MISM 6213- Final Project Report

Northeastern University

Prof. Rajesh Jugulum

Project: Product Sentiment/Text Analysis using Tweets

Team Members: Raghnya Valluru, Sayantani Roy

Contents

Project Overview

Traditionally businesses have used multiple forums to collect customer feedback and also attract and engage with the customers to potentially win a repeat business. In contrary to brick and mortar presence with direct customer touch points in today's world we focus more on digital channels as the points of interactions. Most businesses feel the pressure to be omnichannel and omnipresent. The digital presence can be influential to build a brand or can even measure the success of a new product based on the customer reviews and feedback.

Text analysis helps businesses analyze huge quantities of text-based data in a scalable, consistent, and unbiased manner. Without the need for excessive resources, it analyses data and extracts valuable information, leaving companies free to action on those insights. Due to text analytics, businesses can easily study these texts which can be presented in the form of charts, surveys, and other such formats. On the basis of the results, companies can then make an informed decision about the product concerned and its performance.

Sentiment analysis refers to identifying as well as classifying the sentiments that are expressed in the text source. Tweets are often useful in generating a vast amount of sentiment data upon analysis. These data are useful in understanding the opinion of the people about a variety of topics.

Our project is to develop a machine learning sentiment analysis model using regression and classification models to understand the sentiments of users regarding apple products like iPhone, iPad and other google services using Python on Google Collab.

Problem formulation

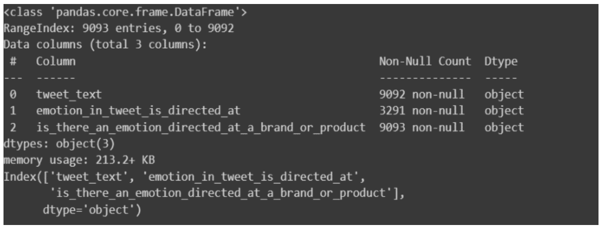
Understanding user's sentiments towards products and services on twitter. We want to analyze twitter data to derive insights about user sentiment using the tweets.

Problem statement- Design a Machine Learning Algorithm that can help us understand user's sentiments towards various products and services using twitter as a platform

Data Collection

We collected data from 'Twitter' containing tweets posted for various services and products along with the emotion contained in the tweet about Apple products and Google services. This data is publicly available (Please find appendix for the dataset attached)

