Operation Stonks

By:
Pathak Siddhant (U2023715K)
Gupta Anant (U2023593G)
Hariharan Dhruv (U2023933G)
Team: 1
Lab Group: FSP8
Tutor Name: Hou Jingwen

TABLE OF CONTENTS

01

INTRODUCTION

Dataset, API,
Problem Definition

02

OUR PROCESS

Machine Learning
models used

03

KEY OUTCOMES AND DATA DRIVEN INSIGHTS

Results, Trends,
Learnings

INTRODUCTION

01

PROBLEM DEFINITION

We aim to prepare a useful model to determine whether to invest in a specific company in the next month, given its past data.

COLLECTION AND CURATION OF THE DATASET

The dataset has been obtained from the Alpha Vantage Stock API. The Alpha Vantage API is a method to obtain historical and real time data for several markets.

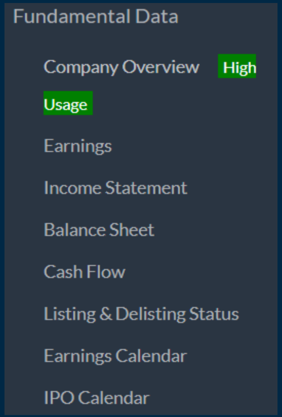
It is a dynamic dataset, i.e. its information is periodically updated on a daily basis. It required importing of the Alpha Vantage API as a Python module.

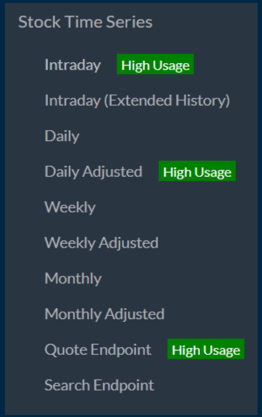
```
pip install alpha_vantage
```

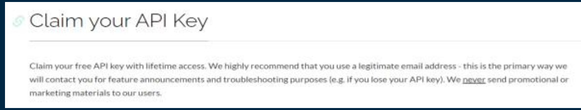
```
Requirement already satisfied: alpha_vantage in c:\programdata\anaconda3\lib\site-packages (2.3.1)
Requirement already satisfied: requests in c:\programdata\anaconda3\lib\site-packages (from alpha_vantage) (2.24.0)
Requirement already satisfied: aiohttp in c:\programdata\anaconda3\lib\site-packages (from alpha_vantage) (3.7.4.post0)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in c:\programdata\anaconda3\lib\site-packages (from requ
ests->alpha_vantage) (1.25.11)
Requirement already satisfied: idna<3,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests->alpha_vantage) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests->alpha_vantage)
(2020.6.20)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\programdata\anaconda3\lib\site-packages (from requests->alpha_vantage)
(3.0.4)
Requirement already satisfied: attrs>=17.3.0 in c:\programdata\anaconda3\lib\site-packages (from aiohttp->alpha_vantage) (20.3.
0)
Requirement already satisfied: multidict<7.0,>=4.5 in c:\programdata\anaconda3\lib\site-packages (from aiohttp->alpha_vantage)
(5.1.0)
Requirement already satisfied: yarl<2.0,>=1.0 in c:\programdata\anaconda3\lib\site-packages (from aiohttp->alpha_vantage) (1.6.
3)
Requirement already satisfied: async-timeout<4.0,>=3.0 in c:\programdata\anaconda3\lib\site-packages (from aiohttp->alpha_vanta
ge) (3.0.1)
Requirement already satisfied: typing-extensions>=3.6.5 in c:\programdata\anaconda3\lib\site-packages (from aiohttp->alpha_vant
age) (3.7.4.3)
Note: you may need to restart the kernel to use updated packages.
```

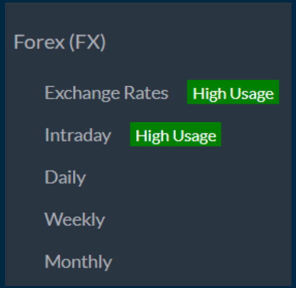
MORE ABOUT THE ALPHA VANTAGE API

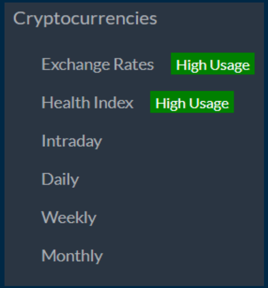
You can access the data directly in Python or any other programming language of your choosing. From there, you can manipulate the data or store it for later use. Alpha Vantage proudly offers its service for free. They provide a generous rate limit of 5 requests per minute and 500 requests per day. In addition to price data, there are more than 50 technical indicators available as well as performance data for 10 US equity sectors.











THE ONE WE USED

TIME_SERIES_MONTHLY

This API returns monthly time series (last trading day of each month, monthly open, monthly high, monthly low, monthly close, monthly volume) of the global equity specified, covering 20+ years of historical data.

API Parameters

❚ Required: `function`

The time series of your choice. In this case, `function=TIME_SERIES_MONTHLY`

❚ Required: `symbol`

The name of the equity of your choice. For example: `symbol=IBM`

❚ Optional: `datatype`

By default, `datatype=json`. Strings `json` and `csv` are accepted with the following specifications: `json` returns the monthly time series in JSON format; `csv` returns the time series as a CSV (comma separated value) file.

❚ Required: `apikey`

Your API key. Claim your free API key here.

DATASET AT A GLANCE

The dynamically-changing dataset is obtained with the help of the Alpha Vantage Stock API.

	1. open	2. high	3. low	4. close	5. volume
date					
1999-12-31	101.00	118.000	91.060	102.81	8.409120e+07
2000-01-31	104.87	121.500	86.500	103.75	1.120998e+08
2000-02-29	104.00	119.940	97.000	114.62	6.535520e+07
2000-03-31	118.56	150.380	114.000	135.81	7.766390e+07
2000-04-28	135.50	139.500	104.870	124.06	7.734290e+07
...	...	...	...	...	...
2020-12-31	121.01	138.789	120.010	132.69	2.319688e+09
2021-01-29	133.52	145.090	126.382	131.96	2.239366e+09
2021-02-26	133.75	137.877	118.390	121.26	1.825487e+09
2021-03-31	123.75	128.720	116.210	122.15	2.650845e+09
2021-04-16	123.66	135.000	122.490	134.16	9.670884e+08


```
{'1. Information': 'Monthly Prices (open, high, low, close) and Volumes',
 '2. Symbol': 'AAPL',
 '3. Last Refreshed': '2021-04-16',
 '4. Time Zone': 'US/Eastern'}
```

DESCRIPTION OF THE COLUMNS

1.open - the price at which a stock first trades upon the opening of an exchange on a trading day/month.

