Data Mining

1st Abdullah Naveed
Computer Science

2nd Karan Kumar
Computer Science

3rd Ahmer Jamil
Computer Science

4th Esha Fatima
Computer Science

***Abstract*—This phase of the project deals with relevant mining techniques applied to our transformed dataset so as to gain relevant trends and information from it. Majorly, the paper focuses on key techniques such as logistic regression and Neural Networks.**

***Index Terms*—Neural Networks, regression, classification**

I. Introduction

Like any social media platform, Twitter has issues related to the authenticity of tweets broadcasted over the platform. The stakes are especially raised when the social media platform in question is considered by many as a mature platform for authentic information, announcements, news, and comments from renowned figures and authorities. In such instances, the platform's usability is severely questioned if a significant proportion of tweets are fake.

A. Corpora of tweets

For the analysis of the issue at hand, fake tweets vs. real tweets, we use the tweets' dataset. The dataset consists of flagged entries of tweets, the users' profile statistics, the originating location, and the tweet entry and whether it is fake or real.

B. Methodology

We split out our work on the corpora of tweets into two major phases:

1) **Exploratory Data Analysis -** In this phase, we analyze the data under various lenses and graphical means to draw out any significant data trends in the dataset. Additionally, the data is pre-processed to make it apt and consistent for analysis.
2) **Classification Model for the Tweets (real or fake)** - During this phase, we train various combinations of classifiers to conclusively derive an appropriate model that best suits the data attributes. The appropriateness of each model tried is gauged from its accuracy, precision, recall, and F1 measure.

C. Research Questions

1) **Research Question - 1 :** Are certain user profiles seen to be more prevalent in sharing fake tweets?
2) **Research Question - 2 :** What could be the major characteristics that can be used to determine if the tweet is real or fake?
3) **Research Question - 3 :** Can we train a model that can detect if the tweet is fake based on its user profile?
4) **Research Question - 4 :** Are there any frequent patterns that can be undermined from our dataset?

Identify applicable funding agency here. If none, delete this.

II. Understanding User Profile

Before going into the details of classification or clustering problems to predict whether a tweet is real or fake, we analyze user profile features and tweets from different aspects. Additionally, to answer our **Research Question -1**, performed extensive Exploratory Data Analysis (EDA) to investigate and summarize the main characteristics of the dataset. In this section, we will present an overview of the major patterns found in the dataset.

A. Assessing the number of real and fake tweets in the context of account followers

The Twitter accounts can be largely split into bins based on the number of followers and can be assessed as to the amount of real or fake tweets each follower bin contributes towards.

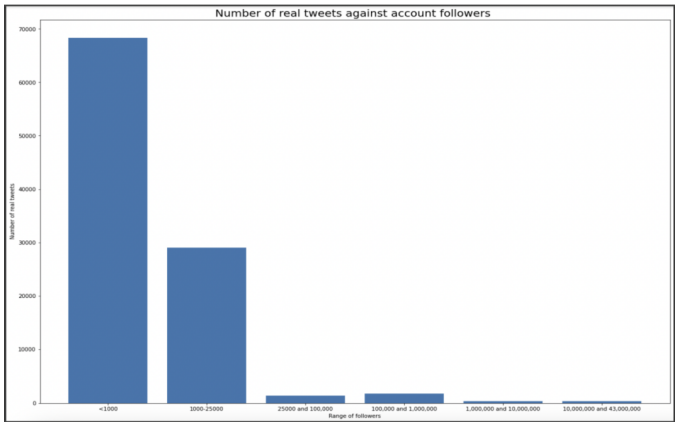


Fig. 1. Assessing the number of real and fake tweets

The graphs (in Fig. 1 and 2) depict that as the number of followers increases, the number of tweets decreases for both real and fake tweets.

B. Assessing the number of real and fake tweets in the context of the account following

The Twitter accounts can be largely split into bins based on the number of followers and can be assessed as to the amount of real or fake tweets each following bin contributes towards.

