CS419M Introduction to Machine Learning, Spring 2021 May 12th 2021
Course Project Report

Predicting Stock Market Price Movement using Sentiment Analysis

Abhishek Pai Angle - 190110101
Neilabh Banzal - 170010014
Pranjal Gupta - 190010053
Sanidhya Anand - 19D170027

Objective

Due to a high degree of uncertainty in the stock market, it is difficult to predict stock price movement. The aim of this project is to model such movements as a complex function of one of various parameters that affect the same; daily world news updates. Neural networking techniques have been implemented in this project to model such complexity.

Problem Statement

The chief problem statement, as mentioned before, was predicting the stock market movement from world news. To develop a model that serves our purpose, we consider daily world news and predict the movement of Dow Jones Industrial Average (DJIA) which is a stock market index that measures the stock performance of 30 large companies listed on stock exchanges in the US. If DJIA moves up, we label the day as +1 and if it moves down, we label it as 0.

Now, before delving into the actual implementation of this approach, we have to consider if our solution has a significant impact in the market. This model would help the investors and traders in getting an idea of market movement, which would assist them in investing in and withdrawing stocks. The financial market is “informationally efficient” i.e. stock prices reflect all known information, and the price movement is in response to news or events.

Datasets

Feature Dataset- For our features, we use a dataset of historical news headlines from 2008-06-08 to 2016-07-01 crawled from Reddit WorldNews Channel (/r/worldnews). Only the top 25 daily headlines ranked by reddit users’ votes have been considered. The corresponding csv file is RedditNews.csv.

We have further scraped our own data using Scrape.py. The corresponding csv file is News.csv. The date column has reference as 1 January 2020 (‘0’).

Label Dataset- For our labels, we use the Dow Jones Industrial Average (DJIA) movement obtained from Yahoo Finance for the same dates. If DJIA moves up, we label the day as +1 and if it moves down we label it as 0. The corresponding csv file is DJIA_table.csv.

For convenience, both datasets have been combined into one csv file Combined*News*DJIA.csv. The columns of this file are the date, the label for that day and the 25 news headlines ranked topic-wise on the basis of how “hot” they are (correlated to engagement and upvotes on reddit). This gives a total of **27 columns** and **1989 rows**. Analysis has been done on this combined dataset.

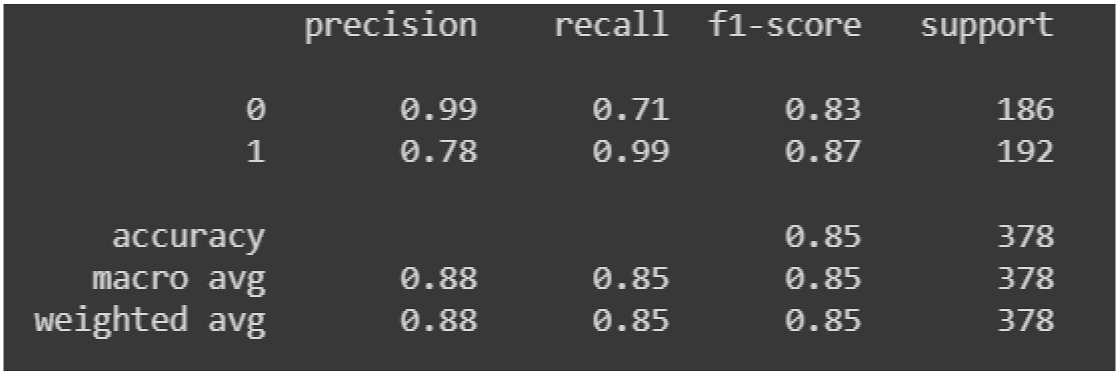
List of techniques used

- LSTM (Long Short Term Memory) architecture for RNNs.
- Random Forests
- Logistic Regression

Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean/average prediction of the individual trees.

<u>Data preprocessing</u>:
For each day, 25 headlines are combined into a single string. Countvectorizer is used to create sparse representations of strings. A word-to-integer mapping dictionary is created first for computing these sparse representations. 200 estimators (individual decision trees) were trained on the training data.



```
              precision    recall  f1-score   support

           0       0.99      0.71      0.83       186
           1       0.78      0.99      0.87       192

    accuracy                           0.85       378
   macro avg       0.88      0.85      0.85       378
weighted avg       0.88      0.85      0.85       378
```

Fig: Results of Random Forest Classifier.

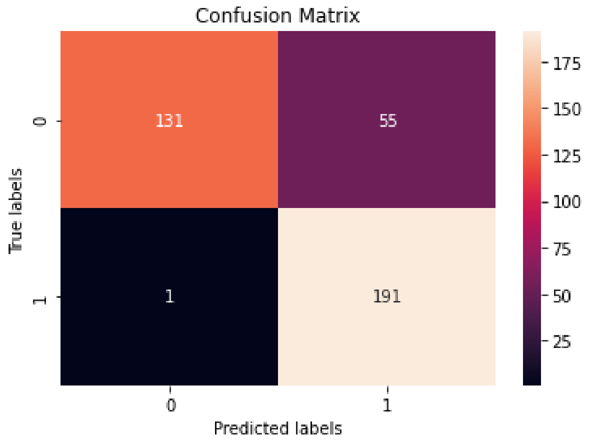


Fig: Confusion matrix for Random Forest results

LSTM

LSTM: Long Short Term Memory networks are a special kind of RNN, capable of learning long-term dependencies. Their advantage over traditional classifiers is that they connect previous data to next data. This is useful for a sequence such as a sentence (headlines in our case). The use of add gates and forget gates to selectively transfer data to subsequent cells is an important advantage over traditional classifiers. Since our sequential data also has a temporal component to it, LSTM is a great fit for modelling time series stock prediction models.