2.high - the highest closing price of a stock over a given period, in this case, a month.

3.low - the lowest closing price of a stock over a given period, in this case, a month.

4.close - the final price at which a stock trades upon the closing of the exchange on a trading day/month.

INTRODUCING ESSENTIAL VARIABLES

	1. open	2. high	3. low	4. close	5. volume	date_time	Volatility	MonthDiff	Invest?
date									
2020-12-31	121.01	138.789	120.010	132.69	2.319688e+09	2020-12-31	18.779	11.68	True
2021-01-29	133.52	145.090	126.382	131.96	2.239366e+09	2021-01-29	18.708	-1.56	False
2021-02-26	133.75	137.877	118.390	121.26	1.825487e+09	2021-02-26	19.487	-12.49	False
2021-03-31	123.75	128.720	116.210	122.15	2.650845e+09	2021-03-31	12.510	-1.60	False
2021-04-21	123.66	135.530	122.490	133.50	1.225012e+09	2021-04-21	13.040	9.84	True

Volatility: difference between high and low

MonthDiff: difference between close and open

Invest?: Categorical variable that determines whether to invest in a certain month or not

OUR PROCESS

02

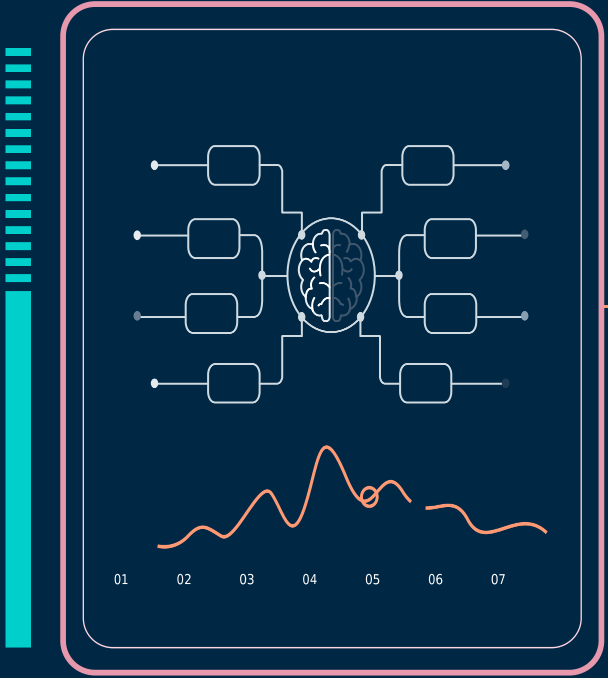
TIME SERIES ANALYSIS

2.1

By Pathak Siddhant

OUR MOTIVATION

Can we predict the opening, closing prices, as well as the highs and the lows of a stock, given its past data?



OUR SOLUTION

The goal is to train a SARIMAX model with optimal parameters that will forecast the four prices with the help of the given data.

FUNDAMENTALS

Understand the basics of Time Series as a Machine Learning model

HYPERPARAMETER OPTIMIZATION

Determine the right combination of various hyperparameters to fine-tune our model

APPLICATION

Implement it to find solution to our problem

MAKE PREDICTIONS

Make predictions to serve our purpose of the model

FUNDAMENTALS

We made use of the inbuilt TimeSeries function in the Alpha Vantage API to plot and observe the behaviour of the data points over the years.

```
import alpha_vantage
from alpha_vantage.timeseries import TimeSeries
```

We imported the PyramidARIMA Python Library to make use of its functions to make predictions and plot basic diagnostics graph to study the correlation of various data points with respect to time. It is a statistical library designed to fill the void in Python's time series analysis capabilities. It self-tunes the various hyperparameters for successful Time Series Analysis.

```
conda install pmdarima
```

```
import pmdarima as pm
```

HYPERPARAMETER OPTIMIZATION

```
Performing stepwise search to minimize aic
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=2667.187, Time=0.21 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=2665.430, Time=0.02 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=2667.315, Time=0.07 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=2667.290, Time=0.08 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=2663.431, Time=0.01 sec

Best model:  ARIMA(0,1,0)(0,0,0)[0]
Total fit time: 0.467 seconds
```

```
Performing stepwise search to minimize aic
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=2606.309, Time=0.13 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=2603.322, Time=0.01 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=2604.346, Time=0.08 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=2604.357, Time=0.08 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=2601.325, Time=0.02 sec

Best model:  ARIMA(0,1,0)(0,0,0)[0]
Total fit time: 0.323 seconds
```

```
Performing stepwise search to minimize aic
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=2698.476, Time=0.18 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=2694.600, Time=0.01 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=2696.569, Time=0.06 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=2696.565, Time=0.09 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=2692.600, Time=0.01 sec

Best model:  ARIMA(0,1,0)(0,0,0)[0]
Total fit time: 0.375 seconds
```

