Abgeschlossen

Project description Python for Finance 2

Celina Junghans 12026256
Sinan-Leon Canbolat 12116162
Manuela Wieser 11919411

1. Motivation

The primary objective of this project is to predict whether the NASDAQ index has risen or fallen based on the analysis of New York Times (NYT) articles from the previous day. We aim to leverage features such as sentiment scores and named entity recognition to enhance the accuracy of our predictions. The project involves implementing three machine learning models to achieve this goal, Perceptron, DecisionTree Classifier and Support Vector Machines.

2. Data Description

2.1 Nasdaq Data

We obtained historical data for the NASDAQ index using the yfinance module from Yahoo Finance. The data includes Open and Close prices, which were used to calculate the daily price difference. Based on the difference, we assigned labels of 0 (fall) or 1 (rise) to each trading day.

```
#getting data from yahoo finance

import yfinance as yf
nasdaq = yf.Ticker("^IXIC")
```

Fig. 1: yfinance ticker

	Open	High	Low	Close	Volume	Dividends	Stock Splits
2022-01-03	15732.500000	15832.799805	15644.089844	15832.799805	4429960000	0.0	0.0
2022-01-04	15852.139648	15852.139648	15512.410156	15622.719727	5131110000	0.0	0.0
2022-01-05	15547.160156	15586.299805	15095.179688	15100.169922	5031850000	0.0	0.0
2022-01-06	15024.150391	15198.450195	14914.870117	15080.860352	4790820000	0.0	0.0
2022-01-07	15095.719727	15171.019531	14877.629883	14935.900391	4238070000	0.0	0.0

Fig. 2:Dataframe of daily NASDAQ OHLC

2.2 NYT Data

For gathering NYT data, we utilized the NYT API. Our analysis focused on three sections of the articles: headlines, lead paragraphs, and the associated entities (named entity recognition).
To preprocess the data, we grouped NYT articles by their publication date and then aligned this data with the NASDAQ data, ensuring compatibility between the two datasets. We restricted our analysis to the year 2022. We end up with

	headline	lead_paragraph	pub_date
0	{'main': 'Alabama Rolls Past Cincinnati, 27-6,...	ARLINGTON, Texas — This was the moment the Cin...	2022-01-01T00:01:35+0000
1	{'main': 'People Magazine's Premature Cover To...	For months, editors at People magazine had bee...	2022-01-01T00:16:10+0000
2	{'main': 'Betty White Recalled as a Trailblaze...	Television stars, comedians, a president and s...	2022-01-01T00:48:48+0000
3	{'main': 'Line Just Before a Comma', 'kicker':...	SATURDAY PUZZLE — One day, when the universe s...	2022-01-01T03:00:05+0000
4	{'main': 'A Renewed Georgia Pummels Michigan t...	It was supposed to be Friday's prizefight, the...	2022-01-01T04:00:35+0000

Fig. 3: NYT article dataframe

3. Methods

3.1 Text Processing

We conducted a preprocessing step by eliminating common stopwords and converting words to lowercase. Subsequently, we tokenized the content of the lead paragraph. This was followed by applying lemmatization alongside part-of-speech (POS) tagging to the individual tokens.

3.2 Sentiment Analysis

We employed the SentimentIntensityAnalyzer() to extract sentiment features from both headlines and lead paragraphs. The analyzer provided polarity scores, including negative, neutral, positive, and compound. The compound score indicates the overall sentiment, where higher values correspond to more positive sentiments and vice versa.

