ASSIGNMENT REPORT

OF

DATA MINING AND DATA WAREHOUSING

LABORATORY

CSPC 328

SENTIMENT ANALYSIS OF TWITTER DATA

SUBMITTED BY:

1) Kaushiki Taru (18103053)

2) Kumari Soni (18103054)

3) Lakshya Toshniwal (18103056)

4) Nikhil Chachan (18103062)

5) Prince Jaiswal (18103072)

SUBMITTED TO:

Dr. Nonita Sharma

Assistant Professor

Dept.: CSE

TABLE OF CONTENTS

1. TITLE:

The title of our project is:

SENTIMENT ANALYSIS OF TWITTER DATA

2. ABSTRACT:

Sentiment Analysis is the method of identifying and categorizing opinions from a bit of text and determining whether or not the writer's angle towards a specific topic is positive, negative, or neutral. It is widely used nowadays as everything is becoming digital. Thus, Sentiment analysis is considered a field of interests to most of the brands in today's time.

The objective of this project is to analyze the sentiments of people based on the emotion recorded in their tweets. In exchange, we can note that the positive reviews lead to rise while the negative ones may hamper the worth or can even be treated as analisi element for improvement.

In this project, we follow a certain set of steps for extracting the result of Sentiment Analysis.

The steps involved are as follows:

- Tokenization of tweets.
- Cleaning the data.
- Remove special characters and stop words.
- Classification using different algorithms.
- Comparative Analysis of Different Classifiers

3. KEYWORDS:

- Classifiers
- Natural Language Processing (NLP)
- Information Gain
- Gini Index
- Naive Bayes
- KNN
- Random Forest
- Gradient Boost
- Visualization
- Confusion Matrix

4. INTRODUCTION:

4.1 REAL LIFE APPLICATION

In the current scenario where the whole world is on the internet, it has become primarily most important for us to utilize this hub of information on social networking sites for analysis to make our brand more famous and sell our product according to the need of the respective person at the best suited time.

Sentiment Analysis is widely being used nowadays in different areas of work like-

- **Government sector**s- The government has to be active inorder to serve its citizens better by utilizing the data as a resource to make its public services powerful
- **Call centres**- as they needed to induce a way on what the tipping purpose of a definite variety of negative interactions was, that was inflicting client attrition.

4.2 DATASET

The dataset is collected from Link and consists of 3085 tuples and 3 attributes/variables namely-

- **Key**- unique ID
- **Tweet**- consisting of special words, characters, numbers
- **Emotion**- output variable which can have values like “happy”, “sad”, “disgust”, etc.

  For some tuples, the Emotion value is “nocode”, and for some, it is “non-relevant” which would be removed in the data cleaning phase as those tuples are of no use in predicting the value of the output variable.

4.3 CLASSIFICATION

Classification is a method of categorization of data into different categories/ classes. In layman language, we can say that it is a method of separation of objects or placing of objects into classes.

It is a supervised learning method, i.e. the output label / container is already defined. It can be performed on both structured and unstructured data.

Classification is widely used nowadays- to classify an email is spam or genuine, in sentiment analysis, disease detection, and in many other areas of study.

The comparison of the classification methods as discussed in section 7 is done on the basis of their accuracy, speed of construction, scalability, interpretability and how the algorithm handles noise and missing values .

4.4 STRUCTURE OF PROJECT REPORT

Section 5: Detailed information about the methods involved in the classification technique. For eg: Information gain, Gini Index, Gradient Boost, etc.

Section 6: Brief description about the implementation of the project.

Section 7: Deals with the evaluation of the experimental results. A comparison between all the classification methods is done so as to get the best model which best fits / classifies our dataset.

Comparison would be done using visualizations (graphs) and also on the basis of different performance metrics such as accuracy, precision, recall, F1 score, etc.

Section 8: Conclusion and future scope are discussed.

5. METHODS OF CLASSIFICATION:

5.1 INFORMATION GAIN

Information gain is a measure of how good an attribute is, for predicting the training data. If we have randomness in our data, it means that we have more information gain.

Here, another term comes into play- "**Entropy**". It is the measure of randomness and entropy leads to information gain. Thus, formula for finding both are same and is given by the following formula-

$$I_H = -\sum_{j=1}^{c} p_j log_2(p_j)$$

For instance, suppose you meet a person. For the first case, let that person be your best friend whom you know very well; so information gain will be very minute.

But, in the second case, let us suppose that the person we met is a completely unknown person. In this case, the information gain would be more as there will be more information exchange and you will be gaining information about a person who you didn't know. Clearly by this example, we can conclude that uncertainty is important and thus, in the dataset, we select those attributes which have the maximum information gain.

5.2 GINI INDEX

Gini index is a measure to quantify the amount of imperfectness of the split. In other words, we may say that it is used to measure the resource inequality or impurity of the split. Thus, lower the gini index, better is the attribute to be considered having more weight.

Gini Index is calculated using the following equation-

$$I_G = 1 - \sum_{j=1}^{c} p_j^2$$

If G=1, then it is perfectly imperfect or 0% perfect.
If G=0, then it is pure or 0% imperfect.
If G=0.5, then we have equal distribution.