The dataset has 9093 entries ,3 columns which become our critical data elements-



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9093 entries, 0 to 9092
Data columns (total 3 columns):
 #   Column                                              Non-Null Count  Dtype 
---  ------                                              --------------  ----- 
 0   tweet_text                                          9092 non-null   object
 1   emotion_in_tweet_is_directed_at                     3291 non-null   object
 2   is_there_an_emotion_directed_at_a_brand_or_product  9093 non-null   object
dtypes: object(3)
memory usage: 213.2+ KB
Index(['tweet_text', 'emotion_in_tweet_is_directed_at',
       'is_there_an_emotion_directed_at_a_brand_or_product'],
      dtype='object')
```

tweet_text- containing the message of the tweet regarding the product/brand
emotion_in_text_directed_at- name of product or brand
is_there_an_emotion_directed_at_a_brand_product-what kind of emotion does the tweet depict (negative, positive, neutral)

This data has been collected after applying sentiment analysis to the extracted twitter data. Twitter provides APIs for developers to extract desired data and further use for data modelling and uploaded on Google drive to use for analysis. This API will give us many fields like timestamp, source of tweet, location and other additional information about the tweet and user.

Our purpose, referring to the problem statement, is to focus only on the Sentiment of the user towards the product, service, or the brand. Therefore, we reduce the data elements to 3.

Data Quality Analysis

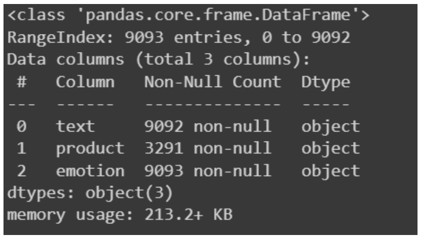
Data is of type object so to perform data quality checks or use for machine learning modelling, we need to transform this data into machine readable format.

Before we clean the data, we want to understand each column and check for its quality using the pandas library in python.

Dimensions of Data Quality

- **Timeliness:** We are not checking for daily or timely trends using time series analysis. Since it's a sentiment analysis, we only want to understand the sentiments of users using their posted tweet. We are not sure of the timeliness of the data on the collection point however, as this is a model project and not a particular product sentiment analysis based on launch date, this won't impact our analysis.
- **Conformity:** Our dataset contains text data of type object. We want to convert this to a machine-readable data type.

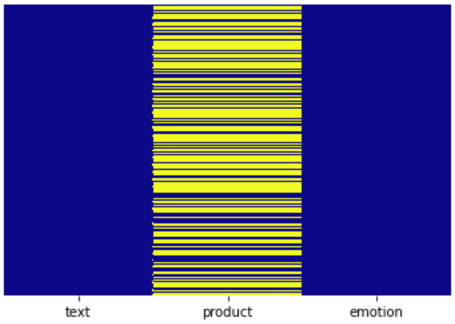
We first change the column names to a simpler name –

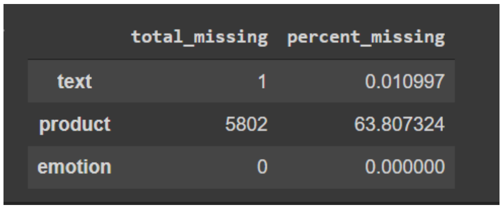


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9093 entries, 0 to 9092
Data columns (total 3 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   text     9092 non-null   object
 1   product  3291 non-null   object
 2   emotion  9093 non-null   object
dtypes: object(3)
memory usage: 213.2+ KB
```

- tweet_text - **text**
- emotion_in_tweet_is_directed_at – **product**
- is_there_an_emotion_directed_at_a_brand_or_product – **emotion**

- **Completeness-** We check for the missing values in our dataset as shown in the figure below. We see that the column product has many missing values and so we can ignore this column and reduce our critical data elements from 3 to 2





	total_missing	percent_missing
text	1	0.010997
product	5802	63.807324
emotion	0	0.000000

- **Validity :** Our final dataset contains two columns

Text & Emotion

We will refer to our problem statement again here to address the importance of our final critical data elements.

Our purpose is to identify emotion from the available text. We can further improve the model after collecting more data to identify specific types of emotion towards specific products.

For now, we are in a way mapping an emotion to the text using the best machine learning model to further automate predicting emotion based on the message on the tweet for products or brands to understand user sentiment.

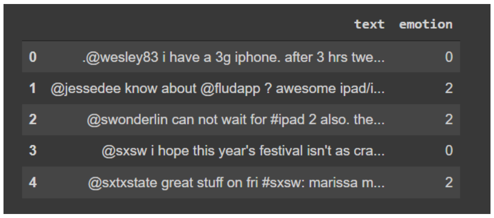
Data Preparation

Now that we have our final clean dataset, we want to prepare the data to use for further advanced analysis using machine learning in Python.

Python has multiple built in libraries that can help convert text or object data to a machine-readable language.

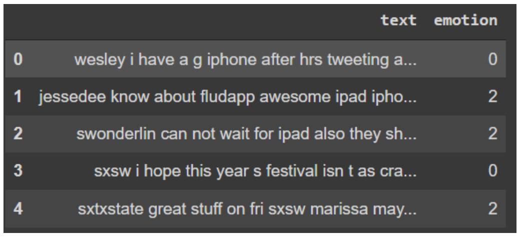
- **Convert emotion to numerical type:** We have mapped each type of emotion to numeric values

0='Negative emotion'
1= 'No emotion toward brand or product'
2= 'Positive emotion'