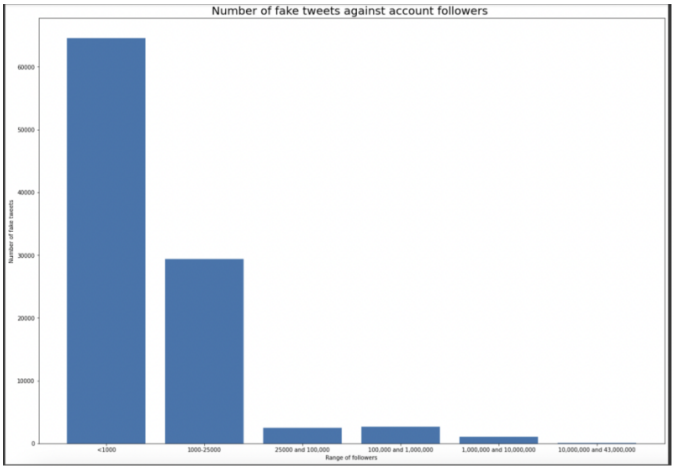


Fig. 2. Number of Fake Tweets against account followers

For both real, fake, and total tweets, the graphs depict that the number of tweets decreases as the number of the following increases.

C. Analysing average status count for real and fake tweets in the context of account following

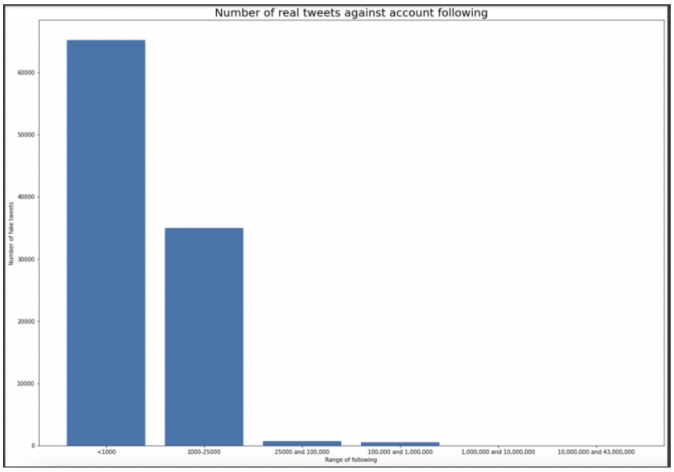


Fig. 3. Number of Real Tweets against account following

When comparing both graphs with each other, users following less than 25,000 accounts are more likely to share real news, whereas users following more than these followers are likely to share more fake news(Fig. 3 and 4).

D. Assessing tweet composition given the account verification status, account background tile, and profile usage of background images

The pie chart (in Fig. 5) shows the distribution of verified and non-verified accounts with their labels. It shows that the majority of the accounts are non-verified, with almost the same proportions of real and fake tweets. The verified accounts only account for 4.4 percent of the total tweets but there is a greater

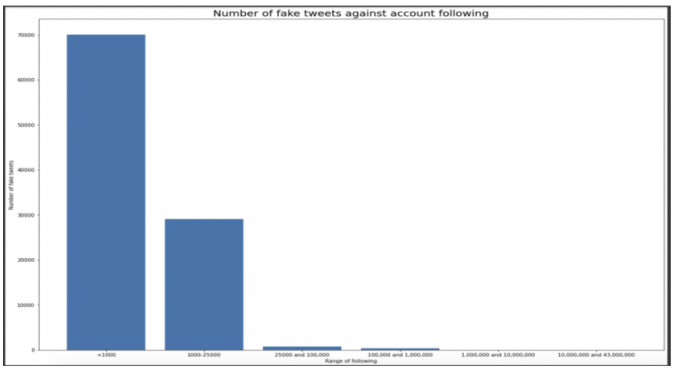


Fig. 4. Number of Fake Tweets against account following

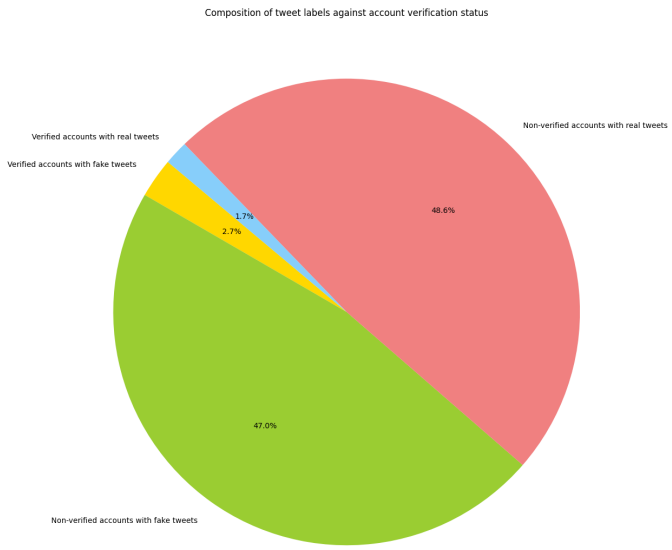


Fig. 5. Comparison of tweet labels

percentage of fake news (61.36 percent) among the verified users than real news.

