IBM-312

Data Mining For Business Intelligence

PROJECT REPORT

On

NEWS ORIENTED STOCK PRICE TREND PREDICTION

Submitted

By-

Ritik Kumar(19115102)
Harshit Kumar Gupta (19115061)
Pawan Kumar (19115086)
Aditya Agrawal (19115009)

Submitted

To-

Prof. Sumit Kumar Yadav

Indian Institute of Technology, Roorkee

2022

ACKNOWLEDGEMENT:

We are grateful to Prof. Sumit Kumar Yadav, our esteemed professor, for his dynamic leadership and knowledge,which enabled us to accomplish this project effectively. We'd also want to express our gratitude to our colleagues who have contributed to this project, whether directly or indirectly.

AIM:

The main goal of our project is to clean and analyse data and train a machine learning model to analyse news data and anticipate market trend movement using two of the most common machine learning methods, Logistic Regression and Support Vector Machine (SVM).

The challenge of predicting market stock values has always been difficult. However, owing to market volatility, making a good prediction purely based on previous stock data is challenging. As a result, in this research, we identify several critical aspects that might be beneficial in stock price prediction and present a machine learning model to capture the dynamics of stock price trend using rich news textual information, based on an examination of daily news' influence on stock markets.

OVERVIEW:

- Introduction and Dataset
- Data Segmentation and Cleaning
- Exploratory Data Analysis using python’s data visualization libraries
- Training the model based on available data

INTRODUCTION:

The price of a stock is inextricably linked to the news of the day. Positive news, such as fresh acquisition prospects, great earnings reports, and good management governance, can boost the price of some companies. Detrimental news such as weak monetary policy, political unrest, and natural disasters may have a negative impact on the stock market as a whole. Depending on the news, the stock market may respond in a variety of ways. As a result, when new information becomes available, it is critical for investors to swiftly assess its impact and make accurate stock price projections. This project's goal is to analyse stock trend movement and train a Machine Learning model that can forecast the trend's value when the relevant prospective data is provided.

DATASET:

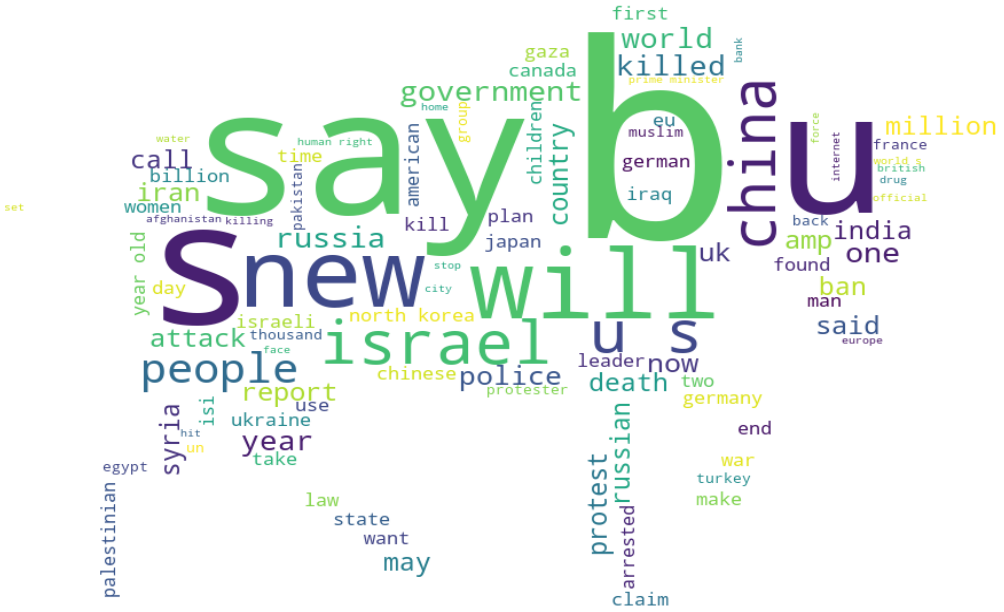
The dataset we're working with is a mix of Reddit news and the stock price of the Dow Jones Industrial Average (DJIA) from 2008 to 2016. From 2008 to 2016, the news dataset covers the top 25 stories on Reddit for each day. Each trading day's basic stock market information, such as Open, Close, and Volume, is contained in the DJIA. The dataset's label indicates whether the stock price increased (labelled as 1) or decreased (labelled as 0) on that particular day. The dataset has a total of 1989 days. We divided the dataset into train and test during our earliest trials. The train dataset makes up 80% of the entire dataset.