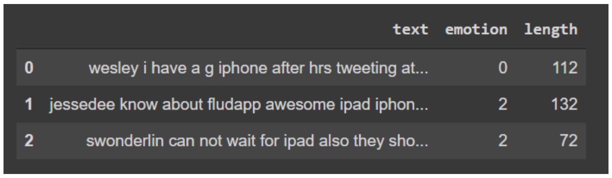


	text	emotion
0	.@wesley83 i have a 3g iphone. after 3 hrs twe...	0
1	@jessedee know about @fludapp ? awesome ipad/i...	2
2	@swonderlin can not wait for #ipad 2 also. the...	2
3	@sxsw i hope this year's festival isn't as cra...	0
4	@sxtxstate great stuff on fri #sxsw: marissa m...	2

- **Convert to lower case:** We have converted any text data to a lower-case using Lambda function in python pandas.
- **Removed hyperlinks, alphabetical characters, and multiple spaces from text:** We have removed special characters, hyperlinks using lambda function
- **Add a column length for better insights:** We have created a column called length to get the length of the values in the column text to get better insights on user's sentiments.

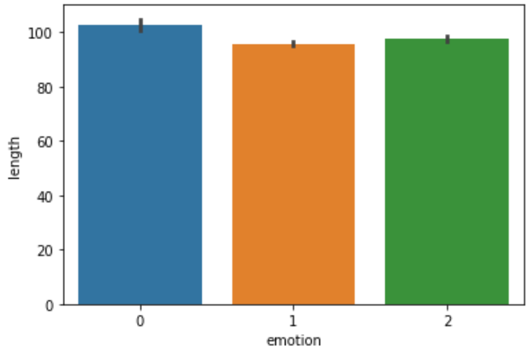


	text	emotion
0	wesley i have a g iphone after hrs tweeting a...	0
1	jessedee know about fludapp awesome ipad ipho...	2
2	swonderlin can not wait for ipad also they sh...	2
3	sxsw i hope this year s festival isn t as cra...	0
4	sxtxstate great stuff on fri sxsw marissa may...	2

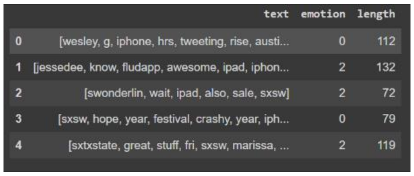


	text	emotion	length
0	wesley i have a g iphone after hrs tweeting at...	0	112
1	jessedee know about fludapp awesome ipad iphon...	2	132
2	swonderlin can not wait for ipad also they sho...	2	72

For example, if a text of a higher length contains a negative emotion, there is a higher correlation between text and emotion. It is however important to note that correlation does NOT imply causation. We cannot say that always texts of larger length are always showing negative emotion. It is simply useful to understand data better.

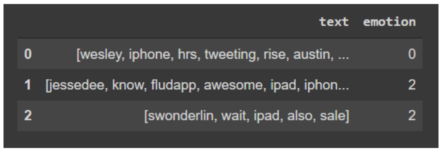


- **Tokenizing and Removing Stop words :** Using the nltk library, we are removing stop words and the resultant text column contains a list of all the words in the given tweet.



	text	emotion	length
0	[wesley, g, iphone, hrs, tweeting, rise, austi...	0	112
1	[jessedee, know, fludapp, awesome, ipad, iphon...	2	132
2	[swonderlin, wait, ipad, also, sale, sxsw]	2	72
3	[sxsw, hope, year, festival, crashy, year, iph...	0	79
4	[sxtxstate, great, stuff, fri, sxsw, marissa, ...	2	119

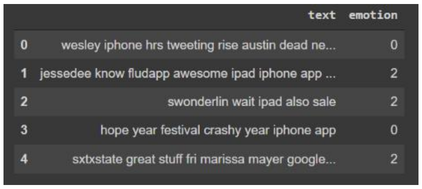
- **Removing one length words:** One length word like ‘a’ will not have a large effect so we have removed those.