This pie chart (in Fig. 6) shows that there is a higher percentage of users without a tile as a background image (54.6 percent) than users with a tile as a background image. However, among the users with a tile as a background image, 14 percent are likely to share real tweets, while the rest are more likely to share fake tweets. Users without a tile as a background image (these may include users who do not have a background image) 80 percent are likely to share real news. Hence, users without a tile as a background image have a higher possibility of sharing real news than fake news.

This pie chart (in Fig. 7) shows that there is a higher percentage of profiles with a background image than there are without a background image. Among the profiles with an image, there is an 80 percent likelihood of sharing real news, whereas, with profiles without an image, there is an 11 percent likelihood of sharing real news.

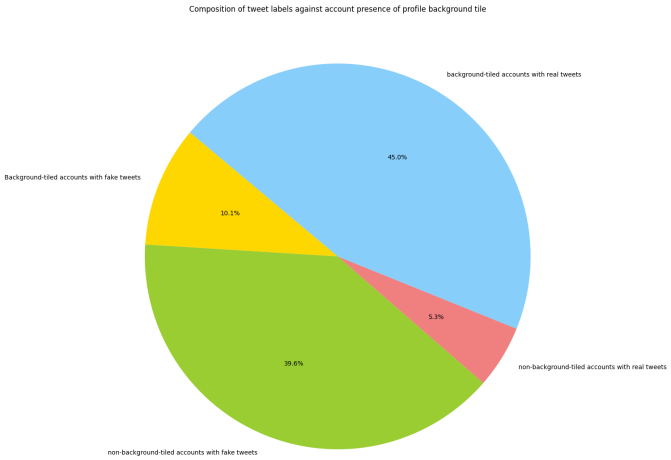


Fig. 6. Comparison of tweet labels

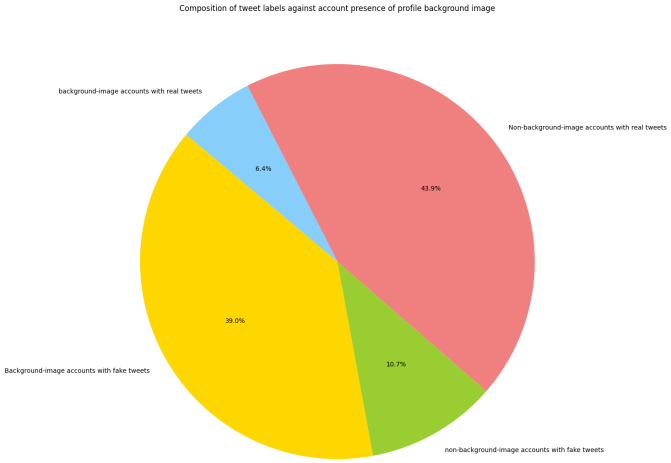


Fig. 7. Comparison of tweet labels against account presence

Hence, for accounts with a profile picture, there is a greater probability of sharing real news. This proves that accounts with a profile picture are likely to be trusted accounts that share more real tweets.

E. Are there any tweet ids exclusively with fake tweets?

This graph (in Fig. 8) shows that there are certain tweet Ids that have a greater count than other tweet ids. Among these, some tweet ids are more likely to share real news than fake news.

F. Are there some words more common in a particular class of tweets?

The graph (in Fig. 9) shows some frequent words in the text and their occurrences among fake and real tweets. The graph clearly depicts that some words are more likely to occur in fake news while some occur more frequently in real tweets. This concept of bag of words could be further extended to find

