BIRMINGHAM CITY
University

Machine learning Report

CMP7228 S1 2023/4

JANUARY 15

Birmingham city university
Student id - 23178059
Authored by: Namagiri Mukul

Index

Domain description -- 3-5

Problem definition -- 5

Dataset description --- 5-7

Dataset exploration --- 8-19

Experiments, Analysis, and Results -- 20-39

Conclusions -- 39-40

References -- 41

Appendixes -- 42

Domain Description

A portfolio is a collection of investments owned by a person/individual or organization. These can range from securities, stocks, individual businesses, bonds, and mutual funds. The objective of a portfolio manager is to evaluate, manage, and optimize the composition of assets dynamically (Johnson,2014). With the advent of machine learning strategies and artificial intelligence algorithms, there is a drastic shift in the way portfolios are managed nowadays. These technologies help in processing large data sets, building models, and predicting market trends. Discretionary portfolio managers or systematic portfolio managers often tend to build theories and rationale with a heavy reliance on judgment and intuition rather than logic and fact-based, empirical evidence (Lopez, de Prado, 2018). Portfolio analysis has several aspects that include Security analysis, Stock selection, Risk management, portfolio construction, Performance Construction, Performance monitoring, Rebalancing, Dividend analysis, Tax considerations, Market and economic analysis. The flow chart below will represent the decision flow of a typical structured portfolio management.

Exploring different aspects of the individual steps presented

1. Security analysis refers to the process of evaluating financial instruments such as stocks to determine investment potential it has two main subcategories namely fundamental analysis and technical analysis.
    - Fundamental analysis – examining the intrinsic value of security like management quality, industry conditions, economic indicators, etc.
    - Technical analysis – analyzing the historical prices and volume data to predict future price movements.
2. Stock selection involves adding stocks based on growth potential, valuation, and risk factors. It also involves diversifying holdings across different industries.
3. Risk management involves assessing the individual stocks, including factors like volatility, beta, and correlation with other assets also includes setting up stopping criteria to hedge against adverse price movements.
4. Portfolio construction involves combining individual stocks with the investor's goals in mind like risk tolerance, and time horizon.

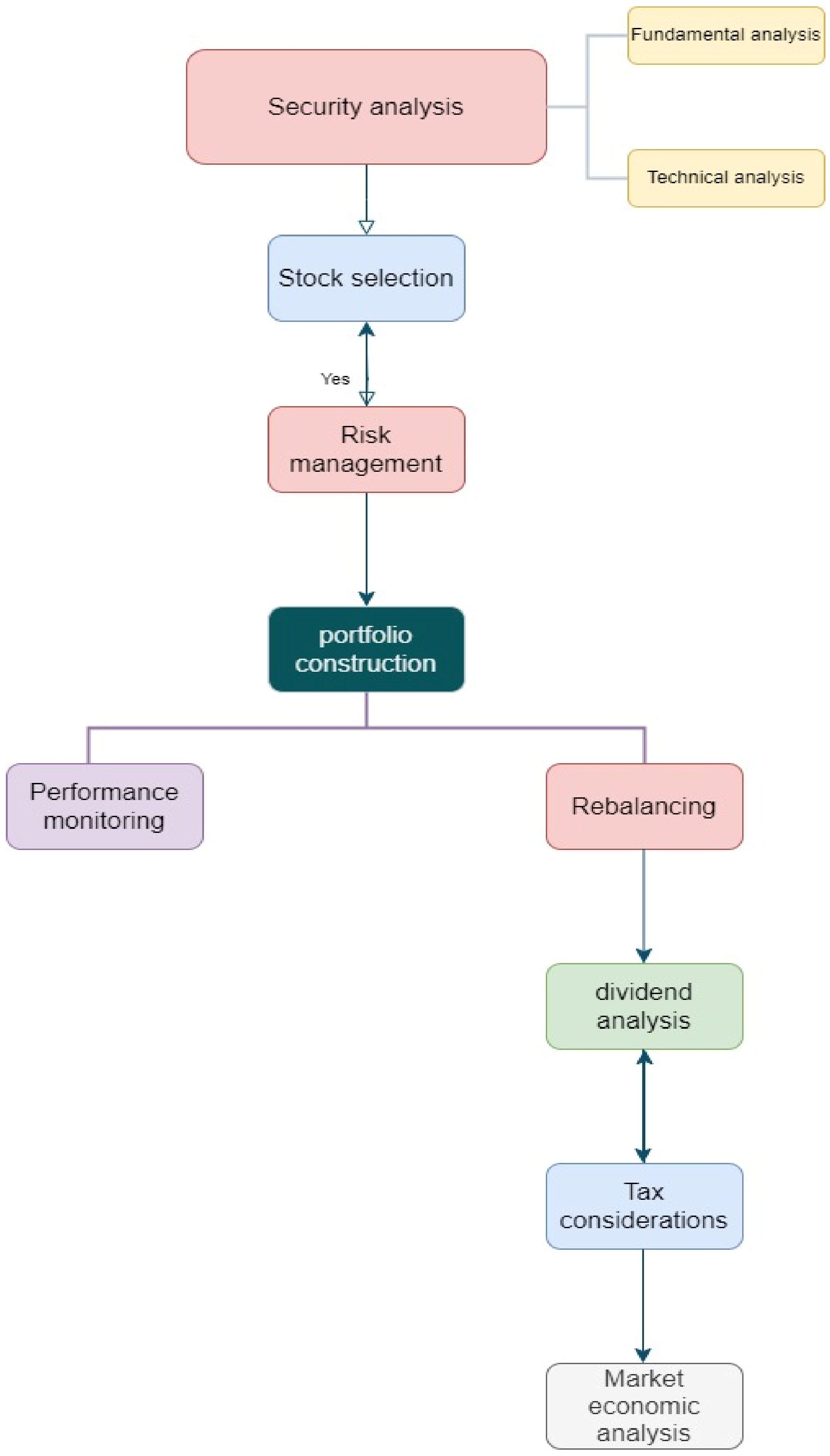

Security analysis
Fundamental analysis
Technical analysis
Stock selection
Yes
Risk management
portfolio construction
Performance monitoring
Rebalancing
dividend analysis
Tax considerations
Market economic analysis

Figure 1

5. Performance monitoring involves regularly reviewing individual companies and assessing the impact of market conditions on stock performance.
6. Rebalancing involves active assessment of assets with investor's objectives and changing market conditions.
7. Dividend analysis involves catering to income-oriented investors making sure that dividend yield and payout ratio are optimal while changing stock allotment.
8. Tax considerations should be optimized considering capital gains tax and dividends.
9. Market and economic analysis involves staying informed about market trends, economic indicators, and geopolitical issues that may impact stocks and also incorporating macroeconomic analysis into portfolio management and stock selection (Johnson and Parente,2013).

Problem definition

The goal of this project is to solve three problems associated with portfolio management using data analysis and machine learning techniques, the primary question of the 3 problems are presented below

Regression problem

Using regression-based machine learning models to find the future values of the adjusted closing value using open, close, high, low, and volume of the selected stocks.

Classification problem

In this section, we use classification techniques to discern whether a stock is likely to experience an upward trend or downward trend enabling us to make an informed decision so as to buy or sell a particular stock.

Clustering problem

Clustering companies based on their risk profile i.e., to identify classes of companies that have the same risk profile we want to diversify our portfolio based on the investor's needs so identifying and clustering them lets us decide which stocks to pick.

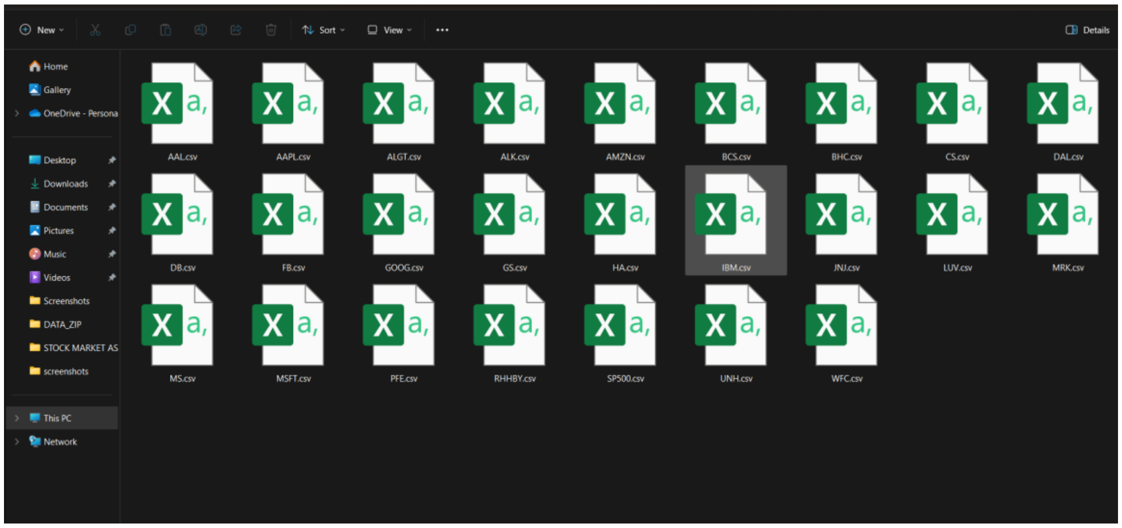
Dataset description

This project uses data from the Kaggle data set (Stock portfolio – financial risk analytics) with a usability rating of 4.71 – tags (Investing)

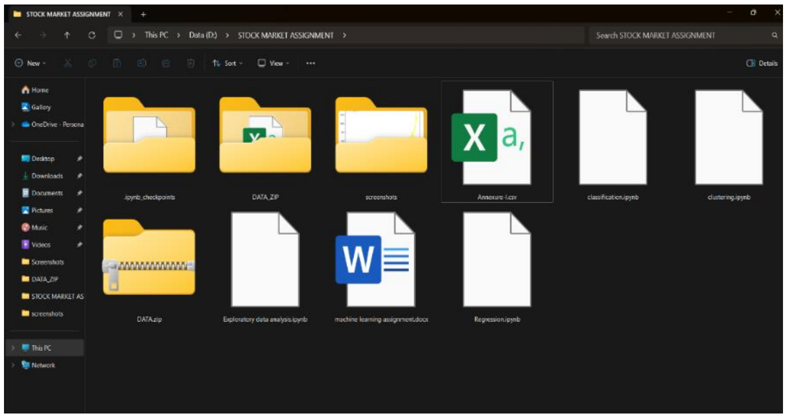
Problem statement presented in Kaggle – how to identify the right investment opportunity and recommend a portfolio as per the client's exact needs.

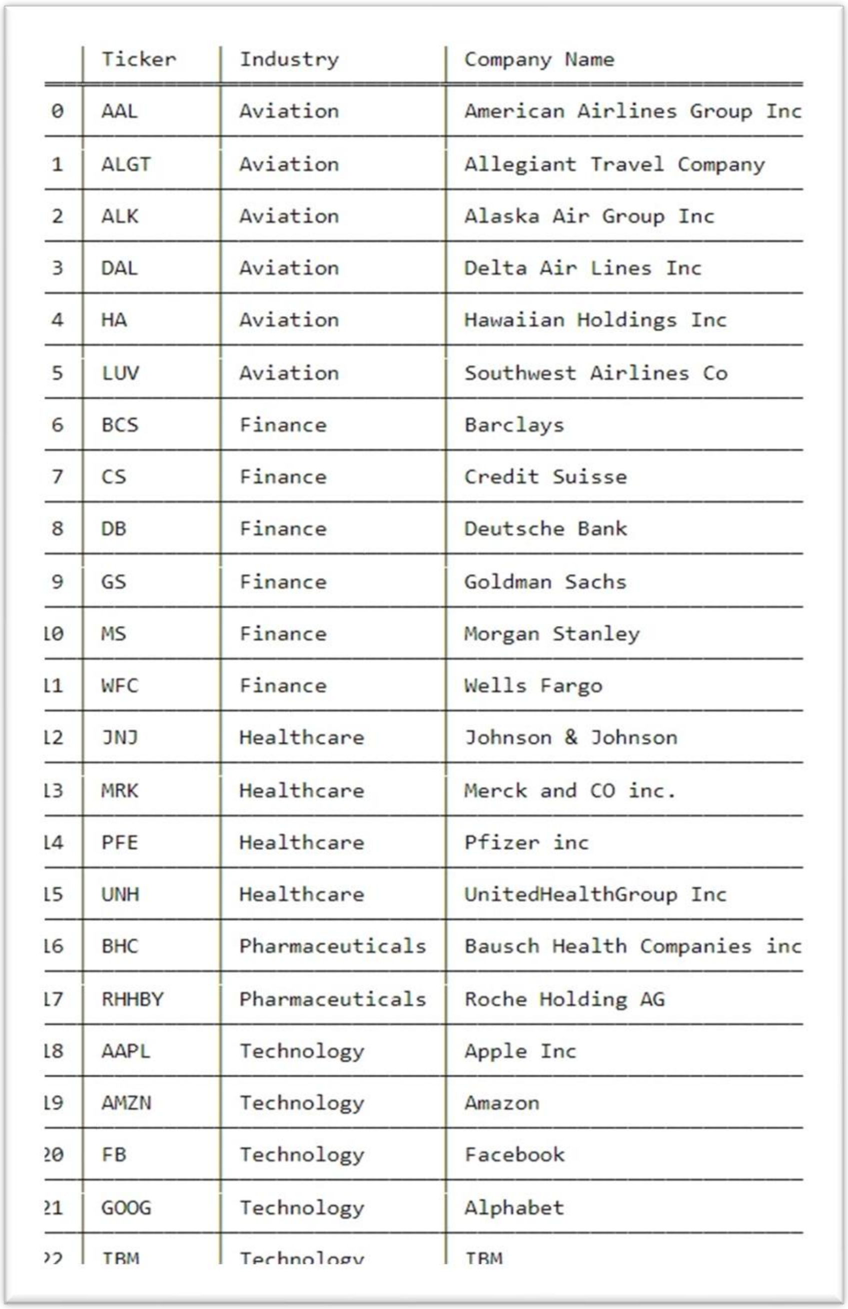
From a broad perspective, there are two types of investors conservative investors, and aggressive investors according to traditional investment norms conservative investors tend to be more cautious and want less risky alternatives even if the returns are not very high but aggressive investors tend to be more risk-taking for better returns.