	text	emotion
0	[wesley, iphone, hrs, tweeting, rise, austin, ...	0
1	[jessedee, know, fludapp, awesome, ipad, iphon...	2
2	[swonderlin, wait, ipad, also, sale]	2

Additional transformations:

- Join list of words to form a review
- Bag of Words Transformer
- Bow Transformer to vector representation of text

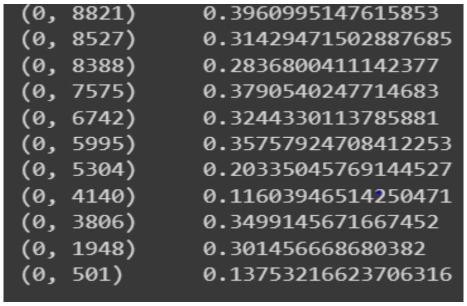


	text	emotion
0	wesley iphone hrs tweeting rise austin dead ne...	0
1	jessedee know fludapp awesome ipad iphone app ...	2
2	swonderlin wait ipad also sale	2
3	hope year festival crashy year iphone app	0
4	sxtxstate great stuff fri marissa mayer google...	2

Converting text object to weights to use in text mining-

- **Inverse document Frequency Transformer**

  tf-idf is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling.

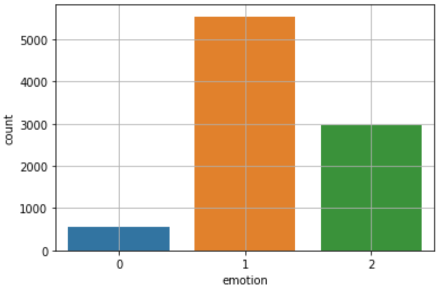


```
(0, 8821)     0.3960995147615853
(0, 8527)     0.31429471502887685
(0, 8388)     0.2836800411142377
(0, 7575)     0.3790540247714683
(0, 6742)     0.3244330113785881
(0, 5995)     0.35757924708412253
(0, 5304)     0.20335045769144527
(0, 4140)     0.11603946514250471
(0, 3806)     0.3499145671667452
(0, 1948)     0.301456668680382
(0, 501)      0.13753216623706316
```

Data Profiling and Exploratory Analysis

We perform some exploratory data analysis to create a profile for our data and understand the given data

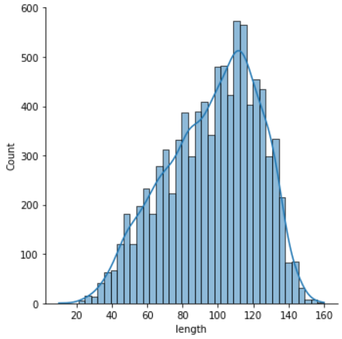
- **Trend in emotion column**



We want to understand the trend in emotions available in the dataset.

We see that most tweets have no emotion towards the brand or product. Positive emotion has a higher count than negative

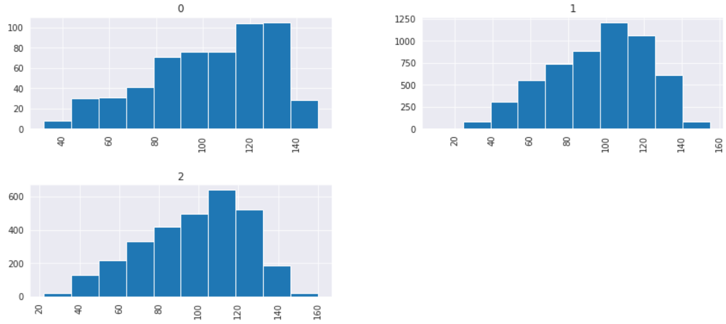
- **Distribution of Length column**



We see the distribution of the length of text counts and see that it follows a normal distribution with a mean of 115 and a minimum of 1 and a maximum of 160.

We choose to ignore the smaller lengths as they wouldn't have much information to contribute to our modeling

- **Distribution of length of text by emotion**



We then see distribution of each emotion with respect to length of text.

