# Stock Data Classification Report

**Phi Lam**
Department of Electrical and Computer Engineering
University of British Columbia
philam@student.ubc.ca

## Abstract

This project focuses on doing binary classification using Logistic Regression and then comparing it with the Random Forest Classifier for stock data news to analyze and predict if a tweet as a positive or negative sentiment to it. It will cover the data set used, methodology used to train the model, the experiments ran, and its results.

## 1 Problem Definition

The objective of this report is to do a binary classification for sentiment analysis to predict if there is positive or negative sentiment on stock news based on Twitter tweets.

## 2 Dataset

The dataset I am using is the **Stock-Market Sentiment Dataset** from Kaggle. The link for this is `https://www.kaggle.com/datasets/yash612/stockmarket-sentiment-dataset/data`. First, I change the sentiment value to 0 or 1 instead of -1 or 1 as the Logistic Regression model only predicts 0 or 1.

The data set starts off like this from Kaggle:

| | Text | Sentiment |
|---|---|---|
| 0 | Kickers on my watchlist XIDE TIT SOQ PNK CPW BPZ AJ trade method 1 or method 2, see prev posts | 1 |
| 1 | user: AAP MOVIE. 55% return for the FEA/GEED indicator just 15 trades for the year. AWESOME. | 1 |
| 2 | user I'd be afraid to short AMZN - they are looking like a near-monopoly in eBooks and infrastructure-as-a-service | 1 |
| 3 | MNTA Over 12.00 | 1 |
| 4 | OI Over 21.37 | 1 |
| ... | ... | ... |
| 5786 | Industry body CII said #discoms are likely to suffer a net revenue loss of around Rs 30,000 crore\n\nhttps://t.co/8c6YJYJajL | -1 |
| 5787 | #Gold prices slip below Rs 46,000 as #investors book profits amid #coronavirus-led #recession fears https://t.co/fSylAJrUqv | -1 |
| 5788 | Workers at Bajaj Auto have agreed to a 10% wage cut for the period between April 15 and till the lockdown is lifted. https://t.co/RgvrKPliNd | 1 |
| 5789 | #Sharemarket LIVE: Sensex off day's high, up 600 points, #Nifty tests 9,200, #TCS, private bank stocks lead\nhttps://t.co/3xgtLroKUI | 1 |
| 5790 | #Sensex, #Nifty climb off day's highs, still up 2%; Key factors driving D-Street higher today https://t.co/jVQcousFp6 | 1 |

5791 rows × 2 columns

Figure 2.1 Raw tweets data for sentiment analysis

After skimming the dataset, I do some natural language processing(NLP) to optimize the data set and help the model comprehend human text. This includes making all the text lower-case, tokenizing the text, removing stop words, symbols, and hyperlinks, and then recombining the tokens into a single text. I excluded the word **over** from the NLTK stop words as I needed it for another step to continue the preprocess for my data. The text column ends up looking like this:

| | Text | Sentiment |
|---|---|---|
| 0 | kickers watchlist xide tit soq pnk cpw bpz aj trade method 1 method 2 , see prev posts | 1 |
| 1 | user : aap movie . 55 % return fea/geed indicator 15 trades year . awesome . | 1 |
| 2 | user 'd afraid short amzn - looking like near-monopoly ebooks infrastructure-as-a-service | 1 |
| 3 | mnta over 12.00 | 1 |
| 4 | oi over 21.37 | 1 |
| ... | ... | ... |
| 5786 | industry body cii said discoms likely suffer net revenue loss around rs 30,000 crore | 0 |
| 5787 | gold prices slip rs 46,000 investors book profits amid coronavirus-led recession fears | 0 |
| 5788 | workers bajaj auto agreed 10 % wage cut period april 15 till lockdown lifted . | 1 |
| 5789 | sharemarket live : sensex day ' high , 600 points , nifty tests 9,200 , tcs , private bank stocks lead | 1 |
| 5790 | sensex , nifty climb day 's highs , still 2 % ; key factors driving d-street higher today | 1 |

5791 rows × 2 columns

Figure 2.2 Text data for sentiment analysis after initial preprocessing

Then I do a little bit of feature engineering by doing one-hot encoding for words that are considered positive or negative in terms of the stock. As I mentioned above, I excluded **over** from the NLTK stop words so that I can use it to do one-hot encoding here. The positive words for the stock analysis include **dip** and **over** $\backslash d+$ where $\backslash d+$ is any number that's one or more digits. Negative words here include **short**. These are all terms that more obviously refers to the financial marketing language. I do this to help my model's accuracy accuracy by having more features to better represent the underlying problem. This feature engineering is shown in Figure 2.3

