

TECHNICAL REPORT  
on  
**REVIEW RANKING SYSTEM BASED ON UTILITY**

by

*Kunal Bharadwaj (D19016)*

*Veera Vignesh (D19036)*

*Vighnesh Tamse (D19037)*

*Durjay Das (D19039)*

Under the guidance of

*Prof. Gourab Nath*

*Assistant Professor, Data Science at Praxis Business School, Bangalore*



Department of Data Science

**Praxis Business School**

## ACKNOWLEDGEMENT

We would like to express our deep and sincere gratitude to our guide and mentor **Prof. Gourab Nath** for giving us the opportunity to do the project and providing invaluable guidance throughout the lifecycle of this project. His sincerity, hard work and motivation has inspired us in completing this project successfully. It was a great privilege and honour to be able to work under his guidance.

We would also like to extend our sincere gratitude to **Utkarsh Vardhan** sir for taking time out of his busy schedule and guiding us throughout this project with his valuable advice and knowledge.

Last but not the least, we extend our heartfelt thanks to all the teaching as well as the non-teaching staff at **Praxis Business School, Bangalore** for providing us with whatever was necessary for this project and to our friends, colleagues and families for always helping and supporting us whenever in need.

Kunal Bharadwaj (D19016)  
Veera Vignesh (D19036)  
Vighnesh Tamse (D19037)  
Durjay Das (D19039)

## TABLE OF CONTENTS

Introduction .....	4
Data Collection.....	5
Part 1: Data Scraping through UI Path Studio.....	5
Part 2: Data Scraping through Selenium WebDriver .....	8
Modelling .....	10
Objective .....	10
Tool used.....	10
Methodology.....	10
Experiments .....	10
1. Problem - To create features for our model from the extracted reviews using UI Path.....	10
2. Problem – Filtering stop words, punctuations and emoticons. Feature engineering using POS tagging, TF-IDF etc. Applying SVR model on the thus created data.....	11
3. Problem – Fitting Random Forest Regressor and XGBoost Regressor models on the data.	12
4. Problem – Remove Upvote and Downvote columns and observe the MAPE score. ....	12
5. Problem – Getting the review sentiment. ....	13
6. Problem – To select the best model from Random Forest or XG Boost or SVR to rank the reviews of products in different categories such as Mobile Phones, Laptops and Cameras. ....	13
Graphical User Interface (GUI).....	19
Objective .....	19
Tools used .....	19
Methodology.....	19
Experiments .....	19
1. Problem – Create 1 <sup>st</sup> version of GUI for Review Ranking System. ....	19
2. Problem – To add warning message.....	20
3. Problem – Change warning message to error message. GUI version 2.....	20
4. Problem – Web scraping the reviews data using Selenium Webdriver.....	21
5. Problem – To create a GUI to accept the URL of the product. GUI version 3.....	21
Deployment .....	23
Objective .....	23
Tools Used.....	23
Experiments .....	23
1. Problem – To create a deployed version to accept the URL of the reviews page of the product and rank them. GUI version 4. ....	23
Conclusion.....	26

# CHAPTER 1

## Introduction

---

The demand for online purchase is increasing day by day because people want to shop without much hassle. People can purchase a product anytime from anywhere without facing the crowd, also having the advantages to explore a variety of different products in the same place at a competitive price. Customers also share their valuable feedback about their purchased products in the form of reviews and ratings. These reviews are also evaluated by other customers by upvoting or downvoting them.

These reviews are helpful in many ways, for example, when someone wishes to buy a new product, they can estimate whether the product is apt for them by reading the reviews. This will make it easy for them to understand the pros and cons of the product and accordingly they can decide whether to buy the product or not. Also, these reviews are helpful for the manufacturers to understand the customer's sentiment. This may help the manufacturers to modify the product as per the customer's needs to sustain in the aggressive market

There may be thousands of different opinions of customers. Some of the reviews are intended to understand the pros and cons of a product while others are not much beneficial. It is cumbersome to go through all the reviews to understand the product. However, we can read a small number of the most relevant and meaningful reviews to have a quicker understanding of the product.

The application can be utilized by customers as well as manufacturers to assess only the most useful reviews to take relevant business decisions.

## Chapter 2

### Data Collection

---

#### Part 1: Data Scraping through UI Path Studio

##### Objective:

To scrape the reviews data for different products from ecommerce websites and store the data in a CSV file using UI Path.

##### Tools Used:

We used **UI Path Studio**, a tool that enables you to design automation processes in a visual manner, through diagrammatic workflows, for data scraping.

##### About UiPath Studio:

**UiPath Studio** - an advanced tool that enables you to design automation processes in a visual manner, through diagrams.

**UiPath Robot** - executes the processes built in Studio, as a human would. Robots can work unattended (run without human supervision in any environment, be it virtual or not) or as assistants (a human initiates the process).

**Workflow Designer:** It is essential for Robotic process automation (RPA) products to give their users an amazing experience. One way to do this is to make visual workflow automation straightforward and intuitive. UiPath is providing automated workflow design which can be used without programming knowledge.

**Activities:** An Activity is an action we can add in your workflow - like clicking the OK button or typing a text in an input box. Activity box is found on the left panel. An activity is added into the workflow by dragging it to the designer workspace

##### Steps Followed:

1. Create a New Process and then a new Flowchart.
2. In the activities tab, search for Open Browser and drag and drop it to the main window below Start.
3. Double click on the Open Browser and enter the Review URL of the product.

4. Click on the Data Scraping button and scrape the features such as Review Title, Review Text, Review Rating, Review Date, Upvote and Downvote of the product. You have to select the data for each of these features only twice.
5. Once data scraping is done, an activity box of Data Scraping will be created. Connect the Data Scraping box with the Open Browser one.
6. Drag and drop the Write CSV activity box below Data Scraping and provide the file path where you wish to save the csv file.
7. Now to run the file, click on the Debug File dropdown and select Run File.
8. Once the process is completed, a csv file of the reviews data will be generated in the location mentioned in the Write CSV activity box.