We see negative emotion shows a larger lengths

Followed by a consistent distribution for no emotion and positive emotion

Data Analytics

Now as we have a cleaned and wrangled dataset, we have performed the data analytics techniques to build the best model to predict the product or business sentiment analysis with the highest accuracy.

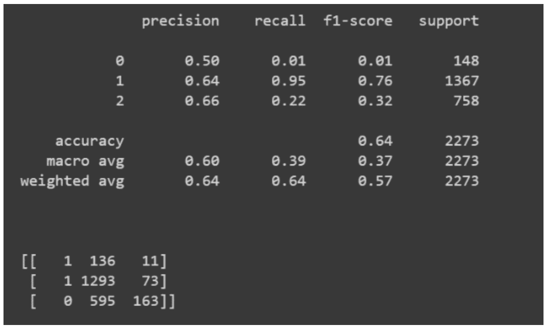
We have first split our dataset in 75-25 ratio where 25% of our dataset is test dataset and the 75% is our train dataset. Our predictor variable y is the 'emotion' data and x is the 'text' column.

- **Model 1: Multinomial Naïve Bayes**

We first performed multinomial Naïve Bayes algorithm as this could be used on both continuous and discrete data. It is also easy to implement using the probability factor. The Multinomial Naive Bayes can be accepted as the probabilistic approach to classifying documents in the case of acknowledging the frequency of a specified word in a text document and could be used as a tends to be **a baseline solution** for sentiment analysis task.

Model 1 Accuracy **64%**

- Negative Emotion '0' : Precision 50%
- No emotion toward brand or product '1': Precision 64%
- Positive emotion '2' : Precision 64%



```
              precision    recall  f1-score   support

           0       0.50      0.01      0.01       148
           1       0.64      0.95      0.76      1367
           2       0.66      0.22      0.32       758

    accuracy                           0.64      2273
   macro avg       0.60      0.39      0.37      2273
weighted avg       0.64      0.64      0.57      2273


[[   1  136   11]
 [   1 1293   73]
 [   0  595  163]]
```

Precision can be used as a measure of quality and recall as a measure of quantity. Higher precision means that an algorithm returns more relevant results than irrelevant ones, and high recall means that an algorithm returns most of the relevant results (whether or not irrelevant ones are also returned).

Based on our model, we need to improve the accuracy and precision score.

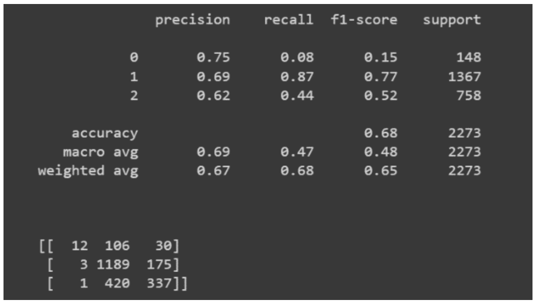
- **Model 2: Logistic Regression**

Logistic regression makes use of the sigmoid function which outputs a probability between 0 and 1. It is one of the basic and very useful classification model in python.

In general, there are two different types of classification models: **generative** models (Naive Bayes, Hidden Markov Models, etc.) and **discriminative** models (Logistic Regression, SVM, etc.). Ultimately, both models try to compute p(class|features), or p(y|x). The key difference is that a generative model tries to model the joint probability distribution p(x,y) first and then compute the conditional probability p(y|x) using **Baye's Theorem**, whereas a discriminative one directly models p(y|x).

Model 2 : accuracy **68%**

- Negative Emotion '0' : Precision 75%
- No emotion toward brand or product '1': Precision 69%
- Positive emotion '2' : Precision 62%



```
              precision    recall  f1-score   support

           0       0.75      0.08      0.15       148
           1       0.69      0.87      0.77      1367
           2       0.62      0.44      0.52       758

    accuracy                           0.68      2273
   macro avg       0.69      0.47      0.48      2273
weighted avg       0.67      0.68      0.65      2273



[[  12  106   30]
 [   3 1189  175]
 [   1  420  337]]
```

Based on our model we can see that we have managed to improve our model accuracy and precision compared to our previous model. However, we tried to conduct few more models to see if further improvement possible.