The file directory is named DATA_ZIP it contains 24 different company stocks and 1 CSV file for S&P 500 index.



One of the CSV file is Annexure – 1 is the index file for the whole data set containing the ticker symbol for the stock being traded, the industry it belongs to, and the name of the company name.





	Ticker	Industry	Company Name
0	AAL	Aviation	American Airlines Group Inc
1	ALGT	Aviation	Allegiant Travel Company
2	ALK	Aviation	Alaska Air Group Inc
3	DAL	Aviation	Delta Air Lines Inc
4	HA	Aviation	Hawaiian Holdings Inc
5	LUV	Aviation	Southwest Airlines Co
6	BCS	Finance	Barclays
7	CS	Finance	Credit Suisse
8	DB	Finance	Deutsche Bank
9	GS	Finance	Goldman Sachs
10	MS	Finance	Morgan Stanley
11	WFC	Finance	Wells Fargo
12	JNJ	Healthcare	Johnson & Johnson
13	MRK	Healthcare	Merck and CO inc.
14	PFE	Healthcare	Pfizer inc
15	UNH	Healthcare	UnitedHealthGroup Inc
16	BHC	Pharmaceuticals	Bausch Health Companies inc
17	RHHBY	Pharmaceuticals	Roche Holding AG
18	AAPL	Technology	Apple Inc
19	AMZN	Technology	Amazon
20	FB	Technology	Facebook
21	GOOG	Technology	Alphabet
22	IBM	Technology	IBM

In this analysis we are only going to look at technology, finance and aviation companies

Dataset download link → https://www.kaggle.com/datasets/ankurnapa/stock-portfolio-financial-risk-analytics

Data set description link → https://www.kaggle.com/datasets/ankurnapa/stock-portfolio-financial-risk-analytics?select=Annexure-I.csv

Dataset Exploration

Describing each of the columns present in the data set

Date:

Date represents the trading day of the company at which the data is recorded.

Open:

Open represents the opening price of the stock for the trading day.

Close:

Close represents the closing price of the stock for the trading day.

High:

High represents the highest price recorded for the trading day.

Low:

Low represents the lowest price recorded for the trading day.

Adjusted closing price:

Adjusted closing price of the stock on a trading day is used to indicate the closing price taking dividends, stock splits, and new stock offerings into account it presents a more accurate value of the actual stock value.

Volume:

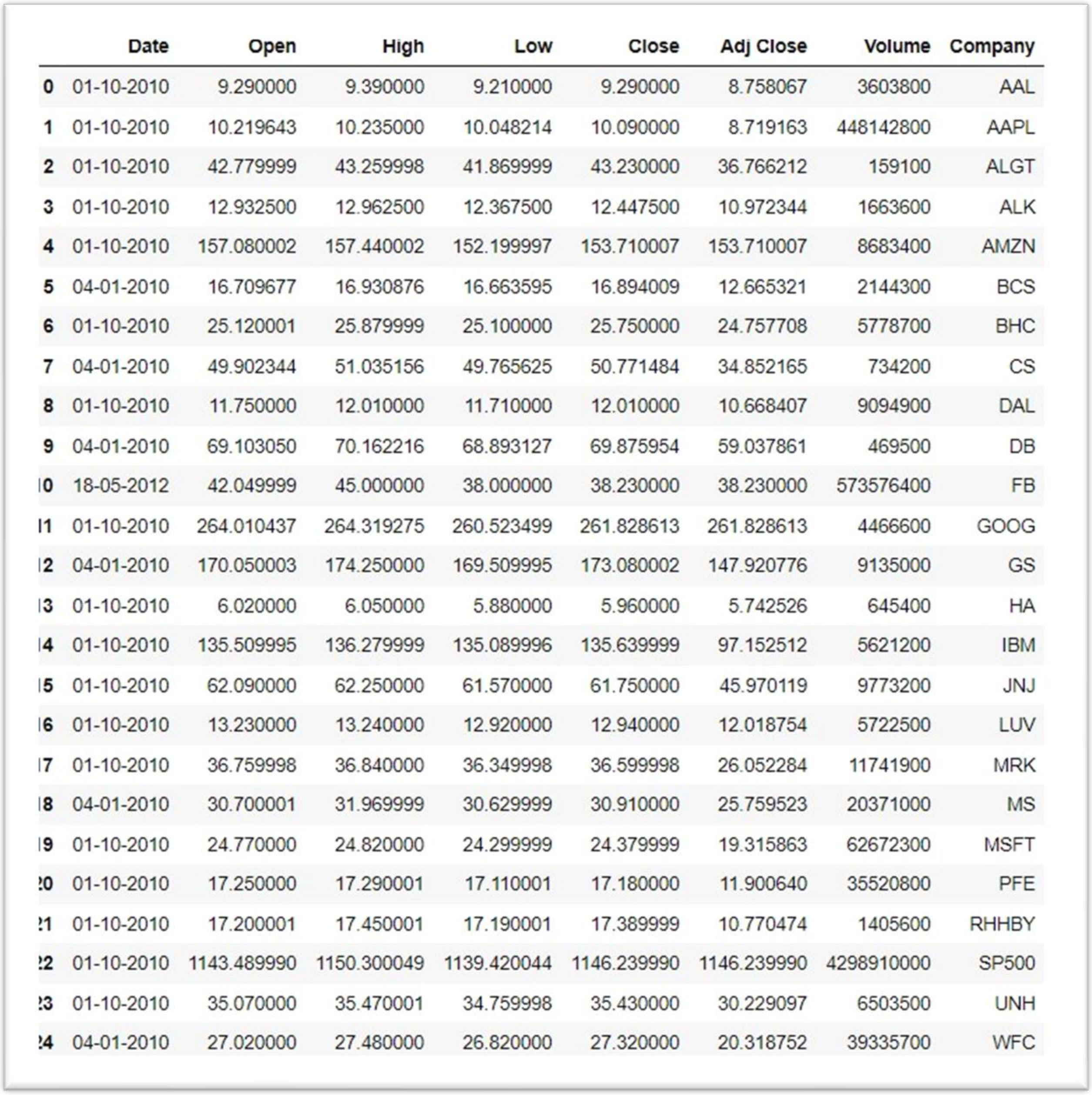
Volume refers to the total number of shares of a particular stock traded during a trading day.

The table below represents the column data for the index fund S&P 500

column	Data type	initial value
Date	**object**	**01-10-2010**
Open	**Float64**	**1143.48**
Close	**Float64**	**1146.23**
High	**Float64**	**1150.30**
low	**Float64**	**1139.42**

volume	**Float64**	**4296910000**
Adjusted closing price	**Float64**	**1146.239990(since there are no stock splits or dividends for INDEX FUND)**

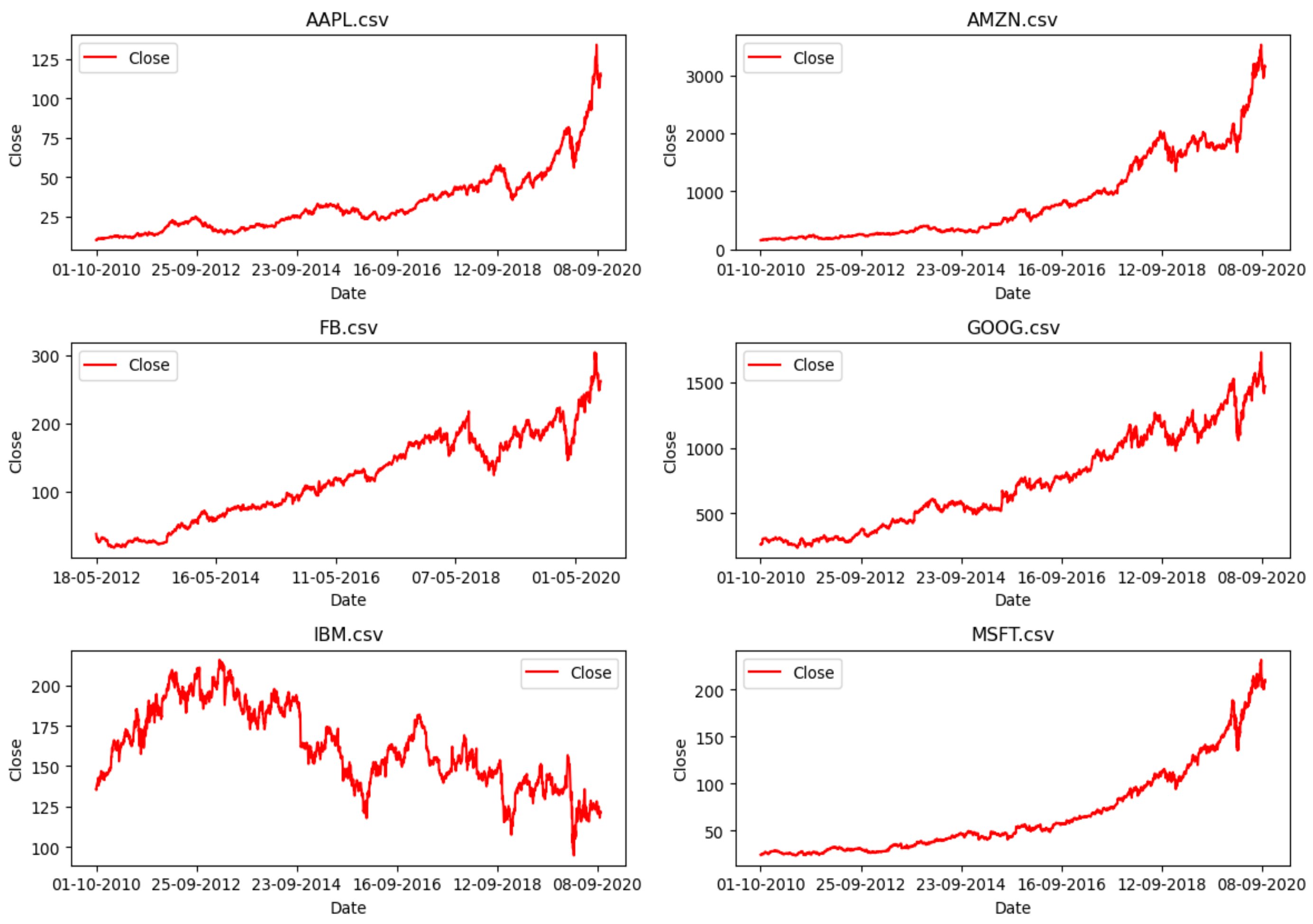
Since manually opening each of the files is not possible using the OS module to read all the files and making a data frame that contains the first row of the CSV files and making a new column named company to identify which stock it belongs to



