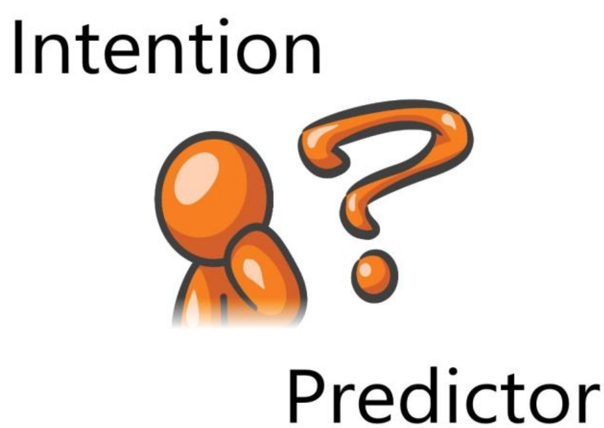
Consumer Intention Prediction using Twitter (CIP)



FINAL YEAR PROJECT

Consumer Intention Prediction using Twitter (CIP)

Intention

Predictor

Supervisor:

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

Group Members:

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

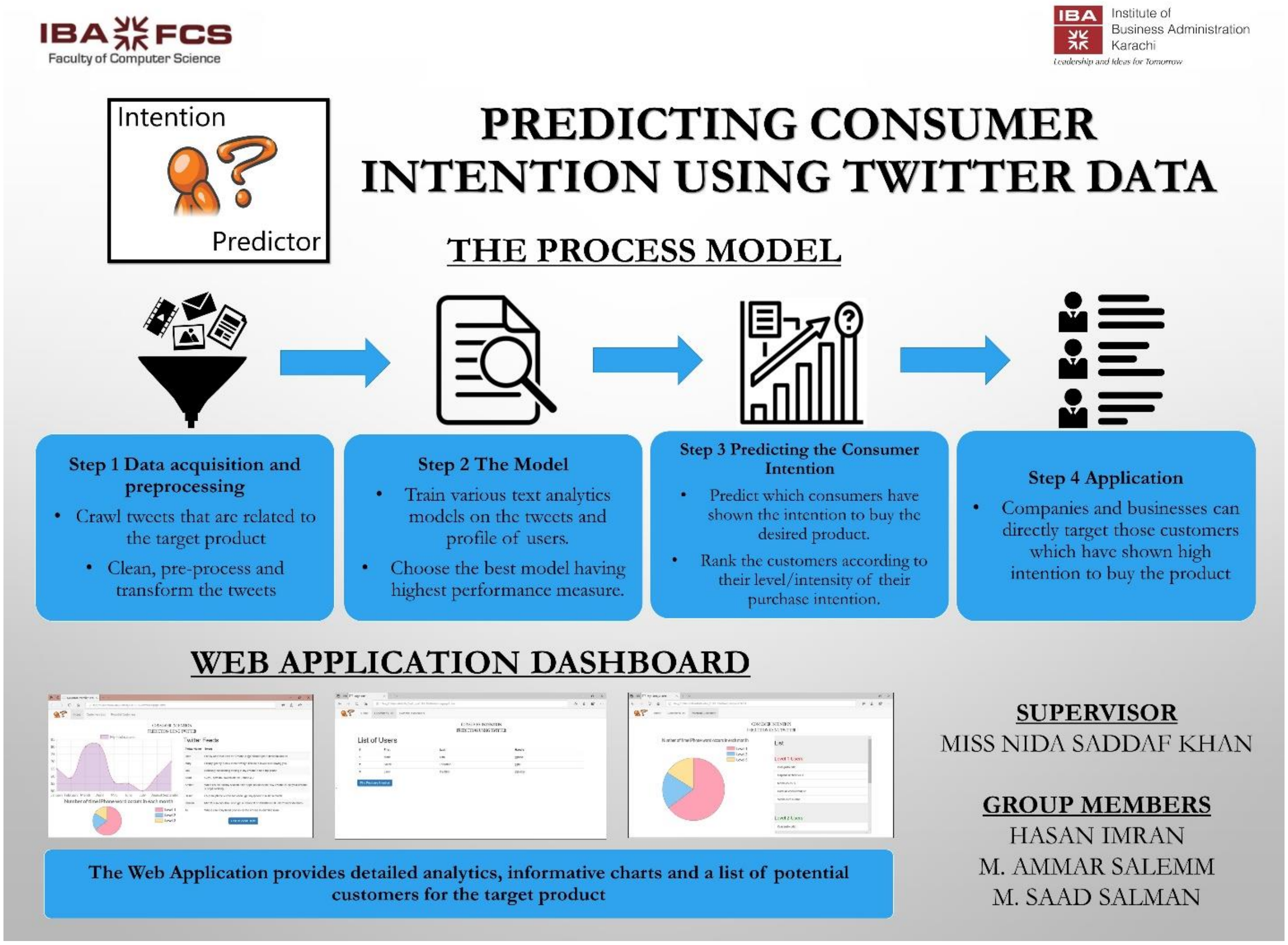
Muhammad Ammar Saleem (msaleem@khi.iba.edu.pk)

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

Introduction:

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.