In simple language, it determines how oftenly a randomly selected variable is incorrectly detected.

5.3 NAIVE BAYES

Naive Bayes classifier is based on the Bayes Theorem. It states that if we have two events, say A and B, then, the probability of occurrence of A given that B has already occurred is written as P(A|B) and is given by the following equation-

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

Here, P(A), P(B) and P(B|A) are the probability of occurrence of event A, the probability of occurrence of event B and the probability of occurrence of B given that A has already occurred respectively.

Main goal: To determine the likelihood of an attribute, provided certain observed characteristics.

5.4 K-NEAREST NEIGHBOURS (KNN)

KNN is one of the easiest yet important classification algorithms used in recent years for machine learning. This algorithm is based on supervised learning methods.

In KNN we are given initial assumptions based on which the steps of algorithms are processed. It is a lazy learning algorithm as for different initial assumptions different outputs of the algorithms can be observed and the model is not actually constructed.

5.5 RANDOM FOREST ALGORITHM

Random Forest is an algorithmic AI program which is supervised. This algorithm can be utilized for each regression and classification problem. They are also called the Black Box Models because one cannot see what is going on inside.

It is a classifier which contains a number of decision trees trained on different subsets from the training dataset. Each tree is prepared in an exceptional manner. At each point, a split is made in the information and added to the tree thinking about just a fixed number of features. Thus, the more the quantity of trees in the forest, the higher is the accuracy.

5.6 GRADIENT BOOST ALGORITHM

It is one among the ensemble techniques. It builds trees and corrects the errors created before. So, the next model is even better than the previous one.

Important terms / parameters used in this algorithm are:

- Number of trees
- Depth of trees
- Learning rate

Major disadvantage of this method is that it is time consuming because of the fact that the trees are built sequentially.

But still we use this algorithm widely because the time is worth giving, as the accuracy is much higher than the normal decision tree.

6. EXPERIMENTATION:

The Tweet data contains unstructured data with layman's language which needs to be converted to structured form for analysis. We can do this using Sentiment Analysis.

It involves 5 major steps as given below-

- **Tokenization:** dividing a statement into a set of words
- **Data Cleaning**: removing the special characters. For eg- !, @, #
- **Removing the stop words**: stop words are those english words that are widely used in any sentence. Removing them as they don't add any value to the analytic result. For eg- the, a,an, is
- **Apply any supervised learning algorithm for Classification:** use any classification technique out of the six mentioned in section 5.
- **Training the model with BOW / Lexicons and testing it on the analysing statement:** More the accuracy, better will be the classification.

Implementation Process:

- First we **imported the dataset** and observed the nature of the dataset.
  We saw that the Tweet attribute consists of different special characters like @, #, etc , different stop words, URLs, punctuations.

```
# Print the top 5 tuples of the dataset using the head() command
#
tweetdata.head()
```

	Key	Tweet	Emotion
0	611857364396965889	@aandraous @britishmuseum @AndrewsAntonio Merc...	nocode
1	614484565059596288	Dorian Gray with Rainbow Scarf #LoveWins (from...	happy
2	614746522043973632	@SelectShowcase @Tate_StIves ... Replace with ...	happy
3	614877582664835073	@Sofabsports thank you for following me back. ...	happy
4	611932373039644672	@britishmuseum @TudorHistory What a beautiful ...	happy

❖ Then, **explored it.**

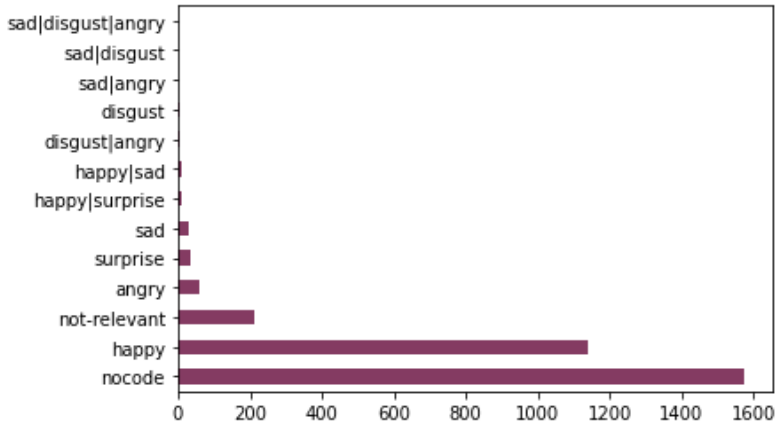
```
# Count the values under the Emotion Attribute
#
tweetdata['Emotion'].value_counts()
```

```
nocode               1572
happy                1137
not-relevant          214
angry                  57
surprise               35
sad                    32
happy|surprise         11
happy|sad               9
disgust|angry           7
disgust                 6
sad|angry               2
sad|disgust             2
sad|disgust|angry       1
Name: Emotion, dtype: int64
```

Visualization makes this much easier, so we made the bar graph to demonstrate the above output-

```
# Above obtained count values can be better visualised as:
#
tweetdata['Emotion'].value_counts().plot(kind='barh',color="#843b62")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f21d6563410>
```



We see that we have 13 different output labels, out of which “nocode” and “not-relevant” are of not much importance. Thus, we removed the tuples containing those output variables. Also, we were told to select only the first value of the output variable if it contains more than a single value.