	Date	Open	High	Low	Close	Adj Close	Volume	Company
0	01-10-2010	9.290000	9.390000	9.210000	9.290000	8.758067	3603800	AAL
1	01-10-2010	10.219643	10.235000	10.048214	10.090000	8.719163	448142800	AAPL
2	01-10-2010	42.779999	43.259998	41.869999	43.230000	36.766212	159100	ALGT
3	01-10-2010	12.932500	12.962500	12.367500	12.447500	10.972344	1663600	ALK
4	01-10-2010	157.080002	157.440002	152.199997	153.710007	153.710007	8683400	AMZN
5	04-01-2010	16.709677	16.930876	16.663595	16.894009	12.665321	2144300	BCS
6	01-10-2010	25.120001	25.879999	25.100000	25.750000	24.757708	5778700	BHC
7	04-01-2010	49.902344	51.035156	49.765625	50.771484	34.852165	734200	CS
8	01-10-2010	11.750000	12.010000	11.710000	12.010000	10.668407	9094900	DAL
9	04-01-2010	69.103050	70.162216	68.893127	69.875954	59.037861	469500	DB
10	18-05-2012	42.049999	45.000000	38.000000	38.230000	38.230000	573576400	FB
11	01-10-2010	264.010437	264.319275	260.523499	261.828613	261.828613	4466600	GOOG
12	04-01-2010	170.050003	174.250000	169.509995	173.080002	147.920776	9135000	GS
13	01-10-2010	6.020000	6.050000	5.880000	5.960000	5.742526	645400	HA
14	01-10-2010	135.509995	136.279999	135.089996	135.639999	97.152512	5621200	IBM
15	01-10-2010	62.090000	62.250000	61.570000	61.750000	45.970119	9773200	JNJ
16	01-10-2010	13.230000	13.240000	12.920000	12.940000	12.018754	5722500	LUV
17	01-10-2010	36.759998	36.840000	36.349998	36.599998	26.052284	11741900	MRK
18	04-01-2010	30.700001	31.969999	30.629999	30.910000	25.759523	20371000	MS
19	01-10-2010	24.770000	24.820000	24.299999	24.379999	19.315863	62672300	MSFT
20	01-10-2010	17.250000	17.290001	17.110001	17.180000	11.900640	35520800	PFE
21	01-10-2010	17.200001	17.450001	17.190001	17.389999	10.770474	1405600	RHHBY
22	01-10-2010	1143.489990	1150.300049	1139.420044	1146.239990	1146.239990	4298910000	SP500
23	01-10-2010	35.070000	35.470001	34.759998	35.430000	30.229097	6503500	UNH
24	04-01-2010	27.020000	27.480000	26.820000	27.320000	20.318752	39335700	WFC

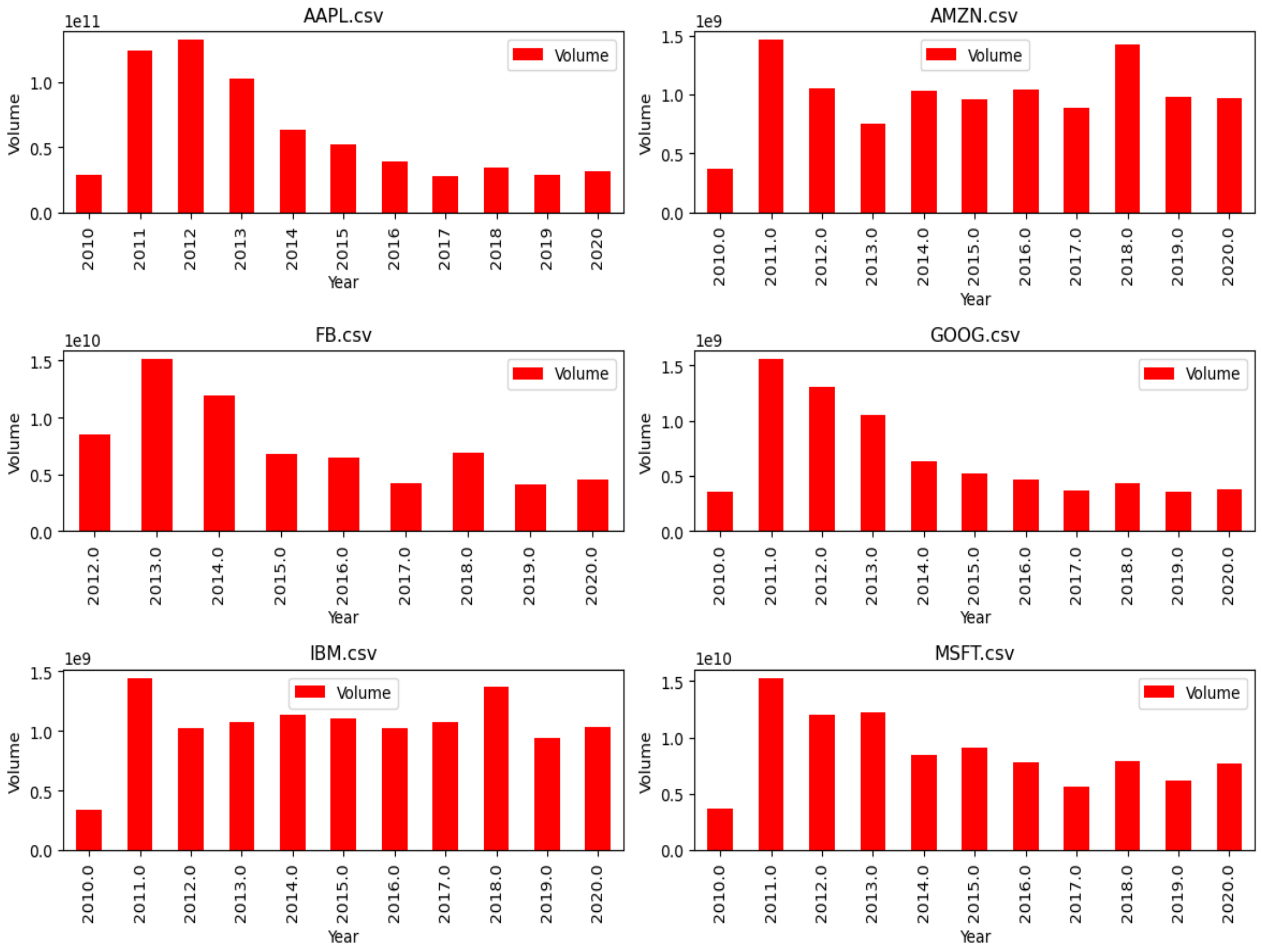
As evident from the table there seems to be inconsistent date formatting and varying start date for different companies of different domains. We want to check the overall general trend of the companies based on the domain of interest.

Analysis of technology companies:



Line plot of different technology companies

As seen from the plots above the technology companies have seen a meteoric rise in recent years with Apple, Amazon, Google, and Microsoft showing very drastic increases whereas IBM shows a more stable plot since IBM has been in the market for a much longer period as expected.



Volume traded per year for different technology companies.

The above plot shows the volume that has been traded across different years apple, google, Microsoft and Facebook have shown a slight downward trend in the end of the decade whereas IBM and Amazon have shown a relatively stable volume of the stock that is being traded.

The peak volume traded for

Apple is around 2011 and 2012

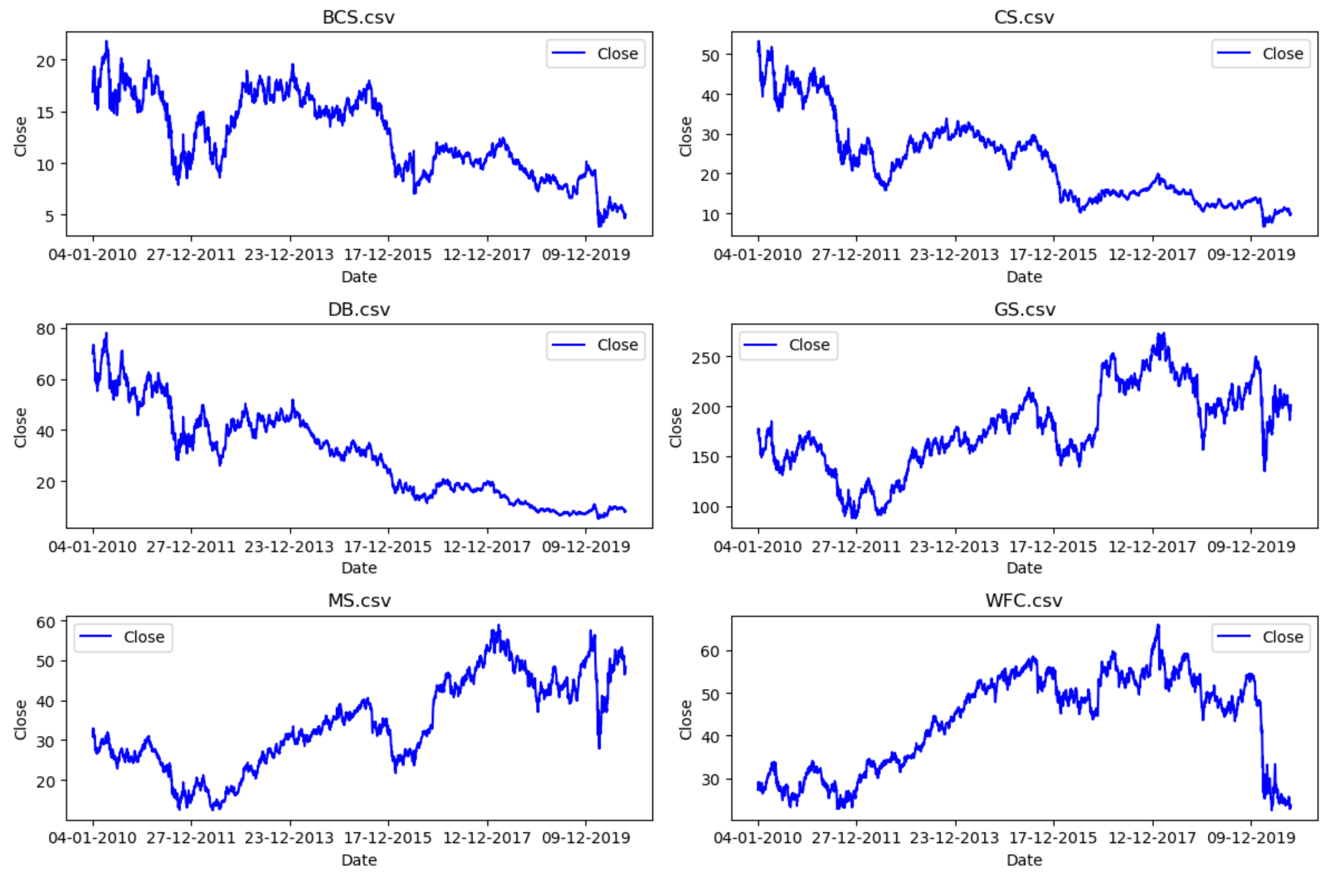
Amazon is in 2011 and 2018

Microsoft in 2011

IBM in 2011 and 2018

Google 2011 and 2012

Analysis of finance companies:

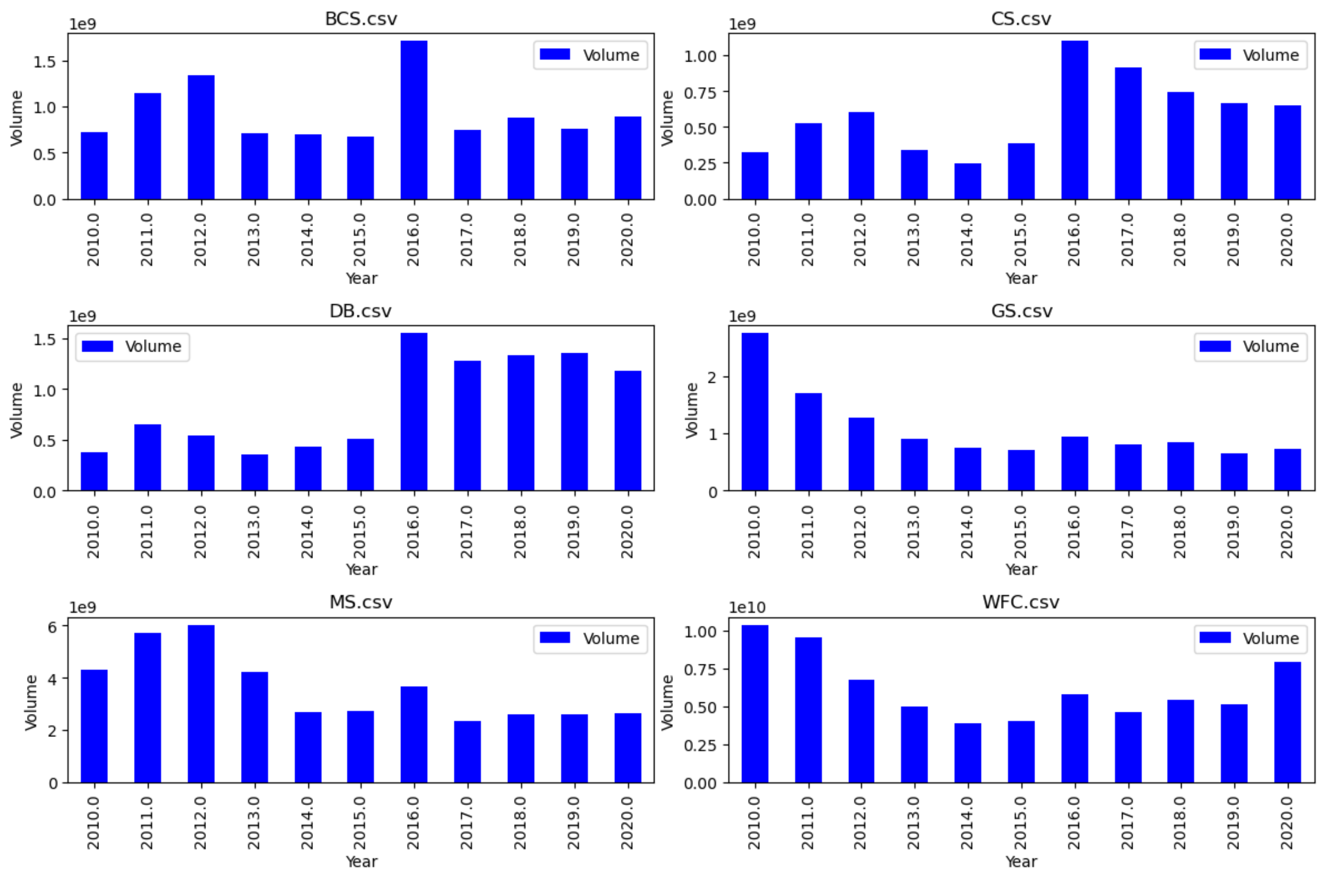


Line plot of different finance companies

As seen from the plots above, finance companies have not had much of a dramatic rise like technology companies and by the late 2020 have shown a dramatic decrease the only companies that did manage to have a relatively stable are Morgan Stanley and Goldman Sachs.