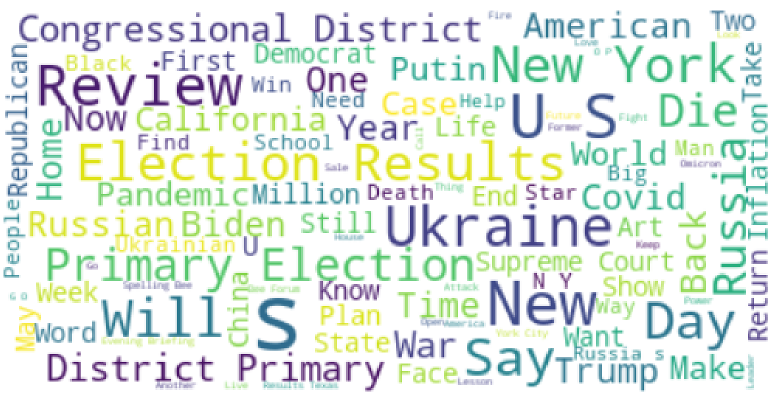

Fig. 4: Headlines WordCloud

```
array([[ 0.072 ,  0.841 ,  0.087 ,  0.9976],
       [ 0.078 ,  0.824 ,  0.098 ,  0.999 ],
       [ 0.088 ,  0.819 ,  0.092 ,  0.8747],
       [ 0.087 ,  0.826 ,  0.086 , -0.8937],
       [ 0.08  ,  0.808 ,  0.112 ,  0.9997],
       [ 0.073 ,  0.838 ,  0.09  ,  0.9987],
       [ 0.08  ,  0.829 ,  0.091 ,  0.9915],
       [ 0.093 ,  0.804 ,  0.103 ,  0.9962],
       [ 0.088 ,  0.806 ,  0.105 ,  0.9989],
       [ 0.089 ,  0.812 ,  0.099 ,  0.9976],
       [ 0.076 ,  0.823 ,  0.101 ,  0.9995],
       [ 0.093 ,  0.812 ,  0.095 ,  0.9035],
       [ 0.099 ,  0.822 ,  0.08  , -0.9995],
       [ 0.079 ,  0.814 ,  0.107 ,  0.9997],
       [ 0.079 ,  0.824 ,  0.098 ,  0.999 ],
       [ 0.085 ,  0.823 ,  0.092 ,  0.9946],
```

Fig. 5: Sample polarity score

Figure 5 illustrates the distinct polarity levels encompassed by the sentiment analysis scores: Negative, neutral and positive compound. The compound score exhibits a polarity continuum. A higher compound score corresponds to a higher positive sentiment, whereas a lower value aligns with a higher negative sentiment. A score proximate to zero indicates a state of neutrality.

3.3 Named Entity Recognition (NER)

Utilizing the spacy module, we performed NER on the NYT articles to identify entities such as organizations, persons, and dates. We further preprocessed these entities to create a feature set that complements the sentiment features.

Fig. 6: NER output

Figure 6 depicts the output of Named Entity Recognition (NER), which served as the foundation for generating a new feature set. This new feature set underwent a preprocessing procedure similar to the methodology applied to standard words within the articles.

4) Models

4.1) Perceptron

As a first classification method, we adopted the perceptron model as our primary methodology. By employing diverse partitions of the dataset into training and test subsets, we embarked upon the exploration of model performance. To fine-tune the perceptron's hyperparameters, we leveraged the GridSearch approach, focusing specifically on the parameters 'max_iter' and 'eta0'. The parameter grid configuration was outlined as follows:

```
param_grid = {
    'max_iter': [10, 20, 30, 40, 50, 100, 500, 1000],
    'eta0': [0.00000000000000000000000001, 0.000000000001, 0.0000001, 0.000001, 0.01, 0.05, 0.1, 0.2]
}
```

In assessing the efficacy of our approach, we employed the accuracy score metric, which facilitated a comparison between the anticipated labels of the test set and their actual counterparts. Notably, as we varied the proportions of training data, the ensuing optimal parameter configurations were as follows:

```
Train size: 70, Best max_iter: 10, Best eta: 1e-26, Accuracy: 0.59
Train size: 75, Best max_iter: 10, Best eta: 1e-26, Accuracy: 0.54
Train size: 80, Best max_iter: 10, Best eta: 1e-26, Accuracy: 0.50
Train size: 85, Best max_iter: 20, Best eta: 0.05, Accuracy: 0.45
Train size: 90, Best max_iter: 10, Best eta: 1e-26, Accuracy: 0.40
```

