Course Project

Authors: Joel Quek (SG)

Contents

Background - The Rise of Sentiment

Stock selection has evolved over the years - from the study of balance sheets Fundamental Analysis to the analysis of chart patterns in Technical Analysis. These two methodologies have been the bread and butter of hedge funds before the emergence of social media. However, with the emergence of content aggregation platforms like Twitter and Reddit, the Market was at the mercy of what is now known as 'Reddit Stocks' (https://finance.yahoo.com/news/12-best-reddit-stocks-invest-233132376.html#:~:text=Leading%20subreddits%20like%20r%2FWallStreetBets,respectively%2C%20as%20c

As an analyst in a hedge fund, I acknowledge the effect of public sentiment on price action, and how the general population is in fact a worthy opponent to the financial institutions when it comes to making market waves. Therefore, we now define a third methodology of stock selection - Sentiment Analysis.

Executive Summary

In the modern era where millenials dominate the thought-space, I propose to leverage on Sentiment Analysis as a worthy complement to Fundamental and Technical Analysis.

As an analyst in a hedge fund, I would like to leverage on the top investing and trading communities (https://www.investopedia.com/reddit-top-investing-and-trading-communities-5189322) on Reddit. The subreddit, r/wallstreetbets, is a clear favourite with more than 10 million members. However, the next in line would be two popular Subreddits

1. r/StockMarket (https://subredditstats.com/r/StockMarket) with 2,493,511 members
2. r/investing (https://subredditstats.com/r/investing) with 2,088,862 members

I would like to find out if these two subreddits are distinct in their content (ie if they are separable/classifiable through modelling), and if so, I would choose to perform Sentiment Analysis on both Subreddits during Stock Selection.

Conversely, if I discover that they are not distinct (ie not clearly separable/classifiable through modelling), then I would just pick one of the two Subreddits in my Sentiment Analysis.

Further Reading: Subreddit Descriptions (https://www.investing.com/academy/stocks/reddit-meme-stocks-to-buy/)

Problem Statement

I would create a few models to perform binary text classification on the two Subreddits. With the intention of attaining a good degree of separation between the two Subreddit classes. This is measured by the Receiver Operating Characteristics Area Under the Curve (ROC-AUC). ROC curve is a probability curve that shows model performance at all classification threshold with 2 parameters which is True Predicted Positive (TPR) and False Predicted Positive (FPR). The larger the area under the curve represents the larger the degree of separability between classes.

Exploratory Data Analysis

I will import the necessary charts in this section. For detailed scraping and EDA, please refer to the following Jupyter Notebooks

1. log-reg-model (Version 2).ipynb
2. random-forest-model (Version 2).ipynb
3. naive-bayes-model.ipynb

Import Libraries

In [1]:

```
#All libraries used in this project are listed here
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

import nltk
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import RegexpTokenizer
from nltk.corpus import stopwords

import re
from bs4 import BeautifulSoup

from sklearn.feature_extraction.text import CountVectorizer,TfidfVectorizer
from sklearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split, GridSearchCV,cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import roc_auc_score, make_scorer, recall_score, precision_score,acc
```

Cleaning Scraped Datasets

Please Refer to the Jupyter Notebook 'reddit-scrape.ipynb' for the full scraping code.

I will describe briefly, the stages of cleaning in my EDA.

In [2]:

```
investing_df = pd.read_csv('datasets/investing.csv')
stockmarket_df = pd.read_csv('datasets/stockmarket.csv')
```

In [4]:

```
investing_df['selftext'].head()
```

Out[4]:

```
0    Hey guys, I’m a 22 year old male, I grew up wi...
1                                            [removed]
2                                            [removed]
3                                            [removed]
4    Hello Redditors 👋 \n\nI work as a Investment C...
Name: selftext, dtype: object
```

In [5]:

```
stockmarket_df['selftext'].head()
```

Out[5]:

```
0                                                   NaN
1    [Link to the full article (4 min read)](https:...
2                                                   NaN
3                                                   NaN
4                                                   NaN
Name: selftext, dtype: object
```

- As you can see from the above datasets, there were a lot of posts with [removed] and NaN. I removed these together with digits an non-letters.
- Missing values were removed as there is no logical way to impute non numerical data.
- I also removed stop words and hot encoded the target vector ('investing': 0, 'StockMarket': 1)