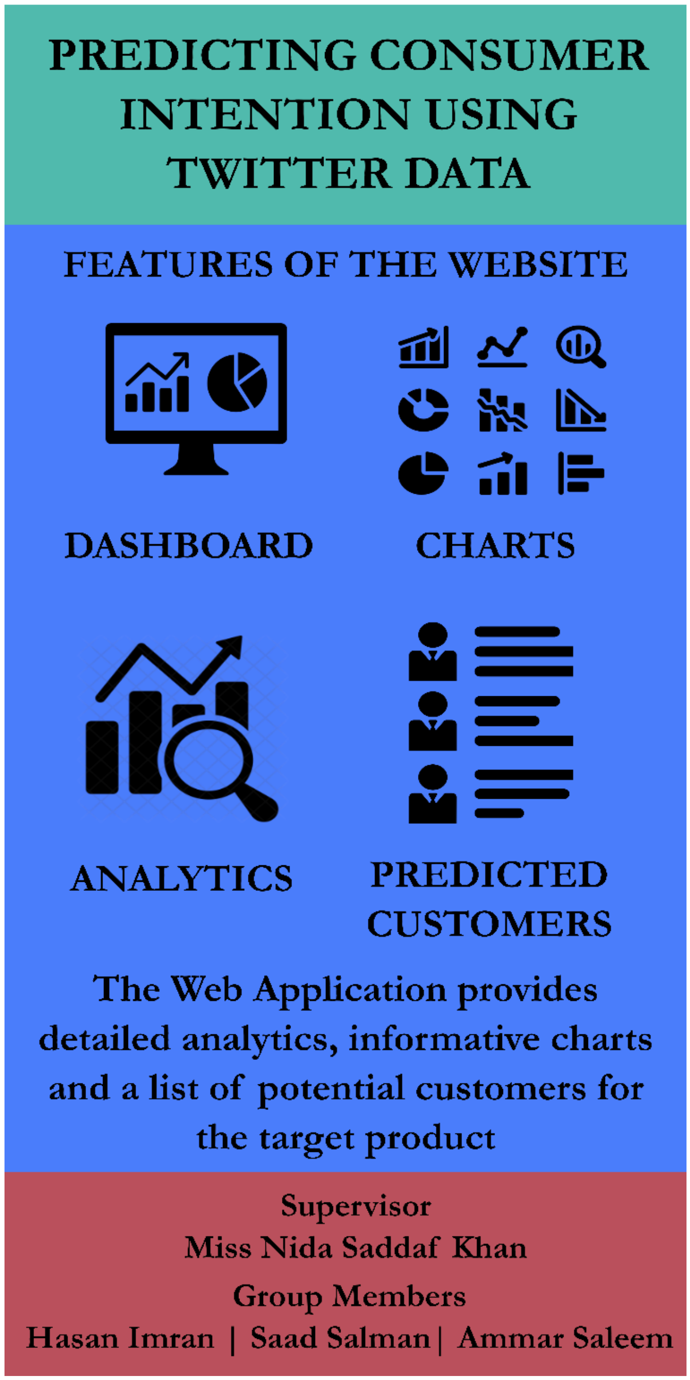
First, we search for Twitter tweets of potential customers wanting to buy a product. And based on those tweets we estimate/predict the likelihood that the customer will buy the product. We then make a model by gathering tweets from users who have already expressed intention to buy the product using their 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.


IBA FCS
Faculty of Computer Science
IBA Institute of Business Administration Karachi
Leadership and Ideas for Tomorrow
Intention
Predictor
PREDICTING CONSUMER INTENTION USING TWITTER DATA
THE PROCESS MODEL
Step 1 Data acquisition and preprocessing
• Crawl tweets that are related to the target product
• Clean, pre-process and transform the tweets
Step 2 The Model
• Train various text analytics models on the tweets and profile of users.
• Choose the best model having highest performance measure.
Step 3 Predicting the Consumer Intention
• Predict which consumers have shown the intention to buy the desired product.
• Rank the customers according to their level/intensity of their purchase intention.
Step 4 Application
• Companies and businesses can directly target those customers which have shown high intention to buy the product
WEB APPLICATION DASHBOARD
The Web Application provides detailed analytics, informative charts and a list of potential customers for the target product
SUPERVISOR
MISS NIDA SADDAF KHAN
GROUP MEMBERS
HASAN IMRAN
M. AMMAR SALEMM
M. SAAD SALMAN

IBA FCS Faculty of Computer Science

IBA Institute of Business Administration Karachi
Leadership and Ideas for Tomorrow

Intention Predictor



High Level Design:

1. Rationale and sources of your project idea

   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.

2. Logical structure

   i. 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.

   ii. Dashboard screen

   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.

   iii. Upload annotated dataset function

   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.

   iv. Analysis function

   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.

3. Background

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.

4. Hardware / Software tradeoffs

Hardware is always preferred over software and in some cases, hardware is regarded as highly optimized form of software.

Since we have focused on developing a machine learning algorithm and deploy it on a website, we must have sophisticated hardware on the server end but if a user is using the website that it is not necessary that the user have a sophisticated and high-end laptop.

We have developed our program on Python and deployed it on the Django Framework, so the software requirement isn't as high. Anyone with a basic python shell can run the code.

5. Relationship with available past projects or standards e,g. IEEE, ANSI, ISO and etc.

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. Studies on identification of wishes from texts, specifically Ramanand et al. (Ramanand, Bhavsar, and Pedanekar 2010) consider the task of identifying 'buy' wishes from product reviews. These wishes include suggestions for a product or a desire to buy a product. They used linguistic rules to detect these two kinds of wishes. Although rule-based approaches for identifying the wishes are effective, but their coverage is not satisfactory, and they can't be extended easily. Purchase Intention detection task is close to the task of identifying wishes in product reviews. Here we don't use the rule-based approach, but we present a machine learning approach with generic features extracted from the tweets.

Past studies have shown that it is possible to apply Natural Language Processing (NLP) and Named Entity Recognition (NER) to tweets (Li et al., 2012) (Liu et al., 2011). However, applying NER to tweets is very difficult because people often use abbreviations or (deliberate) misspelled words and grammatical errors in tweets. Nonetheless, Finin et al. (2010) tried to annotate named entities in tweets using crowdsourcing. Other studies used these techniques to apply sentiment analysis to tweets. The first studies used product or movie reviews because these reviews are either positive or negative. Wang et al. (2011) and Anta et al. (2013) analyzed the sentiment of tweets filtered on a certain hashtag (keywords or phrases starting with the symbol that denote the main topic of a tweet). These studies merely analyze the sentiment of a tweet about a product after the author has bought it. We will however be extracting features from tweets to find whether the user has shown purchase intention towards the product or not.

More recently, research articles like Identifying Purchase Intentions by Extracting Information from Tweets (February 8, 2017, RADBOUD U NIVERSITY NIJMEGEN) and Tweetalyst: Using Twitter Data to Analyze Consumer Decision Process (The Berkeley Institute of Design) investigate if an artificial intelligence approach can predict (from existing user created content on twitter) if someone is a potential customer for a specific company or product and identify users at different stages of the decision process of buying a given product. Further looking at research reports like The Impact of Social Network Marketing on Consumer Purchase Intention in Pakistan: Consumer Engagement as a Mediator (Asian Journal of Business and Accounting 10(1), 2017) give us an insight of the impact of social network marketing on consumer purchase intention and how it is affected by the mediating role of consumer engagement. Based on UGT theory (Uses and Gratification Theory).

Some preprocessing techniques commanly used for twitter data are the sentiment140 API (Sentiment140 allows you to discover the sentiment of a brand, product, or topic on Twitter), the TweetNLP library (a tokenizer, a part-of-speech tagger, hierarchical word clusters, and a dependency parser for tweets), unigrams, bigrams and stemming. There are also some dictionary-based approaches such as using the textBlob library (TextBlob is a Python (2 and 3) library for processing textual data. It provides a consistent API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, and more).