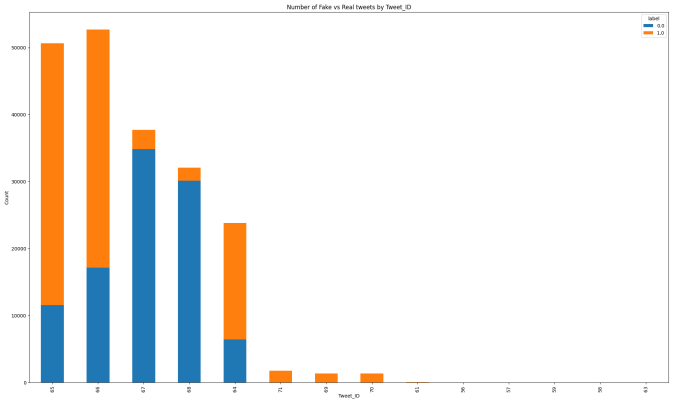


Fig. 8. Number of Fake vs. Real tweets by Tweet ID

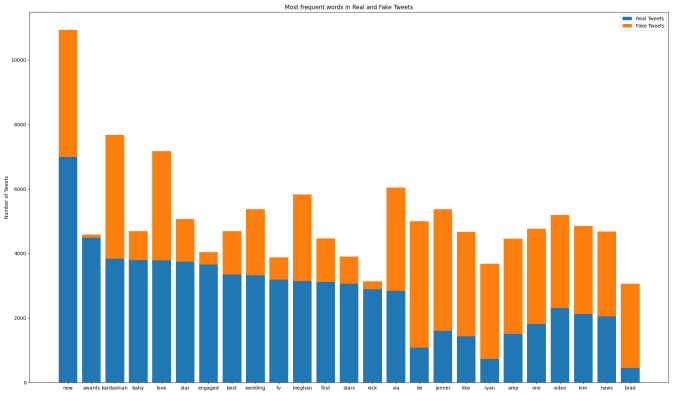


Fig. 9. Common Words

phrases or commonly sequences of words among real and fake tweets.

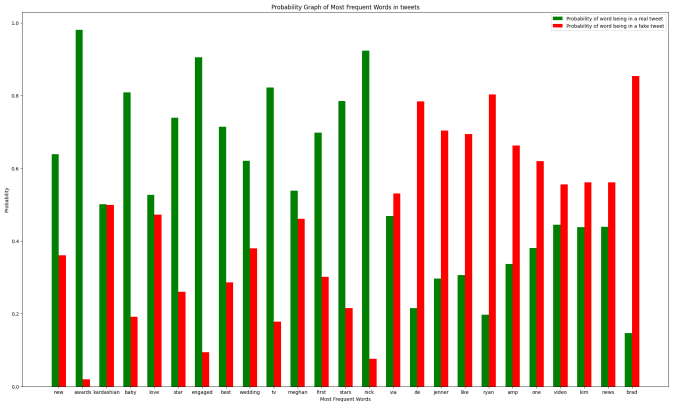


Fig. 10. Most frequent words in real/fake tweets

The graph (in Fig. 10) shows the probabilities of certain frequently repeated words occurring in real and fake tweets. The difference in red and green bars depicts that there is a difference in the probabilities of certain words occurring in fake vs. real tweets.
Hence, this confirms the hypothesis that certain words are

more likely to occur in fake tweets than in real tweets and vice versa. This implies that a word might be more likely to be found in real news than fake and vice versa.

III. Classification Problem

This section will briefly overview several classification techniques applied to the dataset. The goal was to devise different classification models and evaluate their accuracy to decide which is most appropriate for tweet label classification.

A. Naive Baye's classification

The Naïve Bayes classifier is a supervised machine learning algorithm that is extensively used for text classification. It is a generative learning algorithm and hence does not learn which features are most important to differentiate between classes. It uses the basics of Bayes' Theorem, which uses sequential events, where additional information acquired later affects the initial probabilities of events.

We trained a Sci-kit implementation of Naive Baye's classifier to develop a model to classify tweets as real or fake.

Preprocessing

For each of the tweet entries in the dataset, the tweet text was pre-processed to convert the text to lowercase and remove hyperlinks, usernames, digits, next-line symbols, and punctuation marks. Finally, all stop words are removed to ensure that generic verbal words and pronouns do not cause over-fitting.

Train-Test Split

The entire dataset was randomly divided into training data and test data using a 70-30 train-test ratio. Corresponding to each of the training and testing data, the tweet-label column is isolated such that there is a feature vector (X-test and X-train) and a label vector (Y-test and Y-train) for both the training and test data.

Accordingly, the number of samples in each of the training and test data are as follows:

- Training data - 141056
- Test data - 60453

Count Vectorizer

- A list, L, of all unique words in the training dataset's tweet text is extracted.
- A count vectorizer is used to indicate the count of each word in L for each tweet in the training data.
- Consequently, the training dataset is transformed such that each row corresponds to a single tweet, and the columns list the counts of each unique word in L.
- Similarly, the test data set is mapped and transformed onto each of the unique words in L

In our example, the number of unique words in the text of the tweets for training data after pre-processing was 90580. Hence, the dimensions of the transformed test and training matrices are as follows:

- Transformed training data - 141056 x 90580
- Transformed test data - 60453 x 90580

To automate this process, we use Sci-kit's count vectorizer.