Following link gives a detailed step by step guide for data extraction using UI Path:

<https://github.com/veeravignesh1/Capstone-Reviews-Ranker/blob/master/Technical%20Report/UI%20Path%20Steps.docx>

### Data Domain:

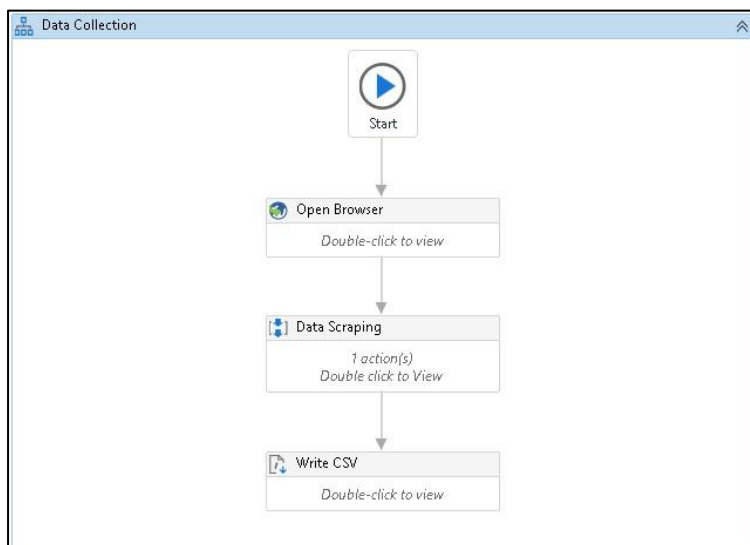
During the initial stages of the project, we extracted the data for some mobile phone models for which there were many reviews. Below are few examples of the mobile phone models:

1. Apple Iphone 7 - [apple-iphone-7-black-32-gb](#)
2. Xiaomi Redmi 8 - [redmi-8-onyx-black-64-gb](#)
3. Samsung Galaxy S10 lite - [samsung-galaxy-s10-lite-prism-blue-512-gb](#)

As mentioned earlier, we have written into the csv files the following features:

1. Review Title
2. Review Text
3. Review Rating
4. Review Date
5. Upvote
6. Downvote

## UI Path Studio workflow:



## Result:

Review_Title	Review_Text	Review_Rating	Review_Date	Upvote	Downvote
Classy product	Too good luv it iPhone is only camera lovers	5	6 months ago	1215	172
Excellent	I click this photo on wall painting	5	6 months ago	1576	234
Must buy!	From last 5 years my younger brother was using iphone 4s and i bought iphone 7 for his birthday gift. When i gave gift packet to him he was	5	5 months ago	2928	460
Pretty good	It was as expected ❤️	4	6 months ago	1560	249
Mind-blowing purchase	it's my first iPhone I love it	5	6 months ago	263	34
Highly recommended	superb product...i liked it very much	5	Dec, 2018	438	69
Best in the market!	black beauty... everything is perfect...	5	Apr, 2018	2027	397
Must buy!	Works perfectly fine even after 1.5 years. Great camera.	5	Aug, 2018	938	181
Excellent	Love Apple ❤️	5	Oct, 2018	442	79
Fabulous!	best	5	Oct, 2018	231	39
Perfect product!	Very happy with the product as well as delivery	5	6 months ago	289	53

## Advantages:

1. The process of scraping is automated which can be achieved without any programming knowledge.
2. The process is fast and efficient
3. URLs of the individual comment can be extracted.

## Limitations:

1. User needs to install the application to his/her system.
2. A working knowledge of UI Path is necessary.
3. The 'READ MORE' part cannot be extracted. It had to be extracted manually.
4. The application cannot extract no. of photos uploaded by the user, etc which is an important metric for our analysis

## Part 2: Data Scraping through Selenium WebDriver

### Objective:

To scrape the reviews data for different products from ecommerce websites using Selenium WebDriver.

### Tools Used:

We used **Python** for implementing data extraction using Selenium WebDriver.

### About Selenium WebDriver:

Selenium WebDriver is one of the most important parts of the Selenium test-suite family.

**Selenium:** Selenium refers to a suite of tools that are widely used in the testing community when it comes to cross-browser testing. Selenium cannot automate desktop applications; it can only be used in browsers. It is one of the most preferred testing tool-suite for automating web applications as it provides support for popular web browsers which makes it very powerful.

**Selenium WebDriver:** Selenium WebDriver is a web framework that permits you to execute cross-browser tests. This tool is used to extract data from websites

Selenium WebDriver allows you to choose a programming language of your choice for data extraction. Selenium WebDriver is not capable of handling window components.

### Data Domain:

We could now extract the reviews for any product just by passing the URL of the product in the ecommerce websites. We extracted data for the following products:

1. Mobile Phones
2. Laptops
3. Digital Cameras.

We created a function that would take in the URL of any product on Flipkart and generate a dataframe or a file consisting of following features:

1. Review Title
2. Review Text
3. Review Rating
4. Number of photos
5. Upvote
6. Downvote



### Advantages:

1. No prior knowledge of any tool is required.
2. User just needs to provide the URL of the product.
3. User manual task is greatly reduced.

## CHAPTER 3

### Modelling

---

#### Objective:

To select the best features and the best model that models the data in such a way that the user gets the ranked reviews for both the positive and the negative ones.

#### Tool used:

Used python's standard Spacy module for advanced NLP tasks. Used Vader Sentiment package for review sentiment. Data modelling was done using sklearn package.