The common machine learning algorithms that are used for text analysis are Linear Regression, Random Forest, Naive Bayes and Support Vector Machine. We will be looking at these models later in detail.

6. Patents, copyright and trademarks

We have not used any copyright software or patents or trademarks. Python is an open source library as well as Django and therefore we have no such requirement.

Software / Hardware Design:

1. Overview

   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.

2. Program Details

   a. Overview

      The 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

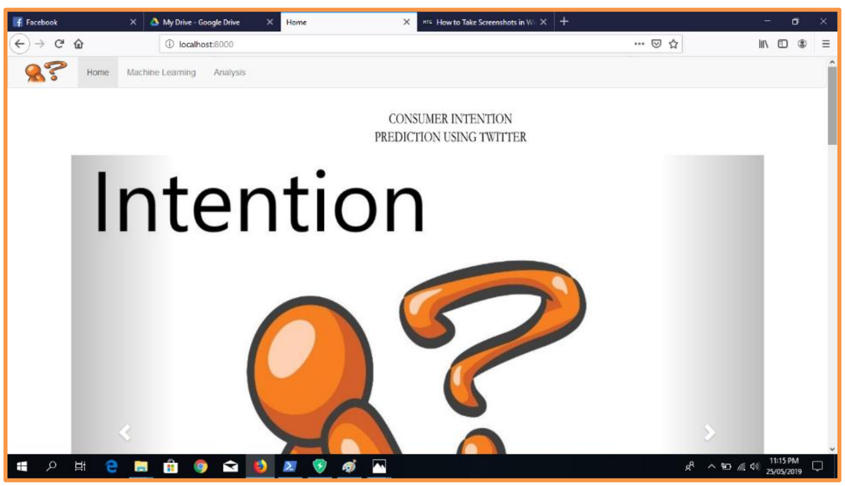
3. Logistic Regression
4. Decision Tree
5. Neural Network

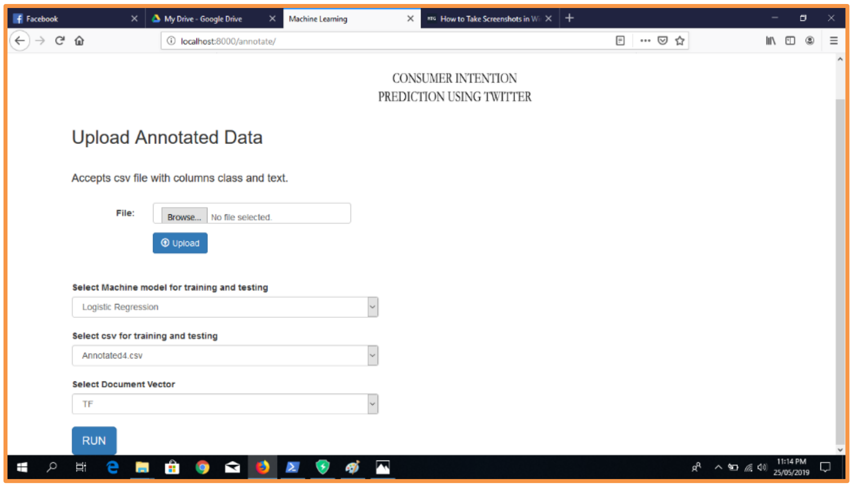
b. User interface

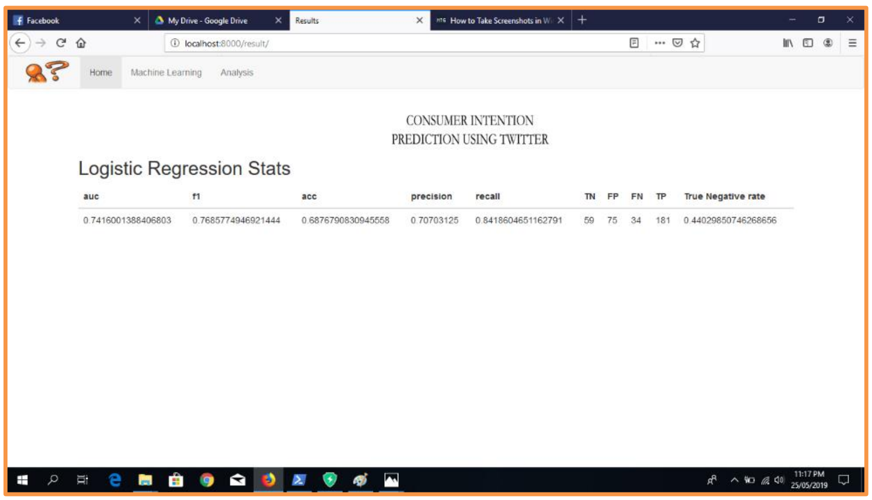
1. Firstly, the user will open the DASHBOARD and see the status of the product through charts and the relevant tweets for the product in a list.

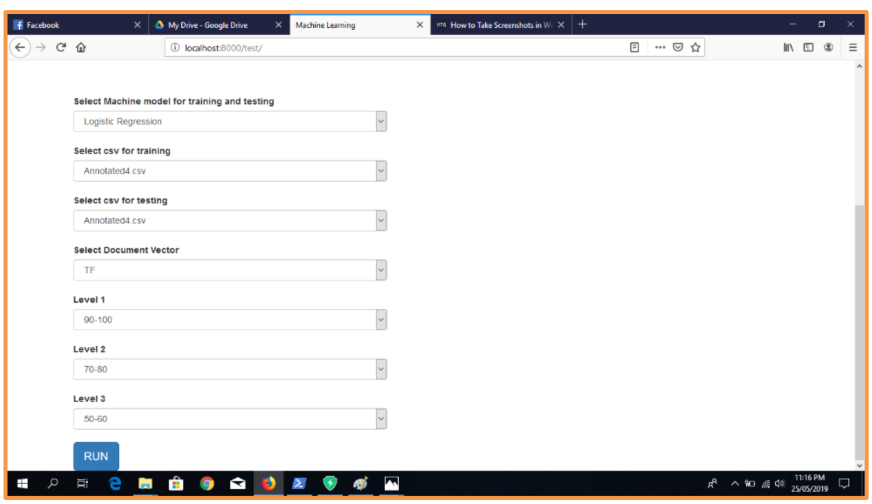
2. Secondly, the user will press the UPLOAD ANNOTATED DATASET button through which he will be asked to upload the dataset he wants to test for his product and find the relevant customers from that dataset. The output will be in a form of a pie chart and a table showing the different level of customers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