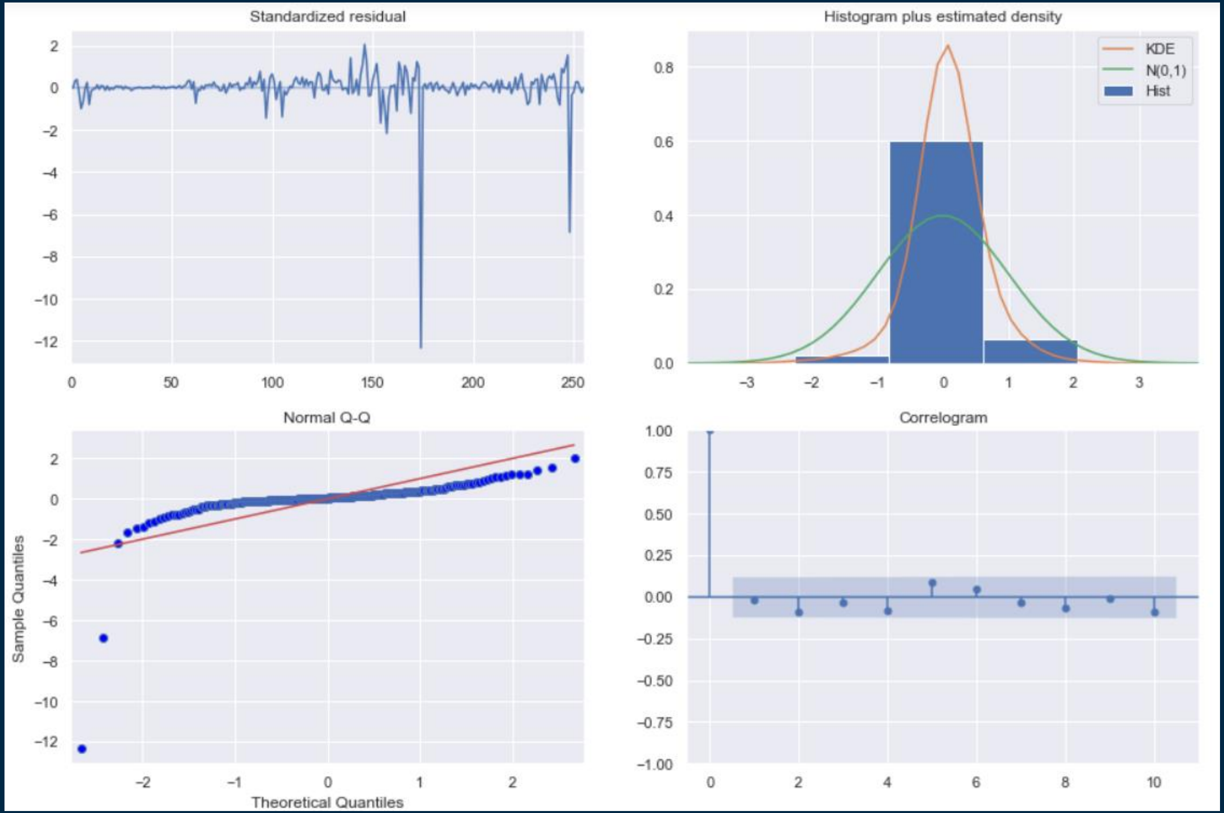
```
Performing stepwise search to minimize aic
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=2663.405, Time=0.17 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=2661.541, Time=0.01 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=2663.405, Time=0.07 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=2663.378, Time=0.07 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=2659.543, Time=0.01 sec

Best model:  ARIMA(0,1,0)(0,0,0)[0]
Total fit time: 0.339 seconds
```

```
Best model:  ARIMA(0,1,0)(0,0,0)[0]
Total fit time: 0.339 seconds
Performing stepwise search to minimize aic
 ARIMA(1,0,1)(0,0,0)[0]             : AIC=10715.456, Time=0.06 sec
 ARIMA(0,0,0)(0,0,0)[0]             : AIC=11224.541, Time=0.01 sec
 ARIMA(1,0,0)(0,0,0)[0]             : AIC=10735.029, Time=0.02 sec
 ARIMA(0,0,1)(0,0,0)[0]             : AIC=11064.918, Time=0.03 sec
 ARIMA(2,0,1)(0,0,0)[0]             : AIC=10711.574, Time=0.20 sec
 ARIMA(2,0,0)(0,0,0)[0]             : AIC=10724.849, Time=0.04 sec
 ARIMA(3,0,1)(0,0,0)[0]             : AIC=10714.664, Time=0.12 sec
 ARIMA(2,0,2)(0,0,0)[0]             : AIC=10711.705, Time=0.18 sec
 ARIMA(1,0,2)(0,0,0)[0]             : AIC=10710.480, Time=0.09 sec
 ARIMA(0,0,2)(0,0,0)[0]             : AIC=11026.305, Time=0.04 sec
 ARIMA(1,0,3)(0,0,0)[0]             : AIC=10712.689, Time=0.12 sec
 ARIMA(0,0,3)(0,0,0)[0]             : AIC=11000.270, Time=0.06 sec
 ARIMA(2,0,3)(0,0,0)[0]             : AIC=10707.222, Time=0.32 sec
 ARIMA(3,0,3)(0,0,0)[0]             : AIC=10697.408, Time=0.38 sec
 ARIMA(3,0,2)(0,0,0)[0]             : AIC=10703.128, Time=0.33 sec
 ARIMA(3,0,3)(0,0,0)[0] intercept   : AIC=10694.064, Time=0.33 sec
 ARIMA(2,0,3)(0,0,0)[0] intercept   : AIC=10706.566, Time=0.26 sec
 ARIMA(3,0,2)(0,0,0)[0] intercept   : AIC=10699.349, Time=0.41 sec
 ARIMA(2,0,2)(0,0,0)[0] intercept   : AIC=10710.583, Time=0.30 sec

Best model:  ARIMA(3,0,3)(0,0,0)[0] intercept
Total fit time: 3.315 seconds
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  257
Model:               SARIMAX(0, 1, 0)   Log Likelihood               -1330.715
Date:                Tue, 20 Apr 2021   AIC                           2663.431
Time:                        00:25:46   BIC                           2666.976
Sample:                             0   HQIC                          2664.856
                                - 257
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
sigma2      1916.6598     24.117     79.473      0.000    1869.391    1963.929
===================================================================================
Ljung-Box (L1) (Q):                   0.12   Jarque-Bera (JB):            103010.20
Prob(Q):                              0.73   Prob(JB):                         0.00
Heteroskedasticity (H):              54.19   Skew:                            -8.59
Prob(H) (two-sided):                  0.00   Kurtosis:                        99.76
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

APPLICATION

ANALYSIS



Standardized residual
Histogram plus estimated density
KDE
N(0,1)
Hist
Normal Q-Q
Sample Quantiles
Theoretical Quantiles
Correlogram

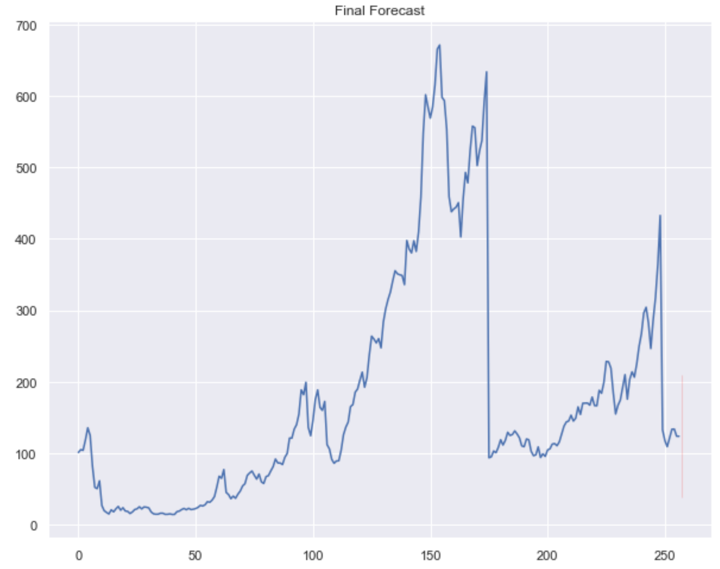
MAKE PREDICTIONS

The SARIMAX model that was trained by auto tuning the hyperparameters was further used to predict the values of the four variables in the upcoming month.

The predictions have been made only for one month. This is because this model does not take into account a lot of external factors that influence the market such as disposal of income, changing social behaviour, international transactions etc., whose study is beyond the scope of this project.