#### Methodology:

NLP techniques were performed using the Spacy module. The best features were selected by fitting various models and calculating the MAPE score. The model that gave the least MAPE score was selected.

#### Experiments:

Date: 06-03-2020

1. Problem - To create features for our model from the extracted reviews using UI Path.

**Github** - <https://github.com/veeravignesh1/Capstone-Reviews-Ranker/blob/master/Experiments/Iphone7.ipynb>

**Experiment No. 1** – Used Spacy library to create features such as Length of review, Number of sentences, Number of exclamation marks, Number of Question marks etc. We have also used separate package to identify number of emoticons used.

**Findings** – For mobile phones, most of the reviews were of length between 1-70 words. The average length of the reviews was about 52 words per review. The average number of sentences was 4. We found that there were very a small number of reviews that used exclamation or question marks. Also, not many reviewers have used Proper Case in their reviews. The maximum number of words per review that we got after web scraping was 500. This was because

there were reviews that had READ MORE section in them which UI Path was not able to handle.

**Experiment No. 2** – Manually copying the reviews that had READ MORE section in them and performing the steps again as done in Experiment No.1.

**Findings** – After manually copying the reviews that had READ MORE section in them, the average length of the reviews came to be 58 words per review. The maximum number of words per review jumped to 633. All the features were extracted in a csv file named 'withfeatures.csv'.

Date: 13-03-2020

2. Problem – Filtering stop words, punctuations and emoticons. Feature engineering using POS tagging, TF-IDF etc. Applying SVR model on the thus created data.

**Github**-<https://github.com/veeravignesh1/Capstone-Reviews-Ranker/blob/master/Experiments/Modelling-Iphone.ipynb>

**Experiment No. 1** – Removing stop words, punctuations, emoticons and spell check. Also, we removed those reviews that had less than 10 words in it.

**Findings** – Using the 'withfeatures.csv' file that we created in Experiment No. 2 in Problem 1, we removed the punctuations (question marks, exclamation marks, full stop etc), stop words and emoticons. For spell correction we experimented using autocorrect package of python and textblob package. We want to display the original comments of the author, so we didn't correct it

**Experiment No. 2** – Removed the stop words and applied POS tagging. Used TF-IDF Vectorizer to create a TF-IDF matrix. Created the target variable named 'h' which is used to rank the reviews and is calculated as below:

$$h = \text{Upvote} / (\text{Upvote} + \text{Downvote})$$

**Findings** – The TF-IDF matrix thus created consisted of all the unique words found in all the reviews as columns plus the columns we created during feature engineering stage. For iphone7, for a total of 264 reviews, we had a total of 1610 unique words. This dataframe was then exported to a csv file named 'Unigram\_Feature.csv'.

**Experiment No. 3** – We split the data in 'Unigram\_Features.csv' thus created in Experiment No. 2 into Train-Validation-Test. We then used the SVR (Support Vector Regression) model to fit our data.

**Findings** – After fitting the SVR model on training data we predicted the test data and calculated the  $R^2$  score which came out to be just 13%.

Date: 15-03-2020

3. Problem – Fitting Random Forest Regressor and XGBoost Regressor models on the data.

**Github -**

- 1) <https://github.com/veeravignesh1/Capstone-Reviews-Ranker/blob/master/Experiments/Modelling Iphone RandomForest.ipynb>
- 2) <https://github.com/veeravignesh1/Capstone-Reviews-Ranker/blob/master/Experiments/Modelling Iphone XGBoost.ipynb>

**Experiment No. 1** – We fitted the Random Forest Regressor model on the Unigram\_Features data and calculated the MAPE score. We also found the most important features by using the feature\_importances\_ function provided by the Random Forest Regressor model.

**Findings** – We found that Upvote and Downvote are the most important features found by our model to predict the outcome. Also, we got a MAPE score of 17.28% means the accuracy of our model was around 82.7%.

**Experiment No. 2** – We fitted the XGBoost Regressor model on the Unigram\_Features data and calculated the MAPE score.

**Findings** – We got a MAPE score of 16.5% means the accuracy of our model was around 83.5%.

Date: 16-03-2020

4. Problem – Remove Upvote and Downvote columns and observe the MAPE score.

**Github –**

<https://github.com/veeravignesh1/Capstone-Reviews-Ranker/blob/master/Experiments/Modelling Exclude Upvote Downvote.ipynb>

**Experiment No. 1** – We removed the Upvote and Downvote columns which our model ranked as the most important features for predicting the outcome and then observed the MAPE score using both Random Forest Regressor and XGBoost Regressor.

**Findings** – When we drop Upvote and Downvote columns, fit our model on train data and test on test data, we found that MAPE score increased to 29% for Random Forest Regressor and 30% for XGBoost Regressor.

Date: 22-03-2020

5. Problem – Getting the review sentiment.

**Github –**

[https://github.com/veeravignesh1/Capstone-Reviews-Ranker/blob/master/Experiments/Review\\_Title\\_Sentiments.ipynb](https://github.com/veeravignesh1/Capstone-Reviews-Ranker/blob/master/Experiments/Review_Title_Sentiments.ipynb)

**Experiment No. 1** – We used the Vader Sentiment Analysis for getting the review sentiments such as positive, negative or neutral.

**Findings** – We found that a few of the users who have rated the product very low (less than 2 ratings) have positive sentiments. On checking the reviews of these users manually we found that the reviews were actually positive but the ratings given were less. This can be misleading.

Date 18-04-2020

6. Problem – To select the best model from Random Forest or XG Boost or SVR to rank the reviews of products in different categories such as Mobile Phones, Laptops and Cameras.

