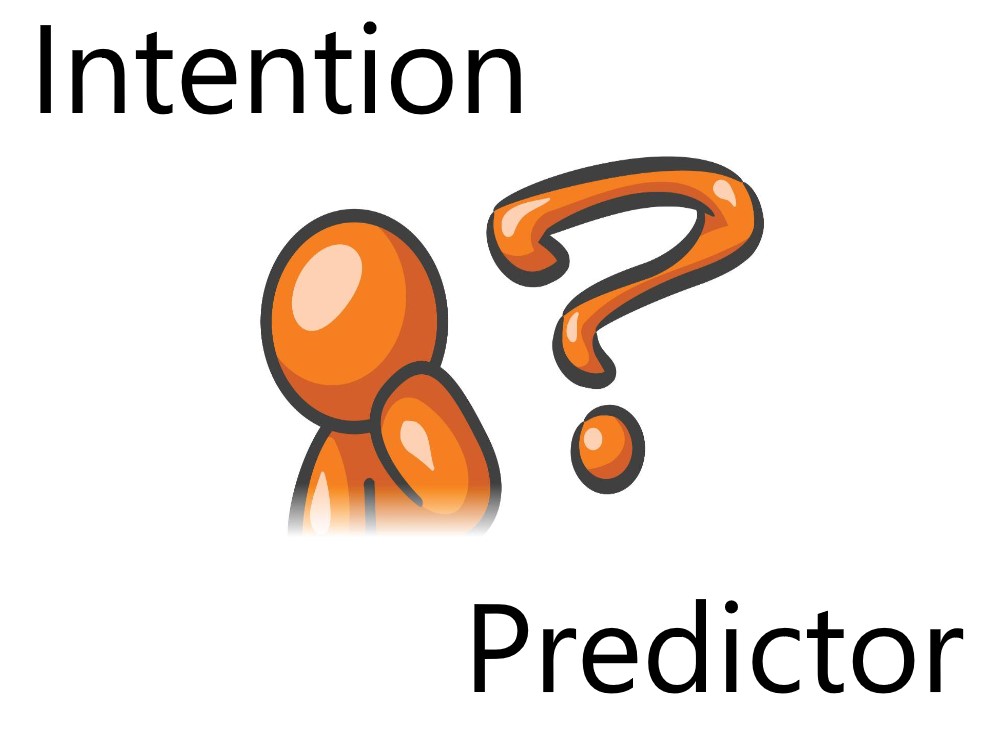
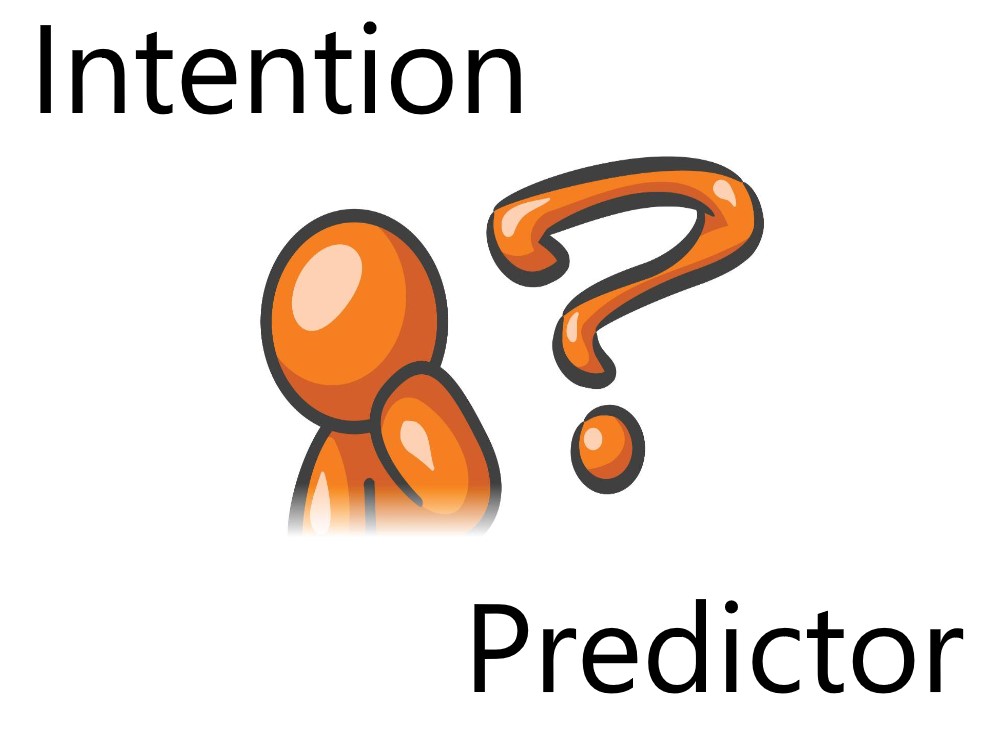
**Consumer Intention Prediction using Twitter**