Thus, we worked on this step by step and got the final values like following-

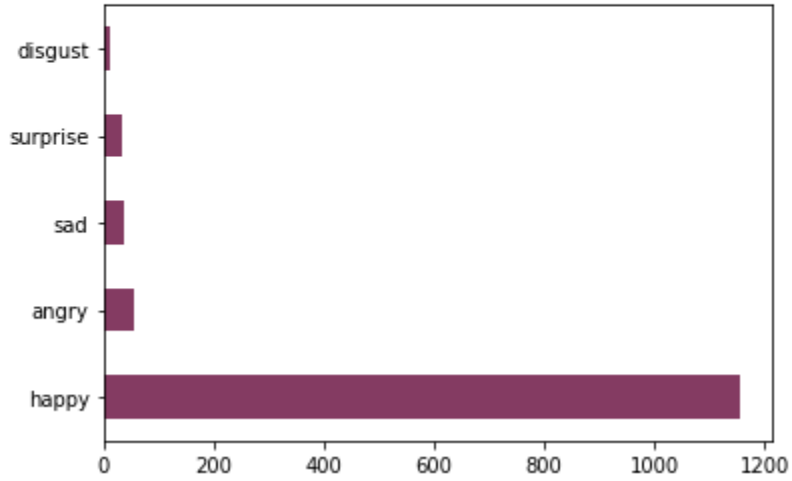
```
# Recheck the values for Emotion
#
tweetdata['Emotion'].value_counts()
```

```
happy       1157
angry         57
sad           37
surprise      35
disgust       13
Name: Emotion, dtype: int64
```

Again, visualizing this output using a bar graph-

```
# Plot the graph to visualise the new distribution of values
#
tweetdata['Emotion'].value_counts().plot(kind='barh',color="#843b62")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f21d645b210>
```



- The next thing we did was **data preprocessing**. So, we removed the tuples containing hyperlinks, stop words, numbers and special characters in the data preprocessing phases and got the output as shown below-

```
# Function call for preprocessing
#
tweetdata["Tweet"] = tweetdata.Tweet.apply(preprocess_tweet_text)
tweetdata.head()
```

	Tweet	Emotion
1	dorian gray rainbow scarf lovewins britishmuseum	happy
2	selectshowcase tatestives replace wish artist ...	happy
3	sofabsports thank following back great hear di...	happy
4	britishmuseum tudorhistory beautiful jewel por...	happy
5	nationalgallery thepoldarkian always loved pai...	happy

- Then the dataset can be split into two parts; first- training ; second- testing in 4:1 ratio.

- And lastly, for each classifier- Information Gain, Gini Index, Naive Bayes, KNN, Random Forest and Gradient Boost; we performed feature extraction using BOW and TF-IDF both and found the confusion matrix, and other performance metrics like following-
  - **Accuracy**: (TP+TN)/ (P+N)
  - **Precision**: TP/(TP+FP)
  - **Recall**: TP/(TP+FN)
  - **F1 score**: 2*P*R/(P+R)

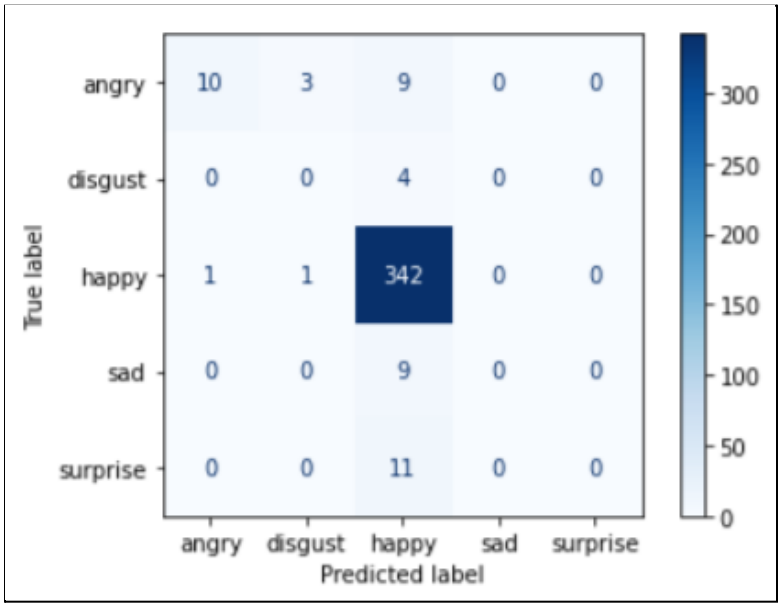
		Predicted class	
		P	*N*
Actual Class	*P*	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