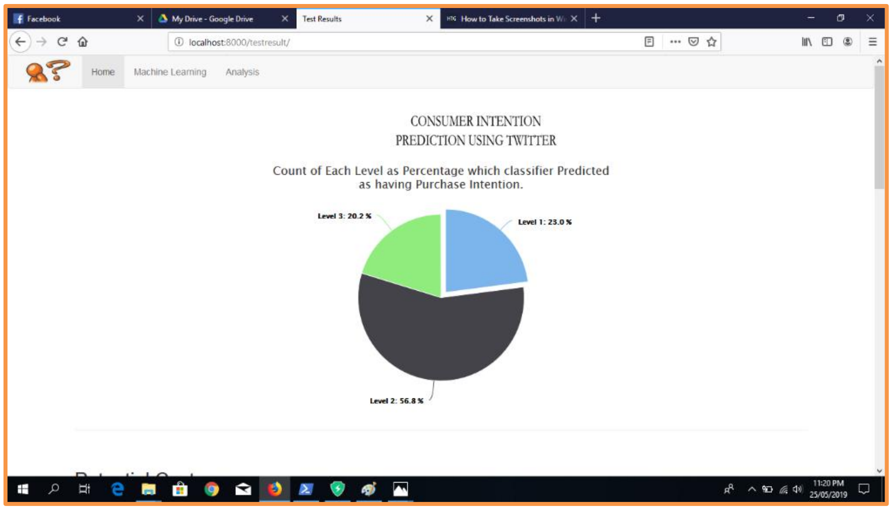
3. Thirdly, when the user presses the ANALYSIS button, he will be taken to the screen which will show the detailed analysis of our dataset that we have built containing word clouds, positive vs negative tweets and the most used words for that product.

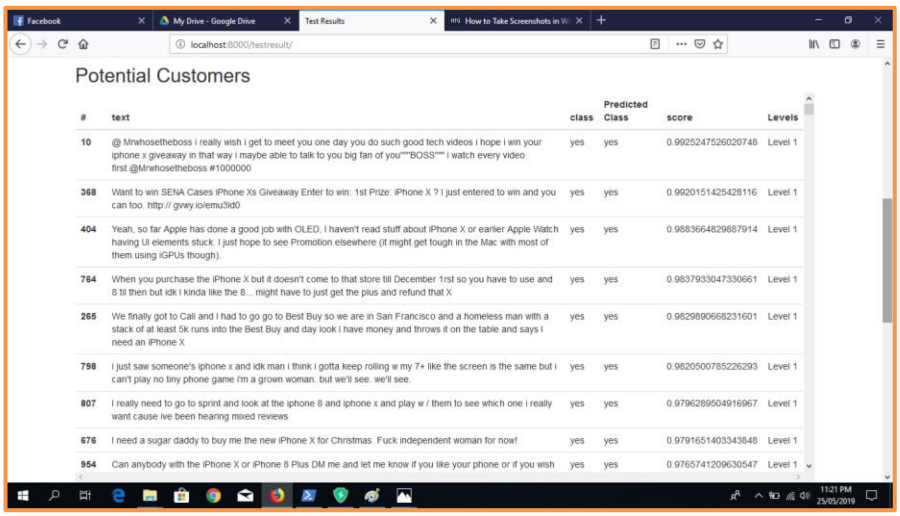


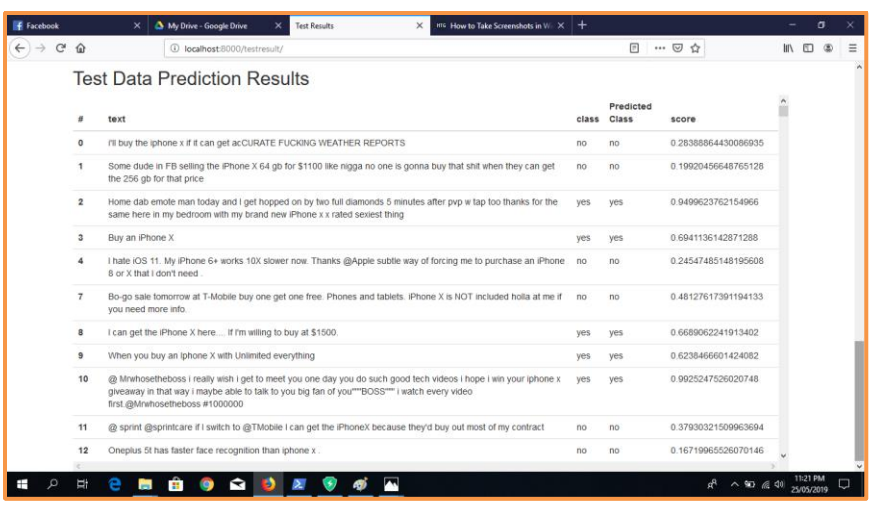

CONSUMER INTENTION
PREDICTION USING TWITTER
Upload Annotated Data
Accepts csv file with columns class and text.
File:
Browse... No file selected.
Upload
Select Machine model for training and testing
Logistic Regression
Select csv for training and testing
Annotated4.csv
Select Document Vector
TF
RUN

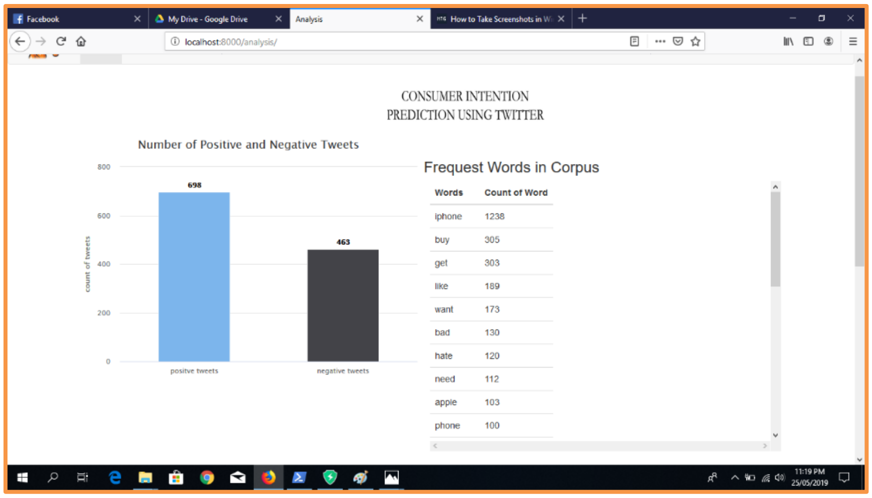

Home
Machine Learning
Analysis
CONSUMER INTENTION
PREDICTION USING TWITTER
Logistic Regression Stats
auc f1 acc precision recall TN FP FN TP True Negative rate
0.7416001388406803 0.7685774946921444 0.6876790830945558 0.70703125 0.8418604651162791 59 75 34 181 0.44029850746268656