**(CIP)**

****

**FINAL YEAR PROJECT**

**Consumer Intention Prediction using Twitter (CIP)**

****

**Supervisor:**

**Miss Nida Saddaf Khan (nskhan@iba.edu.pk)**

**Group Members:**

**Hasan Imran (himran@khi.iba.edu.pk)**

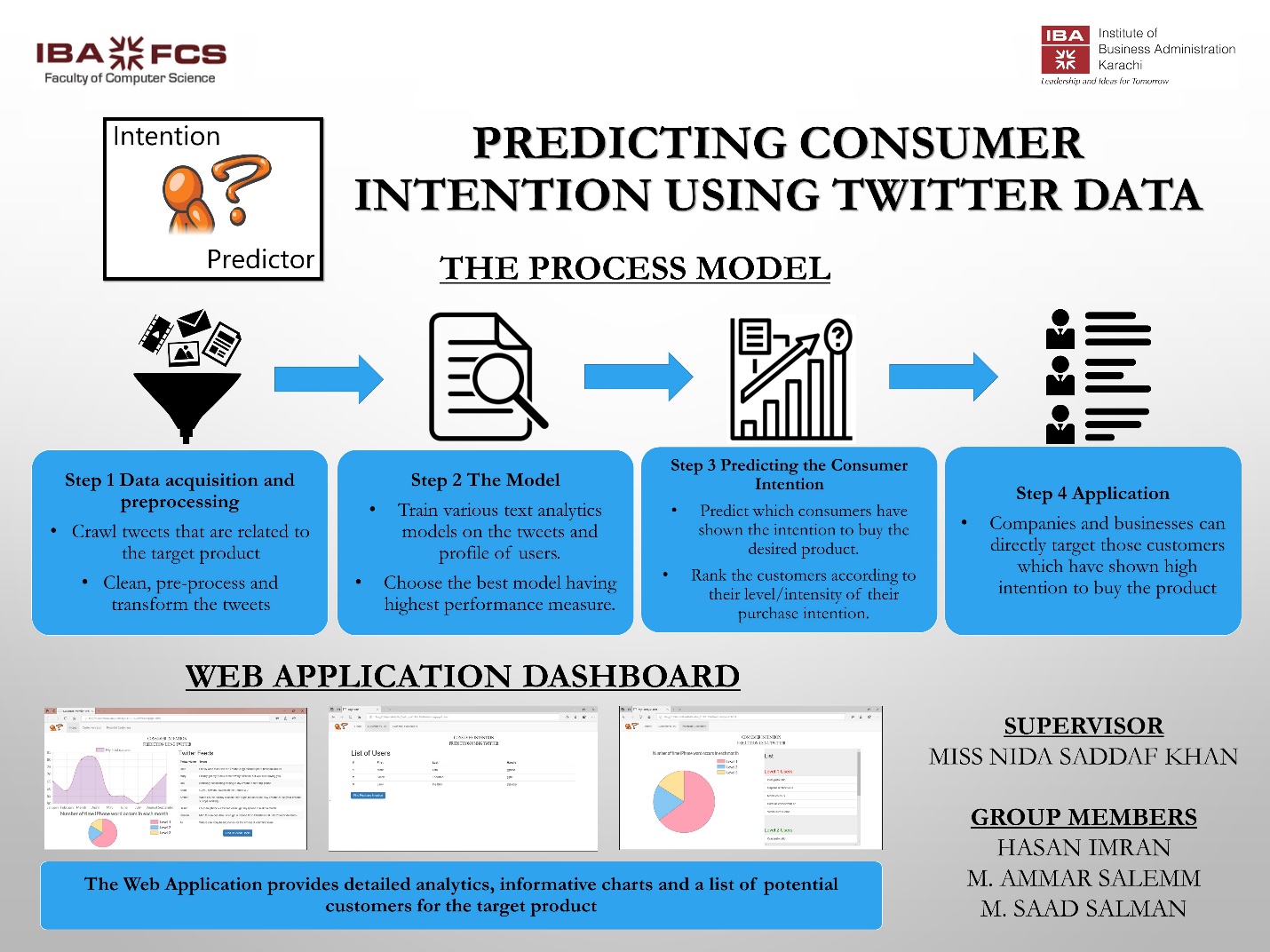
**Muhammad Ammar Saleem (msaleem@khi.iba.edu.pk)**

**Muhammad Saad Salman (muhammadsalman@khi.iba.edu.pk)**

**Executive Summary:**

Our project is a web application that predicts the likelihood/certainty that a customer will buy a product that he is interested in based on his social media posts such as Twitter tweets and user profile data. This will help the company/business target a particular customer more efficiently and boost their sales.

First, we search for Twitter tweets of potential customers wanting to buy a product. And based on those tweets we estimate/predict the likelihood that the customer will buy the product. We then make a model by gathering tweets from users who have already expressed intention to buy the product using their tweet history and if possible, their web search history as well and then training the text analytical model based on those tweets. Using the model, we input potential customers who have tweeted about the product but have not bought it. And based on the training data the model estimates a prediction/likelihood of whether the customer will buy it or not. We have limited the scope of our data to only mobile phones. Our model predicts the consumer intention for the latest upcoming mobile phones. We have tested it on the latest iPhone X variants and our model has provided promising results with accuracies of up to 80% considering that we have limited annotated data.





**Project Description:**

1. Background and Motivation

Currently we have many recommendation systems available which recommend different products to the user, most of which are not efficient. No such effective model for businesses to identify potential customers. Further, there have been several research studies for analyzing the insights of online consumers buying behavior. However, only a few have addressed the customers buying intention for products.

We want to develop an application that will help the businesses identify potential customers for their products by estimating their purchase intention in measurable terms from their tweets and user profile data on twitter. In a way we can say that Purchase Intention detection task is close to the task of identifying wishes in product reviews.

1. Project Goal

We aim to analyze the tweets related to a product and identify the purchase intention in it. In this way we can rank the tweets which have high purchase intention and report the name of the person who tweeted as potential customer of product.

We will make a model by gathering tweets from users who have already expressed intention to buy the product and see their tweet history and if possible, their web search history as well. Using this model, we will input potential customers who have tweeted about the product but have not bought it yet! And based on the training data the model will estimate a prediction/likelihood of whether the customer will buy it or not.