7. RESULTS AND DISCUSSIONS

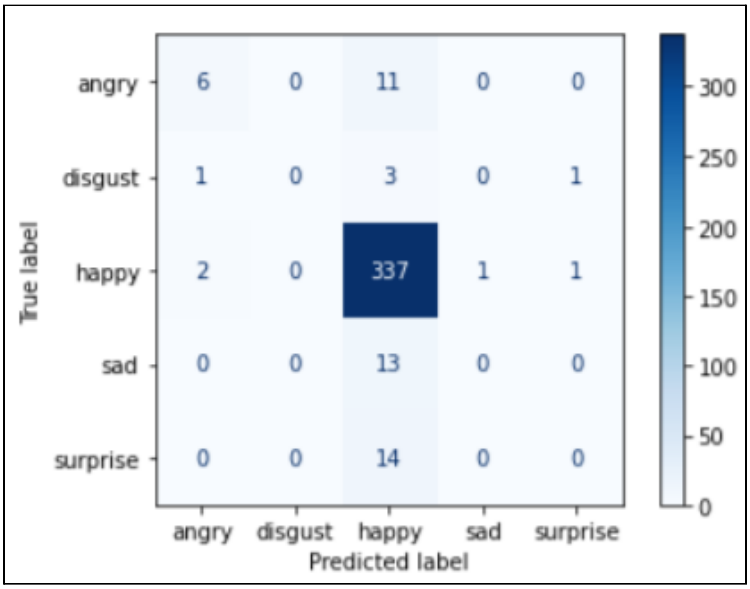
7.1 CONFUSION MATRIX

A) INFORMATION GAIN

i) BOW

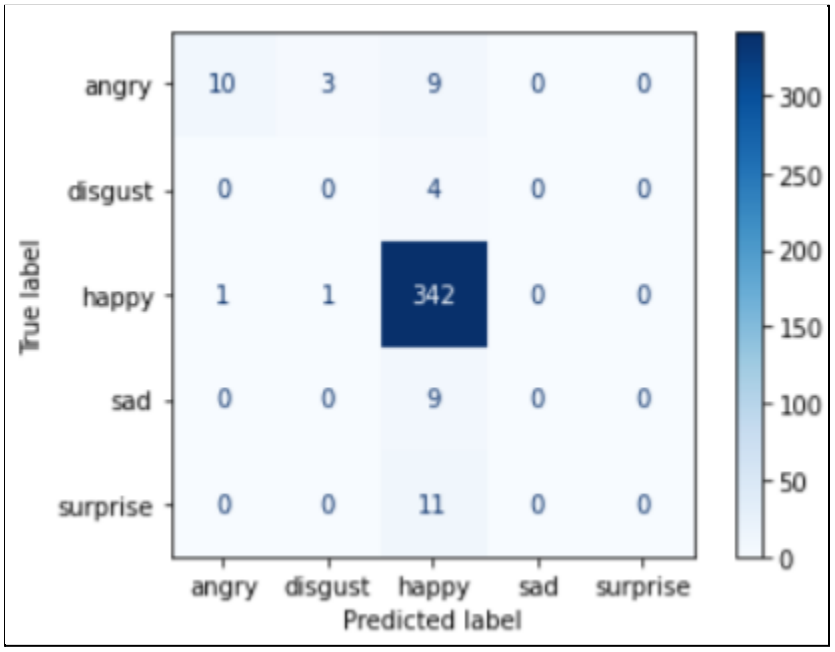


ii) TF-IDF



B) GINI INDEX

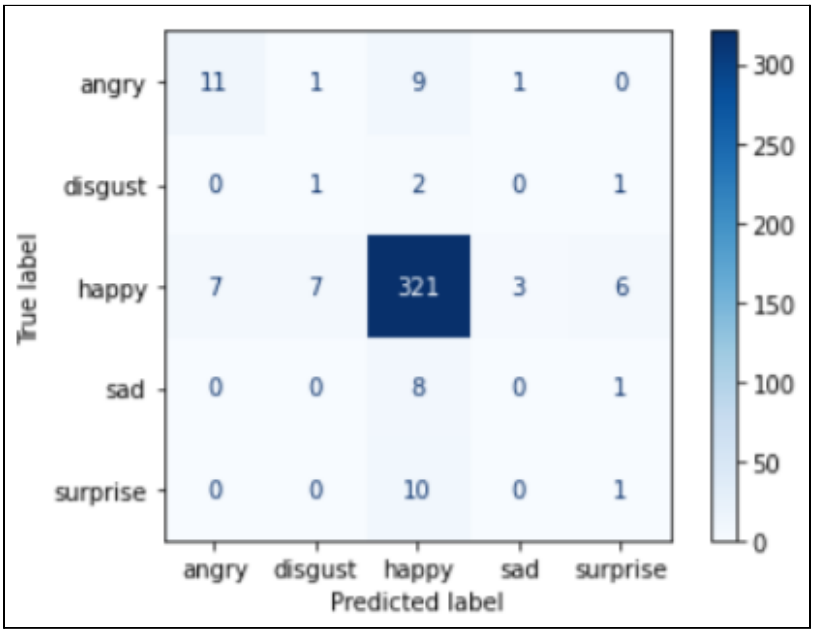
i) BOW



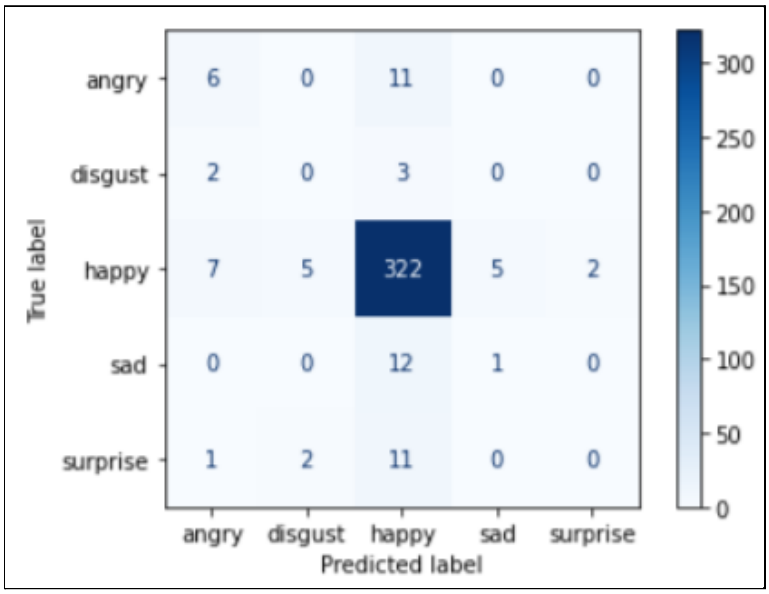
ii) TF-IDF

C) NAIVE BAYES

i) BOW