4.2) Decision Tree

As the second method, we used the decision tree for classification. As before, we employed several partitions of the dataset into training and test subsets and used GridSearch to optimize the model performance. In the GridSearch approach, we focused on the parameters 'max_depth', ‘min_samples_split’, and 'min_samples_leaf'. The parameter grid configuration was outlined as follows:

```
param_grid = {
    'max_depth': [2, 4, 6, 8],           # Different values for max_depth
    'min_samples_split': [2, 5, 10],     # Different values for min_samples_split
    'min_samples_leaf': [1, 2, 4]        # Different values for min_samples_leaf
}
```

Just as before, we used the accuracy score to evaluate the model performance, with the following results for different training sizes:

```
Train size: 70, Best max_depth: 8, Best min_samples_split: 2, Best min_samples_leaf: 2, Accuracy: 0.57
Train size: 75, Best max_depth: 6, Best min_samples_split: 2, Best min_samples_leaf: 2, Accuracy: 0.59
Train size: 80, Best max_depth: 4, Best min_samples_split: 2, Best min_samples_leaf: 4, Accuracy: 0.66
Train size: 85, Best max_depth: 6, Best min_samples_split: 10, Best min_samples_leaf: 4, Accuracy: 0.55
Train size: 90, Best max_depth: 6, Best min_samples_split: 2, Best min_samples_leaf: 1, Accuracy: 0.56
```

4.3) Support Vector machines

As the third method, we used the Support vector machine for binary classification. As before, we employed several partitions of the dataset into training and test subsets and used

GridSearch to optimize the model performance. In the GridSearch approach, we focused on the parameters 'kernel', 'C', and 'gamma'. The parameter grid configuration was outlined as follows:

```
param_grid = {
    'kernel': ['linear', 'rbf'],          # Different kernels to try
    'C': [0.001, 0.1, 1, 10],             # Different values of C
    'gamma': ['scale', 'auto', 0.1, 1]    # Different values of gamma
}
```

Just as before, we used the accuracy score to evaluate the model performance, with the following results for different training sizes:

```
Train size: 70, Best kernel: rbf, Best C: 1, Best gamma: auto, Accuracy: 0.40
Train size: 75, Best kernel: linear, Best C: 0.001, Best gamma: scale, Accuracy: 0.57
Train size: 80, Best kernel: linear, Best C: 10, Best gamma: scale, Accuracy: 0.54
Train size: 85, Best kernel: linear, Best C: 0.001, Best gamma: scale, Accuracy: 0.55
Train size: 90, Best kernel: linear, Best C: 0.001, Best gamma: scale, Accuracy: 0.60
```

5) Results and interpretation

The culmination of our analysis yielded an optimum prediction accuracy of 66%. Notably, this achievement was attained through the application of the decision tree method, specifically with a training size of 80 and the hyperparameters: max_depth: 4, min_samples_split: 2, and min_samples_leaf: 4. Remarkably, the decision tree method emerged as the most proficient model overall.

Conversely, the alternative methods utilized, namely perceptron and support vector machine, displayed varying prediction accuracies, spanning the range of 40% to 60%, contingent upon distinct hyperparameter configurations. Evidently, the perceptron exhibited the weakest performance among these models.

Regrettably, the cumulative outcome of our analysis falls short of our expectations. A potential avenue for enhancing the efficacy of our models involves augmenting the sample size. Notably, our models were trained on a subset of 2022 articles, with a sample size of 250. This aspect underscores the potential for improvements in achieving more robust and satisfactory results.

6. Conclusion

In summary, our project demonstrates the potential of utilizing sentiment analysis and named entity recognition as features for predicting stock market trends based on NYT articles. The combination of sentiment analysis and NER provides valuable insights, however we were unable to improve the accuracy significantly.
There are opportunities for further enhancement and future work could focus on exploring additional feature engineering techniques, experimenting with different machine learning algorithms, and extending the analysis to longer time frames.