Barclays, Credit Suisse, and Deutsche Bank have shown a relatively downward trend

While Wells Fargo has reached all time low by the end of 2019.



Volume traded per year for different finance companies

The volume traded for Goldman Sachs and Morgan Stanley has relatively decreased low trading volume with increased prices signifying a tendency to hold onto the assets.

The peak volume traded for

Barclays is in 2016

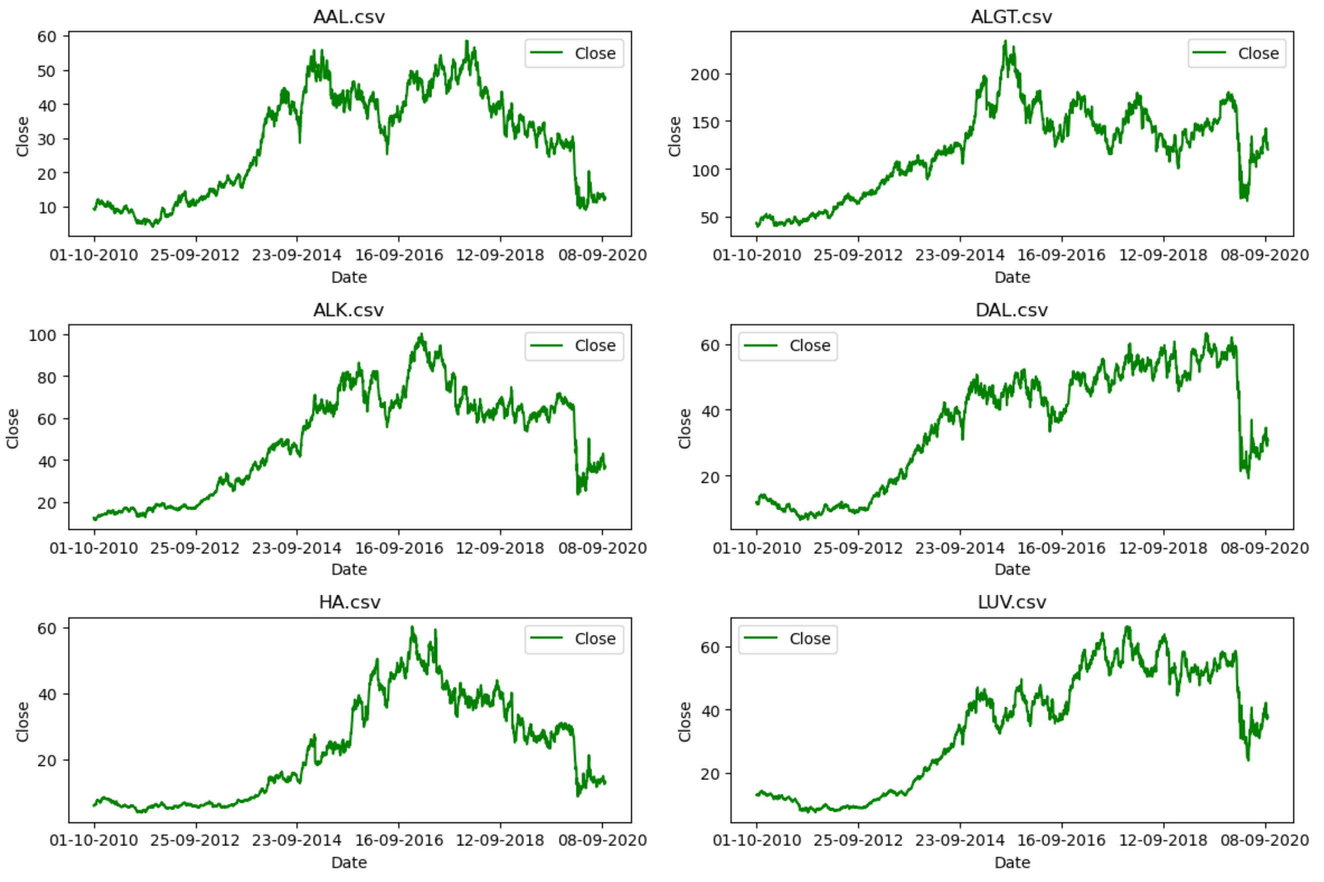
Credit Suisse is in 2016

Morgan Stanley in 2011 and 2012

Wells Fargo in the 2010s and 2020s (with a decreased price indicating loss of value)

Goldman Sachs is around the early 2010s.

Analysis of aviation companies:

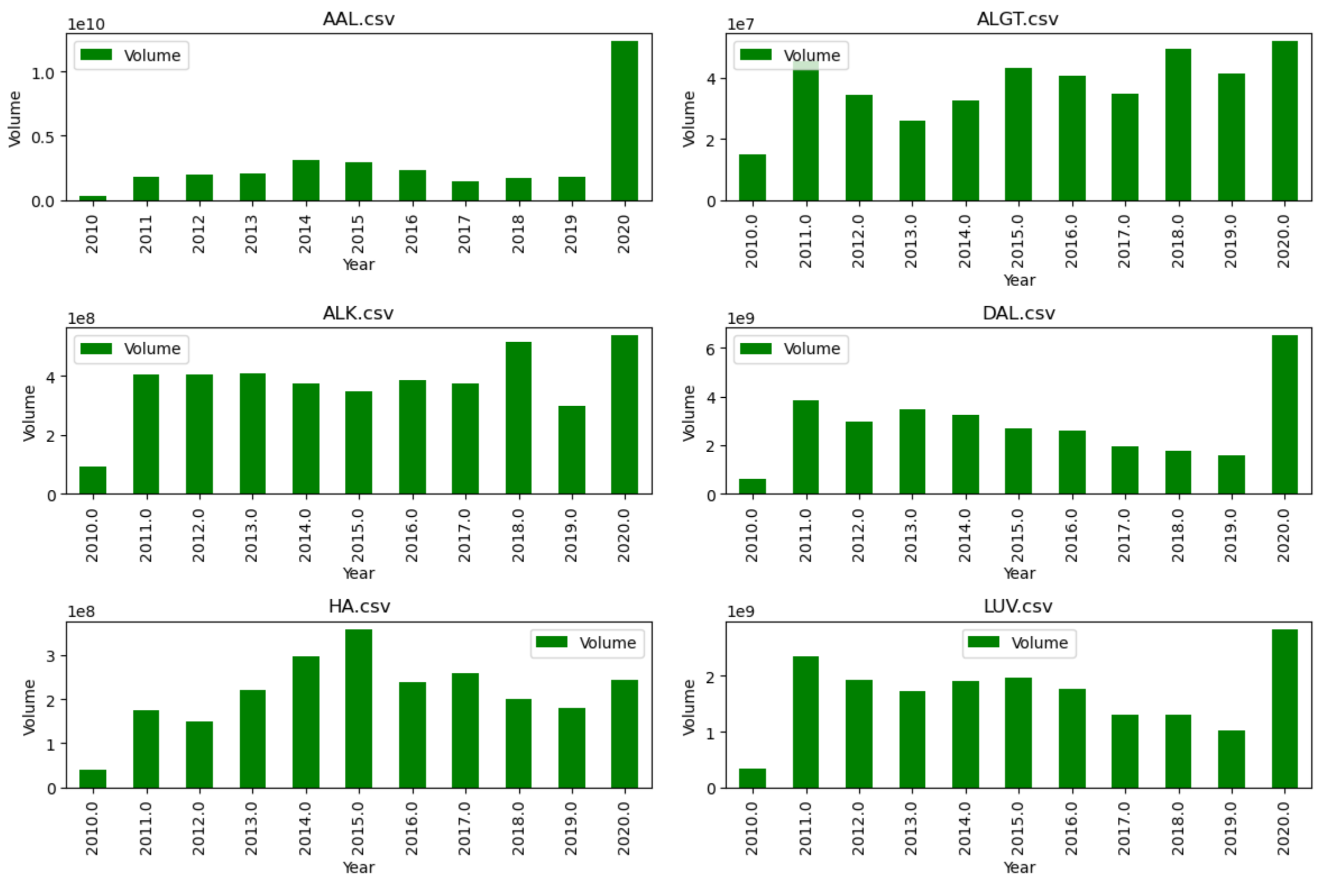


Line plot of different aviation companies

As seen from the plots above, the aviation industry has been relatively stable without any meteoric increase or decreases but the growth seemed to have plateaued by the late 2020s.

Alaska Airlines Group, Hawaiian Holdings Inc., and Allegiant Travel Company have shown a similar trend.

While Delta Airlines and Southwest Airlines seemed to do well until early 2019 by decreased by 2020 like other aviation companies



Volume traded per year for different aviation companies.

The peak volume traded for

American airline group in 2020 [with a drastic decrease in price indicating severe losses.]

Followed by Delta Airlines and Southwest Airlines around 2020 [with a similar trend across each year.]

Alaskan air group and allegiant travel company in 2020.

2020 has been a bad year for the aviation industry with several big companies trading at less price with high volume traded

Correlation analysis of different companies in each sector

Correlation:

Correlation is a statistical measure that quantifies the degree to which two variables move in relation to each other. It ranges from -1 to 1, where:

1 indicates a perfect positive correlation (both variables move in the same direction),

0 indicates no correlation,

-1 indicates a perfect negative correlation (both variables move in opposite directions).

Positive Correlation:

When two stocks have a positive correlation, it means that their prices tend to move in the same direction. If one stock goes up, the other stock is more likely to go up as well.

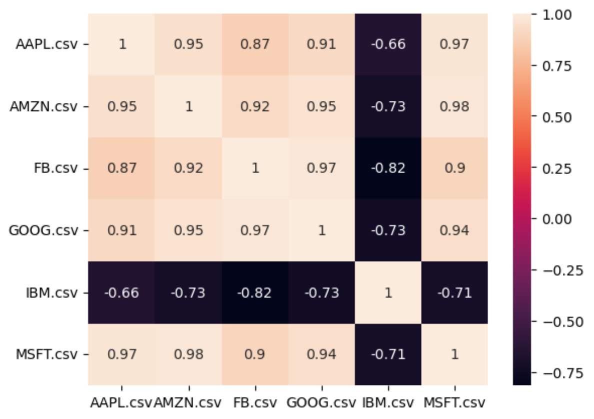
Negative Correlation:

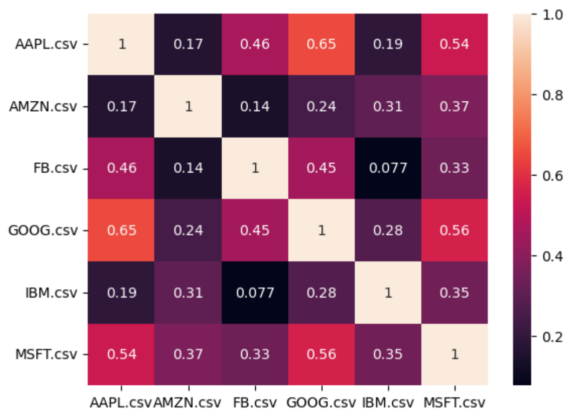
Negative correlation implies that the two stocks move in opposite directions. When one stock's price goes up, the other stock's price tends to go down, and vice versa.

No Correlation:

A correlation of 0 suggests no linear relationship between the two variables. Changes in one variable are not predictive of changes in the other. correlation is one of the important aspects of picking good stocks if the picked companies are highly correlated this might lead to concentration risk and could lead to huge potential losses to avoid this we should analyze the correlation of different companies across each individual sector to find the ideal stock.