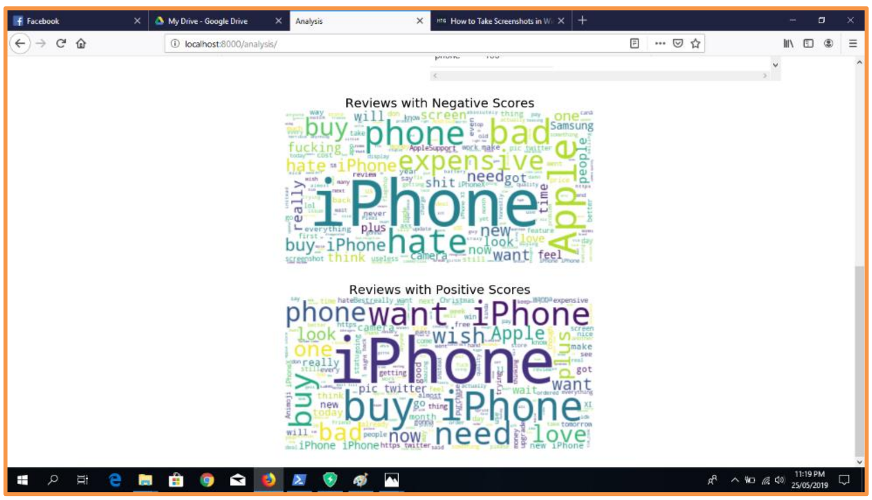

Select Machine model for training and testing
Logistic Regression
Select csv for training
Annotated4.csv
Select csv for testing
Annotated4.csv
Select Document Vector
TF
Level 1
90-100
Level 2
70-80
Level 3
50-60
RUN


Home
Machine Learning
Analysis
CONSUMER INTENTION
PREDICTION USING TWITTER
Count of Each Level as Percentage which classifier Predicted as having Purchase Intention.
Level 3: 20.2 %
Level 1: 23.0 %
Level 2: 56.8 %


Potential Customers
text class Predicted Class score Levels
10 @ Mrwhosetheboss i really wish i get to meet you one day you do such good tech videos i hope i win your iphone x giveaway in that way i maybe able to talk to you big fan of you"""BOSS""" i watch every video first.@Mrwhosetheboss #1000000 yes yes 0.9925247526020748 Level 1
368 Want to win SENA Cases iPhone Xs Giveaway Enter to win: 1st Prize: iPhone X ? I just entered to win and you can too. http:// gvwy.io/emu3id0 yes yes 0.9920151425428116 Level 1
404 Yeah, so far Apple has done a good job with OLED, I haven't read stuff about iPhone X or earlier Apple Watch having UI elements stuck. I just hope to see Promotion elsewhere (it might get tough in the Mac with most of them using iGPUs though). yes yes 0.9883664829887914 Level 1
764 When you purchase the iPhone X but it doesn't come to that store till December 1rst so you have to use and 8 til then but idk I kinda like the 8... might have to just get the plus and refund that X yes yes 0.9837933047330661 Level 1
265 We finally got to Cali and I had to go go to Best Buy so we are in San Francisco and a homeless man with a stack of at least 5k runs into the Best Buy and day look i have money and throws it on the table and says I need an iPhone X yes yes 0.9829890668231601 Level 1
798 i just saw someone's iphone x and idk man i think i gotta keep rolling w my 7+ like the screen is the same but i can't play no tiny phone game i'm a grown woman. but we'll see. we'll see. yes yes 0.9820500785226293 Level 1
807 I really need to go to sprint and look at the iphone 8 and iphone x and play w / them to see which one i really want cause ive been hearing mixed reviews yes yes 0.9796289504916967 Level 1
676 I need a sugar daddy to buy me the new iPhone X for Christmas. Fuck independent woman for now! yes yes 0.9791651403343848 Level 1
954 Can anybody with the iPhone X or iPhone 8 Plus DM me and let me know if you like your phone or if you wish yes yes 0.9765741209630547 Level 1


Test Data Prediction Results
text class Predicted Class score
0 I'll buy the iphone x if it can get acCURATE FUCKING WEATHER REPORTS no no 0.28388864430086935
1 Some dude in FB selling the iPhone X 64 gb for $1100 like nigga no one is gonna buy that shit when they can get the 256 gb for that price no no 0.19920456648765128
2 Home dab emote man today and I get hopped on by two full diamonds 5 minutes after pvp w tap too thanks for the same here in my bedroom with my brand new iPhone x x rated sexiest thing yes yes 0.9499623762154966
3 Buy an iPhone X yes yes 0.6941136142871288
4 I hate iOS 11. My iPhone 6+ works 10X slower now. Thanks @Apple subtle way of forcing me to purchase an iPhone 8 or X that I don't need . no no 0.24547485148195608
7 Bo-go sale tomorrow at T-Mobile buy one get one free. Phones and tablets. iPhone X is NOT included holla at me if you need more info. no no 0.48127617391194133
8 I can get the iPhone X here.... If I'm willing to buy at $1500. yes yes 0.6689062241913402
9 When you buy an Iphone X with Unlimited everything yes yes 0.6238466601424082
10 @ Mrwhosetheboss i really wish i get to meet you one day you do such good tech videos i hope i win your iphone x giveaway in that way i maybe able to talk to you big fan of you"""BOSS""" i watch every video first.@Mrwhosetheboss #1000000 yes yes 0.9925247526020748
11 @ sprint @sprintcare if I switch to @TMobile I can get the iPhoneX because they'd buy out most of my contract no no 0.37930321509963694
12 Oneplus 5t has faster face recognition than iphone x . no no 0.16719965526070146





c. Errors

Initially we encountered some errors in running the python environment and setting up a separate virtual environment for our project. However, we were soon able to overcome this by reading up on the documentation and following tutorials.

We also encountered some errors running the Django framework. Setting up a sperate virtual environment and installing the missing dependencies later fixed this problem.

We also faced some issues in integrating the python code with the website because it was our first time configuring a back-end server to run code scripts and sending the output to the html page and getting it to display correctly.

d. Trails and 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 opted to use unit testing in which individual units of source code, were tested to determine if they are fit to use. Then, we ran integration testing where the individual source was merged and tested as group.