1. Project Requirements

We require data from twitter to analyze purchase intention. For now, we are gathering the data by scraping the tweets of a product through a scraper. After scraping the data, we store the data in mongo DB for further processing and performing analysis.

Secondly, we will need to annotate data retrieved from the scrapper. Since we were not able to find any publicly available dataset related to purchase intention, we had to build our own which was a challenge.

Once model is trained, we will need to develop a website so that users can easily access our application through the graphical interface of the website. We will show on the website the purchase intention rank of tweets on level of 1 to 5 for the desired product of the user.

* 1. Functional Requirements
     1. Web Application

We have developed a web application so that users can easily access our application. A web application also provides portability, speed and is easy to use. It also does not require much hardware to be setup on the user’s end.

* + 1. Dashboard

We have developed a dashboard which shows the relevant statistics according to the dataset of the product which the user can use to evaluate the results and check the trend for the product.

* + 1. Upload annotated dataset

The user will press the upload annotated dataset button through which he will be asked to upload the dataset he wants to test for his product and find the relevant customers from that dataset. The output will be in a form of a pie chart and a table showing the different level of customers and the scores assigned to each customer.

* + 1. Analysis

When the user presses the analysis button he will be taken to the screen which will show the detailed analysis of our dataset that we have built containing word clouds, positive vs negative tweets and the most used words for that product.

* 1. Constraints

Since we are scrapping the data not getting data through twitter Search API, this is sometimes not reliable and robust, so we will need to either get the data from some company which sells such data or will need access to the twitter search API.

Another constraint is that we have a lot of tweets which we will use to train our model and we need annotated data so that we can evaluate our model but since annotating each and every tweet is time-consuming as well as there is no way to verify the annotated tweets, we have used methods of majority voting and averaging.

* 1. Objectives

We aim to analyze the tweets related to a product and identify the purchase intention in it and assign a score to each tweet to show the measurable intention for each tweet.

We have ranked the tweets according to the level of purchase intention and show a list of potential customers to the user which he can use to directly target the customer.

1. Validation and Acceptance Tests

We have tested the model accuracy by confusion matrix (Accuracy, Precision, Recall, F-Measure). Further we have also considered the True Negative Rate, The True Positive Rate and the shape of the ROC curve for more insights. This will give us a percentage of accuracy achieved by our model.

For our application development, we have opted to use unit testing in which individual units of source code, would be tested to determine if they are fit to use. Then, we are going to run integration testing where the individual source codes would be merged and tested as group.

We will also test the usability of our website by carrying out the tasks and functions of the website in different scenarios and checking if they successfully complete.

**Technical Design:**

1. Possible Solutions and Design Alternatives

One approach that we have implemented is to label the tweets text as having Purchase Intention and Not having Purchase intention. We have annotated about 3000 tweets from Twitter using our own web crawler. After preprocessing the tweets, we are left with about 1300 tweets for training data and remaining for testing. We defined definition of Purchase Intention as object that is having action word like (buy, want, desire) associated with it. We have manually annotated the data by reading each tweet and label them as purchase intention and non-purchase intention tweet. We have used this table as a reference to label the tweets:

Criteria for Labelling of tweets

|  |  |  |
| --- | --- | --- |
|  | Tweet | Class |
| 1 | Comparing iphone x with other phone and telling other phone are better? | No PI |
| 2 | Talking about good features of iphone x? | PI |
| 3 | Talking about negative features of iphone x? | No PI |
| 4 | liked video on Youtube about iphone x? | PI |

Each tweet was read by 3 people and final class was decided by maximum voting.

Next, we preprocessed the tweets using these techniques:

1. LOWERCASE

final\_data\_frame["text"] = final\_data\_frame["text"].apply(

lambda x: " ".join(x.lower() for x in x.split())

)

2. REMOVE PUNC

final\_data\_frame["text"] = final\_data\_frame["text"].str.replace(

"[^\w\s]", "")

3. STOPWORDS REMOVAL

stop = stopwords.words("english")

final\_data\_frame["text"] = final\_data\_frame["text"].apply(

lambda x: " ".join(x for x in x.split() if x not in stop)

)

4. COMMON WORD REMOVAL

freq = pd.Series(

" ".join(final\_data\_frame["text"]).split()).value\_counts()[:10]

freq = list(freq.index)

final\_data\_frame["text"] = final\_data\_frame["text"].apply(

lambda x: " ".join(x for x in x.split() if x not in freq)

)

5. RARE WORDS REMOVAL

rare = pd.Series(

" ".join(final\_data\_frame["text"]).split()).value\_counts()[-10:]

rare = list(rare.index)

final\_data\_frame["text"] = final\_data\_frame["text"].apply(

lambda x: " ".join(x for x in x.split() if x not in rare)

)

6. SPELLING CORRECTION

final\_data\_frame["text"][:5].apply(lambda x: str(TextBlob(x).correct()))

7. STEMMING

st = PorterStemmer()

final\_data\_frame["text"][:5].apply(

lambda x: " ".join([st.stem(word) for word in x.split()])

)

8. LEMMATIZATION

final\_data\_frame["text"] = final\_data\_frame["text"].apply(

lambda x: " ".join([Word(word).lemmatize() for word in x.split()])

)

Next, we made 3 types of document vectors:

1. TF

2. IDF

3. TF-IDF

Once the corpus was ready, we then used different text analytical models to test which one gave the best results. We used the following models:

1. Support Vector Machine (SVM)

2. Naive Bayes

3. Logistic Regression

4. Decision Tree

5. Neural Network

To evaluate our models, we used the following techniques:

1. Confusion Matrix

2. Accuracy

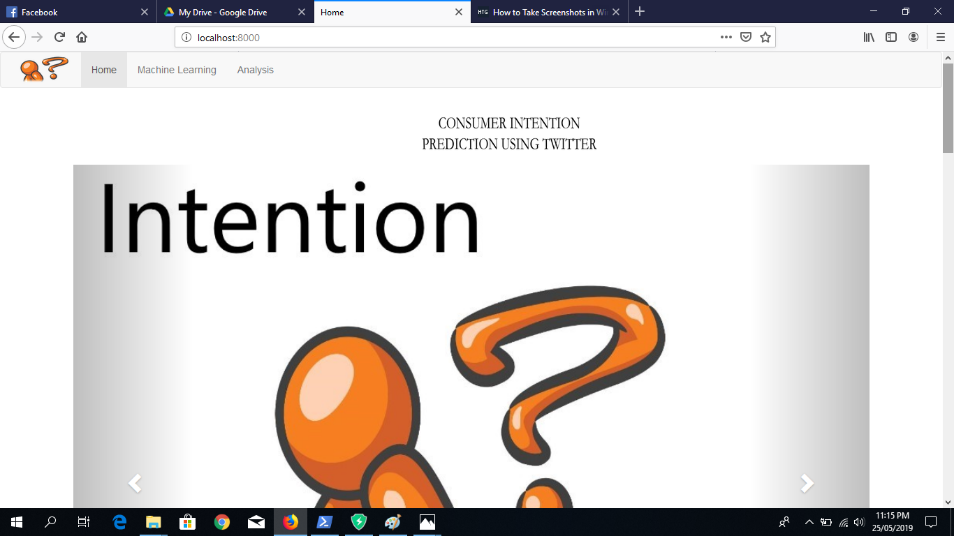
3. Precision

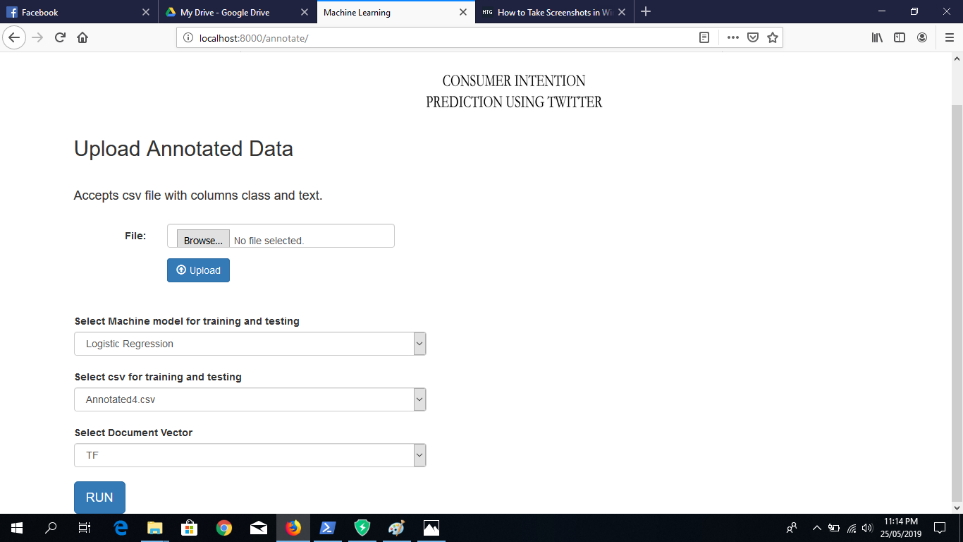
4. Recall

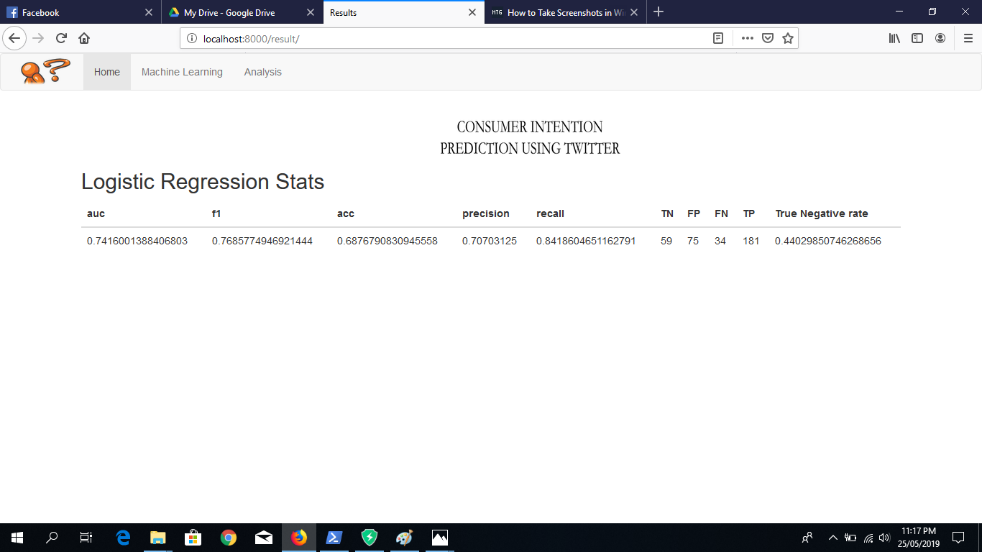
5. F-Measure

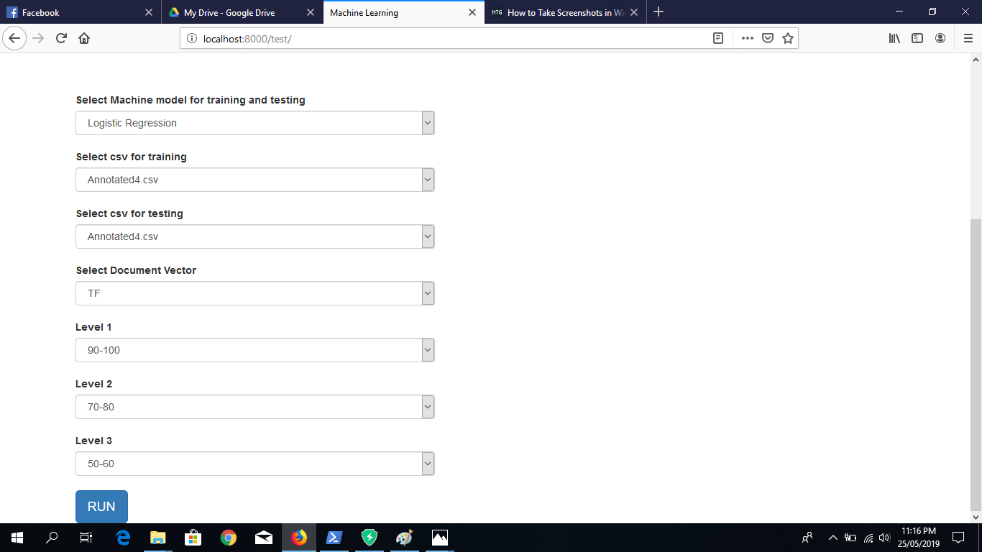
Further we have also considered the True Negative Rate, The True Positive Rate and the shape of the ROC curve for more insights.

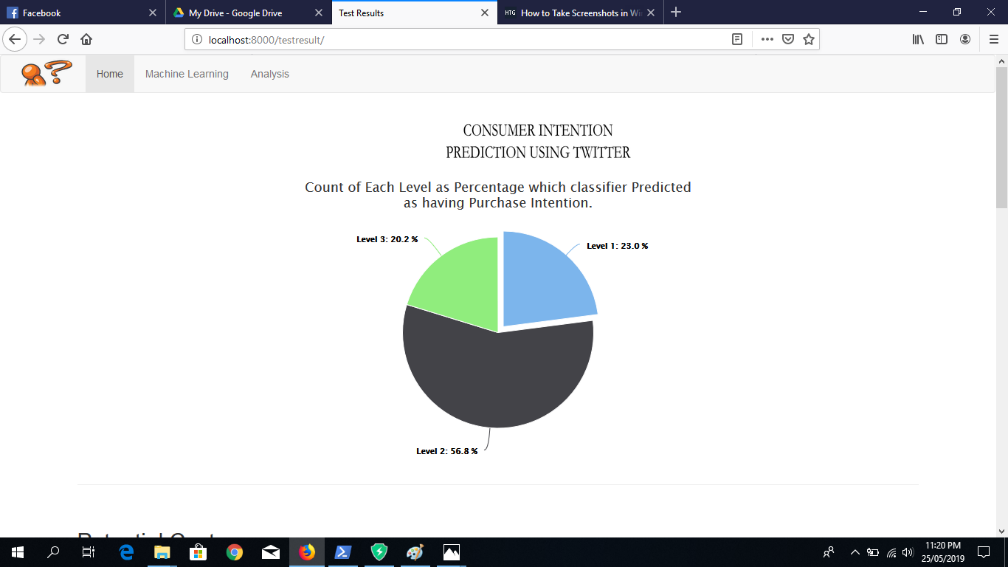
1. System Level Overview
2. Firstly, the user will open the DASHBOARD and see the status of the product through charts and the relevant tweets for the product in a list.
3. Secondly, the user will press the UPLOAD ANNOTATED DATASET button through which he will be asked to upload the dataset he wants to test for his product and find the relevant customers from that dataset. The output will be in a form of a pie chart and a table showing the different level of customers and the scores assigned to each customer.
4. Thirdly, when the user presses the ANALYSIS button, he will be taken to the screen which will show the detailed analysis of our dataset that we have built containing word clouds, positive vs negative tweets and the most used words for that product.