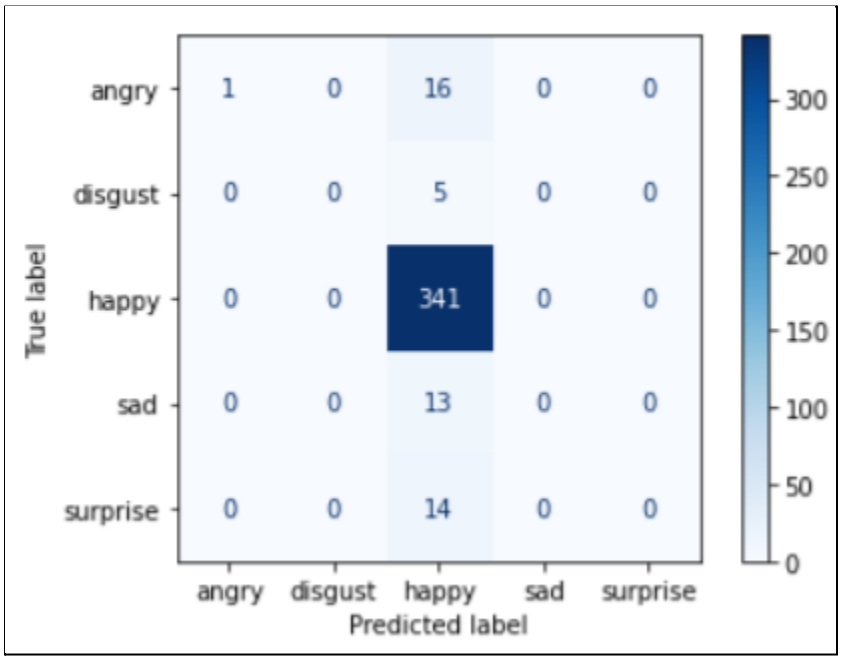
ii) TF-IDF



D) KNN

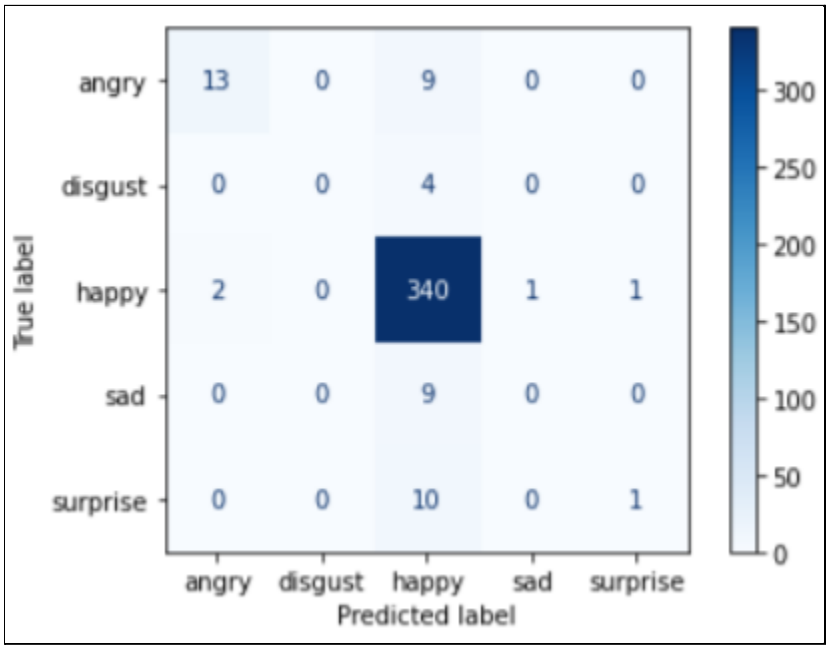
i) BOW

ii) TF-IDF

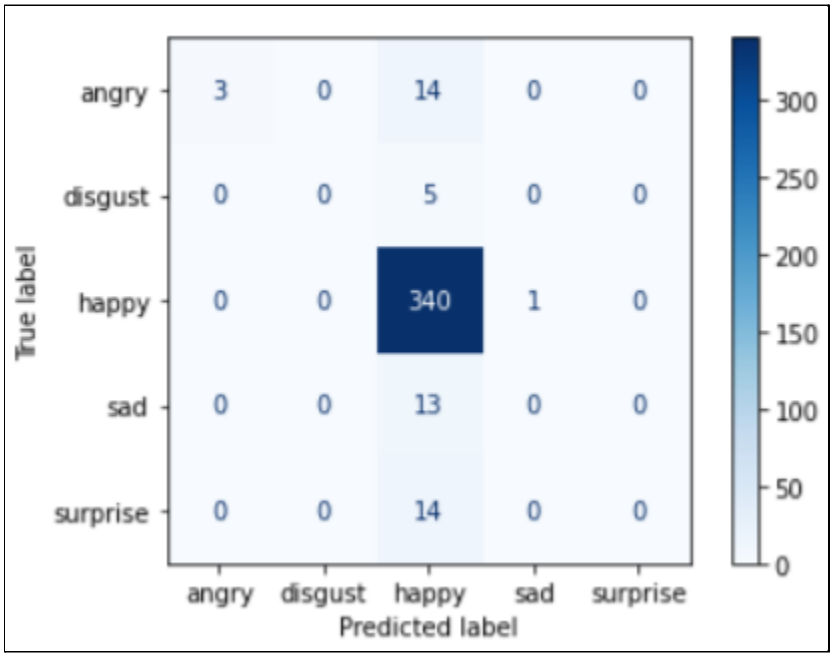


E) RANDOM FOREST

i) BOW



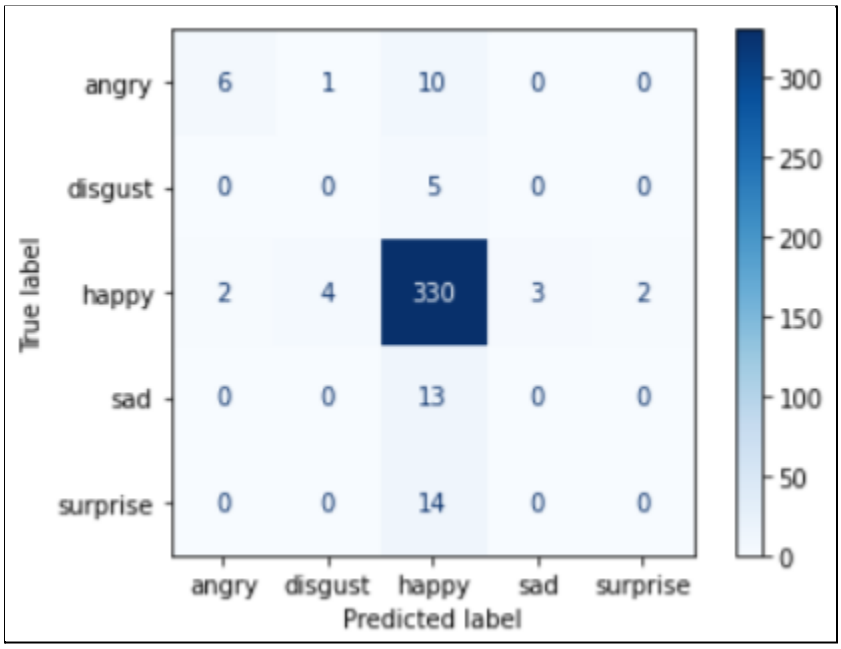
ii) TF-IDF



F) GRADIENT BOOST

i) BOW

ii) TF-IDF

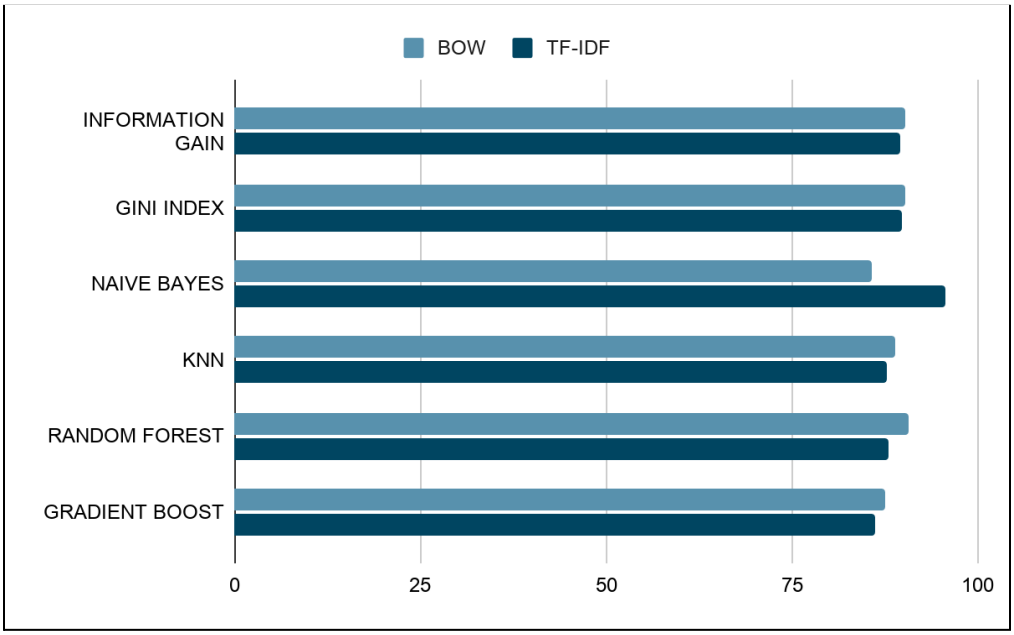