DATA SEGMENTATION AND CLEANING:

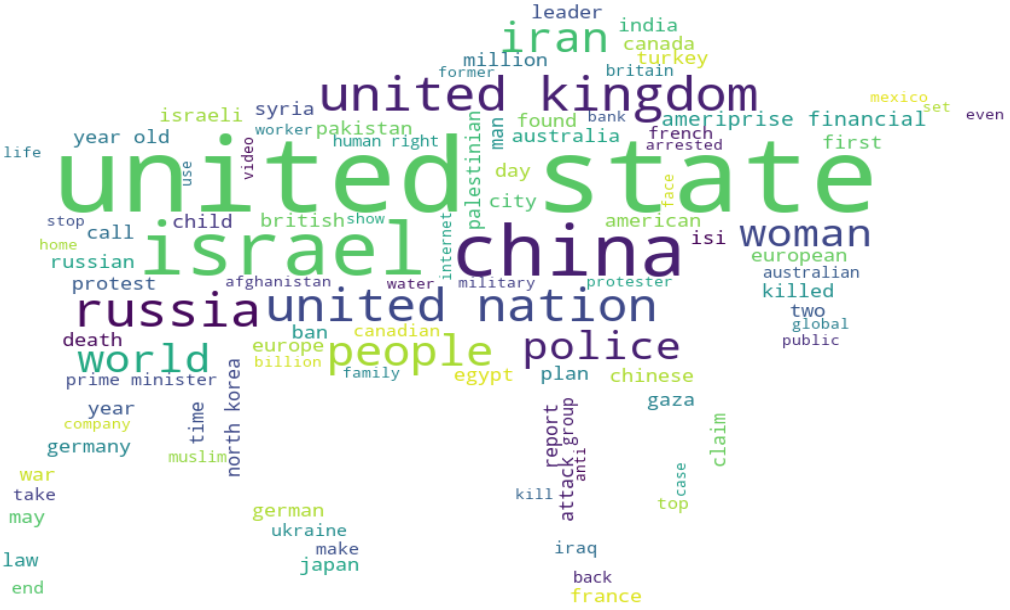
- We created a processed dataset for this project by gathering clear-cut data from the internet.
- Using the pandas data frame, we have loaded the data.
- By using regex expressions we cleaned the dataset by removing punctuations, spaces etc.
- By using the fillna we have filled all the cells with median values.
- We next tokenized the text data, using lemmatization and converting the text to lower case for each token.
- Finally we removed all the stop words present in text data.

EXPLORATORY DATA ANALYSIS:

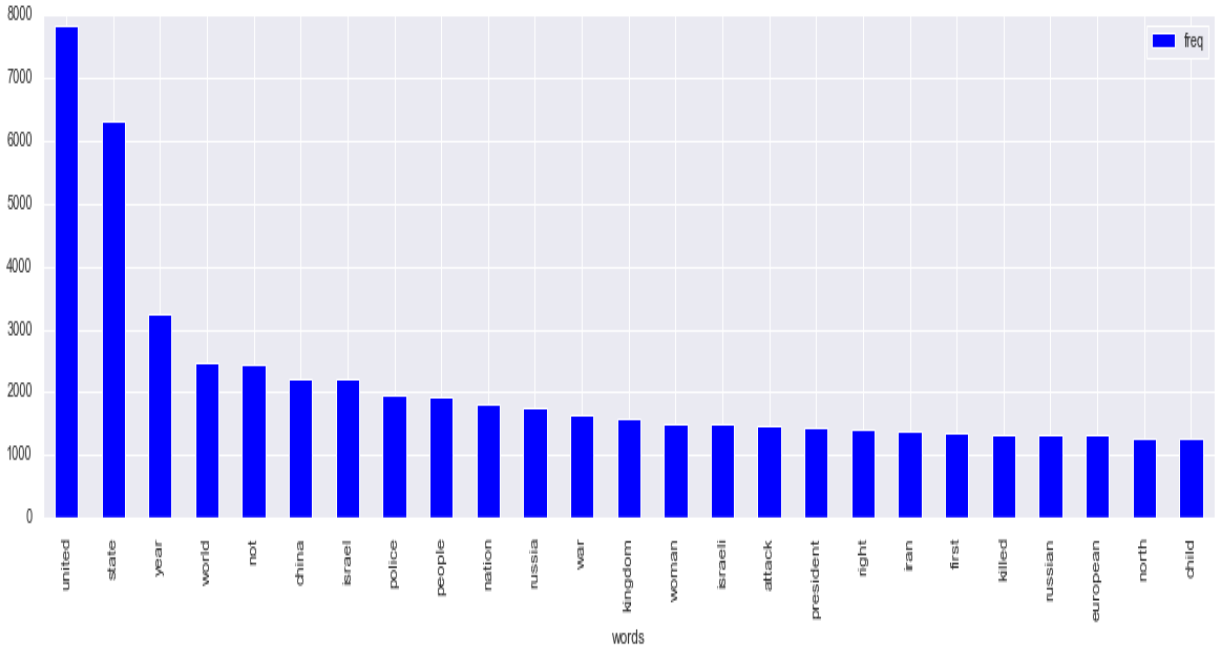
PLOT-1: News wordcloud before removing Stopwords



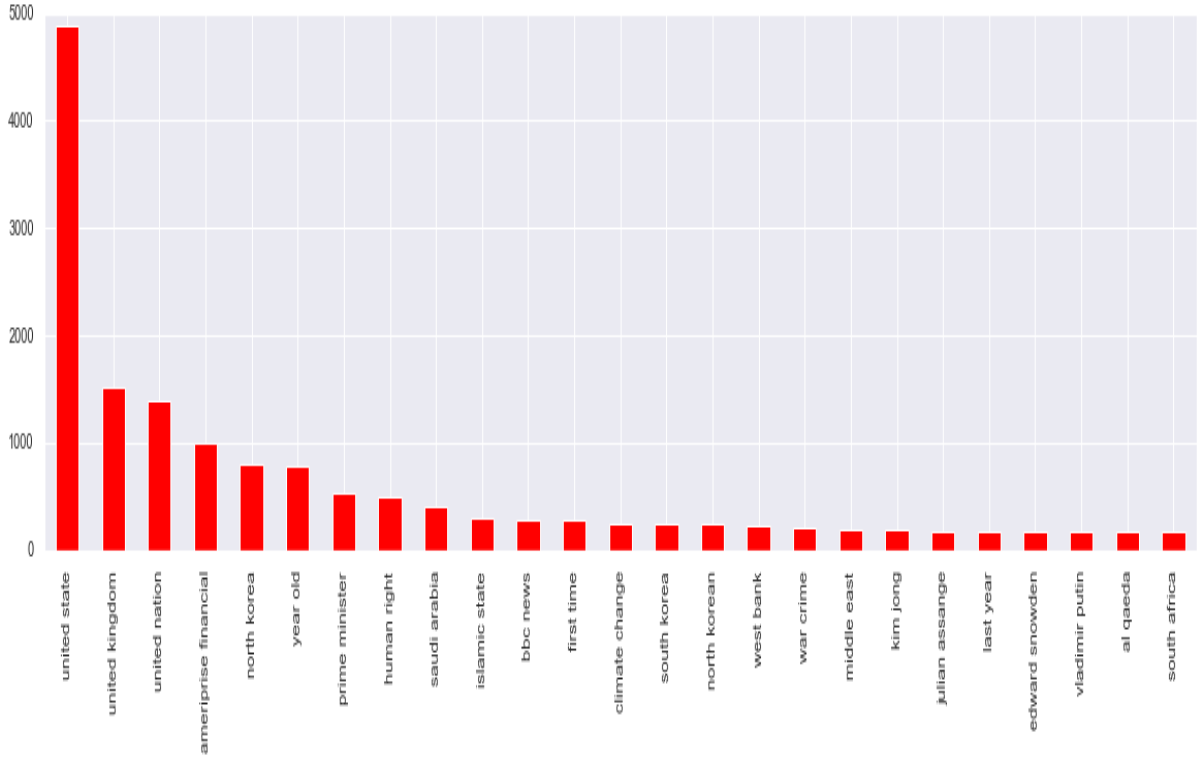
PLOT-2: News wordcloud after removing Stopwords