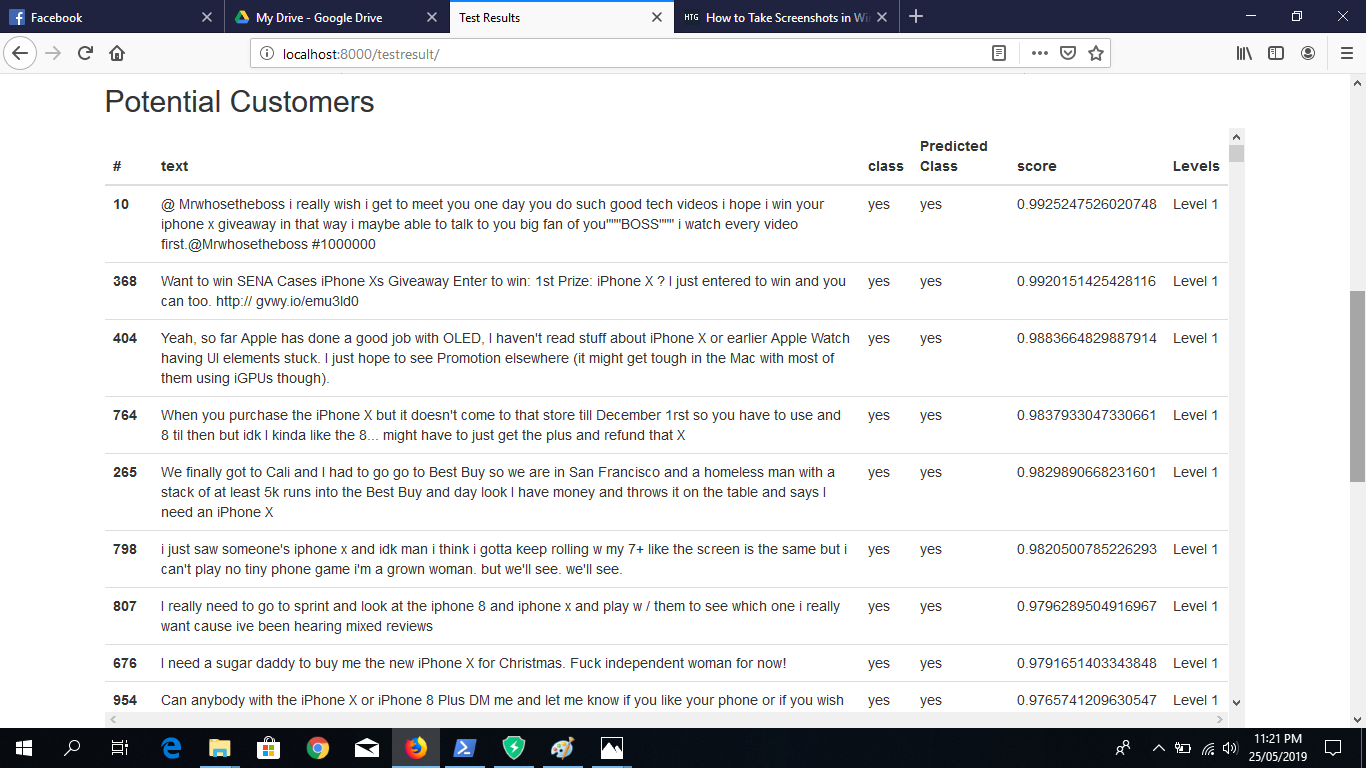


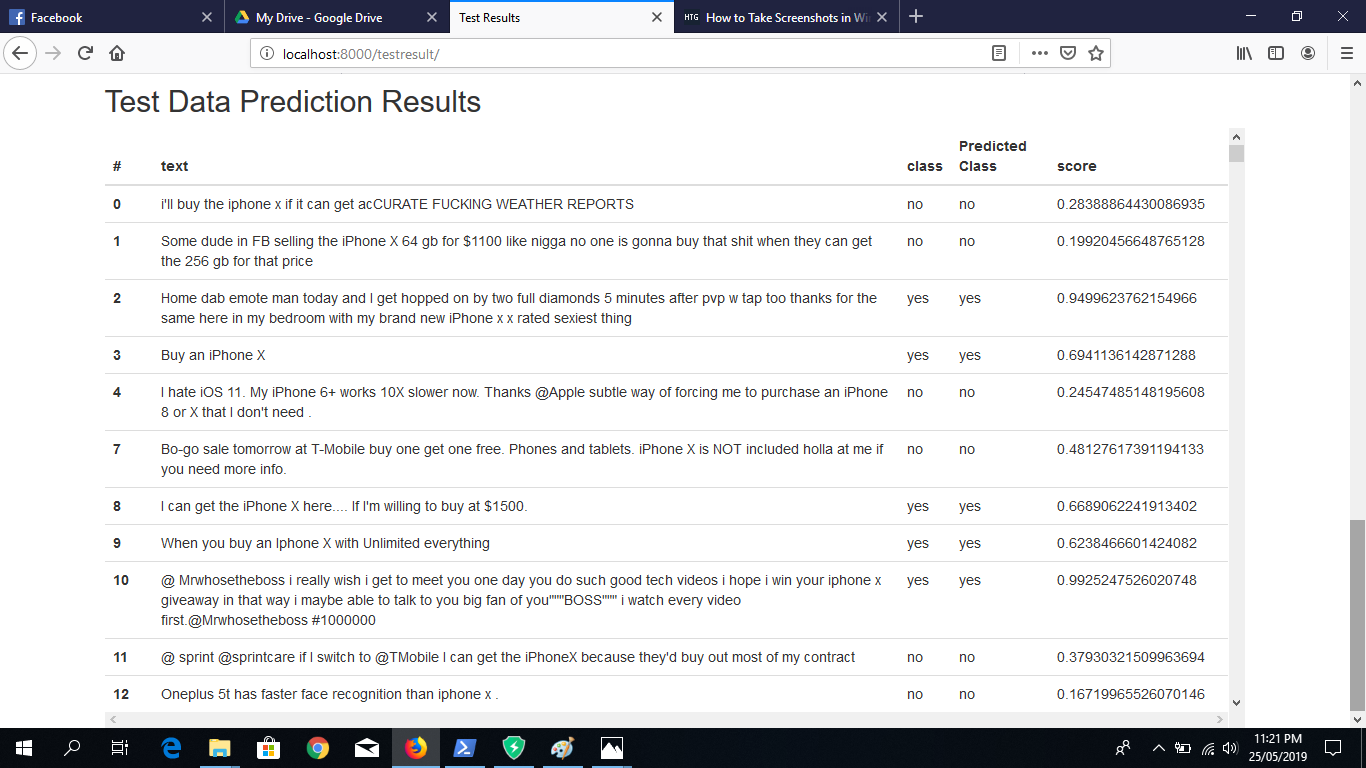


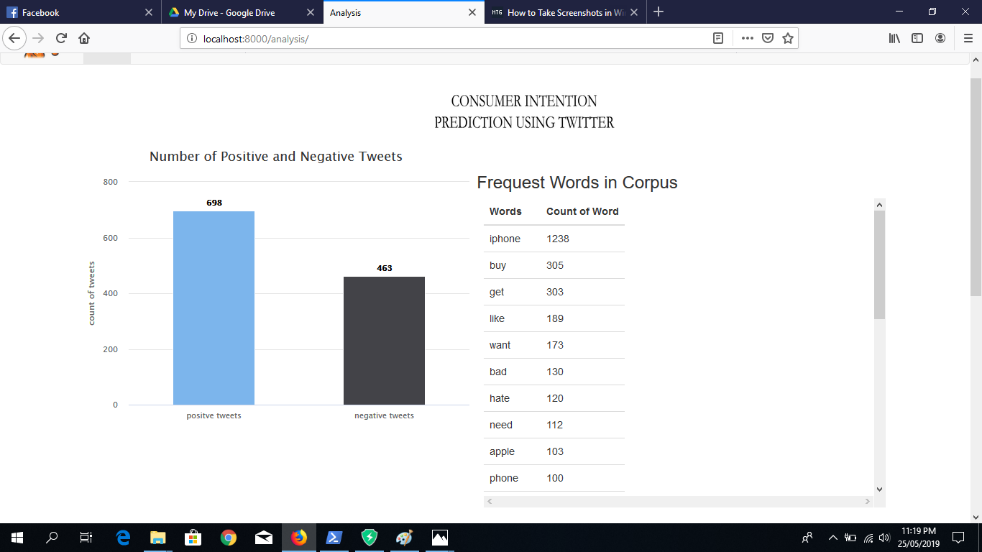


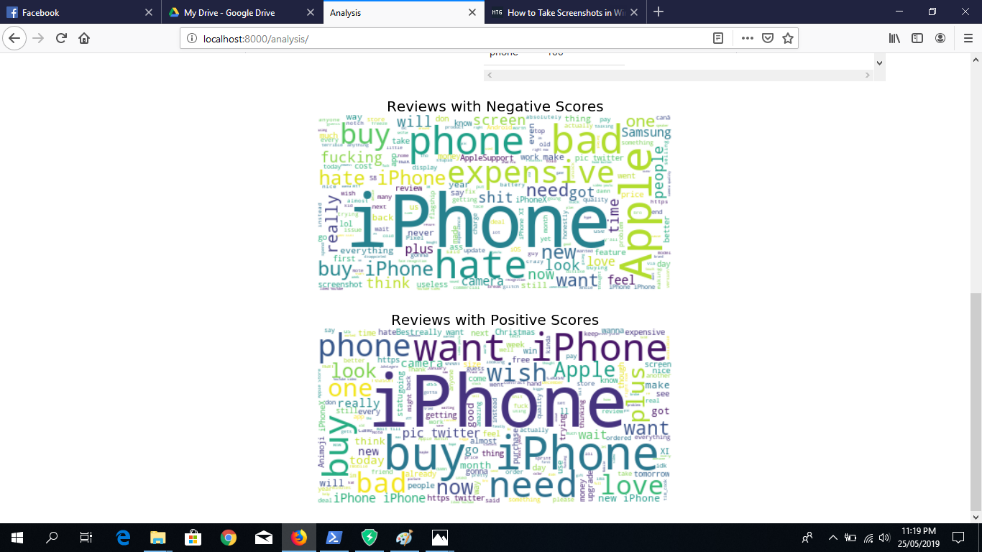












1. Module Level Descriptions
   1. Graphical User Interface Modules

The graphical user interface is a simple website application with each page offering a separate feature.

The first module is the analysis module which shows the detailed analysis of the dataset including word clouds, positive vs negative tweets and the most used words for that product.

The second module is the option to upload a CSV to evaluate the product tweets. The output will be in a form of a pie chart and a table showing the different level of customers and the scores assigned to each customer.

* 1. Control Modules

Scaping module: the scraping module will scrape the tweets for the product specified and store the results in a mongo DB database with each tweet as a separate document.