| | Text | Sentiment | One Hot Encoded |
|---|---|---|---|
| 0 | kickers watchlist xide tit soq pnk cpw bpz aj trade method 1 method 2 , see prev posts | 1 | 0 |
| 1 | user : aap movie . 55 % return fea/geed indicator 15 trades year . awesome . | 1 | 0 |
| 2 | user 'd afraid short amzn - looking like near-monopoly ebooks infrastructure-as-a-service | 1 | 0 |
| 3 | mnta over 12.00 | 1 | 1 |
| 4 | oi over 21.37 | 1 | 1 |
| ... | ... | ... | ... |
| 5786 | industry body cii said discoms likely suffer net revenue loss around rs 30,000 crore | 0 | 0 |
| 5787 | gold prices slip rs 46,000 investors book profits amid coronavirus-led recession fears | 0 | 0 |
| 5788 | workers bajaj auto agreed 10 % wage cut period april 15 till lockdown lifted . | 1 | 0 |
| 5789 | sharemarket live : sensex day ' high , 600 points , nifty tests 9,200 , tcs , private bank stocks lead | 1 | 0 |
| 5790 | sensex , nifty climb day 's highs , still 2 % ; key factors driving d-street higher today | 1 | 0 |

5791 rows × 3 columns

Figure 2.3 Data after including one-hot encoding

Then, I split the data into training and testing dataframes. The split is 75% training and 25% testing.

# 3 Method

## 3.1 Environments

The environments I use for this project includes the following libraries:

**Organization**: numpy, pandas, matplotlib

**Preprocessing**: nltk

**Model Training and Validation**: sklearn

## 3.2 Steps

1. Define the classification I'm trying to achieve, which is classifying if a stock has a positive or negative sentiment based on its tweets. Noted above in the **Problem Statement** section.

2. Preprocess and split into training and test data. Noted above in the **Dataset** section.

3. Do vectorization on the **Text** column of the training data.

I use Sklearn's **TfidfVectorizer** as a step in feature extraction to convert the text I preprocessed into meaningful numerical vectors. It converts raw documents to a matrix of TF-IDF features. It helps normalize the text data, count term frequency, and adjusts counts by the inverse document frequency. Originally I had the vectorization step in the preprocessing of data section until I did more research and understood that doing this could cause data leaks from the testing set to the training model. I capped **max_features=5000** to limit the number of features created from this; the value 5000 yielded the highest accuracy for me.

4. With the X training data fully processed, I go into designing my experiment with the existing ML models: Logistic Regression and Random Forest Classifier. The experiment will be noted down below in **Experiments and Results Analysis**.

# 4 Experiments and Results Analysis

## 4.1 Logistic Regression

I use K-fold cross validation to help prevent overfitting the Logistic Regression model while hyperparameter tuning the model. The hyperparameter I'm tuning is the **C** hyperparameter. The hyperparameter helps tune the regularization of the sigmoid model to train the model well. However, this regularization can also overfit the model.

I range over 5 **C** hyperparameter values while doing 5 folds of the K-fold cross validation. I initially started with a larger range in C, before shortering that range. In the experiment, I range over **C=[4.4, 4.5, 4.6, 4.7, 5]**. After each 5 fold cross validation, I average the evaluation metrics: Accuracy, Precision, Recall, and F1 Scores. Each evaluation metric is plotted below in Figure 4.1 with the varying values based on the hyperparameter C values.

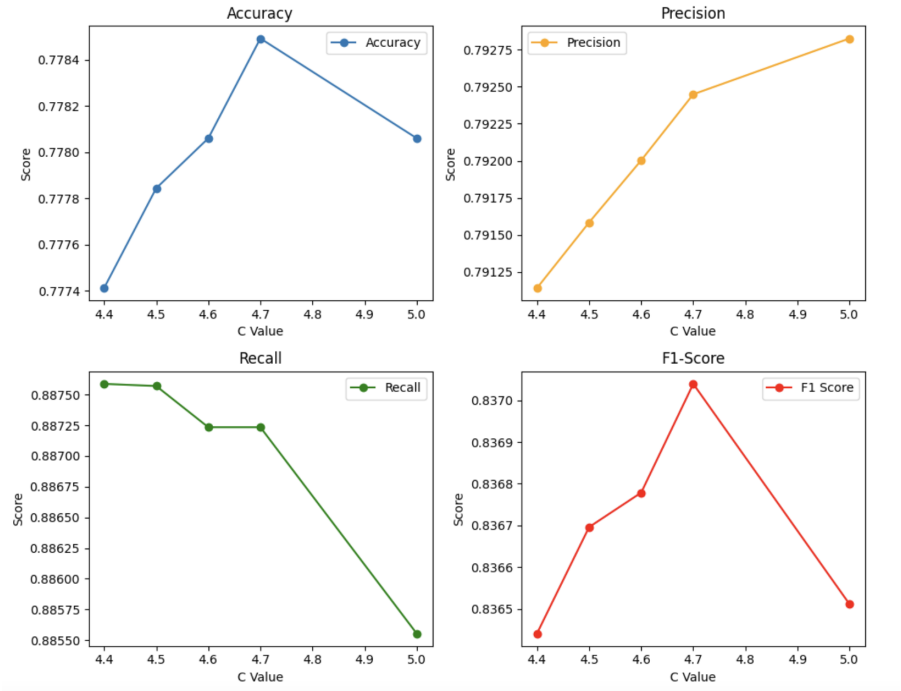


Figure 4.1: Evaluation metrics for the Logistic Regression model with different hyperparameters