Acronyms for the features experimented with

Column_Name	Acronym
Len_after	LA
Len_before	LB
Num_Sentence	NS
No_QMark	NQ
No_ExMarch	NEx
No_Upper	NU
No_proper	NPr
Perc_Noun	PN
Perc_Verb	PV
Perc_Adverb	PAdv
Perc_Adj	PAdj
Target	h

Target Variable:

$$h = \frac{upvotes}{upvotes+downvotes}$$

Acronyms for the available features

Column_Name	Acronym
Review_Title	Title
Review_Text	RT
Review_Rating	RR
Num_Photos	NPh
Upvote	Up
Downvote	Down

**Github** – <https://github.com/veeravignesh1/Capstone-Reviews-Ranker/tree/master/Experiments>

**Experiment No. 1** – Scraped the reviews for a mobile phone, passed it through the `get_features()` function containing TF-IDF vectorizer, created different combinations of feature sets, fitted the Random Forest Regressor Model on all the three products and calculated the MAPE score.

**Experiment No. 2** – Scraped the reviews for a mobile phone, passed it through the `get_features()` function containing TF-IDF vectorizer, created different combinations of feature sets, fitted the XG Boost Regressor Model on all the three products and calculated the MAPE score.

**Experiment No. 3** – Scraped the reviews for a mobile phone, passed it through the `get_features()` function containing TF-IDF vectorizer, created different combinations of feature sets, fitted the Support Vector Machine Regressor Model on all the three products and calculated the MAPE score.

### Model and feature experimentation with Mobile Phone reviews

Sr. No.	Features*	Random Forest		XG Boost		SVR	
		Train_MAPE	Test_MAPE	Train_MAPE	Test_MAPE	Train_MAPE	Test_MAPE
1	[RT,RR,LA,h]	4.76	14.02	0.59	16.18	11.17	12.61
2	[RT,RR,NS,h]	4.8	14.43	0.81	16.27	10.52	12.38
3	[RT,RR,LB,h]	4.61	15.09	0.42	15.09	11.23	12.45
4	[RT,RR,NPh,h]	4.89	14.08	0.64	16.24	9.79	12.79
5	[RT,RR,NPh,LB,LA,h]	4.49	13.19	0.29	15.85	10.88	12.4
6	[RT,PN,PV,h]	5.01	14.9	0.91	17.89	11.81	14.01
7	[RT,PN,PV,PAdv,h]	5.03	14.8	0.9	17.34	11.75	14.17
8	[RT,PN,PV,PAdv,PAdj,h]	4.97	14.86	0.79	16.86	11.81	14.92
9	[RT,NQ,h]	5.22	15.4	1.08	17.67	9.23	14.12
10	[RT,NQ,NEx,h]	5.13	14.98	0.98	17.04	9.49	14.38
11	[RT,NQ,NEx,NU,h]	5.13	14.81	0.98	17.04	9.52	14.38
12	[RT,NQ,NEx,NU,NPr,h]	4.96	14.78	0.56	16.47	9.21	14.03
13	[RT,NPh,NS,h]	4.68	13.62	0.92	15.49	10.23	13.4
14	[RT,NPh,NS,PN,h]	4.65	13.37	0.78	15.61	11.3	13.71
15	[RT,NPh,NS,PV,h]	4.88	13.65	0.9	14.72	10.88	12.16
16	[RT,NPh,NS,PN,PV,h]	4.81	13.52	0.77	15.42	11.11	13.25
17	[RT,NPh,NS,PAdj,h]	4.5	13.27	0.53	15.94	11.94	13.97
18	[RT,NPh,NS,PAdj,PN,h]	4.54	12.95	0.53	15.52	11.64	14.95
19	[RT,NPh,NS,PAdj,PN,PV,h]	4.67	13.21	0.53	14.39	11.58	15.01
20	[RT,RR,NPh,NS,PAdj,PN,h]	4.58	12.85	0.47	15.38	11.64	15.01

\* Review\_Text (RT) column is needed in every combination. 'h' is target variable

### Model and feature experimentation with Laptop reviews

Sr. No.	Features	Random Forest		XG Boost		SVR	
		Train_MAPE	Test_MAPE	Train_MAPE	Test_MAPE	Train_MAPE	Test_MAPE
1	[RT,RR,LA,h]	5.37	14.69	0.08	17.22	15.06	16.59
2	[RT,RR,NS,h]	5.42	14.6	0.26	17.02	13.05	15.58
3	[RT,RR,LB,h]	5.32	14.9	0.07	17.18	15.01	16.94
4	[RT,RR,NPh,h]	4.95	14.88	0.07	15.56	10.59	15.47
5	[RT,RR,NPh,LB,LA,h]	4.93	15.04	0.07	16.48	15.16	16.8
6	[RT,PN,PV,h]	5.53	15.8	0.25	18.89	14.42	16.74
7	[RT,PN,PV,PAdv,h]	5.58	15.66	0.25	18.15	14.18	16.68
8	[RT,PN,PV,PAdv,PAdj,h]	5.66	15.6	0.24	17.96	14.05	16.03
9	[RT,NQ,h]	5.65	14.55	0.49	15.29	9.97	15.44
10	[RT,NQ,NEx,h]	5.69	14.92	0.49	15.19	9.75	15.54
11	[RT,NQ,NEx,NU,h]	5.71	14.64	0.49	15.19	9.75	15.54
12	[RT,NQ,NEx,NU,NPr,h]	5.71	14.79	0.49	15.19	9.77	15.48
13	[RT,NPh,NS,h]	5.16	14.72	0.08	15.4	13.62	16.19
14	[RT,NPh,NS,PN,h]	5.21	16.01	0.08	17.54	14.15	16.88
15	[RT,NPh,NS,PV,h]	5.16	14.81	0.08	15.16	13.84	16.07
16	[RT,NPh,NS,PN,PV,h]	5.26	15.78	0.08	17.87	14.05	16.93
17	[RT,NPh,NS,PAdj,h]	5.21	14.22	0.08	15.44	14.17	15.72
18	[RT,NPh,NS,PAdj,PN,h]	5.31	15.9	0.07	16.74	13.97	16.51
19	[RT,NPh,NS,PAdj,PN,PV,h]	5.38	15.48	0.07	17.4	13.93	16.35
20	[RT,RR,NPh,NS,PAdj,PN,h]	5.06	15.78	0.07	16.21	13.5	15.86