Correlation analysis of technology companies



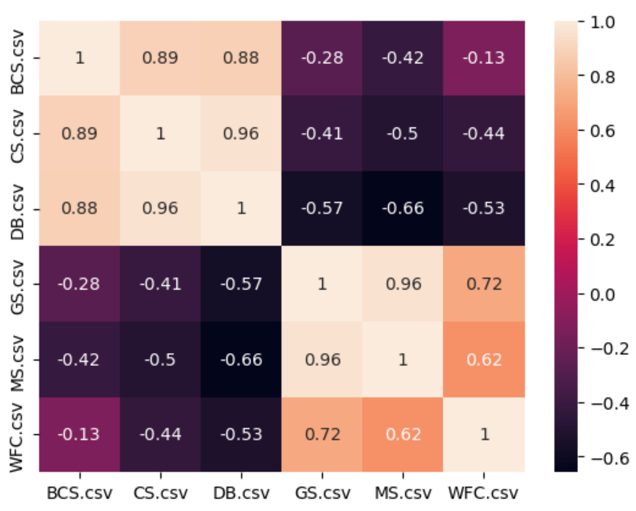


Heatmap of closing prices correlation   Heatmap of volume traded correlation

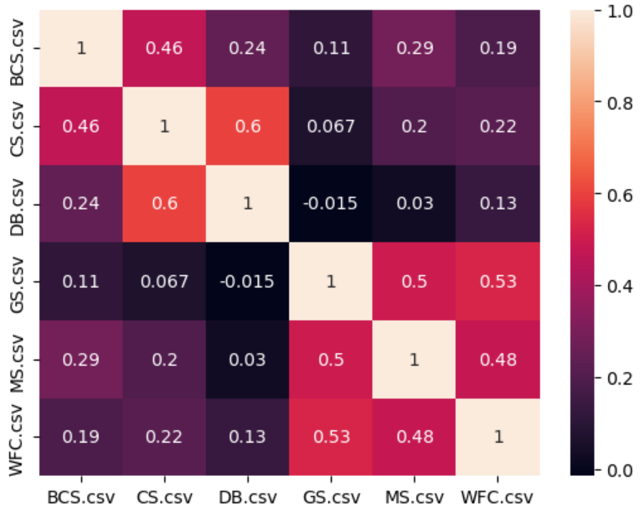
As it can be inferred from the correlation matrix Amazon and Apple are highly correlated also, Microsoft and Apple are highly correlated with respect to prices of the trading period, and Google and Apple and Microsoft and Apple are highly correlated in terms of volume.

IBM seems to be negatively correlated with several other technology companies indicating a good opportunity to diversify portfolio.

Correlation analysis of finance companies



	BCS.csv	CS.csv	DB.csv	GS.csv	MS.csv	WFC.csv
BCS.csv	1	0.89	0.88	-0.28	-0.42	-0.13
CS.csv	0.89	1	0.96	-0.41	-0.5	-0.44
DB.csv	0.88	0.96	1	-0.57	-0.66	-0.53
GS.csv	-0.28	-0.41	-0.57	1	0.96	0.72
MS.csv	-0.42	-0.5	-0.66	0.96	1	0.62
WFC.csv	-0.13	-0.44	-0.53	0.72	0.62	1



	BCS.csv	CS.csv	DB.csv	GS.csv	MS.csv	WFC.csv
BCS.csv	1	0.46	0.24	0.11	0.29	0.19
CS.csv	0.46	1	0.6	0.067	0.2	0.22
DB.csv	0.24	0.6	1	-0.015	0.03	0.13
GS.csv	0.11	0.067	-0.015	1	0.5	0.53
MS.csv	0.29	0.2	0.03	0.5	1	0.48
WFC.csv	0.19	0.22	0.13	0.53	0.48	1

Heatmap of closing prices correlation

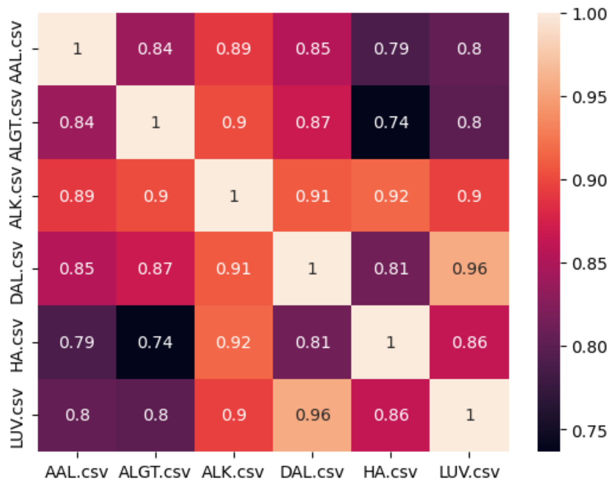
Heatmap of volume traded correlation

The correlation analysis of closing prices of finance companies shows both positive and negative correlation, unlike most of the technology companies.

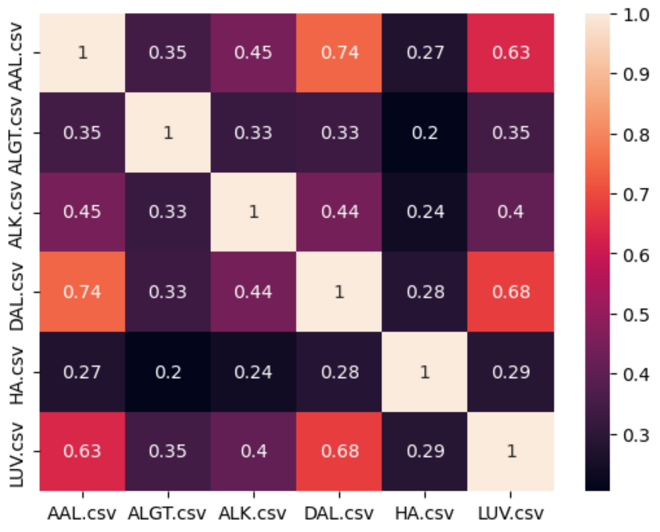
As it can be inferred from the above charts credit Suisse is highly correlated with Deutsche Bank and Morgan Stanley is highly correlated with Goldman Sachs.

But there seems to be a very high negative correlation between traditional banks and investment banks as expected which presents us with a chance to categorize the finance companies into two subsets and a potential opportunity to diversify the assets.

Correlation analysis of Aviation companies



	AAL.csv	ALGT.csv	ALK.csv	DAL.csv	HA.csv	LUV.csv
AAL.csv	1	0.84	0.89	0.85	0.79	0.8
ALGT.csv	0.84	1	0.9	0.87	0.74	0.8
ALK.csv	0.89	0.9	1	0.91	0.92	0.9
DAL.csv	0.85	0.87	0.91	1	0.81	0.96
HA.csv	0.79	0.74	0.92	0.81	1	0.86
LUV.csv	0.8	0.8	0.9	0.96	0.86	1



	AAL.csv	ALGT.csv	ALK.csv	DAL.csv	HA.csv	LUV.csv
AAL.csv	1	0.35	0.45	0.74	0.27	0.63
ALGT.csv	0.35	1	0.33	0.33	0.2	0.35
ALK.csv	0.45	0.33	1	0.44	0.24	0.4
DAL.csv	0.74	0.33	0.44	1	0.28	0.68
HA.csv	0.27	0.2	0.24	0.28	1	0.29
LUV.csv	0.63	0.35	0.4	0.68	0.29	1

Heatmap of closing prices correlation

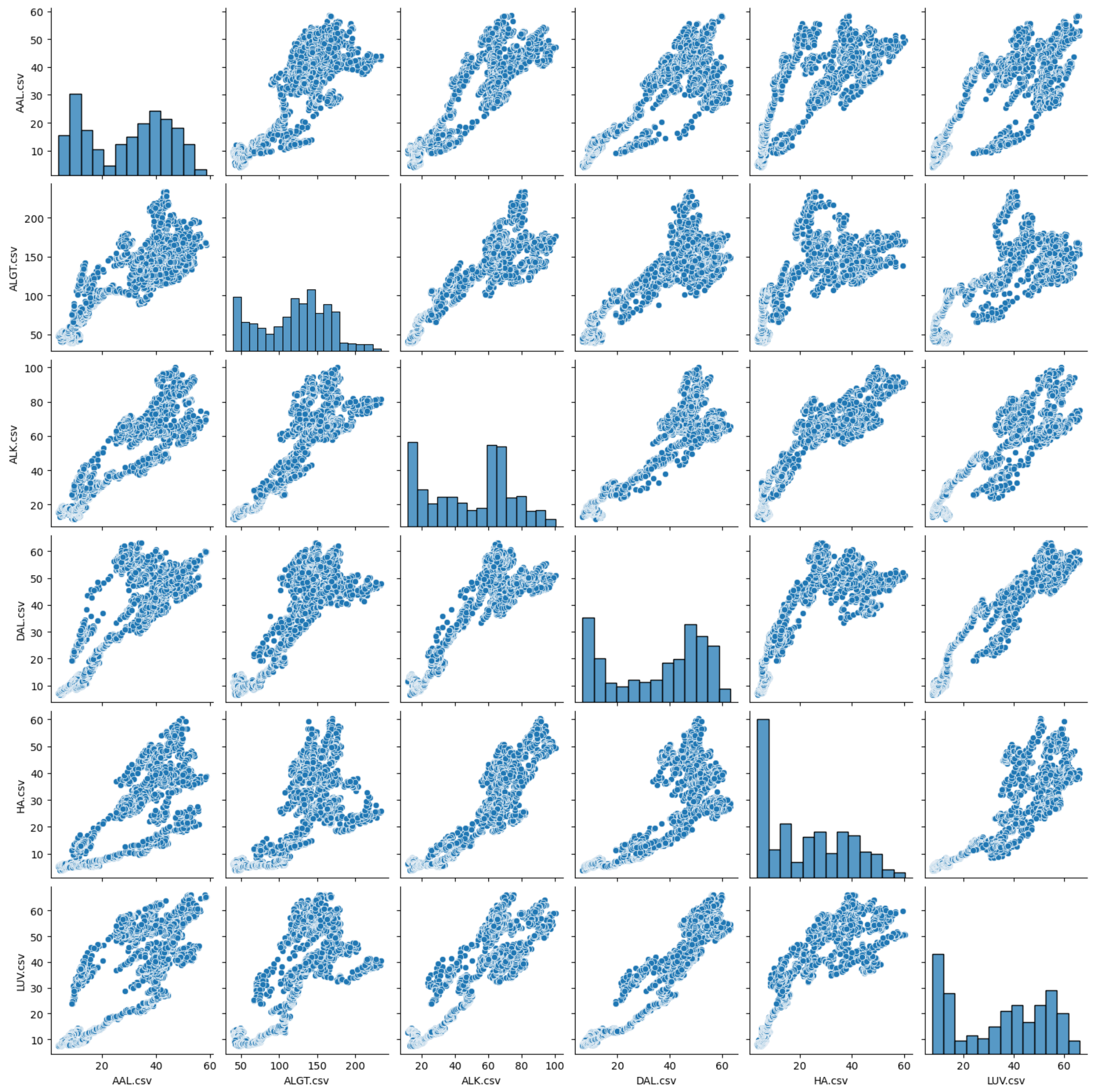
Heatmap of volume traded correlation

The correlation analysis of closing prices of aviation companies shows positively correlated values with values ranging from 74 percent to 96 percent.

Here Delta Airlines seems to be highly correlated with Southwest Airlines cooperation.

Even in terms of volume Delta Airlines is highly correlated with Southwest airline corporation and so is American Airlines.

There seem to be no negative correlations between any of the airline companies indicating a relatively similar trend further investigation is carried out below using a pair plot to check the trend of the data points.



pair of closing prices of aviation companies

The pair plot does show that there are no negatively correlated columns in the aviation industry.

Experiments and machine learning modeling

Classification task

Stock selection

Having conducted thorough Exploratory Data Analysis (EDA) and drawn meaningful inferences from companies operating in the Technology, Finance, and Aviation sectors, we have meticulously selected one exemplary company from each domain. The overarching objective of this undertaking is to employ predictive modeling through binary classification. Specifically, we aim to predict whether the stocks of these chosen companies are poised to move upwards or downwards in the future. This predictive analysis serves as a crucial decision-making tool, guiding us in determining whether investing in these stocks aligns with our strategic financial goals.

From the Technology sector – Since Apple is highly correlated with several other technology companies Apple is selected as the representative company.

From the finance sector - Morgan Stanley has been selected since it has a relatively stable stock with less volume traded and good price elevation over the years

From the Aviation sector – Allegiant travel company is selected since it has relatively stable stock with less meteoric decline unlike the other companies in the sector.

