1. NLTK'S sentiment analyzer, VADER.
2. Textblob

Two target columns are created. Multinomial naive Bayes model is used to classify the text into the given classes.
Model is fitted for two different target columns , for VADER scores and textblob scores. In a way we also tried to compare which sentiment analyser would be better for our reviews. After predicting and evaluating the results we conclude that textblob sentiment and analyser works better for our model since these reviews are more detailed. I also conclude that our Navie Bayes model would predict the emotion behind a review with 59% accuracy.

Future Scope

The model that we created might work good for the reviews which have one or two sentences but for reviews with more sentences , this model might not be as accurate in predicting the emotion. Since we have taken the reviews of a car , reviews might be subjective to many aspects of the car. The review might talk positively about one aspect of the car, whilst it might be talking negatively about two other aspects of the car. Since a car is a huge investment or asset the reviews are more detailed , doing just a fine grained sentiment analysis wouldn't be sufficient in determining whether a car is good or bad.
During the webscraping portion we faced many issues as many autoportals had employed measures to limit the bot movement through their content which caused us a lot of time. We tried numerous websites and methods, and were fortunate to find this website.
To understand more about the market value of the car we need to understand how the customer feels about each aspect of the car. To build such a model it is necessary to do aspect based sentiment analysis where we try to understand the sentiment behind each aspect of the car review. This model will be more useful if it would understand the customer's needs and recommend a car that is closer to his requirement.Additionally, we can correlate and eventually compare how a car is performing in the market based on the customer sentiment.

Bibliography

[1] A. Ankush Sharma, “A comparative study of sentiments analysis using rule based and support vector machine.”

[2] B. Liu, “Sentiment analysis and opinion mining.”

[3] W. A. K. Aisah Rini Susanti, Taufik Djatna, “Twitter’s sentiment analysis on gsm services using multinomial naïve bayes.”

[4] Y. S. Arif Abdurrahman Farisi and S. A. Faraby, “Sentiment analysis on hotel reviews using multinomial naïve bayes classifier.”

[5] Selenium, “Selenium documentation.”