It can be concluded that the best **C** hyperparameter value is 4.7. The F1-score is a combination of the recall and precision scores. So although the other hyperparameter values have larger precision and recall values, they are not deemed a better model as it has not made correct prediction across the entire dataset. The accuracy predicts overall how correct the model scored and as it can be seen **C=4.7** scores the highest.

**The evaluation metrics for C=4.7:**
Accuracy = 0.7784914294235101, Precision = 0.7924489314029295, Recall = 0.8872341793238945, and F1-Score = 0.8370394403564717

### 4.2 Random Forest Classifier

I compare my tuned model with the **RandomForestClassifier** by first doing **GridSearchCV** to find the best hyperparameters for the model. The best hyperparameter values for the model: max_depth=15, max_leaf_nodes=24, n_estimators=5

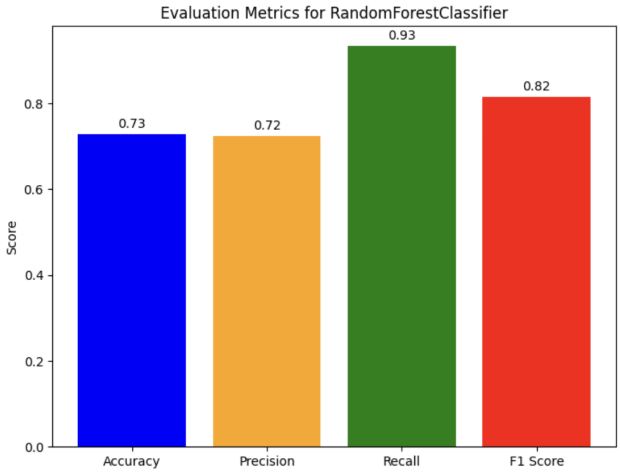


Figure 4.1: Evaluation metrics for the RandomForestClassifier model

Above is the training data evaluation metrics results. For the most tuned RandomForestClassifier model, it performs worst than the Logistic Regression model in everything but recall.

### 4.3 Comparing Models with Test Results

Using the most tuned hyperparameters for Logistic Regression and RandomForestClassifier, I train the models with the training data. Before doing predictions with the different models, I have to additionally vectorize the X_test data. The results with the test data for both models side by side is plotted down below in Figure 4.3.

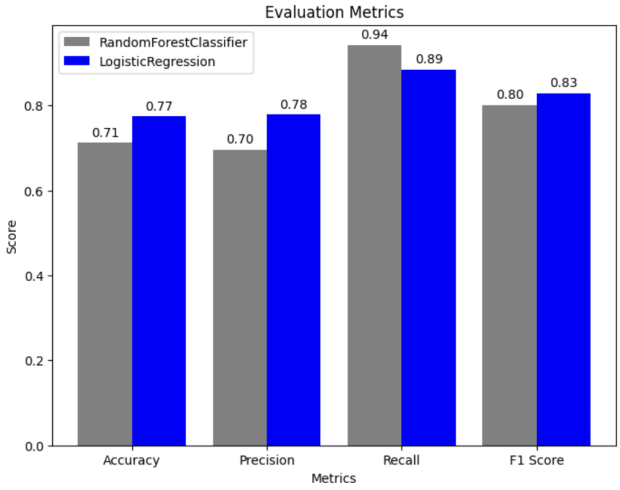


Figure 4.1: Test data evaluation metrics for both models

## 5 Discussion

With the findings of this experiment, the simpler Logistic Regression model, with an accuracy of 0.77, outperformed the RandomForestClassifier model, which achieved a 0.71 accuracy. This suggests that the complexity of RandomForestClassifier was not advantageous for our stock text dataset, which is possibly due to the complexities of NLP. We find that both models did not overfit during the hyperparameter tuning as the test scores are similar to the training scores. While the accuracy achieved by the Logistic Regression model is lower than the ideal for a robust model, it represents the best outcome that can come from the data set used. My potential next steps includes more developed feature engineering to perform a better NLP for the data set and doing vectorization for key phrases and words in the data set. I'd also want to extract the stock names and preprocess the data so that I can predict whether the stock would have a positive or negative sentiment and not just the tweet itself.