Training and Evaluation

The transformed feature vector for training data is fed into the Sci-kit's Multinomial Classifier to produce a classification model. For evaluation, we predict the labels for the transformed feature matrix for the test data.

Results

The trained classifier reported an overall accuracy of 92.6% on the test data. Additionally, the model reports an F1- score of 0.926 as

—	Actual Positives	Actual Negatives
Predicted Positives	27125	2952
Predicted Negatives	1509	28867

B. Logistic Regression

Logistic Regression is a classification technique used in machine learning. It uses a logistic function to model the dependent variable. The dependent variable can only be two classes; hence, this technique is used when dealing with binary data.

Data Extraction

The location column is transformed via a label encoder so that a unique numerical ID tags each of the unique values in the location column. This is done via a Sci-kit library's Label Encoder.

We extract relevant columns from the dataset so as to ensure that only relevant attributes are used in the training process of our classifier. Consequently, a heat map that depicts the correlation between different numerical attributes is created.

The relevant attributes meaningful to our classifier are:

- user-id
- tweet-id
- followers
- following
- location
- verified
- statuses-count

Additionally, it is worth noting that there are very few unique user-ids and tweet-ids, which implies that our model can make use of this in our training process.

Data Normalization

The values in the numerical attributes are scaled down to values between 0 and 1 for the purposes of uniformity between various attributes.

Train-Test Split

The entire dataset was randomly divided into training and test data using an 80-20 train-test ratio. Corresponding to each of the training and testing data, the tweet-label column is isolated such that there is a feature vector (X-test and X-train) and a label vector (Y-test and Y-train) for both the training and test data.

Training and Evaluation

The training data was trained using the Logistic Regression Classifier of the Sci-kit library.

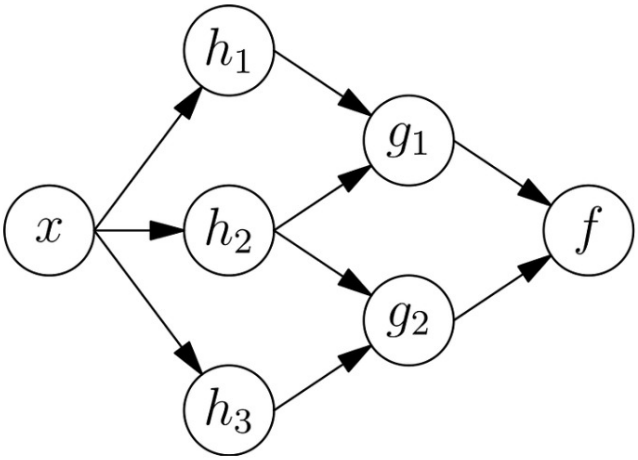
Results

The trained classifier reported an overall accuracy of 76.5% on the test data. Additionally, the model reports an F1- score of 0.765 as

—	Actual Positives	Actual Negatives
Predicted Positives	15078	4966
Predicted Negatives	4479	15779

C. Neural Net Classifiers

Neural nets are inspired by the learning process of the human brain. They consist of an artificial network of functions that allows the computer to learn and fine-tune itself by analyzing new data. Each parameter is a function that produces an output after receiving one or multiple inputs. Those outputs are then passed to the next layer of neurons, which use them as inputs of their own function, and produce further outputs.



Classification types

In this section, we discuss and implement classification using deep-learning-based models..

Primarily, we train classifiers of 3 types:

- Multilayer Perceptron
- Support Vector Machine
- Linear Perceptron

A linear perceptron is a two-layer network that has only one input and output layer. However, a multi-layer perceptron has at least one hidden layer.

Results

The trained classifiers reported an overall accuracy and F1 score as given below:

- Multilayer Perceptron-
  - Accuracy - 81.3%
  - F1-score - 0.813
- Linear Perceptron
  - Accuracy - 49.7%
  - F1-score - 0.335
- Support Vector Machine
  - Accuracy - 76.9%
  - F1- score - 0.769

Confusion Matrix for Multilayer Perceptron

—	Actual Positives	Actual Negatives
Predicted Positives	15832	4212
Predicted Negatives	3308	16950

Confusion Matrix for Linear Perceptron

—	Actual Positives	Actual Negatives
Predicted Positives	19982	62
Predicted Negatives	20193	65

Confusion Matrix for Support Vector Machine

—	Actual Positives	Actual Negatives
Predicted Positives	16004	4040
Predicted Negatives	5252	15006