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

3. Hardware Details

   a. Overview

   Since our project is purely software based and we have developed a web application as a final deliverable we did not have any hardware requirements.

   We tested all the algorithms and the web application on multiple *for daily use laptops* and the functions and outputs were all displaying correctly.

   b. User interface hardware

   Our user interface hardware would be the laptop and so the web application includes graphical controls, which the user can select using a mouse or keyboard.

   c. Things that did not work

   Since our project is purely a software-based web application we did not have to test any hardware but rather just checked on multiple laptops if the website was functioning properly.

   d. Trails and tests

   We assessed 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.

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

Results:

Since we are using a machine learning (artificial intelligence) based approach we needed to set an accuracy standard for our model and evaluate the results by matching the desired standard.

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.

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

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

Accuracy Table

	Naive Bayes	Logistic Regressio	Support Vector Machin	Decision Tree	Artificial Neural Networ	Naive Bayes CUSTOM
TF	78.2	80.2	80.5	69.3	76	79.4
TF-IDF	65.6	78.2	78.2	72.3	77.6	76.7
binary doc	77.5	80.8	80.2	72.6	78.9	79.4
text-blob + TF		79.5	78.5	66	75.2	72.7
text-blob + TF-IDF		78.9	76.9	69.6	75.6	70.75
text-blob + binary doc		79.5	78.5	72.3	79.2	73.12

True Negative Rate

	Naive Bayes	Logistic Regressio	Support Vector Machin	Decision Tree	Artificial Neural Networ	Naive Bayes CUSTOM
TF	32.8	29.7	42.2	45.3	43.8	46
TF-IDF	48.4	37.5	48.4	46.9	46.9	52
binary doc	28.1	32.8	45.3	48.4	46.9	46
text-blob + TF		31.2	39.5	54.7	40.6	48
text-blob + TF-IDF		40.6	43.7	51.6	50	54
text-blob + binary doc		31.2	39	48.4	32.8	50

Precision						
	Naive Bayes	Logistic Regressio	Support Vector Machin	Decision Tree	Artificial Neural Networ	Naive Bayes CUSTOM
TF	83.4	83.2	85.4	83.8	84.9	86.8
TF-IDF	83.5	84.2	86.2	84.7	85.8	87.5
binary doc	82.5	83.8	85.9	85.1	86	86.8
text-blob + TF		83.4	83.9	85	84.2	86
text-blob + TF-IDF		84.8	85	85.2	86	86.85
text-blob + binary doc		83.4	84.5	85	83.6	86.48

Recall						
	Naive Bayes	Logistic Regressio	Support Vector Machin	Decision Tree	Artificial Neural Networ	Naive Bayes CUSTOM
TF	90.3	93.7	90.8	75.7	84.5	87.7
TF-IDF	70.3	89.1	86.2	79.1	85.8	82.8
binary doc	90.7	93.7	89.5	79.1	87.5	87.7
text-blob + TF		92.5	89.9	69	84.5	78.8
text-blob + TF-IDF		89.1	85.8	74.5	82.4	74.87
text-blob + binary doc		92.4	89.1	78.6	91.6	78.81

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

Accuracy Table					
	Naive Bayes	Logistic Regression	Support Vector Machine	Decision Tree	Artificial Neural Network
TF + neg handling + kfold	75.2	76.9	74	69	74.2
TF-IDF + neg handling + kfold	70.2	74.4	77.7	70.4	67.8
TF + neg handling + lemmatization + kfold	75.4	77.4	74.4	70.9	72.7
TF-IDF + neg handling + lemmitization + kfold	69.6	72.8	75.9	70.4	73.7
TF + lemmitization	75.6	76.9	73.6	73.6	71.3
TF-IDF + lemmitization	73.9	74.2	79.2	69.3	73.6

True Negative Rate					
	Naive Bayes	Logistic Regression	Support Vector Machine	Decision Tree	Artificial Neural Network
TF + neg handling + kfold	45.6	47	48.6	48.6	51
TF-IDF + neg handling + kfold	11.4	26.9	49.1	46.2	0
TF + neg handling + lemmatization + kfold	43.3	47.6	48.3	51.3	51
TF-IDF + neg handling + lemmitization + kfold	11.4	24.9	46	52.7	49.3
TF + lemmitization	49.4	46	47.1	57.5	51.7
TF-IDF + lemmitization	13.8	24.1	46	47.1	52.9

Conclusions:

Our results were quite promising since we had created our own dataset and were building the model from scratch. We had to create our own dataset because there does not exist a publicly available dataset for purchase intention based on twitter tweets.

The 2 major problems that we faced were:

1. The imbalance class problem: Since our dataset was manually annotated by us, we had about 2000 positive tweets and 1200 negative tweets. Due to this we were getting a very low True Negative Rate and our model was not accurately predicting the negative class.
2. Limited annotated data: Since we had to manual annotate each tweet in the dataset and this process takes a lot of time, we were only able to annotate about 3200 tweets.

Looking at the other researches that are done in the similar field, our project also stands apart since we have implemented 5 different models and after evaluating them, we choose the best one customized to the product data.

We were not able to get more than 80% accuracy because of the two problems highlighted above. To achieve even 80% accuracy with an imbalance class data and such a small dataset is a victory.

After showing the website to a few potential clients we have received positive remarks about our product and people seem interest in this new approach to customer identification and targeted marketing.

.

Appendix:

Appendix 1: Equations

1. Naïve Bayes Algorithm

   Bayes theorem provides a way of calculating the posterior probability, P(c|x), from P(c), P(x), and P(x|c). Naive Bayes classifier assume that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. This assumption is called class conditional independence.

Likelihood

Class Prior Probability

$$P(c \mid x) = \frac{P(x \mid c)\,P(c)}{P(x)}$$

Posterior Probability

Predictor Prior Probability

$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

- *P*(*c*|*x*) is the posterior probability of *class* (*target*) given *predictor* (*attribute*).

- *P*(*c*) is the prior probability of *class*.
- *P*(*x*|*c*) is the likelihood which is the probability of *predictor* given *class*.
- *P*(*x*) is the prior probability of *predictor*.