\* Review\_Text (RT) column is needed in every combination. 'h' is target variable



### Model and feature experimentation with Camera reviews

Sr. No.	Features	Random Forest		XG Boost		SVR	
		Train_MAPE	Test_MAPE	Train_MAPE	Test_MAPE	Train_MAPE	Test_MAPE
1	[RT,RR,LA,h]	5.03	12.43	1.64	13.44	10.2	11.88
2	[RT,RR,NS,h]	5.43	12.68	2.12	13.23	9.55	11.69
3	[RT,RR,LB,h]	4.95	12.6	1.48	13.3	10.24	12.42
4	[RT,RR,NPh,h]	4.75	12.51	0.89	12.31	9.42	12.59
5	[RT,RR,NPh,LB,LA,h]	4.48	12.3	0.68	12.27	10.23	12.27
6	[RT,PN,PV,h]	5.33	12.64	2.06	12.6	10.67	12.23
7	[RT,PN,PV,PAdv,h]	5.31	12.7	2.06	12.37	10.53	12.08
8	[RT,PN,PV,PAdv,PAdj,h]	4.93	13.21	1.58	12.96	10.63	11.84
9	[RT,NQ,h]	5.65	12.5	2.55	12.77	8.6	11.65
10	[RT,NQ,NEx,h]	5.66	12.33	2.55	12.77	8.6	11.75
11	[RT,NQ,NEx,NU,h]	5.62	12.38	2.55	12.77	8.6	11.77
12	[RT,NQ,NEx,NU,NPr,h]	5.54	12.51	2.36	12.85	8.56	11.83
13	[RT,NPh,NS,h]	4.71	12.6	0.94	12.8	8.73	12.39
14	[RT,NPh,NS,PN,h]	4.58	12.64	0.74	13.17	10.87	11.73
15	[RT,NPh,NS,PV,h]	4.57	12.7	0.98	12.44	9.72	12.81
16	[RT,NPh,NS,PN,PV,h]	4.48	12.86	0.74	12.16	10.57	12.31
17	[RT,NPh,NS,PAdj,h]	4.59	13.05	0.79	13.43	10.78	12.07
18	[RT,NPh,NS,PAdj,PN,h]	4.52	12.94	0.72	12.84	10.9	11.78
19	[RT,NPh,NS,PAdj,PN,PV,h]	4.4	13.1	0.71	12.81	10.55	12.21
20	[RT,RR,NPh,NS,PAdj,PN,h]	4.57	12.88	0.72	12.99	10.79	11.73

\* Review\_Text (RT) column is needed in every combination. 'h' is target variable

#### Model Selection:

Analysing the minimum test MAPE for all the 3 products across all the model and deriving their statistic

	Random Forest	XG Boost	SVR
Mobile Phones	14.089	16.1205	13.705
Laptop	15.1385	16.554	16.167
Camera	12.678	12.814	12.0515

	Random Forest	XG Boost	SVR
Average	13.9685	15.16283333	13.9745
Min	12.678	12.814	12.0515
Max	15.1385	16.554	16.167
Span	2.4605	3.74	4.1155

**Findings** – Out of all the three experiments, we found that Random Forest model gave least MAPE scores and best statistic among all the model, so we went ahead with Random Forest as our final model.

#### Feature Selection:

After selection of the Random Forest model, we analysed the model across all the product and calculated the statistics for best features to be selected. The below table represents experimented value of test MAPE obtained for the given features

Sr. No.	Features	Random Forest				
		Mobile Phones	Laptop	Camera	Average Across Product Groups	Span
1	[RT,RR,LA,h]	14.02	14.69	12.43	13.71	2.26
2	[RT,RR,NS,h]	14.43	14.6	12.68	13.90	1.92
3	[RT,RR,LB,h]	15.09	14.9	12.6	14.20	2.49
4	[RT,RR,NPh,h]	14.08	14.88	12.51	13.82	2.37
5	[RT,RR,NPh,LB,LA,h]	13.19	15.04	12.3	13.51	2.74
6	[RT,PN,PV,h]	14.9	15.8	12.64	14.45	3.16
7	[RT,PN,PV,PAdv,h]	14.8	15.66	12.7	14.39	2.96
8	[RT,PN,PV,PAdv,PAdj,h]	14.86	15.6	13.21	14.56	2.39
9	[RT,NQ,h]	15.4	14.55	12.5	14.15	2.9
10	[RT,NQ,NEx,h]	14.98	14.92	12.33	14.08	2.65
11	[RT,NQ,NEx,NU,h]	14.81	14.64	12.38	13.94	2.43
12	[RT,NQ,NEx,NU,NPr,h]	14.78	14.79	12.51	14.03	2.28
13	[RT,NPh,NS,h]	13.62	14.72	12.6	13.65	2.12
14	[RT,NPh,NS,PN,h]	13.37	16.01	12.64	14.01	3.37
15	[RT,NPh,NS,PV,h]	13.65	14.81	12.7	13.72	2.11
16	[RT,NPh,NS,PN,PV,h]	13.52	15.78	12.86	14.05	2.92
17	[RT,NPh,NS,PAdj,h]	13.27	14.22	13.05	13.51	1.17
18	[RT,NPh,NS,PAdj,PN,h]	12.95	15.9	12.94	13.93	2.96
19	[RT,NPh,NS,PAdj,PN,PV,h]	13.21	15.48	13.1	13.93	2.38
20	[RT,RR,NPh,NS,PAdj,PN,h]	12.85	15.78	12.88	13.84	2.93

**Findings** – Out of all the three experiments, we found that Random Forest model gave least test MAPE scores and best statistic among all the model, so we went ahead with Random Forest as our final model.