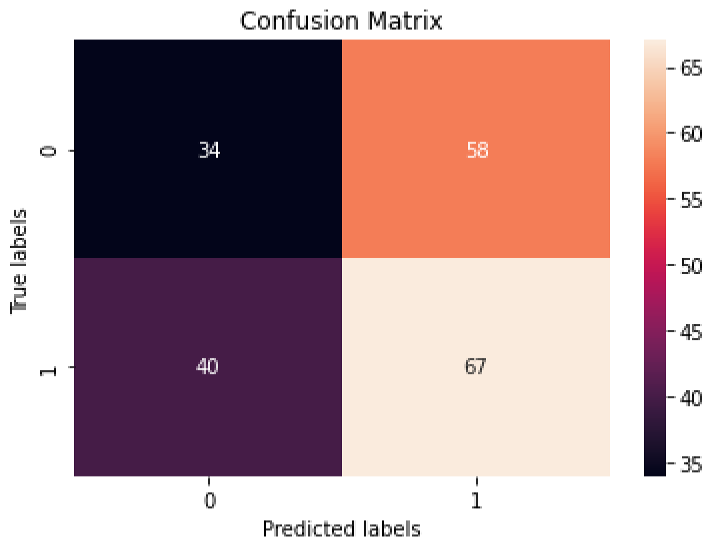
Methodology:

1. We have considered a combined set of 25 news headlines as input and we predict the market movement as upward or downward as the output.
2. We convert the headline text into integers by tokenizing them using a dictionary based on frequency of the word in the corpus.
3. These tokens are later converted into Embeddings of specific dimension to feed into the LSTM model.
4. The output of the LSTM model is fed into an artificial neural network which gives sigmoid output which is the probability of the market going up.

Challenges

Finding the right embedding technique and dimension plays an important role in model performance. After choosing a model architecture, our model was overfitting. The performance of the model was improved by simplifying model architecture, using dropout layers and regularization.

Results



The model on a test set gives accuracy of ~50%.

Logistic Regression

We also try another classification technique on our data, known as Logistic Regression. It is a linear binary classification algorithm for linearised inputs and the output is characterized by the sigmoid function, which gives a value between 0 and 1, akin to the probability of the output being true (i.e, 1). For the purpose of this project, we have used the LogisticRegression class of the sklearn library. However, instead of using the headlines directly as input, we have first performed sentiment analysis on the headlines and then used a two-feature input as training data. The sentiment analysis is done using the TextBlob library, and the two features are known as Polarity and Subjectivity.

TextBlob library
TextBlob is a Python library which is used for common NLP tasks such as Speech Tagging, Speech Translation, Sentiment Analysis.

Polarity - Polarity is a measure of the sentiment of a sentence. The value is in the range of [-1,1], -1 being a highly negative sentence (eg: 10 people were killed in a crash) and +1 being a highly positive sentence (eg: More than 25000 people recovered from Covid-19 today)

Subjectivity - Subjectivity is a measure of sentiment which shows how subjective a sentence is. A subjective sentence means that the sentence is most probably a personal opinion which may not be applicable in every situation or to every individual (eg: Kanye west is a good presidential candidate) and an objective sentence is a universal truth (eg: Covid-19 is a pandemic).

Convolutional Neural Network

Implementing a CNN on textual data is difficult, mainly due to unavailability of relevant dimensions derived from embedding the words in the text. (data, in this case, being news headlines and covers). A possible implementation has been theoretically explored.

Textual data events can be analysed as a fixed tuple E = (O1, P, O2, T), where P is the action, O1 is the actor and O2 is the object on which the action is performed. T is the timestamp of the event, which is mainly used for aligning stock data with news data.
A conversion to event embeddings which represent the actor, action and object as the average of its word embeddings follows. This gives greater accuracy than simply using word embeddings. Convolution operations are done on time series of input event embeddings. The output of these convolution operations is a feature vector that can correspond to daily, weekly, as well as yearly news. In our scenario, a feature vector corresponding to daily news may be chosen. To correlate the feature vector and stock prices, we use a feedforward neural network.

Conclusion

In conclusion, we explored three different techniques extensively. Results have been extensively shown, with the best model being Random Forest Classifier, followed by LSTM and Logistic Regression.

Literature Review

The chief purpose of our literature reviews was to pinpoint the ideal implementation/method to be used for stock data. Not much study has been done particularly with NLP on news based data, but for general temporal prediction, LSTM was targeted as the most commonly used method.

Bibliography

Ding, X., Zhang, Y., & Liu, T. (2015). Deep Learning for Event-Driven Stock Prediction. *International Joint Conference on Artificial Intelligence*, *2015*. 10.5555/2832415.2832572

Hu, Z., Zhao, Y., & Khushi, M. (2021). A Survey of Forex and Stock Price Prediction Using Deep Learning. *Applied System Innovation*, *4*(9). Retrieved May 13, 2021, from https://www.mdpi.com/2571-5577/4/1/9

Lakshminarayanan, S. K. (n.d.). A comparative study of svm and lstm deep learning algorithms for stock market prediction. http://ceur-ws.org/Vol-2563/aics_41.pdf