PLOT-3: Top 25 Unigram words in News after removing Stopwords



PLOT-4: Top 25 Bigram words in News after removing Stopwords



MODEL :-

1. Logistic Regression

In [46]:
```
from sklearn.linear_model import LogisticRegression
# initialize data pipeline
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
```

In [47]:
```
pipeline = Pipeline([
        ('vect', CountVectorizer(ngram_range=(1, 1))),
        ('tfidf', TfidfTransformer()),
        ('clf', LogisticRegression())
    ])
# fit on the pipeline
pipeline.fit(X_train, y_train)
y_pred = pipeline.predict(X_test)
print(classification_report(y_test, y_pred));
```

```
              precision    recall  f1-score   support

           0       0.89      0.72      0.80       186
           1       0.77      0.91      0.84       192

    accuracy                           0.82       378
   macro avg       0.83      0.82      0.82       378
weighted avg       0.83      0.82      0.82       378
```

Including Unigrams, Bigrams, Trigrams

In [49]:
```
pipeline = Pipeline([
        ('vect', CountVectorizer(ngram_range=(1, 3))),
        ('tfidf', TfidfTransformer()),
        ('clf', LogisticRegression())
    ])

# fit on the pipeline
pipeline.fit(X_train, y_train)

y_pred = pipeline.predict(X_test)

print(classification_report(y_test, y_pred));
```

```
              precision    recall  f1-score   support

           0       0.98      0.70      0.82       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
```

The accuracy of the Logistic Regression model using only unigrams is **82%** and has been increased to **85%** after including bigrams and trigrams also.

2. Support Vector Machine(SVM)

In [50]:
```
from sklearn.svm import SVC
```

In [51]:
```
pipeline = Pipeline([
        ('vect', CountVectorizer(ngram_range=(1, 1))),
        ('tfidf', TfidfTransformer()),
        ('clf', SVC(kernel='rbf', random_state = 4520))
    ])


# fit on the pipeline
pipeline.fit(X_train, y_train)

y_pred = pipeline.predict(X_test)

# import the class report function for benchmark model evaluation.
from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.97      0.70      0.82       186
           1       0.77      0.98      0.86       192

    accuracy                           0.84       378
   macro avg       0.87      0.84      0.84       378
weighted avg       0.87      0.84      0.84       378
```

Including Unigrams, Bigrams, Trigrams

In [53]:
```
pipeline = Pipeline([
        ('vect', CountVectorizer(ngram_range=(1, 3))),
        ('tfidf', TfidfTransformer()),
        ('clf', SVC(kernel='rbf', random_state = 4520))
    ])


# fit on the pipeline
pipeline.fit(X_train, y_train)

y_pred = pipeline.predict(X_test)

# import the class report function for benchmark model evaluation.
from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      0.70      0.82       186
           1       0.77      1.00      0.87       192

    accuracy                           0.85       378
   macro avg       0.89      0.85      0.85       378
weighted avg       0.89      0.85      0.85       378
```

The accuracy of the SVM model using only unigrams is **84%** and has been increased to **85%** after including bigrams and trigrams also.

Conclusion:-

- In this project we performed News Oriented Stock Price Trend Prediction.
- To train our model on Clean data, we did some simple data cleaning actions and deleted Stopwords.
- We used two models, Logistic Regression and Support Vector Machine(SVM), after looking over the clean data .
- For both the models we achieved the maximum accuracy of 85% , and this was achieved when we included all unigrams, bigrams and trigrams.

References:-

1. Dataset_Source:- https://data.world/finance/daily-news-for-stock-market

2. Image_Source:- https://www.vectorstock.com/royalty-free-vector/stock-market-bull-symbol-vector-14167666

3. Data Mining for Business Intelligence- by Galit Shmueli, Nitin R. Patel, Peter C. Bruce