7.2 PERFORMANCE METRICS COMPARISON

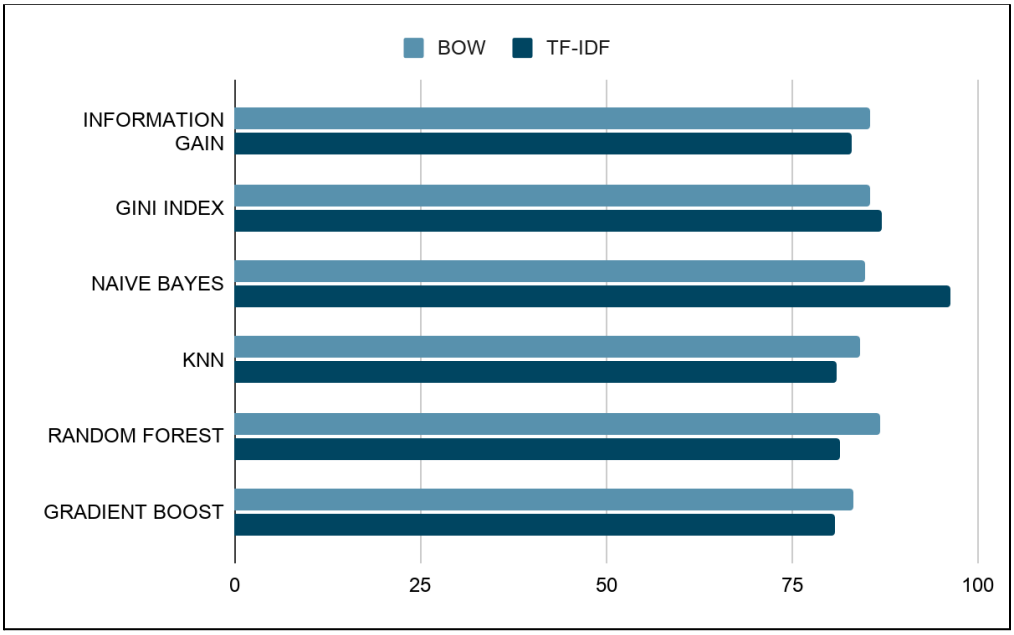
Sr. no	Classifier	Feature	Accuracy	Precision	Recall	F1 score
1	Information Gain	BOW	90.256	85.571	90.256	87.33
		TF-IDF	89.487	83.107	89.487	85.66
2	Gini Index	BOW	90.256	85.571	90.256	87.33
		TF-IDF	89.744	86.986	89.744	86.238
3	Naive Bayes	BOW	85.641	84.771	85.641	85.139
		TF-IDF	95.641	96.336	95.641	95.879
4	KNN	BOW	88.974	84.046	88.974	84.37
		TF-IDF	87.692	81.006	87.692	82.171
5	Random Forest	BOW	90.769	86.917	90.769	88.168
		TF-IDF	87.949	81.375	87.949	83.091
6	Gradient Boost	BOW	87.436	83.266	87.436	85.044
		TF-IDF	86.154	80.833	86.154	83.029