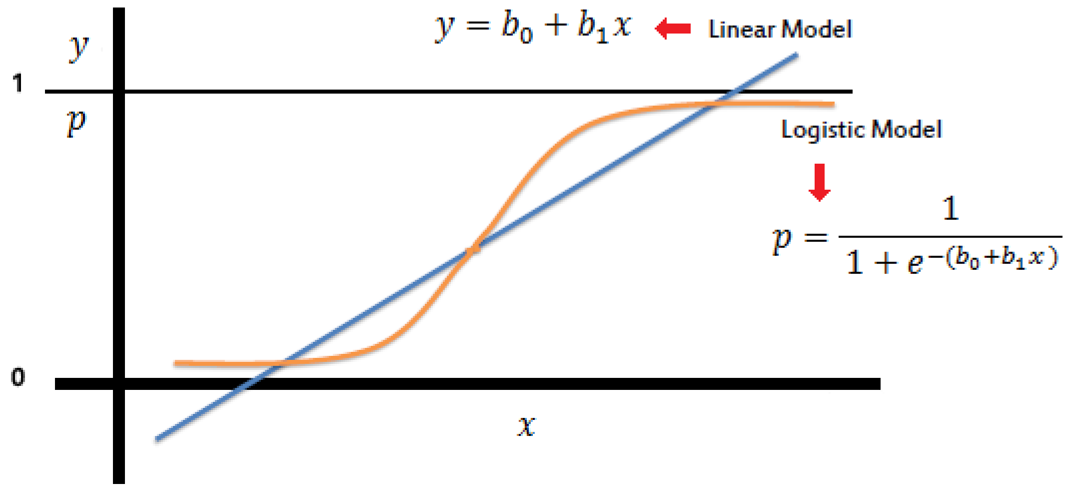
2. Support Vector Machine:

   Support Vector Machines are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships.

   For this type of SVM, training involves the minimization of the error function:
   $$\frac{1}{2}w^T w + C\sum_{i=1}^{N}\xi_i$$
   subject to the constraints:
   $$y_i\left(w^T\phi(x_i)+b\right)\geq 1-\xi_i \text{ and } \xi_i \geq 0, i=1,\ldots,N$$
   where C is the capacity constant, w is the vector of coefficients, b is a constant, and ξ_i represents parameters for handling nonseparable data (inputs). The index i labels the N training cases. Note that $y \in \pm 1$ represents the class labels and xi represents the independent variables. The kernel ϕ is used to transform data from the input (independent) to the feature space. It should be noted that the larger the C, the more the error is penalized. Thus, C should be chosen with care to avoid over fitting.

3. Logistic Regression

   Logistic regression predicts the probability of an outcome that can only have two values (i.e. a dichotomy). The prediction is based on the use of one or several predictors (numerical and categorical).



In the logistic regression the constant (b_0) moves the curve left and right and the slope (b_1) defines the steepness of the curve. By simple transformation, the logistic regression equation can be written in terms of an odds ratio.

$$\frac{p}{1-p} = \exp(b_0 + b_1 x)$$

Finally, taking the natural log of both sides, we can write the equation in terms of log-odds (logit) which is a linear function of the predictors. The coefficient (b_1) is the amount the logit (log-odds) changes with a one unit change in x.

$$ln\left(\frac{p}{1-p}\right) = b_0 + b_1 x$$

As mentioned before, logistic regression can handle any number of numerical and/or categorical variables.

$$p = \frac{1}{1 + e^{-(b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p)}}$$

4. Decision tree

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.

To build a decision tree, we need to calculate two types of entropy using frequency tables as follows:

a) Entropy using the frequency table of one attribute:

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

b) Entropy using the frequency table of two attributes:

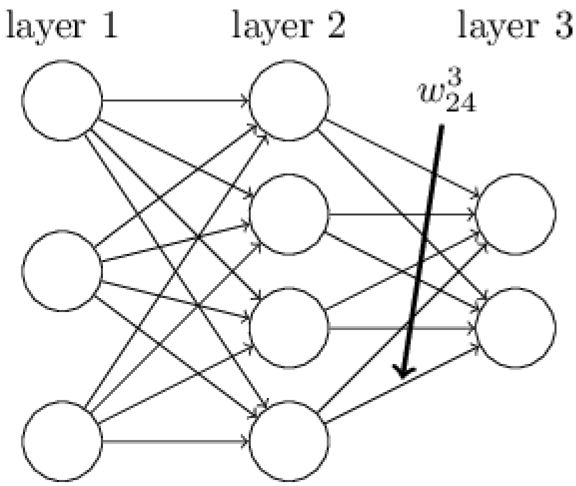
$$E(T, X) = \sum_{c \in X} P(c) E(c)$$

Constructing a decision tree is all about finding attribute that returns the highest information gain (i.e., the most homogeneous branches).

$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

5. Neural Network

Backpropagation algorithms are a family of methods used to efficiently train artificial neural networks (ANNs) following a gradient descent approach that exploits the chain rule. The main feature of backpropagation is its iterative, recursive and efficient method for calculating the weights updates to improve in the network until it is able to perform the task for which it is being trained.



Appendix 2: Code

1. Code for models

```
def output_to_results(pathData):
    final_data_frame, data_frame_undefined = extract(pathData)
    final_data_frame["text"] = final_data_frame["text"].apply(
        lambda x: " ".join(x.lower() for x in x.split())
    )
    final_data_frame["text"] = final_data_frame["text"].str.replace(
        "[^\w\s]", "")
    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)
    )
    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)
    )
    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)
    )
    final_data_frame["text"][:5].apply(lambda x: str(TextBlob(x).correct()))
    st = PorterStemmer()
```

```
    final_data_frame["text"][:5].apply(
        lambda x: " ".join([st.stem(word) for word in x.split()])
    )
    final_data_frame["text"] = final_data_frame["text"].apply(
        lambda x: " ".join([Word(word).lemmatize() for word in x.split()])
    )
    corpus = []
    for text in final_data_frame["text"]:
        corpus.append(text)
    final_data_frame.rename(columns={"class": "class_label"}, inplace=True)
    Class_Label = {"yes": 1, "no": 0}
    final_data_frame.class_label = [
        Class_Label[item] for item in final_data_frame.class_label
    ]
    final_data_frame.rename(columns={"class_label": "class"}, inplace=True)
    count_vectorizer = CountVectorizer()
    count_vectorized_data = count_vectorizer.fit_transform(corpus)
    tfidf_vectorizer = TfidfVectorizer()
    tfidf_vectorized_data = tfidf_vectorizer.fit_transform(corpus)
    vectorized_data = tfidf_vectorized_data
    X_train, X_test, Y_train, Y_test = train_test_split(
        vectorized_data, final_data_frame["class"], test_size=0.3, random_state=0
    )
    SVM = svm.SVC(probability=True, C=1.0,
                  kernel="linear", degree=3, gamma="auto")
    SVM.fit(X_train, Y_train)
    Naive = naive_bayes.MultinomialNB()
    Naive.fit(X_train, Y_train)
    logisticReg = linear_model.LogisticRegression(C=1.0)
    logisticReg.fit(X_train, Y_train)
    dtc = DecisionTreeClassifier(min_samples_split=7, random_state=252)
    dtc.fit(X_train, Y_train)
    neural_network = MLPClassifier(
        solver="lbfgs", alpha=1e-5, hidden_layer_sizes=(5, 2), random_state=1
    )
    neural_network.fit(X_train, Y_train)
    stats_SVM = report_results(SVM, X_test, Y_test)
    stats_Naive = report_results(Naive, X_test, Y_test)
    stats_logistic = report_results(logisticReg, X_test, Y_test)
    stats_dtc = report_results(dtc, X_test, Y_test)
    stats_neural = report_results(neural_network, X_test, Y_test)
    stats = []
    stats.append(stats_SVM)
    stats.append(stats_Naive)
    stats.append(stats_logistic)
```

```
    stats.append(stats_dtc)
    stats.append(stats_neural)
    return stats


# output_to_results("data/AnnotatedData3.csv")
```