Model Evaluation

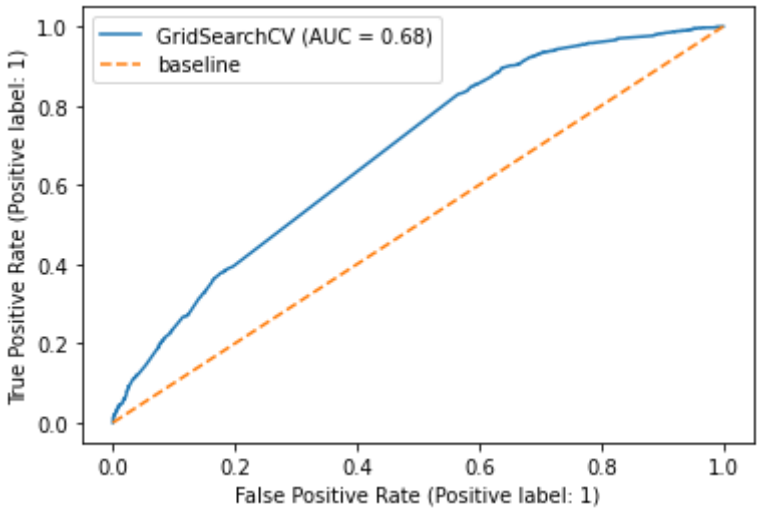
Stats	Logistic Regression (CVEC)	Logistic Regression (TVEC)	Random Forest (CVEC)	Random Forest (TVEC)	Naive Bayes (CVEC)	Naive Bayes (TVEC)
Recall/Sensitivity	0.7036	0.7222	0.76	0.368	0.534	0.681
Precision	0.6402	0.718	0.778	0.629	0.834	0.773
TP	952	2020	2168	1048	1320	1683
TN	2980	2725	2521	2896	2376	2144
FP	535	790	994	619	263	495
FN	1899	831	683	1803	1153	790

Observations from Model Evaluation

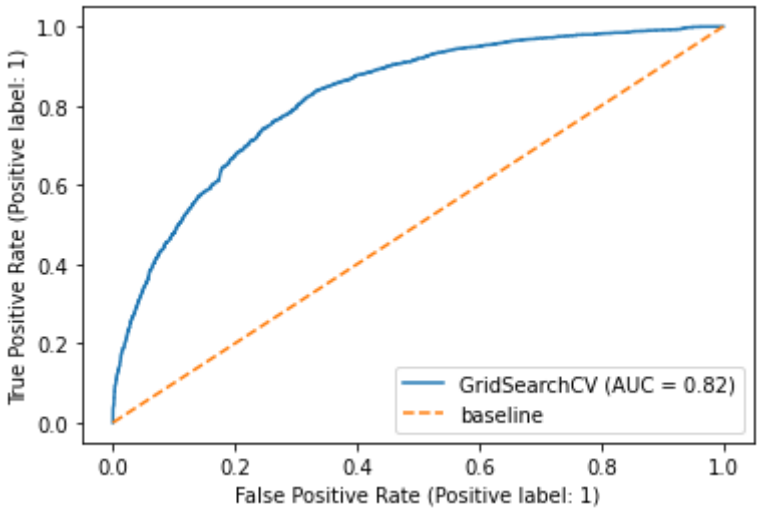
- We can see from the stats that both Logistic Regression (TVEC) and Random Forest (CVEC) models performed similarly. While Naive Bayes (TVEC) performed relatively well.
- In our case, precision score is important as we want to accurately identify r/StockMarket posts. In this case, our models have only slight differences in our most scores.

ROC AUC Scores

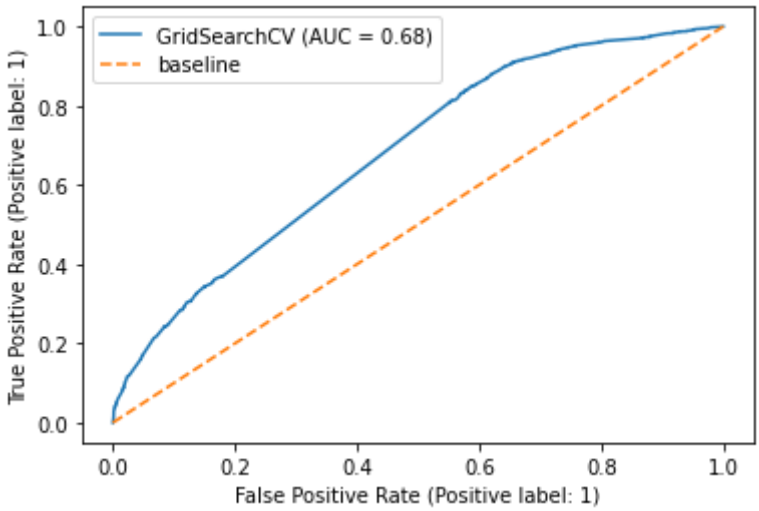
Logistic Regression (Count Vectorizer)