The selected Features are:

**Review\_Title, No. of Photos, No. of Sentence, Percentage of Adj., Target**

## CHAPTER 4

### Graphical User Interface (GUI)

---

#### Objective:

To create a Graphical User Interface (GUI) for our product 'Review Ranker'.

#### Tools used:

**Tkinter:** We used Tkinter, a standard python package for developing GUI, for developing the 1<sup>st</sup>, 2<sup>nd</sup> and the 3<sup>rd</sup> version of our GUI. User needs to install this particular application in his/her local system in order to get the ranked reviews.

#### Methodology:

We integrated the functions that were created during the model development part with the skeleton of the GUI such that on click of the Submit button the main function should be called and provide the user with the ranked reviews.

#### Experiments:

Date: 27-03-2020

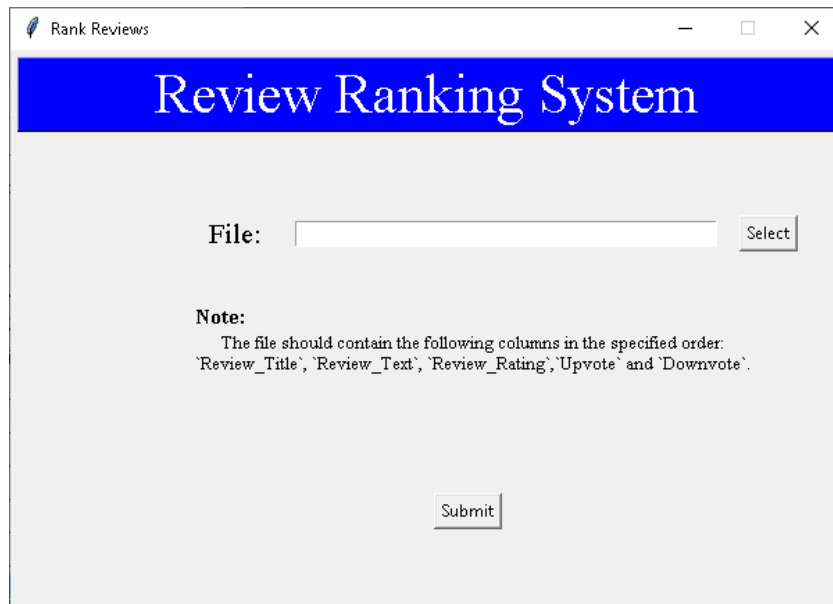
1. Problem – Create 1<sup>st</sup> version of GUI for Review Ranking System.

##### Github –




- 1) [https://github.com/veeravignesh1/Capstone/blob/master/Experiments/Review\\_GUI\\_v1.ipynb](https://github.com/veeravignesh1/Capstone/blob/master/Experiments/Review_GUI_v1.ipynb)
- 2) [https://github.com/veeravignesh1/Capstone/blob/master/Experiments/Review\\_GUI\\_v1.py](https://github.com/veeravignesh1/Capstone/blob/master/Experiments/Review_GUI_v1.py)

##### Experiment No. 1 –

- Used 'Tkinter' library to create GUI for our project. It is designed such that it takes the extracted reviews file from the user as input with columns in order of: Review\_Title, Review\_Text, Review\_Rating, Upvote and Downvote. Converted the ipython notebook (.ipynb) to python script (.py)



- This system outputs 3 csv files namely: allreviews, negative and positive each containing reviews which are ranked.

 allreviews.csv	4/6/2020 1:43 PM	Microsoft Excel C...	25 KB
 negative.csv	4/6/2020 1:43 PM	Microsoft Excel C...	2 KB
 positive.csv	4/6/2020 1:43 PM	Microsoft Excel C...	22 KB

Date: 04-04-2020

## 2. Problem – To add warning message.

**Experiment No. 1** – Added a warning message that pops up when a user tries to click on Submit button without selecting any file.

**Findings** – The system threwed the warning message even if the file was selected by the user.

**Experiment No. 2** – Tweaked the program to rectify the defect encountered in Experiment No. 1 to give warning message only when file is not selected.

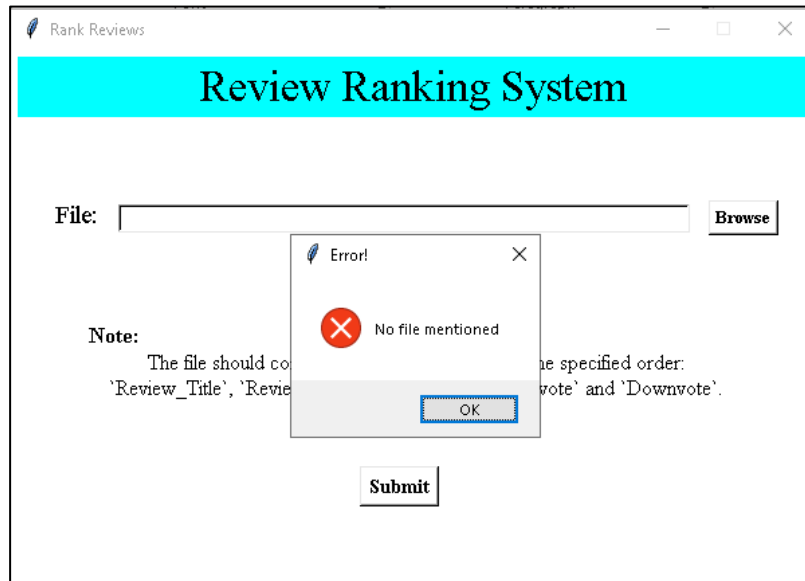
Date: 05-04-2020

## 3. Problem – Change warning message to error message. GUI version 2.

**Github** - [https://github.com/veeravignesh1/Capstone/blob/master/Experiments/Review\\_GUI\\_v2.ipynb](https://github.com/veeravignesh1/Capstone/blob/master/Experiments/Review_GUI_v2.ipynb)

**Experiment No. 1** – Changed the warning message to an error message when user does not select any file. Changed the look and feel of the UI.