```
forecasted_open_company=plot_forecasted(1, data_df_company['1. open'])
```

```
Performing stepwise search to minimize aic
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=2667.187, Time=0.17 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=2665.430, Time=0.01 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=2667.315, Time=0.06 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=2667.290, Time=0.06 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=2663.431, Time=0.01 sec

Best model:  ARIMA(0,1,0)(0,0,0)[0]
Total fit time: 0.321 seconds
```



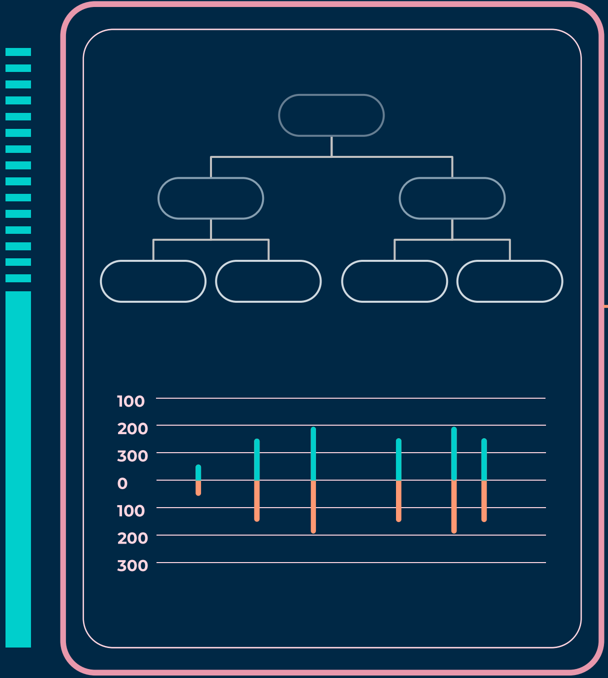
RANDOM FOREST CLASSIFICATION

2.2

By Hariharan Dhruv

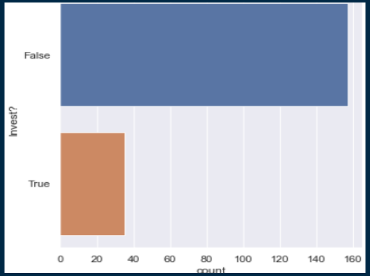
OUR MOTIVATION

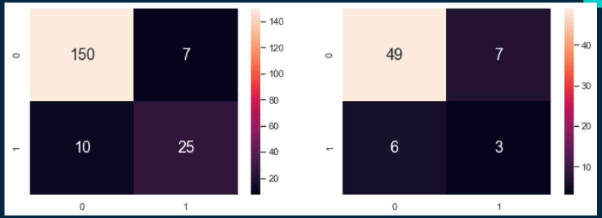
Predicting if the next month is a good time to invest in a particular company or not based on the Random Forest Classification Model.



EXPLORATORY DATA ANALYSIS AND DATA VISUALIZATION

```
False     213
True       44
Name: Invest?, dtype: int64
```

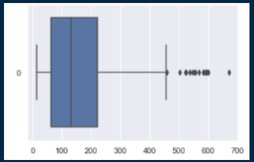


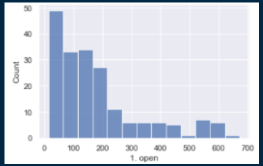


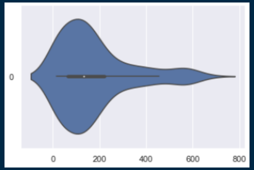
Confusion Matrix

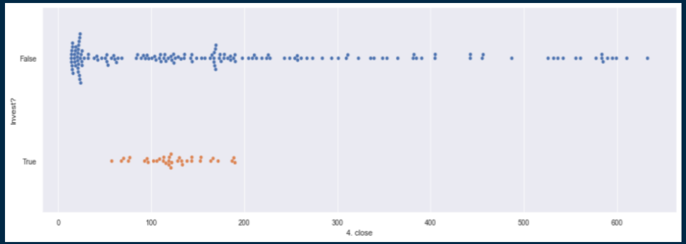
Univariate Statistics

Bi-Variate Statistics









MOST IMPORTANT DATA VISUALIZATION: OHLC CHARTS

OHLC Charts consists of Open, High, Low and Close values in a given timeframe.
Vertical segments represent the high and low values.
Horizontal segments determine the open and close values.
In this example, red represent decreasing momentum and green lines represent increasing momentum.

RANDOM FOREST CLASSIFIER: SUPERVISED LEARNING

Splitting Dataset

Splitting the dataset into train and test, we made the test size 0.25.

Fitting the Model

We fit the model on train and test data.

Prediction

We predict the response variable based on the OHLC values predicted by the time series values.

Checking Accuracy

We check the accuracy and goodness of fit of model on the test and train predictions.

Goodness of Fit of Model Classification Accuracy	Train Dataset : 0.9114583333333334
Goodness of Fit of Model Classification Accuracy	Test Dataset : 0.8307692307692308

```
#Response
y = pd.DataFrame(data_company["Invest?"])
#Predictors
X = pd.DataFrame(data_company[["1. open", "2. high", "3. low", "4. close"]])
#Then we proceed to split the dataset into train and test where the size of test_size is 0.25 and train is 0.75
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25)
```

SOLUTION AND ANALYSIS

```
In [64]: company_values = pd.concat([forecasted_open_company,forecasted_high_company,forecasted_low_company,forecasted_close_company])
         company_values = company_values.T
         company_values.columns=['Open','High','Low','Close']
         time = data_company['date_time'].iloc[-1]
         print("Predicted values for the following month: ", time.month+1,time.year)
         company_values