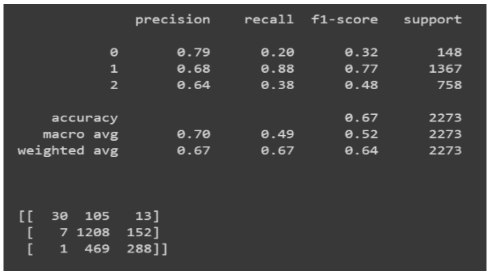
- **Model 3 : Random Forest Classifier**

Random forest is a commonly used machine learning algorithm which combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems.

Algorithmic simplicity makes it an attractive choice for text classification. In addition, its capability to handle the high dimensional data and high performance under imbalanced datasets are significant advantages over other machine learning models

Model 3 : accuracy **67%**

- Negative Emotion '0' : Precision 79%
- No emotion toward brand or product '1': Precision 68%
- Positive emotion '2' : Precision 64%



```
              precision    recall  f1-score   support

           0       0.79      0.20      0.32       148
           1       0.68      0.88      0.77      1367
           2       0.64      0.38      0.48       758

    accuracy                           0.67      2273
   macro avg       0.70      0.49      0.52      2273
weighted avg       0.67      0.67      0.64      2273



[[  30  105   13]
 [   7 1208  152]
 [   1  469  288]]
```

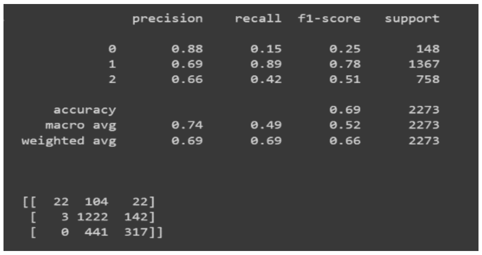
Based on our data, Random forest has abled to improve the precision score however, the model accuracy score has not improved.

- **Model 4: Support Vector Machine (SVM-Classifier)**

Support vector machines is an algorithm that determines the best decision boundary between vectors that belong to a given group (or category) and vectors that do not belong to it. It can be applied to any kind of vectors which encode any kind of data. Using SVM classifiers for text classification tasks might be a really good idea, especially if the training data available is not much (~ a couple of thousand tagged samples).

Model 4 : accuracy **69%**

- Negative Emotion '0' : Precision 88%
- No emotion toward brand or product '1': Precision 69%
- Positive emotion '2' : Precision 66%



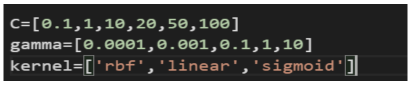
```
              precision    recall  f1-score   support

           0       0.88      0.15      0.25       148
           1       0.69      0.89      0.78      1367
           2       0.66      0.42      0.51       758

    accuracy                           0.69      2273
   macro avg       0.74      0.49      0.52      2273
weighted avg       0.69      0.69      0.66      2273


[[  22  104   22]
 [   3 1222  142]
 [   0  441  317]]
```

This is so far the best accuracy score along with the precision score. So, we can choose SVM classifier for our final model.

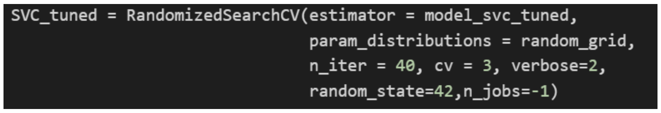
We also decided to tune our dataset for further accuracy.

Tuning the best model- SVM Classifier :

We used 3-fold grid search cv with below specifications to tune our model.