Data processing module: this module will clean and process the tweets which have purchase intention words associated with the desired product.

The machine learning module: this module will use the model built to check if the new tweet has purchase intention, and if so, will calculate the estimated amount of purchase intention for the user of the tweet for the product.

* 1. Miscellaneous Modules

Other modules will be used to generate lists and graphs which we will use to give some amount of analysis easy interpretation of the data for the users.

1. Assessment of Proposed Solution

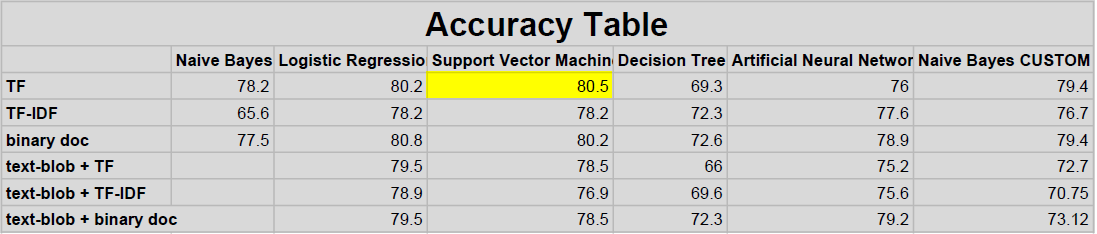
We have established different scenarios for our web application. We will check each scenario and the outcome and see if it matches with the expected result.

