Stock Sentiment Analysis

The goal of the this project was to analyze a dataset pertaining to the Dow Jones Industrial Average(DJIA), with the hopes of exploring and processing the dataset, drawing meaningful statistical conclusions, as well as building a machine learning model using state of the art frameworks like Keras and BERT to predict price movement of the DJIA.

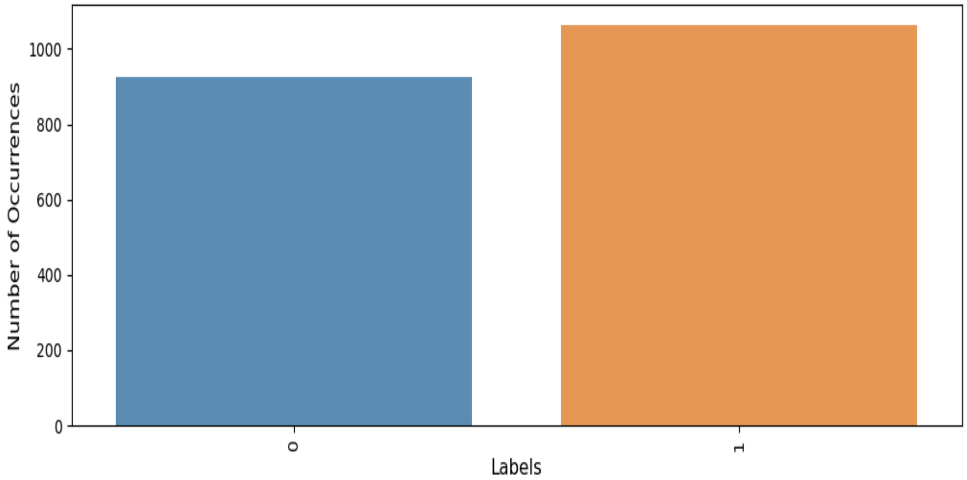
Dataset

The dataset, while comprehensive, was very simple and easy to follow. Given a specific date that falls in the range of the collected data, each row contains the top 25 news headlines for that date along with a label indicating whether the DJIA increased(1) or decreased(0). The dataset contains 1989 rows meaning that roughly 50000 news headlines were collected between August 8th, 2008 and July 1st, 2016. The dataset can be found in the following link:

https://www.kaggle.com/datasets/waseemalastal/combined-news-and-djia-dataset

Data Exploration and Preprocessing

First, I wanted to see the distribution of classes throughout the dataset, which can be seen as follows:



From there I began to preprocess the data. The first step in preprocessing the data was to clean the text such that it was of a neater and easier to follow quality. Specifically, regular expressions were used to remove unwanted punctuation and characters, lingering html tags, additional spaces, as well as leading and trailing spaces. From there I consolidated all the news headlines pertaining to a given date into a single list. The next step was to run these cleaned news headlines through a Spacy NER model with the goal of filtering out any irrelevant or unwanted headlines. The rationale for this step is as follows: considering that the DJIA is a major stock market index, only news pertaining to major world or economic events should be relevant. For instance, the headline, *“A 117-year-old woman in Mexico City finally received her birth certificate, and died a few hours later. Trinidad Alvarez Lira had waited years for proof that she had been born in 1898”*, realistically has no bearing on the movement of any index fund. It is just an interesting story. However the headline, *“Georgia downs two Russian warplanes as countries move to brink of war”*, definitely raises alarm bells that could contribute to a change in the DJIA.

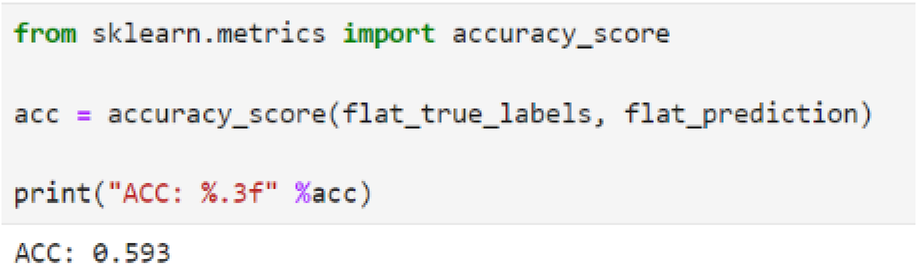
Following the NER filtering, I combined all the individual news headlines for each respective date into a single string. The final step was to remove stopwords from the text and format everything to a consistent format. The data was now ready to be passed through the ML pipeline for classification.

Building Model Pipeline

The first step in building the pipeline for classification was to tokenize the text. To do this I used the BertTokenizer from Huggingface’s transformers library. Using the tokenized text I then built embeddings for each tokenized text block. The next step was to pad/truncate the embedded text such that all inputs were of a consistent length. This was done using the pad_sequences method from Keras. All sequences were padded/truncated to a length of 256 tokens. From there I also built an attention mask using the padded sequences. The data was ready to be split into training, testing, and validation sets and prepped for input into the classification model.

Training and Evaluating the Model

The results after five training epochs are as follows, with a sequence length of 256, batch size of 8, and a learning rate of 1e-5:

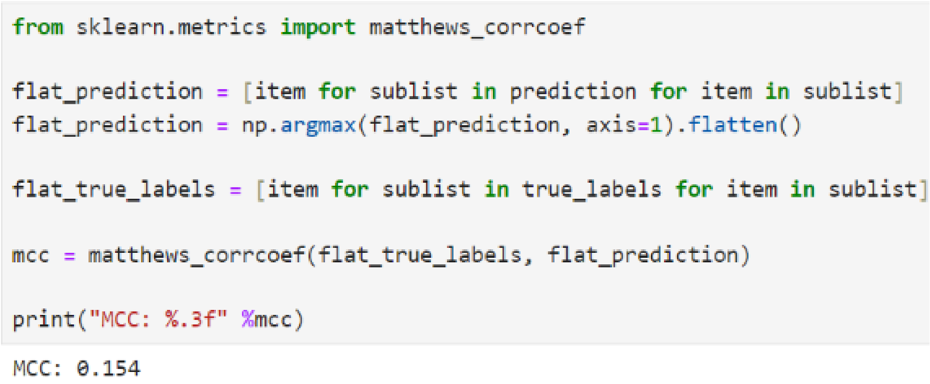


```
from sklearn.metrics import accuracy_score

acc = accuracy_score(flat_true_labels, flat_prediction)

print("ACC: %.3f" %acc)
```

```
ACC: 0.593
```



```
from sklearn.metrics import matthews_corrcoef

flat_prediction = [item for sublist in prediction for item in sublist]
flat_prediction = np.argmax(flat_prediction, axis=1).flatten()

flat_true_labels = [item for sublist in true_labels for item in sublist]

mcc = matthews_corrcoef(flat_true_labels, flat_prediction)

print("MCC: %.3f" %mcc)
```

```
MCC: 0.154
```

Moving Forward and Expansion

I’d like to continue with this project with the goal of improving upon the accuracy and MCC. I’d also like to try using some different models and comparing their performance to the BERT based approach that I took in this project. This was a great learning experience and I want to continue building upon the foundation that I set for myself in this project.