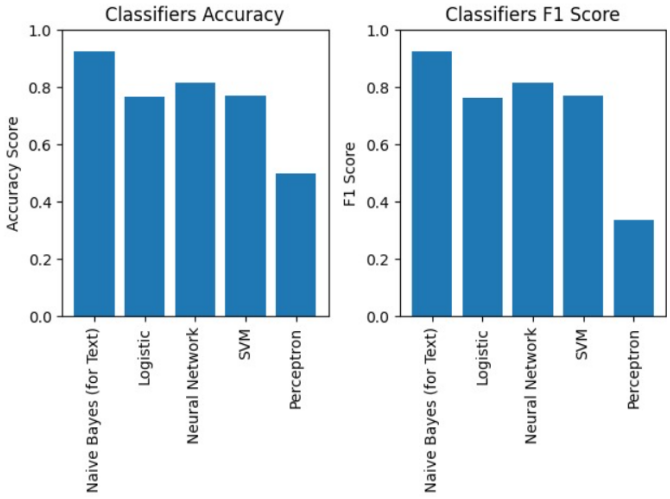


Fig. 11. Results

D. Feature Selection for Multi-layer Perceptron

From the above-observed results, it is evident that a multi-layer perceptron obtains the highest accuracy and F1 score. This gives us an exploratory room for improved accuracy for the multi-layer perceptron.

Hence, we employ feature selection to choose the best possible combination of features that would prevent the model from over-fitting to training instances

. The three major techniques that we employ for feature selection are as follows:

1) Filter method
2) Wrapper selection and Decision tree classifier
3) Wrapper selection and Gradient Boosting Regressor

Additionally, the tweet text column is obviously unique and is not considered in our analysis. Incorporating tweet text through one hot encoding resulted in an excessive number of dimensions (201509 rows × 118160), making it impractical to compute and causing memory problems.

Filter Method

In this approach, we set a variance threshold of 0.1. Any attribute that observes a variance value lower than this threshold is stagnant for our classifier and does not add any new

information. Hence, such attributes are discarded.
Given this approach, the attributes that remain after filtering are as follows:

- user-id
- tweet-id
- followers
- following
- location
- statuses-count
- profile-background-tile
- profile-use-background-image
- favorite-count

Wrapper selection and Decision tree classifier
For this approach, we use Recursive Feature Selection (RFE) along with a Decision Tree Classifier as the estimator for (RFE). Additionally, the number of features that we wish to obtain is 7.
The selected features by our model as given below:

- user-id
- tweet-id
- followers
- following
- location
- statuses-count
- favorite-count

Wrapper selection and Gradient Boosting Regressor Similar to the previous approach, an RFE model is used to extract 7 features. However, the estimator used is Gradient Boosting Regressor. The features selected by our model are as listed below:

- user-id
- tweet-id
- followers
- statuses-count
- profile-background-tile
- profile-use-background-image
- favorite-count

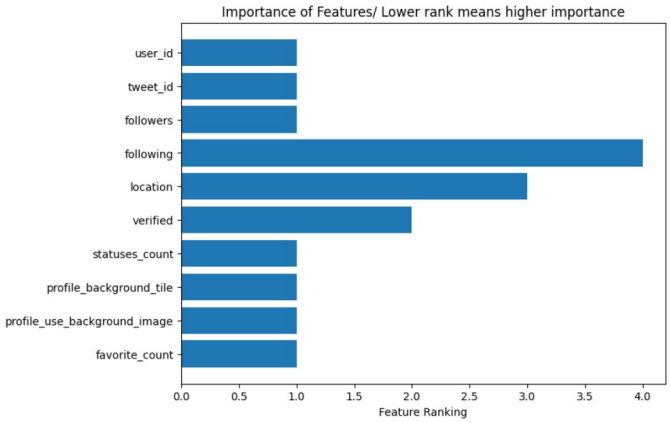


Fig. 12. Relative importance of features

Training and Evaluation
We train and evaluate the Multilayer Perceptron on 4 possible feature sets.