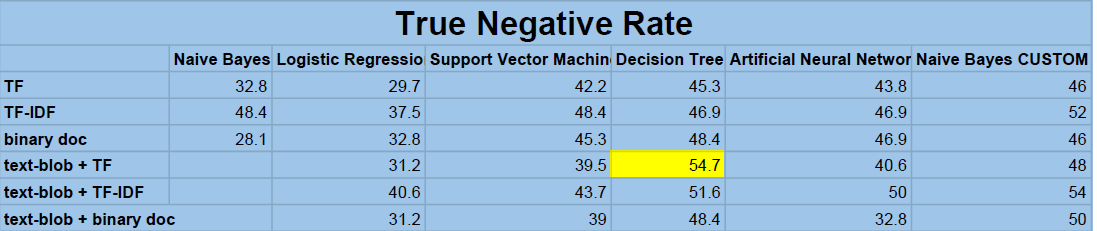
We will set an accuracy standard for our model and check if the results match the desired standard.

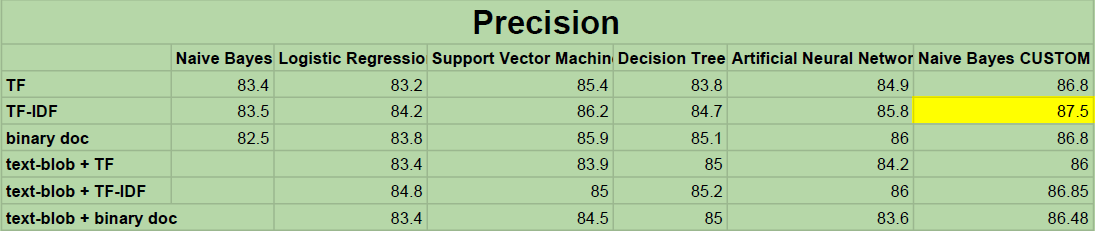
We will also assess the non-functional requirements of the website by evaluating the security, scalability and flexibility of the website in terms of users who use it and for the admins who will monitor the website.

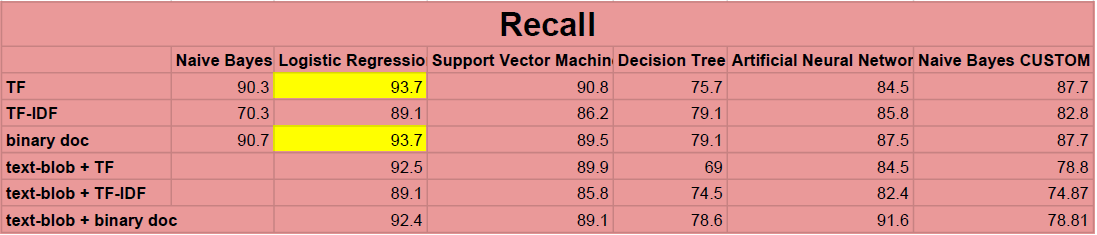
After evaluating our model here are the following results that we have gotten:

For our first attempt this is the results that we got:

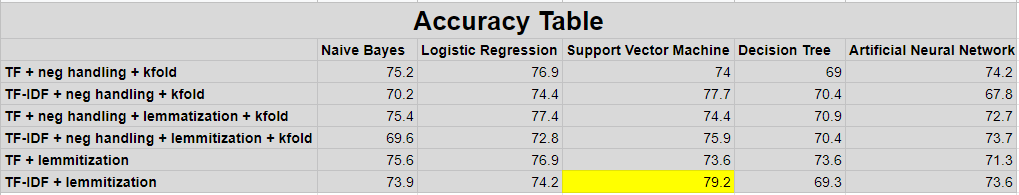


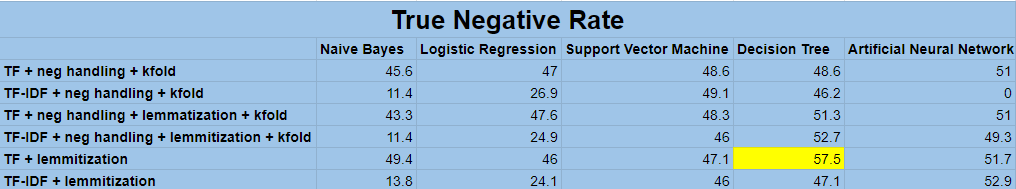






For our second attempt after reorganizing the data preprocessing steps and adding some customized steps specific to our data we got these results:





**Work Plan:**

1. Feasibility Assessment

This tool can be used by any organization’s marketing department. This application will help increase the sales of the company as it will target those users who have shown interest in the product. Therefore, it can get positive response in the market.

One of the major concerns of the tool is to input data from different sources, which currently is being done through a scraper which is not reliable in the long term. We will need access to the Search API of Twitter if we want to reliably gather tweets from the website or we will have to buy the data from companies which sell similar products.

Financial requirements for this tool are minimal because right now we are using a scraper to gather data.

* 1. Skill and Resources

This project requires us to have knowledge about data scraping, data cleaning and processing, applying machine learning models and web application development and design.

* 1. Risk Assessment
* **Technical Risk:** High.
* **Timing Risk:** High.
* **Budget Risk:** Low.

We have highlighted the technical and timing risk as high because it took us a long time to prepare the annotated dataset and to learn the various text analytical models and how to apply them. Further, we also had to learn the Django framework because that is the platform on which we have developed our website.