         Predicted values for the following month:  5 2021

Out[64]:
```

	Open	High	Low	Close
forecasted	123.66	135.53	122.49	133.5

```
In [30]: company_pred_value = rforest.predict(company_values)
         company_pred_value

Out[30]: array([ True])
```

We display the forecasted values for the next month. The model then displays True or False, a categorical variable that determines if the next month is a good or bad month to invest in. The predicted values should just be used as an indication because stock prices are subject to market risk and other external factors.

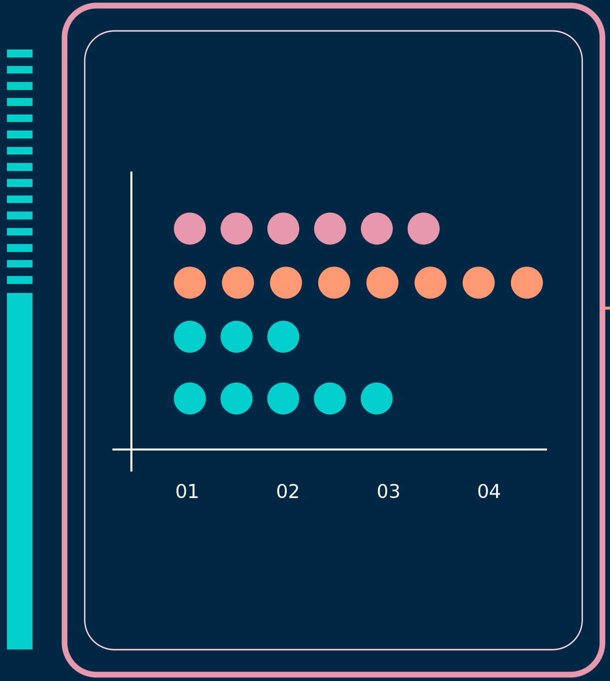
CLASSIFICATION & ANOMALY DETECTION

2.3

By Gupta Anant

OUR MOTIVATION

Is it possible to derive a relationship between stock volatility and volume based on past data?



FUNDAMENTALS

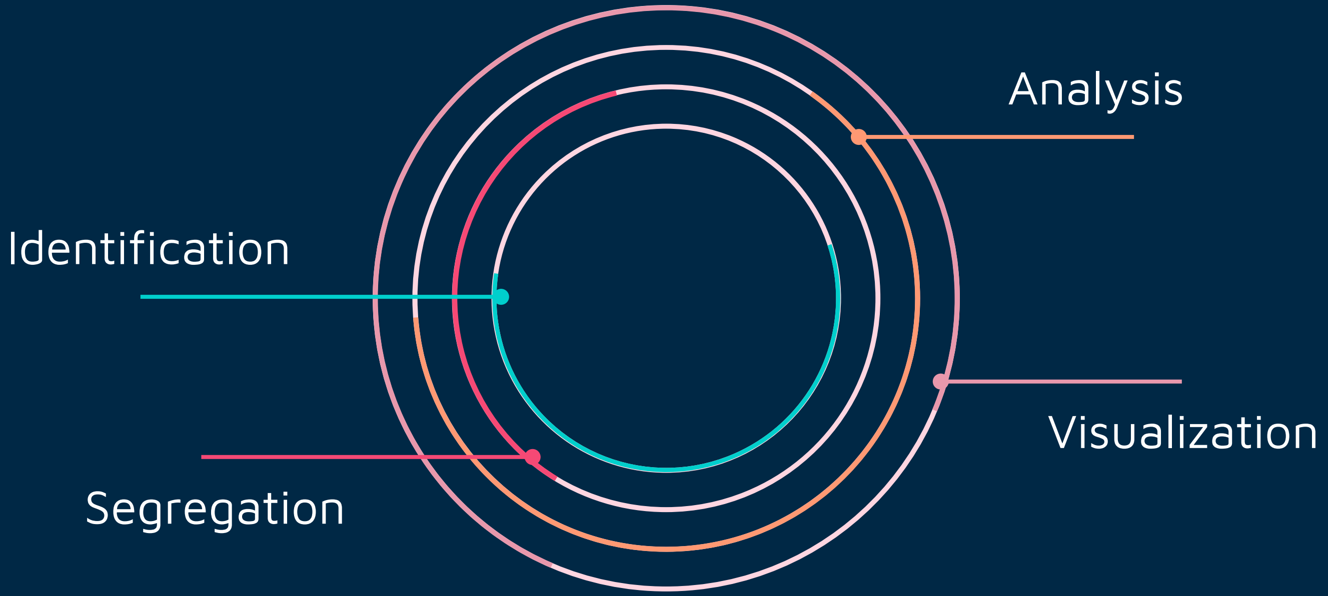
We utilised the in built package from the scikit-learn library in order to check the stability of stocks based on clustering of volume.

```
# Import KMeans from sklearn.cluster
from sklearn.cluster import KMeans
```

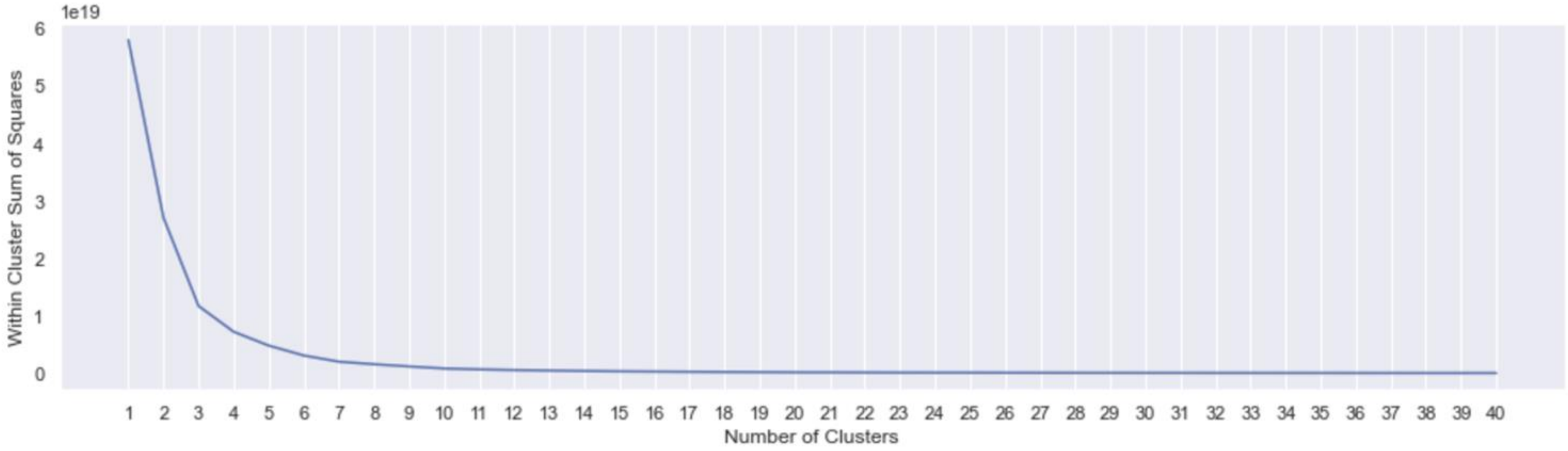
We used anomaly detection to detect abnormally large volumes and determine whether volatility directly affects purchase and selling of stocks.