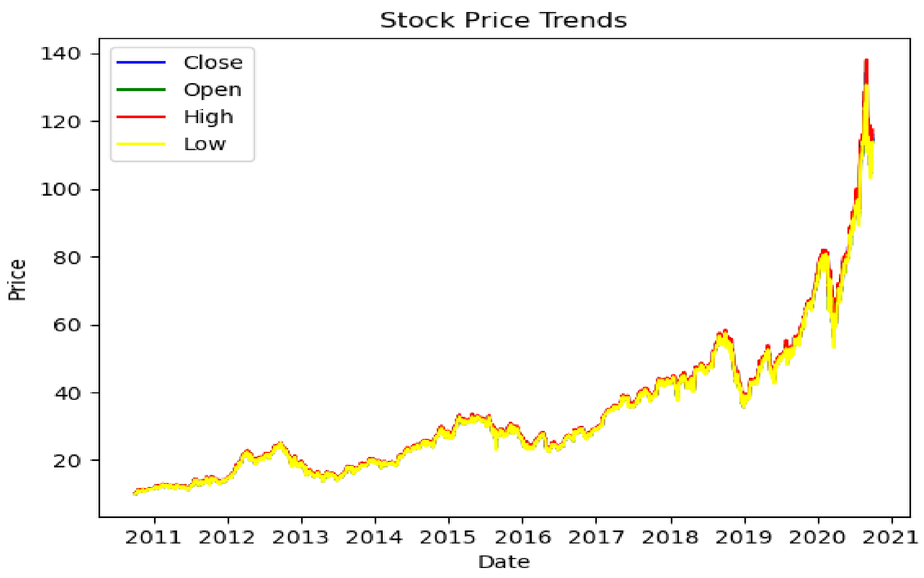
In this section, we import many base models on a sample of training data and compare these performance metrics using the traditional classification metrics like Accuracy, Precision, Recall, and f1 score.

Importing the required libraries

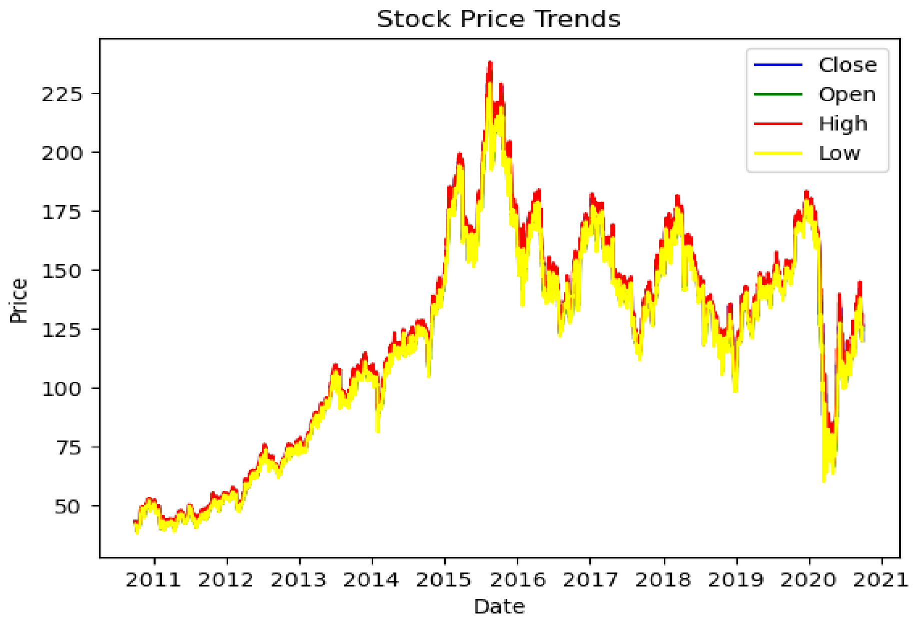
```
from sklearn.preprocessing import StandardScaler
import time
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import ElasticNet
from sklearn.model_selection import train_test_split
from sklearn.metrics import ConfusionMatrixDisplay, confusion_matrix
from sklearn.metrics import recall_score, precision_score, f1_score
from sklearn.linear_model import LogisticRegression
model1 = LogisticRegression()
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.svm import SVC
model2 = SVC()
from sklearn.neighbors import KNeighborsClassifier
model4 = KNeighborsClassifier()
from sklearn.ensemble import RandomForestClassifier
model6 = RandomForestClassifier(n_estimators=30)
from sklearn.model_selection import train_test_split
from sklearn.ensemble import AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier
base_classifier = DecisionTreeClassifier(max_depth=1)
```

We have imported all the required models from the sci kit learn library which is an open-source library used for machine-learning tasks.

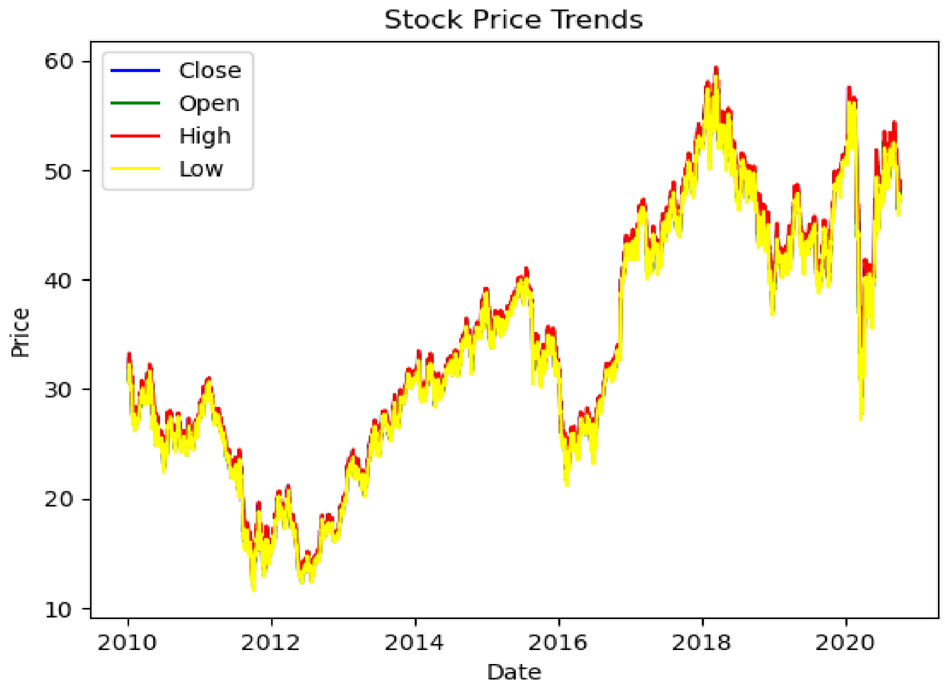
Apple stock trend



Allegiant travel company stock trend

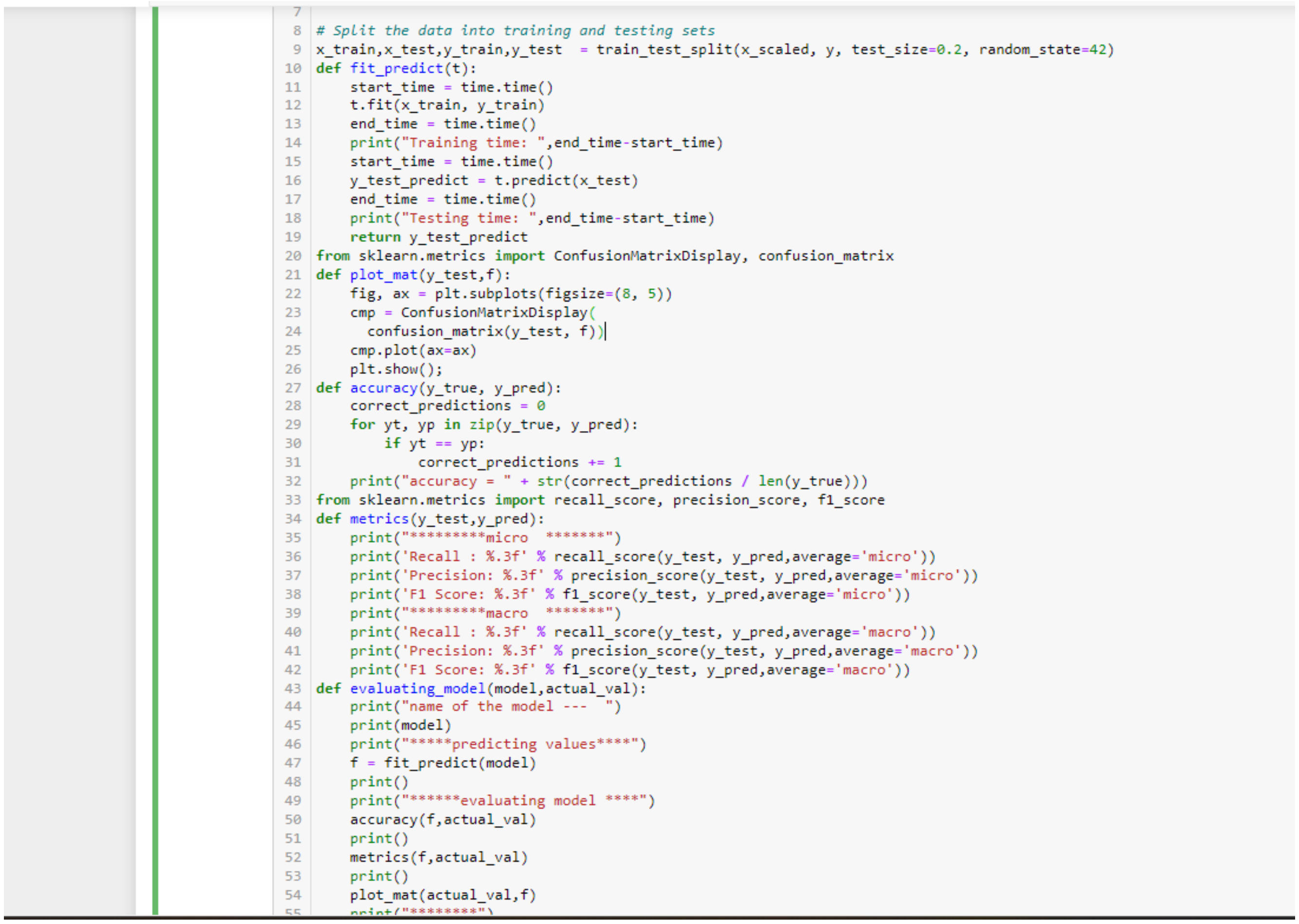


Morgan Stanley stock trend



The stock trend for each of the selected stocks is presented above.

Modeling and code for each of the stocks.



```
# Split the data into training and testing sets
x_train,x_test,y_train,y_test  = train_test_split(x_scaled, y, test_size=0.2, random_state=42)
def fit_predict(t):
    start_time = time.time()
    t.fit(x_train, y_train)
    end_time = time.time()
    print("Training time: ",end_time-start_time)
    start_time = time.time()
    y_test_predict = t.predict(x_test)
    end_time = time.time()
    print("Testing time: ",end_time-start_time)
    return y_test_predict
from sklearn.metrics import ConfusionMatrixDisplay, confusion_matrix
def plot_mat(y_test,f):
    fig, ax = plt.subplots(figsize=(8, 5))
    cmp = ConfusionMatrixDisplay(
      confusion_matrix(y_test, f))
    cmp.plot(ax=ax)
    plt.show();
def accuracy(y_true, y_pred):
    correct_predictions = 0
    for yt, yp in zip(y_true, y_pred):
        if yt == yp:
            correct_predictions += 1
    print("accuracy = " + str(correct_predictions / len(y_true)))
from sklearn.metrics import recall_score, precision_score, f1_score
def metrics(y_test,y_pred):
    print("*********micro  *******")
    print('Recall : %.3f' % recall_score(y_test, y_pred,average='micro'))
    print('Precision: %.3f' % precision_score(y_test, y_pred,average='micro'))
    print('F1 Score: %.3f' % f1_score(y_test, y_pred,average='micro'))
    print("*********macro  *******")
    print('Recall : %.3f' % recall_score(y_test, y_pred,average='macro'))
    print('Precision: %.3f' % precision_score(y_test, y_pred,average='macro'))
    print('F1 Score: %.3f' % f1_score(y_test, y_pred,average='macro'))
def evaluating_model(model,actual_val):
    print("name of the model ---  ")
    print(model)
    print("*****predicting values****")
    f = fit_predict(model)
    print()
    print("******evaluating model ****")
    accuracy(f,actual_val)
    print()
    metrics(f,actual_val)
    print()
    plot_mat(actual_val,f)
    print("*********")
```

The code given above is used for the classification on stock price data using various machine learning models. The data is preprocessed to create a binary target variable, 'Label', which indicates whether the stock price will increase (1) or decrease (0) in the next period. Features such as 'Open', 'High', 'Low', and 'Volume' are used for prediction. The dataset is split into training and testing sets, with 80% used for training and 20% for testing.