1) All columns except text - (retained for comparison) - **Set-1**
2) Feature selection using variance filter method with threshold 0.1 - **Set-2**
3) Feature selection using wrapper method on decision tree classifier - **Set-3**
4) Feature selection using wrapper method and gradient boosted regression - **Set-4**

Confusion Matrix for Set - 1

—	Actual Positives	Actual Negatives
Predicted Positives	15832	4212
Predicted Negatives	3308	16950

Confusion Matrix for Set - 2

—	Actual Positives	Actual Negatives
Predicted Positives	18078	1966
Predicted Negatives	5933	14325

Confusion Matrix for Set - 3

—	Actual Positives	Actual Negatives
Predicted Positives	15651	4393
Predicted Negatives	2711	17547

Confusion Matrix for Set - 4

—	Actual Positives	Actual Negatives
Predicted Positives	16276	3768
Predicted Negatives	3633	16625

Accuracy and F1 Scores

—	Accuracy	F1 score
Set - 1	81.3%	0.813
Set - 2	80.4%	0.804
Set - 3	82.3%	0.823
Set - 4	81.6%	0.816

From our evaluation and the above-provided results, it is evident that feature selection using the wrapper method on the decision tree classifier performs best on the given test data instances.

In order to expand our exploration and incorporate additional popular models for testing Ensemble learning combinations, we proceeded to train additional models, including Decision Trees and Polynomial Regression, using scikit-learn implementation. The accuracies of these models, as compared to others, are presented below.

E. Ensemble Model

Ensemble modeling is a technique in which we combine two or more related but similar machine-learning models to devise a new model we a higher accuracy. We combined the following four classification models:

- Logistic Regression
- Neural Network
- Decision Tree
- Polynomial Regression

in order to get a single Ensemble Model that can outperform the individual models.

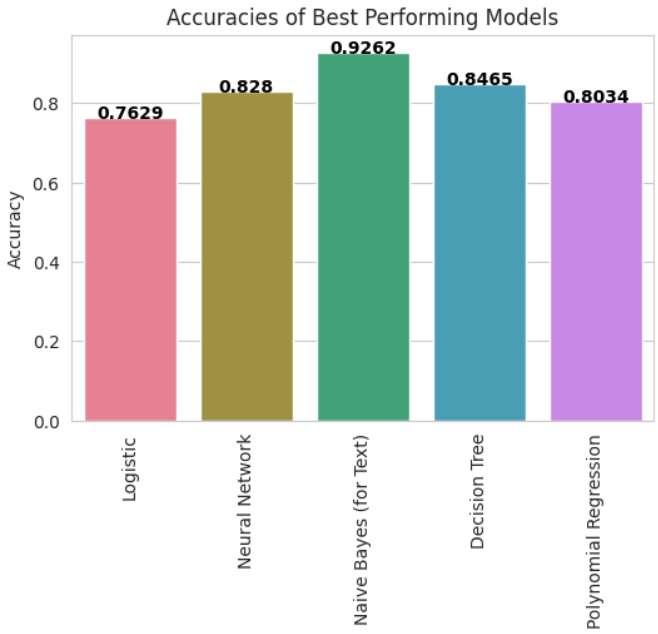


Fig. 13. Accuracies of various Classification models

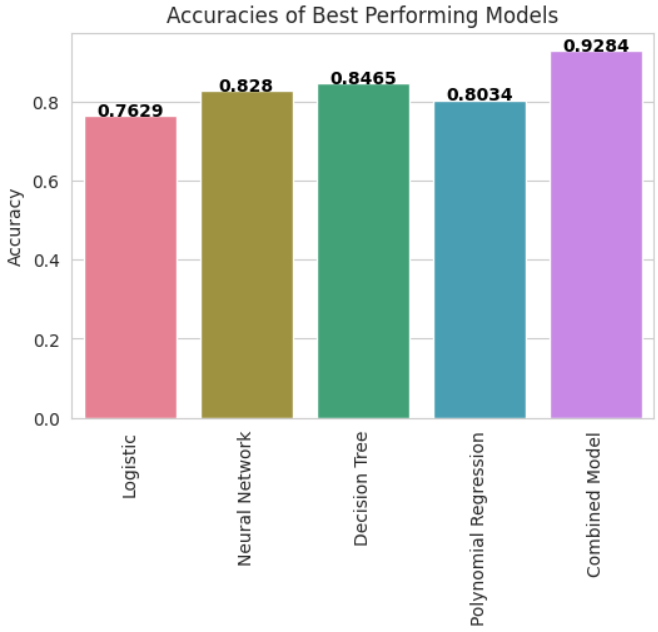


Fig. 14. Accuracies of different classification models

The ensemble modeling approach addresses the technical obstacles that arise when constructing a single estimator:

1) High variance/over-fitting: The model is extremely responsive to the inputs supplied to the learned features.
2) Inadequate precision: Employing a single model or algorithm to fit the entire training data may fail to meet expectations.
3) Inaccuracies due to feature noise and bias: The model depends heavily on one or a small number of features when making predictions.