**Findings** – The system showed the error message accurately when user intends to click the submit button without selecting any file.



Date 15-04-2020

4. Problem – Web scraping the reviews data using Selenium Webdriver.

**Github** - <https://github.com/veeravignesh1/Capstone-Reviews-Ranker/blob/master/Experiments/Selenium%20with%20one%20browser%20instance.ipynb>

**Experiment No. 1** – In order to rank the reviews using just the URL of the reviews page of the product for the new version of the GUI, we scraped the reviews data by using Selenium Webdriver.

**Findings** – We created a function that would get the reviews just by passing the URL of the reviews page of the product.

Date 16-04-2020

5. Problem – To create a GUI to accept the URL of the product. GUI version 3.

**Github** – <https://github.com/veeravignesh1/Capstone-Reviews-Ranker/blob/master/Product%20GUI/Tkinter/RR%20-Tkinter.py>

**Experiment No. 1** – Created a GUI that will accept the URL of the reviews page of the product and a folder location where you want to place the files in. On click of Submit button, the ranked reviews (all, positive and negative) will be stored in the specified folder.

## Review Ranker

URL

Location

Browse

Clear

Submit

Open  
Folder

### Advantages:

1. User just needs to provide the URL of the product from Flipkart and the location of the folder where he/she wants to save the files.
2. User manual task is greatly reduced as the application itself takes care of review extraction and processing.

### Limitations:

1. Time to obtain the results is high.
2. User has to install the application in order to get the ranked positive and negative reviews.

## CHAPTER 5

### Deployment

---

The application, Review Ranker, that we created is useful for both the manufacturers as well the customers. It will help the manufacturers to better assess their product and come up with innovative ideas which may meet the customer needs based on the reviews.

#### Objective:

To create a web application of our product to benefit the users who want to use it on the fly.

#### Tools Used:

**Bootstrap:** It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. It is an open source toolkit for developing with HTML, CSS, and JS.

**Flask:** Flask is a web framework. This means it provides us with tools, libraries and technologies that allow us to build a web application. It has little or no dependencies on external libraries.

**Heroku:** Heroku is a container-based cloud Platform as a Service (**PaaS**). It is used to deploy, manage, and scale modern apps.

#### Experiments:

Date 22-04-2020

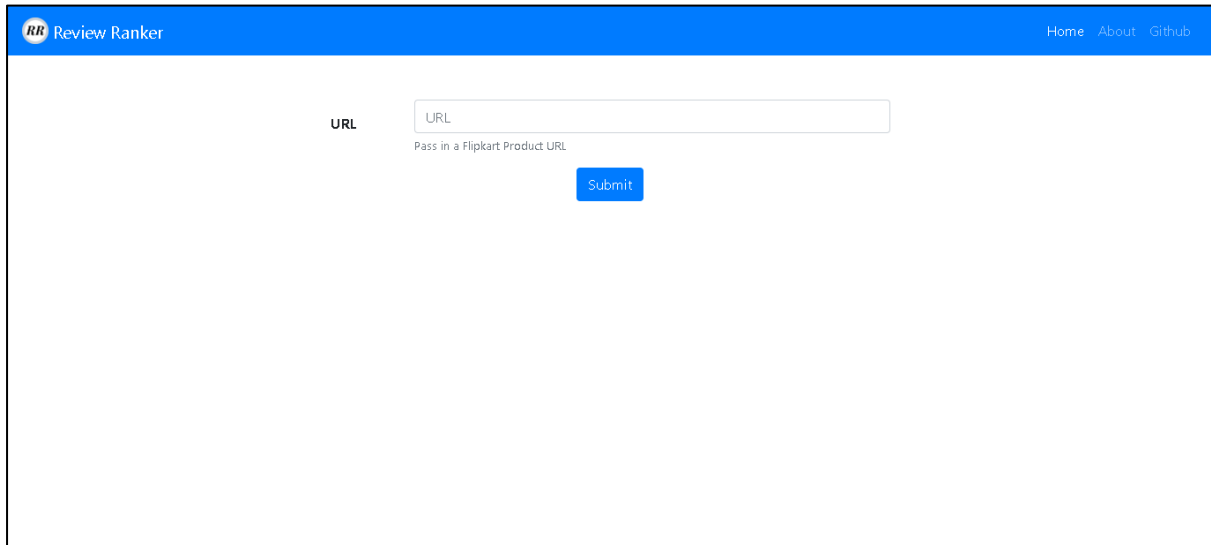
1. Problem – To create a deployed version to accept the URL of the reviews page of the product and rank them. GUI version 4.

**Github** – <https://github.com/veeravignesh1/Reviews-Ranker>

**Experiment No. 1** – Created a GUI and deployed on Heroku app that will accept the URL of the reviews page of the product on click of Submit button, the ranked reviews (positive and negative) will be displayed.

