

# Yahoo Finance and Crash Like Dates

Allana Coleman

April 28, 2024

## Abstract

In this programming assignment we were to use Python and Yahoo Finance to predict crash dates and use supervised learning models.

## 1 Summary

This project consisted of 7 sections. 1-3 consisted of scraping Yahoo Finance using Python, gathering the information for a specific "ticker"(Stock), saving it to an excel document, and visualizing the data. Sections 4-5 consisted of identifying dates with large decreases in closing price, determining a "K" value, creating a new data frame that will identify columns of 10 consecutive stock close prices and identify the row as 0 - no crash like dates or 1 - crash like dates.

## 2 Methodology

### 2.1 Exploratory Data Analysis(EDA)

Exploratory Data Analysis(EDA) or Data Exploration is a step in the process to understand the data being worked with. Understanding the dataset includes: identifying useful variables and discarding the others, identifying erroneous and or missing data, and understanding the relationship between the variables. This important step helps to have clean and understandable data before manipulation.

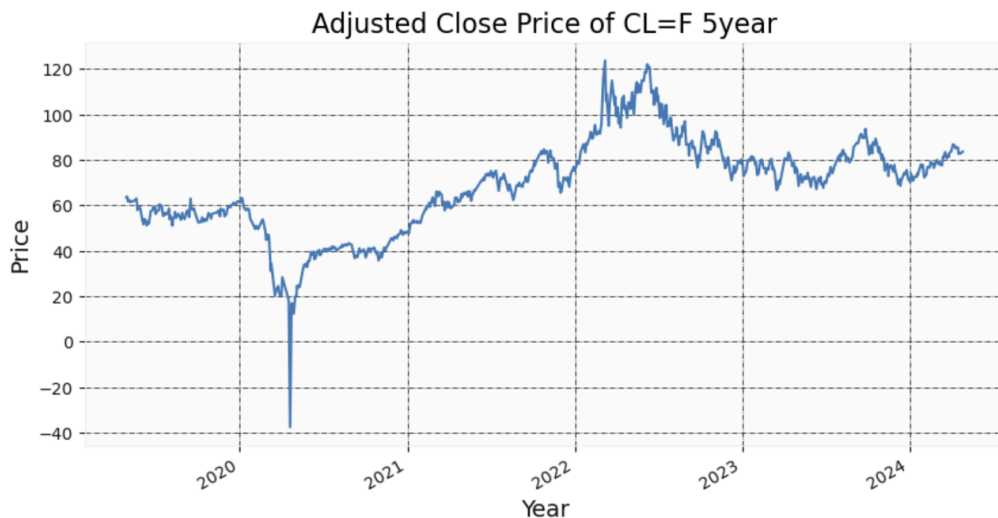


Figure1 – 5YearClosePricesforCrudeOil

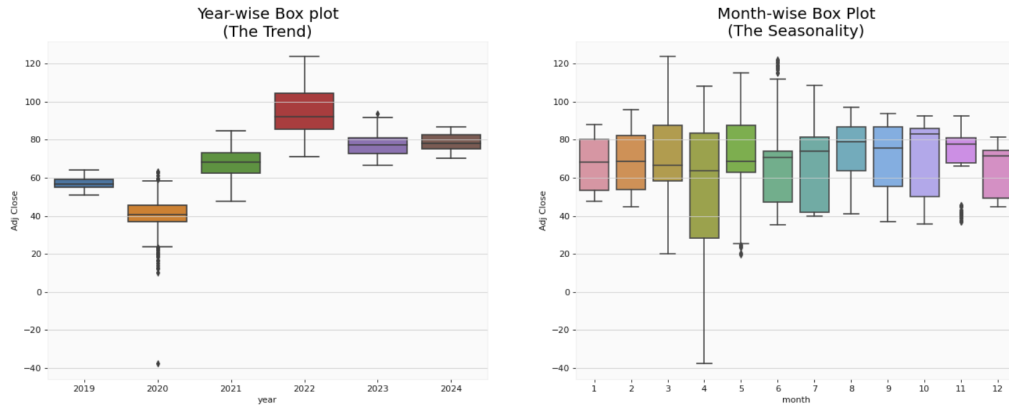


Figure2 – SeasonalityBoxPlots

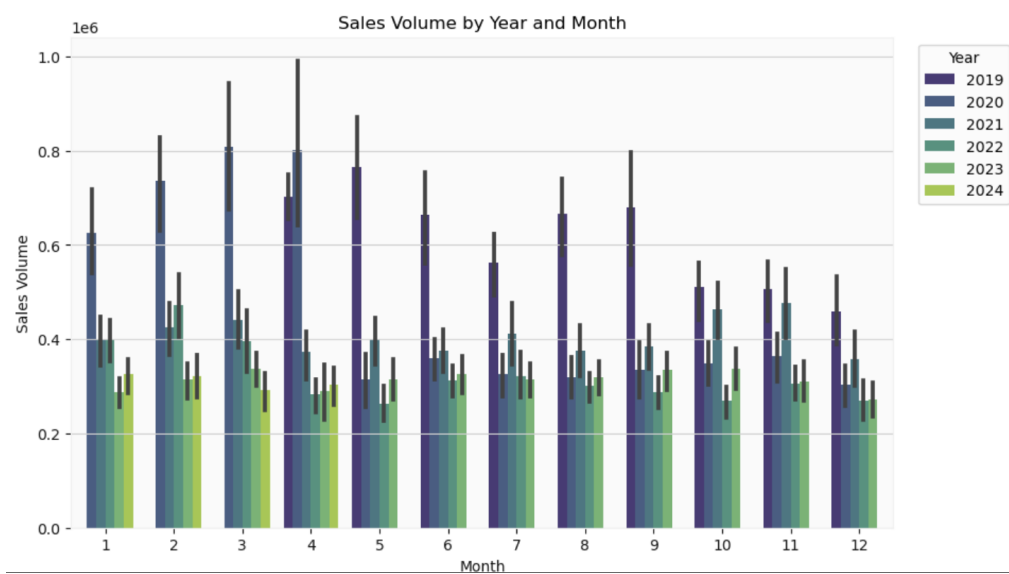


Figure3 – SalesVolume

## 2.2 Grouping and Manipulating the Dataset

Grouping the dataset is a great function of the Pandas library in Python, that allows us to group the data frame by the index name, and plot or analyze any other desired correlation data. In this project, resetting the index to the, "Date" attribute was most useful as this data was based on dates and adjusted close prices for a particular stock. We had to identify "crash-like" dates in this dataset by using a K value that would identify when the stock price dropped a certain (-K) value. Out of 1251 data points 126 crash dates were identified using this K value.