7.3 GRAPHICAL VISUALIZATIONS

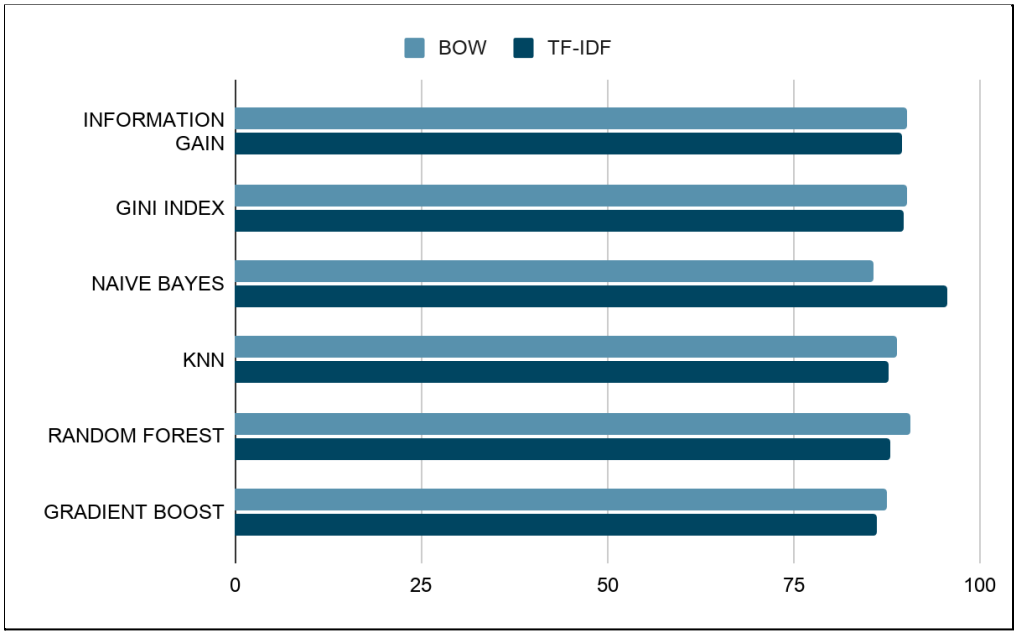
A) ACCURACY OF EACH CLASSIFIER



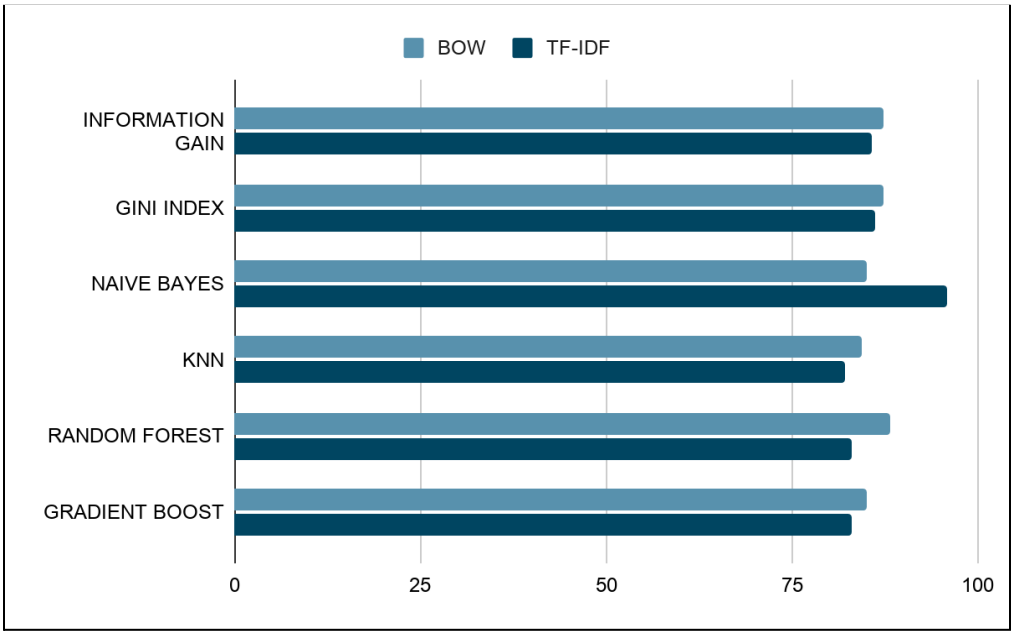
B) PRECISION OF EACH CLASSIFIER



C) RECALL OF EACH CLASSIFIER



D) F1- SCORE OF EACH CLASSIFIER



8. CONCLUSION AND FUTURE SCOPE:

8.1 CONCLUSION

The chart above clearly shows the comparison of Accuracy of all the Classifiers for both feature extraction methods separately. Although all of them have pretty good predictive accuracy, yet for the Sentiment Annotation dataset, Naive Bayes Classifier gives better results, especially when we follow the TF-IDF feature extraction method.

This varying accuracy output for Naive Bayes Classifier, i.e., 95.641 % (highest among all) is due to the fact that Naive Bayes performs better than other classifiers in case of categorical input values when compared to numeric data. The Sentiment Annotation Twitter dataset has categorical value in the form of words recorded, when converted to structured form.

Thus, a sentiment analysis model driven by the given dataset can be constructed by training with Naive Bayes Classification.

8.2 FUTURE SCOPE

In the future, we can work on exploring new technologies having better accuracy rates than our current model.

Link to Code
(Google Collab Notebook)