The fit_predict function is defined to train the given model (t) on the training data and make predictions on the test data. The training and testing times are recorded and printed. The plot_mat function utilizes ConfusionMatrixDisplay from scikit-learn to visualize the confusion matrix for the predicted and actual values.

The accuracy function calculates and prints the accuracy of the model by comparing the predicted values to the actual values. The metrics function computes and prints various evaluation metrics such as micro and macro recall, precision, and F1 score.

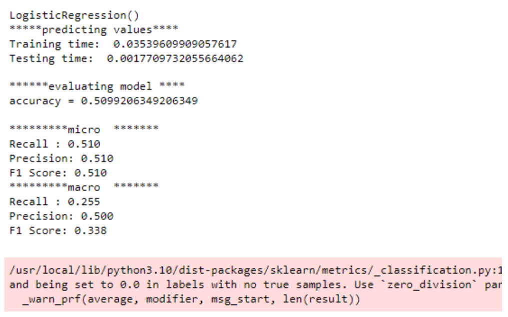
The evaluating_model function takes a model and the actual test values, prints the model name, predicts values using the fit_predict function, evaluates the model using accuracy and metrics, and visualizes the confusion matrix using plot_mat. It a comprehensive framework for training, predicting, and evaluating multiple classification models on stock price data.

Brief description of each of the models that have been selected for modeling and the Evaluation results.

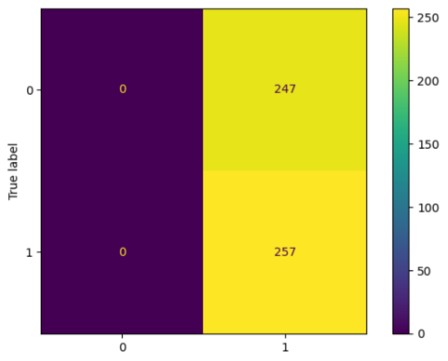
Logistic regression

Logistic regression is a linear classification algorithm that is widely used for binary classification tasks. It is computationally efficient and easy to model it uses a sigmoid function to squish the best-fit line and probabilities are calculated and classified based on if the points fall below or above the point of inflexion.

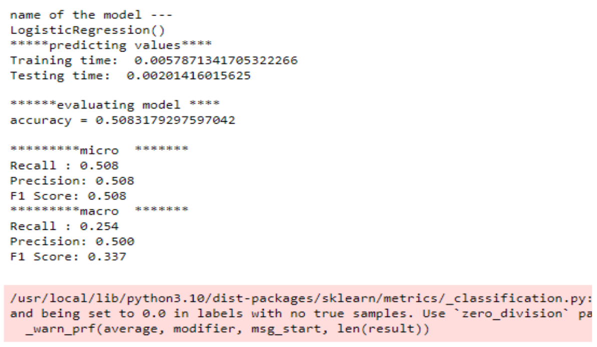
Performance on Apple stock



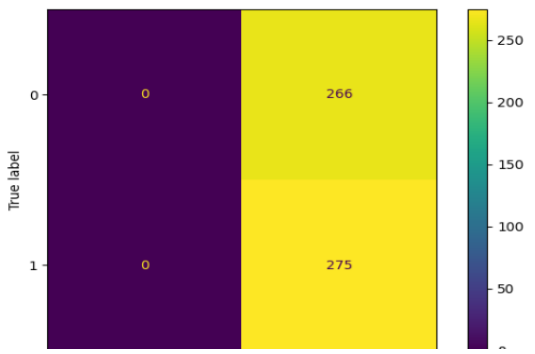
```
LogisticRegression()
*****predicting values****
Training time:  0.03539609909057617
Testing time:  0.0017709732055664062

******evaluating model ****
accuracy = 0.5099206349206349

*********micro  *******
Recall : 0.510
Precision: 0.510
F1 Score: 0.510
*********macro  *******
Recall : 0.255
Precision: 0.500
F1 Score: 0.338

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1
and being set to 0.0 in labels with no true samples. Use `zero_division` par
  _warn_prf(average, modifier, msg_start, len(result))
```



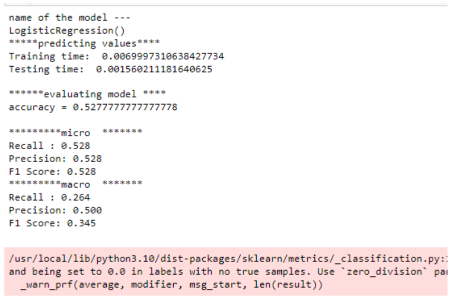
Performance on Morgan Stanley stock



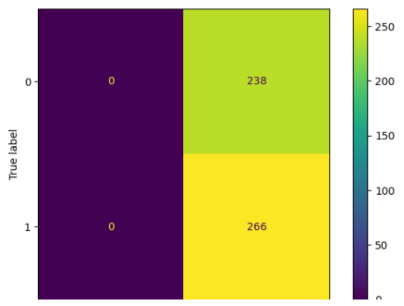
```
name of the model ---
LogisticRegression()
*****predicting values****
Training time:  0.0057871341705322266
Testing time:  0.00201416015625

******evaluating model ****
accuracy = 0.5083179297597042

*********micro  *******
Recall : 0.508
Precision: 0.508
F1 Score: 0.508
*********macro  *******
Recall : 0.254
Precision: 0.500
F1 Score: 0.337

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:
and being set to 0.0 in labels with no true samples. Use `zero_division` pa
  _warn_prf(average, modifier, msg_start, len(result))
```



Performance on Allegiant Travel Company stock



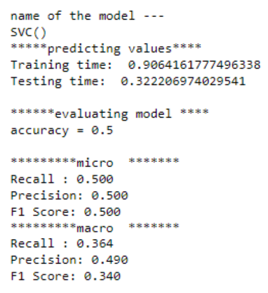
```
name of the model ---
LogisticRegression()
*****predicting values****
Training time:  0.0069997310638427734
Testing time:  0.0015602111816406250

******evaluating model ****
accuracy = 0.5277777777777778

*********micro  *******
Recall : 0.528
Precision: 0.528
F1 Score: 0.528
*********macro  *******
Recall : 0.264
Precision: 0.500
F1 Score: 0.345

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:
and being set to 0.0 in labels with no true samples. Use `zero_division` pa
  _warn_prf(average, modifier, msg_start, len(result))
```



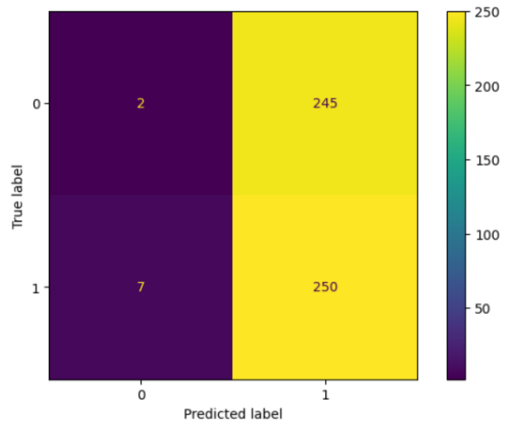
Support vector machine classifier

The support vector machine aims to find the hyperplane that best separates the data into different classes it is one of the best models for both linear and non-linear relationships in the data using different kernels.

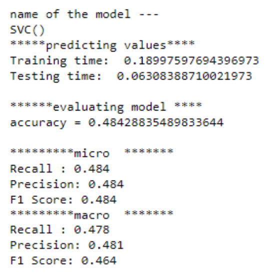
Performance on Apple stock



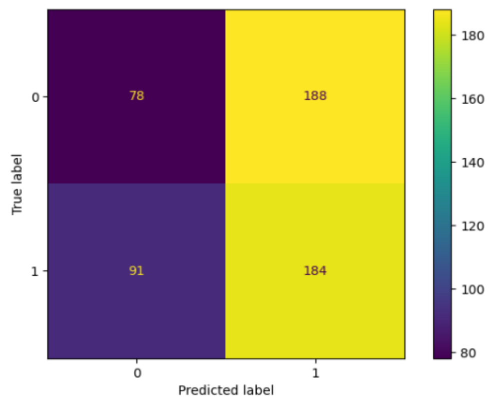
```
name of the model ---
SVC()
*****predicting values****
Training time:  0.9064161777496338
Testing time:  0.322206974029541

******evaluating model ****
accuracy = 0.5

*********micro  *******
Recall : 0.500
Precision: 0.500
F1 Score: 0.500
*********macro  *******
Recall : 0.364
Precision: 0.490
F1 Score: 0.340
```



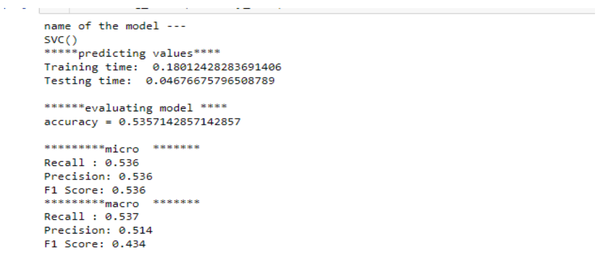
Performance on Morgan Stanley stock



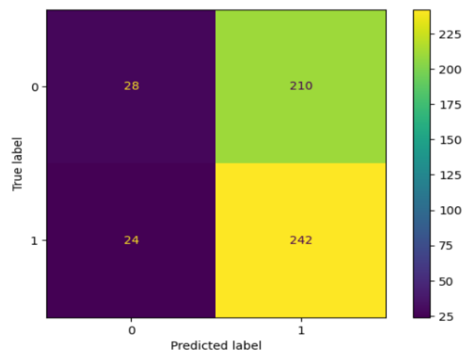
```
name of the model ---
SVC()
*****predicting values****
Training time:  0.18997597694396973
Testing time:  0.06308388710021973

******evaluating model ****
accuracy = 0.48428835489833644

*********micro  *******
Recall : 0.484
Precision: 0.484
F1 Score: 0.484
*********macro  *******
Recall : 0.478
Precision: 0.481
F1 Score: 0.464
```



Performance on Allegiant Travel Company stock



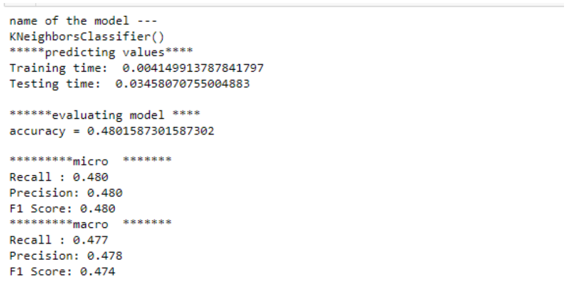
```
name of the model ---
SVC()
*****predicting values****
Training time:  0.18012428283691406
Testing time:  0.04676675796508789

******evaluating model ****
accuracy = 0.5357142857142857

*********micro  *******
Recall : 0.536
Precision: 0.536
F1 Score: 0.536
*********macro  *******
Recall : 0.537
Precision: 0.514
F1 Score: 0.434
```



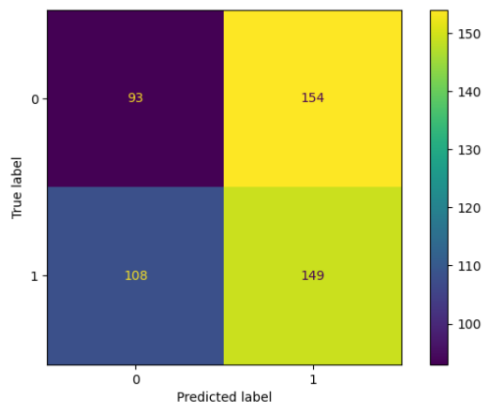
K nearest neighbor

K nearest neighbors is a simple and intuitive algorithm also called as lazy learner the advantages of k nearest neighbors it does not make any assumptions about the underlying data distribution and is nonparametric and only computes predictions at the time of the predictions. The results below are presented after hyperparameter tuning of the model

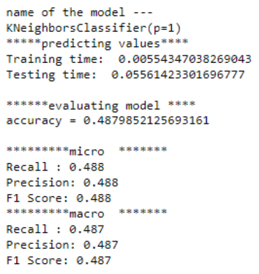
Performance on Apple stock



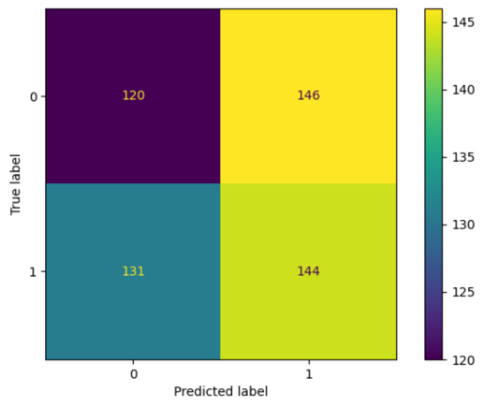
```
name of the model ---
KNeighborsClassifier()
*****predicting values****
Training time:  0.004149913787841797
Testing time:  0.03458070755004883

******evaluating model ****
accuracy = 0.4801587301587302

*********micro  *******
Recall : 0.480
Precision: 0.480
F1 Score: 0.480
*********macro  *******
Recall : 0.477
Precision: 0.478
F1 Score: 0.474
```