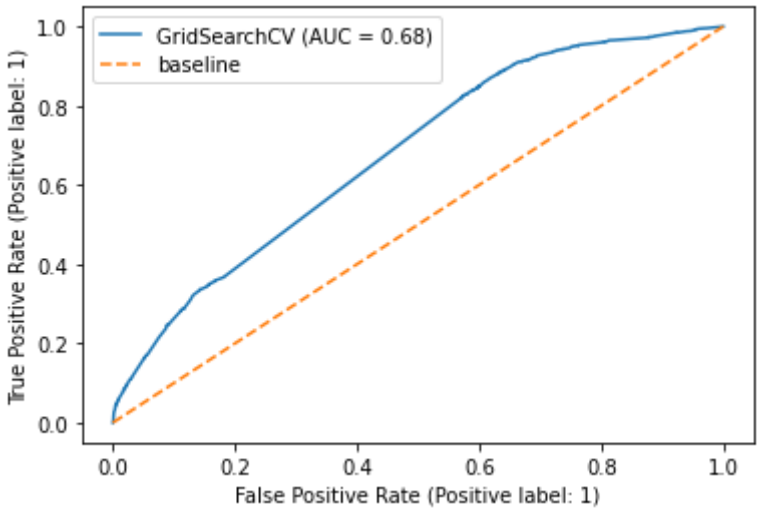
Logistic Regression (TFID Vectorizer)



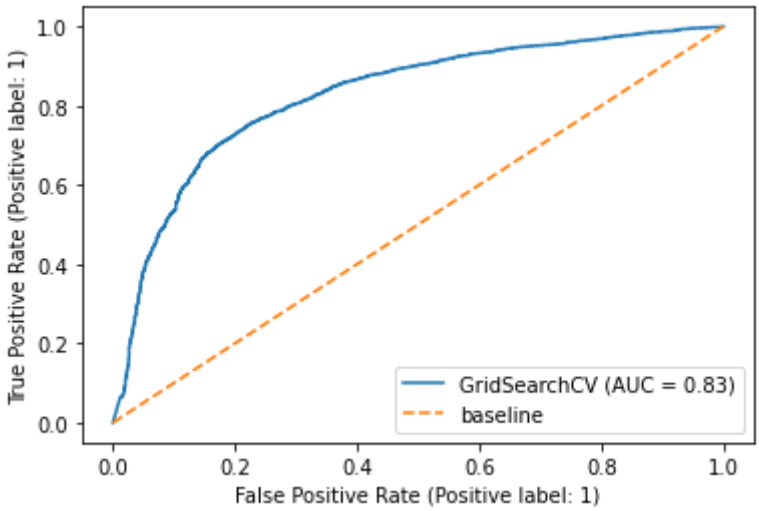
Random Forest (Count Vectorizer)



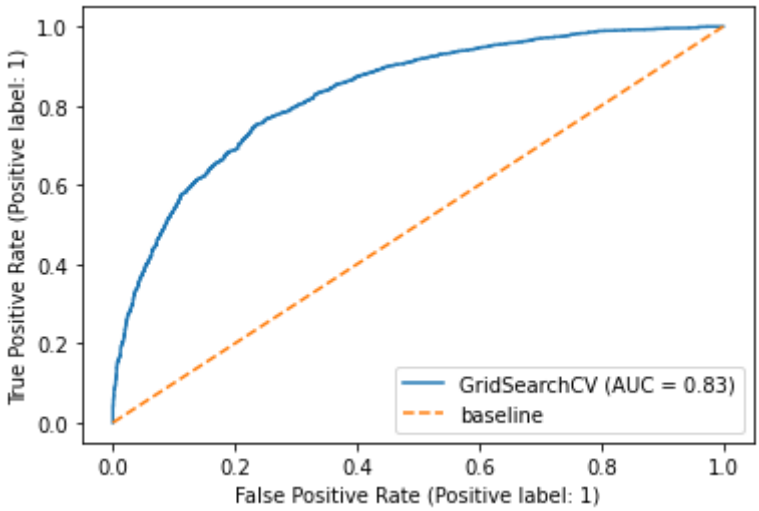
Random Forest (TFID Vectorizer)



Naive Bayes (Count Vectorizer)



Naive Bayes (TFID Vectorizer) ¶



Interpreting ROC Curve

The more area under a curve means better better separated our distributions our model give. When our ROC AUC is closer to 1, then our positive and negative populations are better separated which means the model is better. From this graph, we can see that Logistic Regression gives a much better curve.

Conclusions and Recommendations

The final model chosen is Naive Bayes with Count or TFID Vectorizer

- From the model stats and ROC AUC curve, Naive Bayes Models (both CVEC and TVEC) and Logistic Regression with TFID Vectorizer performed the best. But NAive Bayes had a slightly higher ROCAUC score of 0.83
- We can look at other models (KNN) to see if they can do better than our current models.
- To further build on this project, we can look at sentiment analysis on the 2 topics. We can also look at specific topics in each subreddit that are unique.