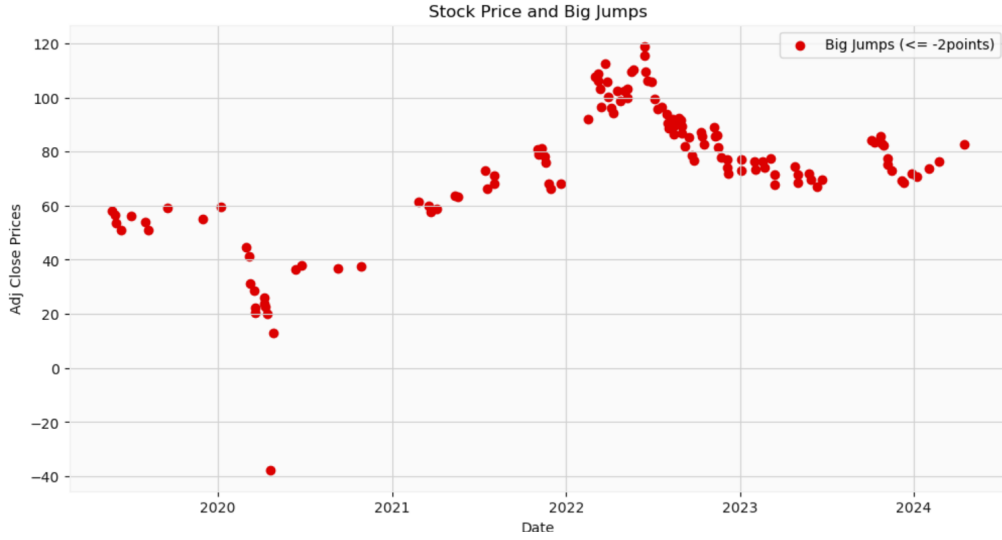


Figure4 – *CrashLikeDates*

### 2.3 Supervised Learning Models(SVMs)

Logistic Regression is an algorithm used in solving classification problems, used for predictive analysis that describes data and explains the relationship between variables. LR uses the sigmoid function, which is an S-shaped curve to find the relationship between variables 0 and 1.

Decision Tree Classifier is used in solving classification and regression tasks. It uses a flowchart-like tree structure where the internal nodes are features, the branch signify the rules and the leaf denotes the result of the algorithm. This process is also used in Random Forest which is a useful and powerful algorithm that trains different subsets of training data.

A Neural Network is a model that mimics processing data similarly as the human brain. Neural Networks rely on training data to learn and improve accuracy over time. Neural Networks are deep learning models within machine learning.

## 3 Results

In our dataset 0 represents a non crash-like date and 1 represents crash-like dates. The following Supervised Learning models were used to test the dataset, Logistic Regression, Decision Tree Classifier, Random Forest Classifier, and MLP Classifier which is a Neural Network SVM.

The classification report contains: Precision which is the ratio of correctly predicted positive observations to the total predicted positives Recall is the ratio of correctly predicted positive observations to all observations in a class F1-score is the weighted average of Precision and Recall. It takes both false positives and false negatives into account. It is a good way to show that a classifier has a good value for both recall and precision Support is the actual number of actual occurrences of the class in the dataset. Accuracy is the ratio of correctly predicted observations. Confusion matrix is a table used to describe the performance of the model TN: True Negative — FP: False Positive FN: False Negative — TP: True Positive

```

----- LogisticRegression -----
Classification Report:
              precision    recall  f1-score   support

         0       0.66       0.75       0.70       115
         1       0.76       0.68       0.72       136

   accuracy          0.71
  macro avg       0.71       0.71       0.71
weighted avg       0.72       0.71       0.71

Confusion Matrix:
[[86 29]
 [44 92]]

```

Figure5

This logistic regression model is under-performing as the precision is low for non-crash-like dates at 0.66 and the overall accuracy of the model is 0.71

```

----- DecisionTreeClassifier -----
Classification Report:
              precision    recall  f1-score   support

         0       0.74       0.83       0.79       115
         1       0.84       0.76       0.80       136

   accuracy          0.79
  macro avg       0.79       0.80       0.79
weighted avg       0.80       0.79       0.79

Confusion Matrix:
[[ 96  19]
 [ 33 103]]

```

Figure6

This model performed better than the Logistic Regression as there are improvements to precision in both 0 and 1 and the overall accuracy has increased to 0.79.

```

----- RandomForestClassifier -----
Classification Report:
              precision    recall  f1-score   support

         0       0.79      0.90      0.84        115
         1       0.90      0.80      0.85        136

   accuracy          0.84        251
  macro avg       0.85      0.85      0.84        251
weighted avg       0.85      0.84      0.84        251

Confusion Matrix:
[[103  12]
 [ 27 109]]

```

Figure7

Random Forest performed better than Decision Tree and Logistic Regression as the precision also increased from 0.74 to 0.79 for 0 and 0.84 to 0.90 for 1. The overall accuracy also increased to 0.84

```

----- MLPClassifier -----
Classification Report:
              precision    recall  f1-score   support

         0       0.73      0.90      0.81        115
         1       0.90      0.71      0.80        136

   accuracy          0.80        251
  macro avg       0.81      0.81      0.80        251
weighted avg       0.82      0.80      0.80        251

Confusion Matrix:
[[104  11]
 [ 39  97]]

```

Figure8

The Neural Network performed not as good as the other models as the recall ratios decreased in percentage and precision decreased in comparison to the other models.