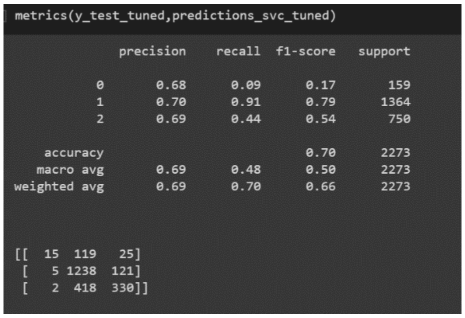
```
C=[0.1,1,10,20,50,100]
gamma=[0.0001,0.001,0.1,1,10]
kernel=['rbf','linear','sigmoid']
```



```
SVC_tuned = RandomizedSearchCV(estimator = model_svc_tuned,
                               param_distributions = random_grid,
                               n_iter = 40, cv = 3, verbose=2,
                               random_state=42,n_jobs=-1)
```

Tuned SVM Model : accuracy 70%

- Negative Emotion '0' : Precision 68%
- No emotion toward brand or product '1': Precision 70%
- Positive emotion '2' : Precision 69%



```
metrics(y_test_tuned,predictions_svc_tuned)

              precision    recall  f1-score   support

           0       0.68      0.09      0.17       159
           1       0.70      0.91      0.79      1364
           2       0.69      0.44      0.54       750

    accuracy                           0.70      2273
   macro avg       0.69      0.48      0.50      2273
weighted avg       0.69      0.70      0.66      2273


[[  15  119   25]
 [   5 1238  121]
 [   2  418  330]]
```

Though the precision score has dropped but overall accuracy has improved to 70% after tuning the dataset.

Conclusion

With our analysis we could say that with SVM model we could analyze the text sentiment with 70% accuracy.

Project Leaning: Sentiment analysis is a highly effective tool for a business to not only take a look at the overall brand perception, but also evaluate customer attitudes and emotions towards a specific product line or service. This data-driven approach can help the business better understand the customers and detect subtle shifts in their opinions in order to meet changing demand.

Our model could help the business to understand the positive or negative sentiment analysis for any product or service. This could also be useful to understand customer feedback for a new product or service launch

Limitations: We are not aware of the date or year of the data and hence timeliness of the data and implication is an issue however the model is relevant for any business. We also have a limited number of entries (approx.

9000 data) and hence model accuracy is low. However, on business implication this would not be a challenge and hence the model accuracy could be further improved.

References

[1] *Sentiment Classification with Logistic Regression—Analyzing Yelp Reviews*. [online] Available at: <https://towardsdatascience.com/sentiment-classification-with-logistic-regression-analyzing-yelp-reviews-3981678c3b44> [Accessed 27 June 2022].

[2] *Text Analytics and its importance - Aryng's Blog*. [online] Available at: <https://aryng.com/blog/blogmain/text-analytics-and-its-importance/> [Accessed 27 June 2022].

[3] 2022. [online] Available at: <https://monkeylearn.com/text-analysis/> [Accessed 27 June 2022].

[4] *A Guide: Text Analysis, Text Analytics & Text Mining*. [online] Available at: <https://towardsdatascience.com/a-guide-text-analysis-text-analytics-text-mining-f62df7b78747> [Accessed 27 June 2022].

[5] Analytics Vidhya. 2022. *Twitter Sentiment Analysis | Implement Twitter Sentiment Analysis Model*. [online] Available at: <https://www.analyticsvidhya.com/blog/2021/06/twitter-sentiment-analysis-a-nlp-use-case-for-beginners/#:~:text=Sentiment%20analysis%20refers%20to%20identifying,about%20a%20variety%20of%20topics.> [Accessed 27 June 2022].

[6] Developer.twitter.com. 2022. *Twitter API Documentation*. [online] Available at: <https://developer.twitter.com/en/docs/twitter-api> [Accessed 27 June 2022]

[7] [online] Available at: <https://monkeylearn.com/text-classification-support-vector-machines-svm/#:~:text=Support%20vector%20machines%20is%20an,encode%20any%20kind%20of%20data> [Accessed 27 June 2022].

Appendix

Data source:

final_data.csv

Python code:

Project_Sentiment_Analysis.ipynb