```
# Import LocalOutlierFactor from sklearn.neighbors
from sklearn.neighbors import LocalOutlierFactor
```

CLUSTERING MODEL



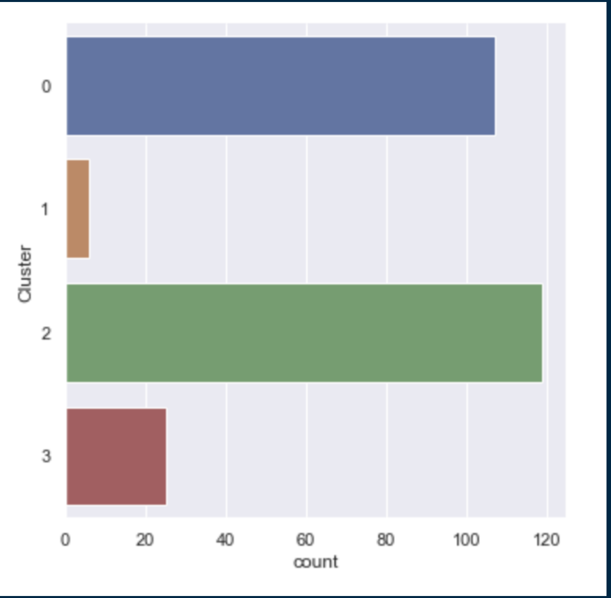
APPLICATION

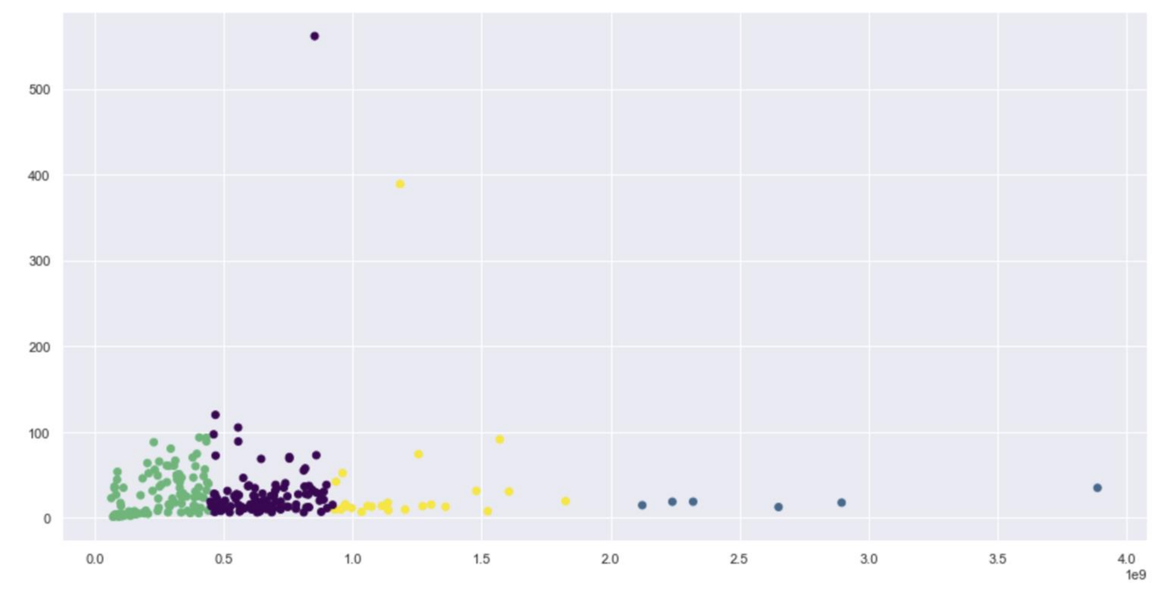


Features	Volatility	Volume
Cluster 0:	30.47	656343392.88
Cluster 1:	19.54	2685789032.0
Cluster 2:	24.93	233431845.29
Cluster 3:	37.42	1193201603.36

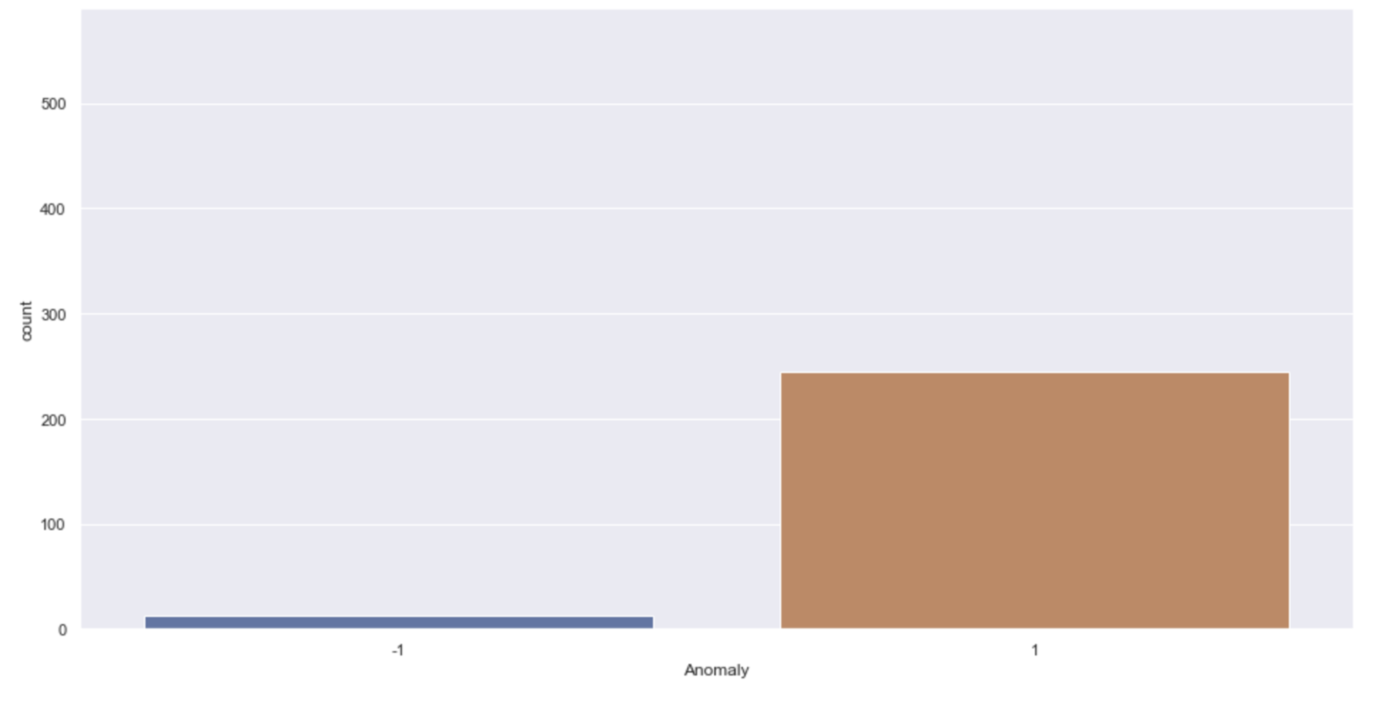
Within Cluster Sum of Squares : 7.223495049764538e+18

APPLICATION

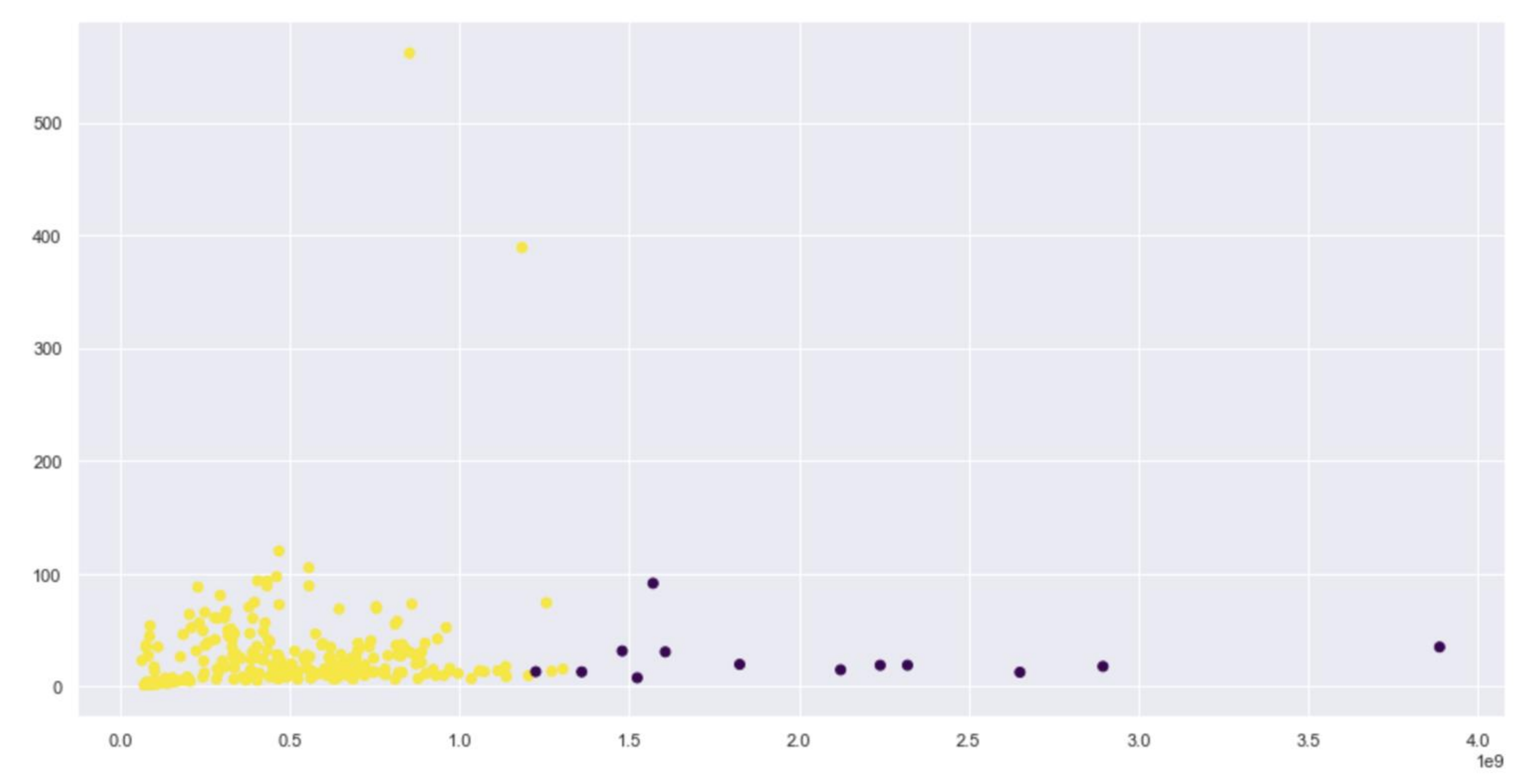




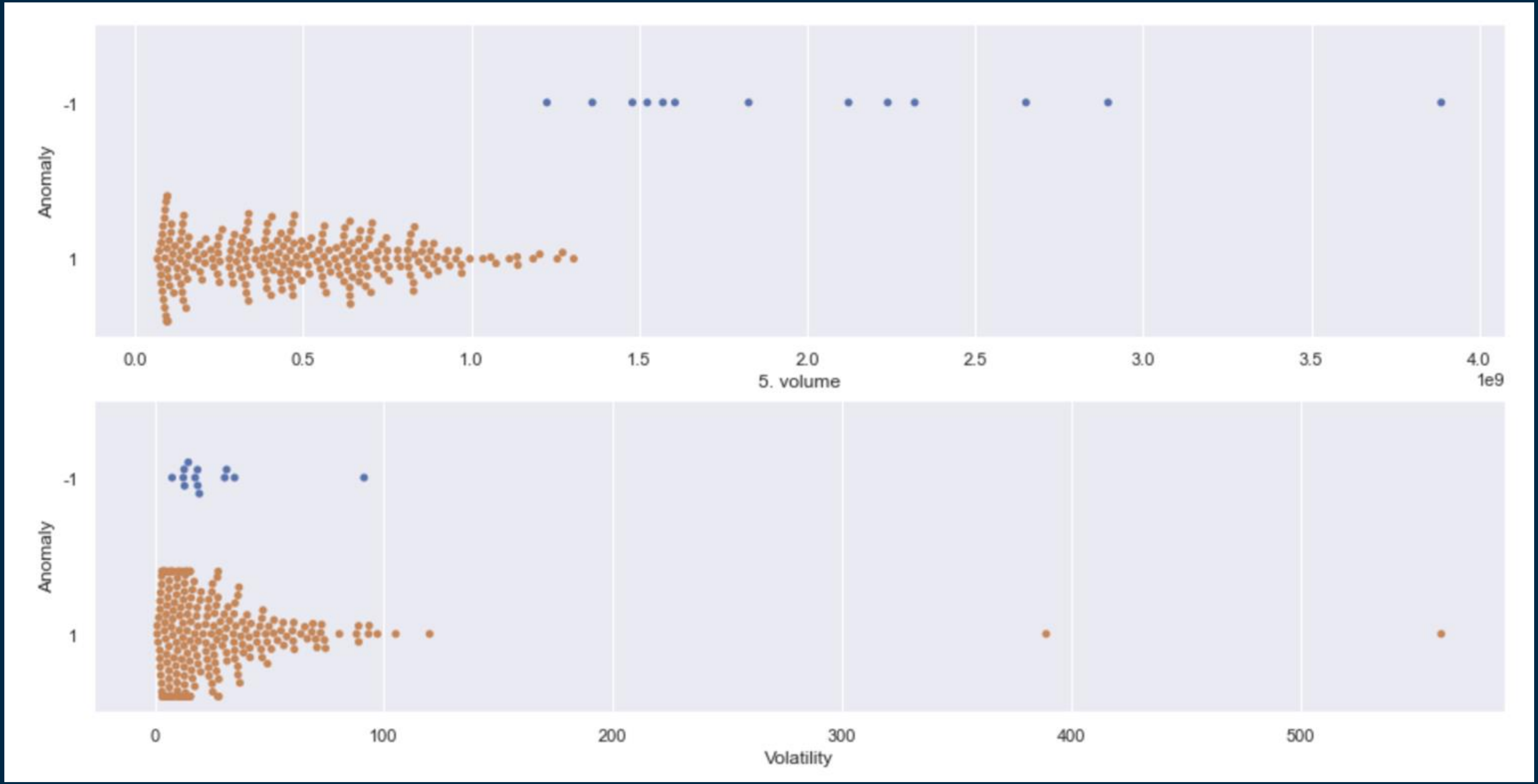
APPLICATION



APPLICATION



APPLICATION



ANALYSIS

- Univariate and Multivariate Linear Regression on 5 variables failed to yield results.
- Further analysis into the economic element of the stock market revealed that market volatility is one of the most important indicators to investors.
- Clustering and Anomaly Detection were more beneficial to depict a tangible relationship between the two variables and find outliers which might have caused by external factors like recession.

OUTCOMES AND DATA DRIVEN INSIGHTS

03

KEY OUTCOME

- Significant events in history have affected the stock volume but volatility is still a good predictor.
- This model is flexible to many companies.
- For any company we choose, we not only get the OHLC values for the next month but the model tells us should we invest or not.