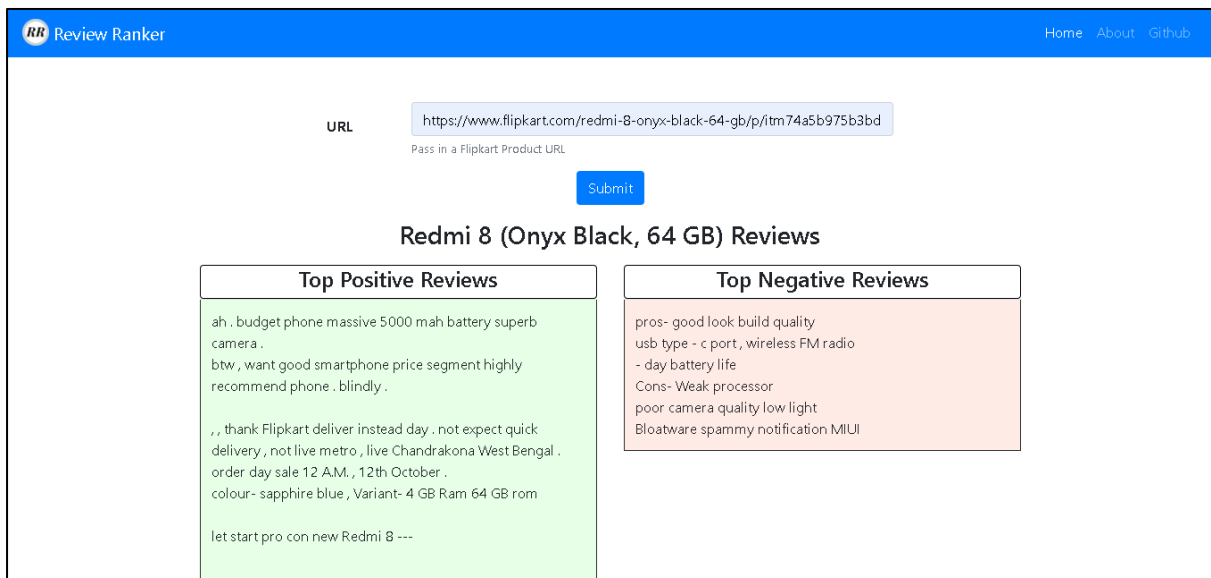
**Application:** <https://reviews-ranker.herokuapp.com/>

Open the application URL in a browser:



Paste the URL of the product on flipkart in the URL section for which you wish to get the reviews ranked and click on Submit.

e.g.: [Redmi 8](#)



Top Positive Reviews	Top Negative Reviews
ah . budget phone massive 5000 mah battery superb camera . btw , want good smartphone price segment highly recommend phone . blindly .  ,, thank Flipkart deliver instead day . not expect quick delivery , not live metro , live Chandrakona West Bengal . order day sale 12 A.M. , 12th October . colour- sapphire blue, Variant- 4 GB Ram 64 GB rom  let start pro con new Redmi 8 ---	pros- good look build quality usb type - c port , wireless FM radio - day battery life Cons- Weak processor poor camera quality low light Bloatware spammy notification MIUI

If the user tries to click on submit button without providing the URL, a warning message will be shown:



The screenshot shows the 'Review Ranker' web application interface. At the top, there is a blue header with the 'RR' logo and the text 'Review Ranker' on the left, and links for 'Home', 'About', and 'Github' on the right. The main content area has a light gray background. On the left, the label 'URL' is displayed. To its right is a text input field containing the placeholder text 'URL'. Below the input field, there is a small gray box with the text 'Pass in a Flipkart Product URL'. A blue 'Submit' button is positioned below the input field. A yellow warning message box with an exclamation mark icon is overlaid on the input field, containing the text 'Please fill out this field.'

If the user tries to provide a text other than URL, again a warning message will be shown:

This screenshot shows the same 'Review Ranker' web application interface. The text input field now contains the text 'Redmi note 8'. The gray box below the input field still says 'Pass in a Flipkart Product URL'. The blue 'Submit' button remains. A new yellow warning message box with an exclamation mark icon is overlaid on the input field, containing the text 'Please enter a URL.'

### Advantages:

1. No need to install any external application.
2. On the go results with just a click of a button.
3. Results are provided at a faster rate.

### Future Scope:

In the web application that we have created, user has to manually copy and paste the URL of the product and then get the ranked reviews. In order to avoid this, we can provide our product as a browser extension such that when user is on the product page of flipkart, just by clicking on the extension the user can get the ranked positive and negative reviews.

Also, we can extend this product to rank reviews of products from any ecommerce site instead of just flipkart.

## CHAPTER 6

### Conclusion

---

This technical report lists down different chapters showcasing the various steps, experiments and the challenges we faced during the journey of developing our product 'Review Ranker'.

The first chapter presented as an introduction defines our problem statement, the current scenario of online shopping and how our product will help both the customers as well as the manufacturers. The second chapter lists down the two approaches we followed for data collection and the steps for each of these methods. Data Collection forms an integral part of any data science project. The third chapter talks about the various modelling techniques we applied and the experiments we performed to achieve our objective. Through these experiments we finalized our best performing model and the best feature sets to get accurate results. The fourth chapter walks us through our journey of developing a Graphical User Interface (GUI) for our product and how we kept on improving from the previous versions. And finally, in the fifth chapter, we present our final product where we have hosted our product on the web for the ease of use for our users where users can get the results on the fly without the need to install any application.

And, finally, we conclude by saying that after facing numerous challenges in the journey of creating 'Review Ranker', we have been successful in achieving our objective. Many improvements can be done on the same as our understanding of the problem and the solution to it is limited. Tasks like this have no easy solutions. Having said that, we are happy that we have made a significant progress from what we were in the initial stage and we have come a long way in developing our product. We are looking forward to increase the scope of this project in near future.