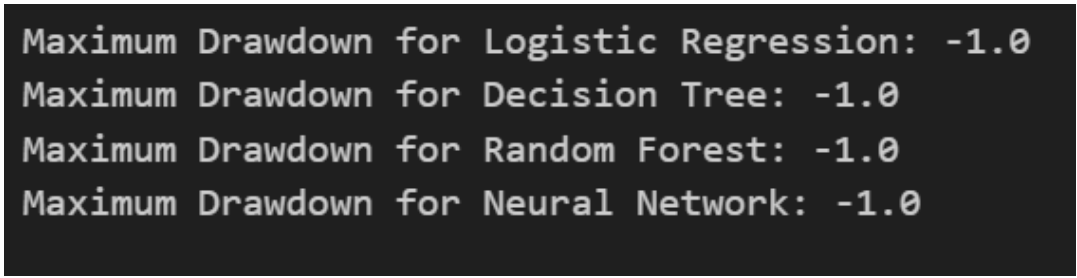
From these results one can conclude that Random Forest was the best model for the dataset and provided more accuracy overall in all categories of the results.

### 3.1 Optimization: Maximum DrawDown(MDD)

The maximum distance from a relative maximum to a relative minimum found in a specific time range of time-series data, defines the Maximum DrawDown (MDD). The MDD is a measure of risk and is used to evaluate the performance of Trading systems and portfolios. The MDD's significance is it represents the maximum loss that a trader or investor would experience if going to market and exiting at the worst time(low-prices).

The MDD depends on: the average return, the standard deviation, the sampling frequency, the sequence of all the values of the series, and the width of the time range.

From a statistical perspective MDD can not be shaped to a Gaussian probability distribution, an ideal Brownian motion time-series probability distribution is similar to log-normal distributions, which are asymmetric random variables that are positive only.



```
Maximum Drawdown for Logistic Regression: -1.0
Maximum Drawdown for Decision Tree: -1.0
Maximum Drawdown for Random Forest: -1.0
Maximum Drawdown for Neural Network: -1.0
```

Figure9

The results for the Maximum DrawDown performed state there are no drawdowns and for all of them to be -1.0 is highly unlikely. There are many problems with the code and calculations for the MDD as I do not understand how to implement the method that well in Python. The code was generated through AI and users from StackOverflow have yet to offer suggestions to help make the code more accurate.

## 4 Conclusion

The code produced from this project can be used again to improve upon itself and find better methods and ensure that grouping the data and analysis the data is accurate. Also to further improve this project and results, the MDD code must be revisited and improved upon, and possibly rewritten.

## 5 References:

- [1]<https://towardsdatascience.com/an-extensive-guide-to-exploratory-data-analysis-ddd99a03199e>.
- [2]<https://www.dataquest.io/blog/portfolio-project-predicting-stock-prices-using-pandas-and-scikit-learn/>
- [3]<https://dev.to/bshadmehr/navigating-financial-insights-analyzing-stock-data-with-python-and-visualization-hd6>
- [4]<https://datagy.io/pandas-groupby/>
- [5]<https://towardsdatascience.com/forecasting-time-series-data-stock-price-analysis-324bcc520af5>.
- [6]<https://www.overleaf.com>
- [7]<https://www.mpinvestit.it/en/predicting-the-drawdown-a-machine-learning-model-to-measure-risk-part-1/>
- [8]<https://medium.com/axum-labs/logistic-regression-vs-support-vector-machines-svm-c335610a3d16>
- [9]<https://www.geeksforgeeks.org/decision-tree/>
- [10]