```
Apple="AAPL"
Microsoft="MSFT"
Google="GOOGL"
Amazon="AMZN"
Facebook="FB"
```

```
data_company, meta_company, data_df_company = get_data(Apple)
data_df_company
```

```
data_company, meta_company, data_df_company = get_data(Microsoft)
data_df_company
```

```
company_pred_value = rforest.predict(company_values)
company_pred_value
```

```
array([ True])
```

```
company_pred_value = rforest.predict(company_values)
company_pred_value
```

```
array([False])
```

KEY DATA DRIVEN INSIGHTS

- The time-series model consists of values taken over a period of 20 years, which include financial discontinuities, making the model reliable but not for long term predictions.
- R^2 values in univariate and multivariate regression were too low but in random classifier it was high.
- OHLC values cannot be sole predictor of stock volume but good indicator to determine if a company can be used to invest in or not.

11,308,000

This model can be extended to
these many data points available on
Alpha Vantage API

THANK YOU!

REFERENCES

- Adam, P. (n.d.). *Markdown Cheatsheet*. GitHub. https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet
- *Are Stocks With Large Daily Volume Less Volatile?* (n.d.). Investopedia. https://www.investopedia.com/ask/answers/09/daily-volume-volatility.asp#:%7E:text=There%20is%20a%20relationship%20between,stock%20experiences%20a%20sharp%20decrease
- Matt Macarty. (2021, January 11). *How to Use Alpha Vantage Free Real Time Stock API & Python to Extract Time of Daily Highs and Lows*. YouTube. https://www.youtube.com/watch?v=WJ2t_LYb__0

REFERENCES

- *OHLC Chart Definition and Uses*. (n.d.). Investopedia. https://www.investopedia.com/terms/o/ohlcchart.asp
- *pmdarima.arima.ARIMA documentation*. (n.d.). Alkaline-ML. https://alkaline-ml.com/pmdarima/modules/generated/pmdarima.arima.ARIMA.html
- *sklearn.cluster.KMeans documentation*. (n.d.). Scikit-Learn 0.24.1. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
- *Stock Market Forecasting Using Time Series Analysis*. (n.d.). KDNuggets. https://www.kdnuggets.com/2020/01/stock-market-forecasting-time-series-analysis.html

REFERENCES

- *The 4 Basic Elements of Stock Value*. (n.d.). Investopedia. https://www.investopedia.com/articles/fundamental-analysis/09/elements-stock-value.asp
- *Understanding Random Forests Classifiers in Python*. (n.d.). DataCamp. https://www.datacamp.com/community/tutorials/random-forests-classifier-python
- *Volatility*. (n.d.). Investopedia. https://www.investopedia.com/terms/v/volatility.asp