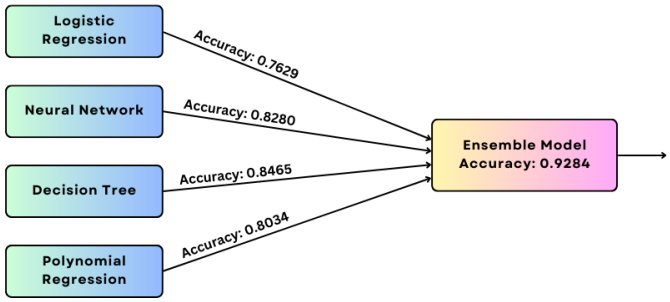


Fig. 15. Combining different classification models

IV. Frequent Pattern Mining

Frequent pattern mining is a data mining technique that aims to identify recurring patterns in large datasets. The process involves detecting sets of items that frequently appear together in a dataset, such as commonly purchased items in a supermarket. In this project, we employ frequent pattern mining on our dataset to explore the relationship between repetitions and labels.

Apriori

Due to limitations in processing power, we are unable to run the Apriori algorithm on the entire dataset. Instead, we break down the data into smaller chunks and analyze repeated patterns within each chunk. Before running Apriori, we preprocess the data by converting all values in columns to strings and appending the column names to the values. This is necessary to differentiate between identical values in different columns since Apriori tries to establish relationships across columns.

We run the Apriori algorithm twice. First, we analyze the Boolean attributes, tweet-id, and favorite count, resulting in a total of six attributes: tweet-id, verified, profile-background-tile, profile-use-background-image, favorite-count, and label. By focusing on these columns, we are able to run Apriori without encountering memory issues. We identify a list of repeated patterns, which we further divide to determine the frequent patterns for labels 1 and 0, respectively.

One possible approach is to execute Apriori with various combinations and consolidate the findings to identify the most frequently occurring patterns. Nonetheless, considering time constraints and the priority of specific attributes highlighted by feature selection, we proceed with the previously selected features and run Apriori. To enhance the likelihood of discovering frequent patterns, we eliminated the user-Id and statuses-count columns from our data frame, which contained a greater number of unique values and fewer repetitions. We then apply Apriori to the remaining attributes and generate a list of the most prevalent patterns evident in the data.

FP-Growth

FP-growth builds a compact data structure called a frequent pattern tree (FP-tree) to represent the transnational database,

which allows it to efficiently mine frequent item sets without generating candidate item sets explicitly. Compared to Apriori, FP growth is generally faster and uses less memory. Therefore, we apply the FP growth algorithm to our complete dataset, where we specify the column names along with their respective values for clarity, as we have done before. Running the algorithm generates a list of all frequent patterns. However, we discard any frequent patterns that do not involve the label since our aim is to identify the connection between frequent patterns and labels.

By following the same procedure and setting the support threshold to 0.9, we identify the most frequently occurring patterns. Then, we generate separate outputs for the frequent patterns associated with labels 1 and 0, which enables us to gain insights into the prevalent patterns in each label. Moreover, this analysis reveals that some patterns are more likely to occur in one label than the other, which could be leveraged to enhance classification tasks.

Future Work

As an extension to frequent pattern mining, it is expected that more patterns can be sought out by assigning weights to different frequent patterns extracted from the subsets of data. Eventually, they can be merged to gather significant insights on the trends that are most common in profiles for real and fake tweets both.

V. Conclusion

Using the analysis we can conclude that Neural Networks based classifier performs significantly better than other other individual classification techniques. Furthermore, there exists potential room for overfitting which needs to be avoided via feature selection.

Additionally, the multinomial classifier (Naive Baye's) reports attainably the highest accuracy as it relies on the bag of words implementation focusing solely on the crux of tweet text. There is little attention paid on extraneous attributes and the probabilistic modelling is conditioned over tweet text. Hence, it can be safely labelled as the most appropriate classification technique for the given data semantics which include a heavy reliance on tweet text.

However, an ensemble model that uses a combination of multiple models results in the highest accuracy and is able to predict the best results owing to the combination of predictions of multiple models to improve the overall accuracy of the prediction. The individual models in the ensemble may have different strengths and weaknesses, and by combining their predictions, the ensemble model can compensate for these weaknesses and provide a more accurate prediction as evident in our results.