2. Code for website

```
{% extends 'navbar.html' %}
{% block title  %}Analysis{% endblock %}
{% block content  %}
<script src="https://code.highcharts.com/highcharts.js"></script>
<script src="https://code.highcharts.com/modules/data.js"></script>
<script src="https://code.highcharts.com/modules/exporting.js"></script>
<script src="https://code.highcharts.com/modules/export-data.js"></script>
<script src="https://code.highcharts.com/highcharts.js"></script>
<script src="https://code.highcharts.com/modules/wordcloud.js"></script>

<div id="container" style="min-width: 310px; height: 400px; margin: 0
auto"></div>
<br>

<table id="datatable"></table>

<div id="word-cloud"></div>

    var lines = text.split(/[,\. ]+/g),
        data = Highcharts.reduce(lines, function (arr, word) {
            var obj = Highcharts.find(arr, function (obj) {
                return obj.name === word;
            });
            if (obj) {
                obj.weight += 1;
            } else {
                obj = {
                    name: word,
                    weight: 1
                };
                arr.push(obj);
            }
            return arr;
```

```
        }, []);

    Highcharts.chart('word-cloud', {
        series: [{
            type: 'wordcloud',
            data: data,
            name: 'Occurrences'
        }],
        title: {
            text: 'Wordcloud of Lorem Ipsum'
        }
    });
</script>

<script>
    $(document).ready(function () {
        Highcharts.chart('container', {
            data: {
                table: 'datatable'
            },
            chart: {
                type: 'column'
            },
            title: {
                text: 'Data extracted from a HTML table in the page'
            },
            yAxis: {
                allowDecimals: false,
                title: {
                    text: 'Units'
                }
            },
            tooltip: {
                formatter: function () {
                    return '<b>' + this.series.name + '</b><br/>' +
                        this.point.y + ' ' + this.point.name.toLowerCase();
                }
            }
        });
    });
</script>
{% endblock %}
```

Appendix 3: Schematic of your hardware

We did not require any specialized hardware for building the machine learning model and the website.

Appendix 4: Software/parts list

1. Python 3.6
2. Python Django Framework
3. Mongo DB
4. scikit-learn library for Python
5. Internet browser eg Google Chrome
6. A Python code editor

Appendix 5: Work distribution

1. Hasan Imran: in charge of building and testing the various text analytical models and preparing them to be incorporated in the website.
2. M Ammar Saleem: in charge of developing the website for the whole project from the front end to the back end and providing complete integration.
3. M Saad Salman: in charge of the data gathering and preprocessing task. From scraping the data, to storing it in a database and then applying preprocessing techniques on the data.

Appendix 6: Project timeline

- ✓ Project started on August 20th, 2018 and completed on May 10th, 2019.
- ✓ First major milestone was on December 2018 when we completed the proposal documentation and started working on the next phase of development.
- ✓ Second major milestone was on March 15th, 2019 when we finally got a good accuracy for our model.

References:

Books:

1. Speech and Language Processing (3rd ed. draft), Dan Jurafsky and James H. Martin.

Inspirations for code and designs:

1. Building a prediction model, https://www.kaggle.com/gpayen/building-a-prediction-model
2. Sentiment analysis, https://www.kaggle.com/laowingkin/amazon-fine-food-review-sentiment-analysis.
3. TEXT PREPROCESSING USING PYTHON, https://www.kaggle.com/shashanksai/text-preprocessing-using-python.

Papers:

1. Identifying Purchase Intentions by Extracting Information from Tweets, February 8, 2017, RADBOUD U NIVERSITY NIJMEGEN, BACHELOR 'S THESIS IN ARTIFICIAL INTELLIGENCE.
2. Tweetalyst: Using Twitter Data to Analyze Consumer Decision Process, The Berkeley Institute of Design.
3. The Impact of Social Network Marketing on Consumer Purchase Intention in Pakistan: Consumer Engagement as a Mediator, Asian Journal of Business and Accounting 10(1), 2017.
4. Using Twitter Data to Infer Personal Values of Japanese Consumers, 29th Pacific Asia Conference on Language, Information and Computation pages 480 – 487 Shanghai, China, October 30 - November 1, 2015, Copyright 2015 by Yinjun Hu and Yasuo Tanida.

Datasheets:

1. Amazon Fine Food Reviews, 500,000 food reviews from Amazon, Stanford Network Analysis Project, https://www.kaggle.com/snap/amazon-fine-food-reviews

Vendor:

None

Background sites:

1. https://www.kaggle.com/snap/amazon-fine-food-reviews

2. https://scikit-learn.org/stable/

Acknowledgements:

We would like to thank our Supervisor Miss Nida Sadaf Khan. Without her support and guidance, we would not have been able to complete the project. She has been a constant mentor through out our project and has always been there to help. She has never said no about meeting and discussing our project approach. From guiding us, to correcting us she has steered the project into the right direction.

We are grateful to her as she is the one who motivated us when we were stuck and constantly inspired us to do better due to which our project has developed extremely well.

We would also like to thank Sir Waseem Arain, who has co ordinated the Final Year Projects so effectively and efficiently and always strives for making things easier for students.

We would also like to thank Mr Ayaz Ahmed who has always remained available for any query regarding the administrative aspect of our Final Year Project. His dedication towards the IBA FCS Department is inspirational.

Lastly, we would like to thank IBA for giving us this opportunity to work on our final year project and always providing top of the art facilities for getting work done.