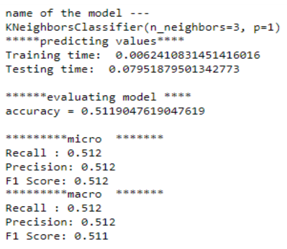
Performance of Morgan Stanley bank



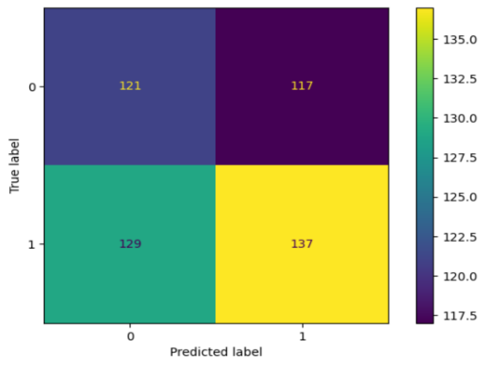
```
name of the model ---
KNeighborsClassifier(p=1)
*****predicting values****
Training time:  0.00554347038269043
Testing time:  0.05561423301696777

******evaluating model ****
accuracy = 0.4879852125693161

*********micro  *******
Recall : 0.488
Precision: 0.488
F1 Score: 0.488
*********macro  *******
Recall : 0.487
Precision: 0.487
F1 Score: 0.487
```



Performance on Allegiant Travel Company stock



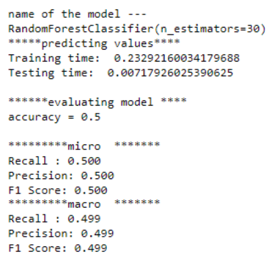
```
name of the model ---
KNeighborsClassifier(n_neighbors=3, p=1)
*****predicting values****
Training time:  0.0062410831451416016
Testing time:  0.07951879501342773

******evaluating model ****
accuracy = 0.5119047619047619

*********micro  *******
Recall : 0.512
Precision: 0.512
F1 Score: 0.512
*********macro  *******
Recall : 0.512
Precision: 0.512
F1 Score: 0.511
```



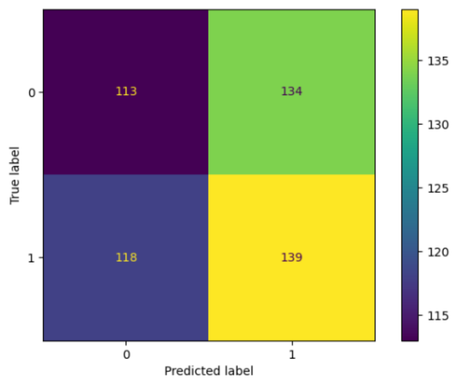
Random forest

Random forest is an ensemble learning method that constructs a multitude of decision problems it has several small decision trees embedded in it and is used for classification tasks.

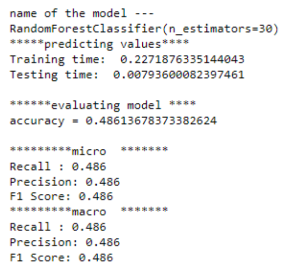
Performance on Apple stock



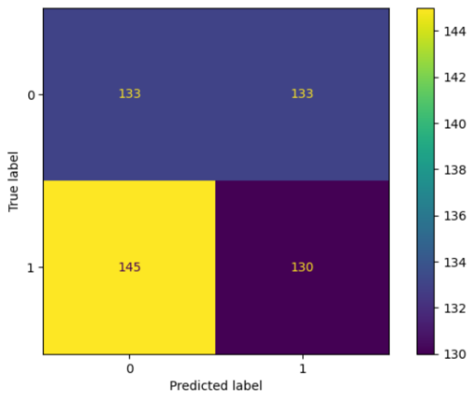
```
name of the model ---
RandomForestClassifier(n_estimators=30)
*****predicting values****
Training time:  0.23292160034179688
Testing time:  0.00717926025390625

******evaluating model ****
accuracy = 0.5

*********micro  *******
Recall : 0.500
Precision: 0.500
F1 Score: 0.500
*********macro  *******
Recall : 0.499
Precision: 0.499
F1 Score: 0.499
```



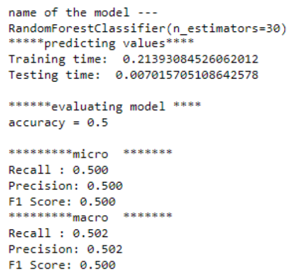
Performance on Morgan Stanley



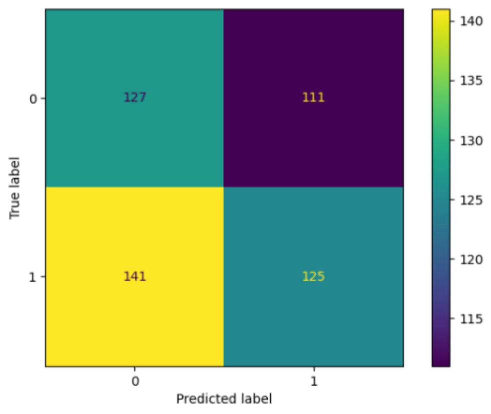
```
name of the model ---
RandomForestClassifier(n_estimators=30)
*****predicting values****
Training time:  0.2271876335144043
Testing time:  0.00793600082397461

******evaluating model ****
accuracy = 0.48613678373382624

*********micro  *******
Recall : 0.486
Precision: 0.486
F1 Score: 0.486
*********macro  *******
Recall : 0.486
Precision: 0.486
F1 Score: 0.486
```



Performance on Allegiant Travel Company stock



```
name of the model ---
RandomForestClassifier(n_estimators=30)
*****predicting values****
Training time:  0.21393084526062012
Testing time:  0.007015705108642578

******evaluating model ****
accuracy = 0.5

*********micro  *******
Recall : 0.500
Precision: 0.500
F1 Score: 0.500
*********macro  *******
Recall : 0.502
Precision: 0.502
F1 Score: 0.500
```

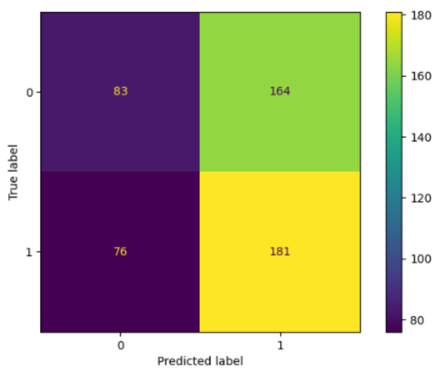


Ada boost

Ada boost combines weak learners into strong learners with a sequential training strategy and puts more emphasis on the misclassified instances.

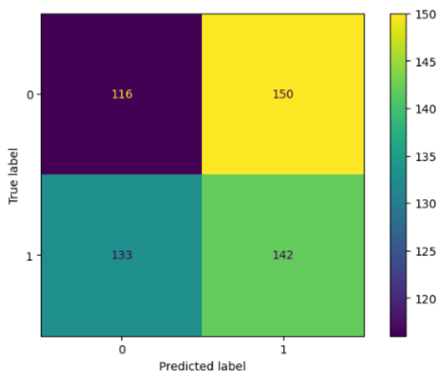
Performance on Apple stock

```
name of the model ---
AdaBoostClassifier(estimator=DecisionTreeClassifier(max_depth=1),
                   random_state=0)
*****predicting values****
Training time:  0.1649003028869629
Testing time:  0.010760784149169922

******evaluating model ****
accuracy = 0.5238095238095238

*********micro  *******
Recall : 0.524
Precision: 0.524
F1 Score: 0.524
*********macro  *******
Recall : 0.523
Precision: 0.520
F1 Score: 0.505
```



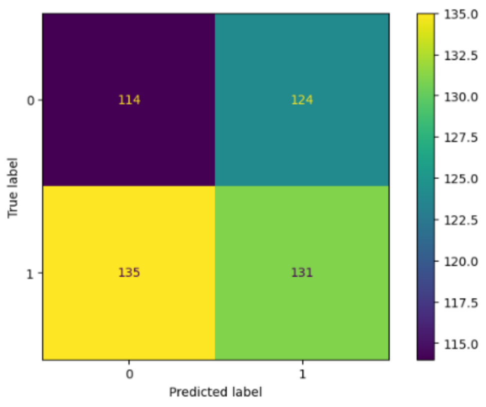
Performance on Morgan Stanley

```
name of the model ---
AdaBoostClassifier(estimator=DecisionTreeClassifier(max_depth=1),
                   random_state=0)
*****predicting values****
Training time:  0.15132355690002441
Testing time:  0.012334108352661133

******evaluating model ****
accuracy = 0.47689463955637706

*********micro  *******
Recall : 0.477
Precision: 0.477
F1 Score: 0.477
*********macro  *******
Recall : 0.476
Precision: 0.476
F1 Score: 0.476
```



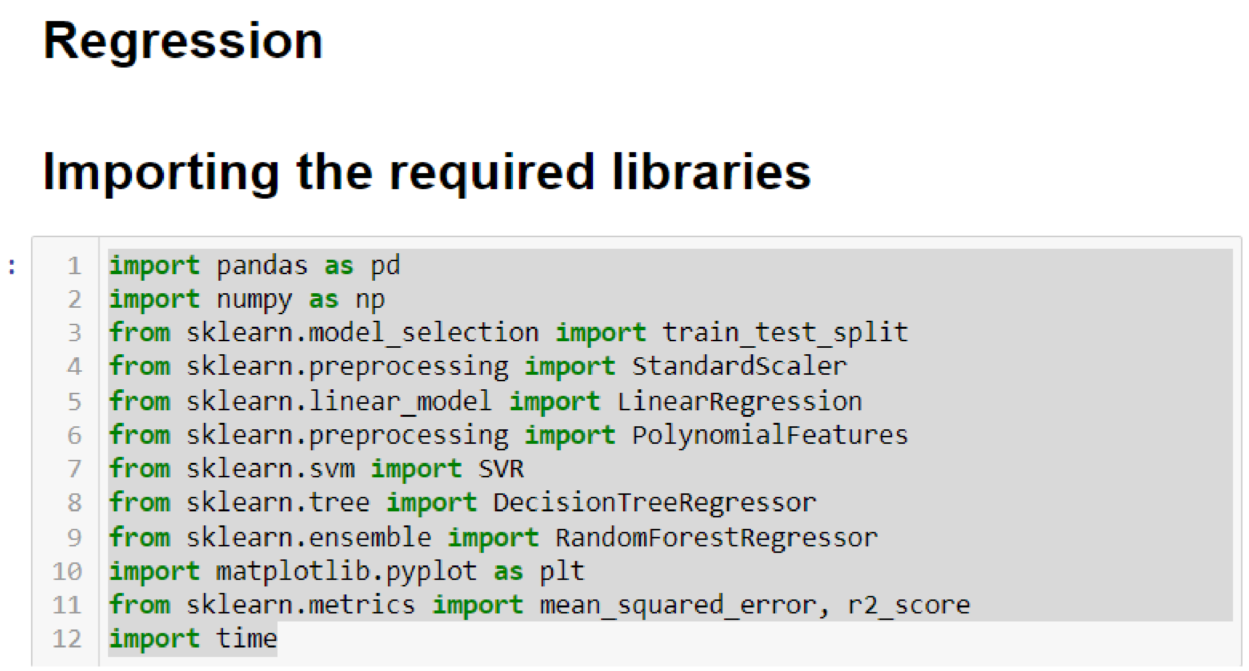
Performance on Allegiant Travel Company stock

```
name of the model ---
AdaBoostClassifier(estimator=DecisionTreeClassifier(max_depth=1),
                   random_state=0)
*****predicting values****
Training time:  0.16290950775146484
Testing time:  0.014379501342773438

******evaluating model ****
accuracy = 0.48611111111111111

*********micro  *******
Recall : 0.486
Precision: 0.486
F1 Score: 0.486
*********macro  *******
Recall : 0.486
Precision: 0.486
F1 Score: 0.486
```



Regression task

In this section, we aim to leverage a selection of exemplary stocks for each of 3 different sectors as presented above to predict the adjusted closing price based on various attributes such as open, close, volume traded for a particular day, high, and low values of each day. The primary objective is to discern the optimal values for the adjusted closing price it also provides a right check-in accessing the worthiness of a stock after the company issues dividends and stock splits (Ravi Kumar & Saraf, 2020)

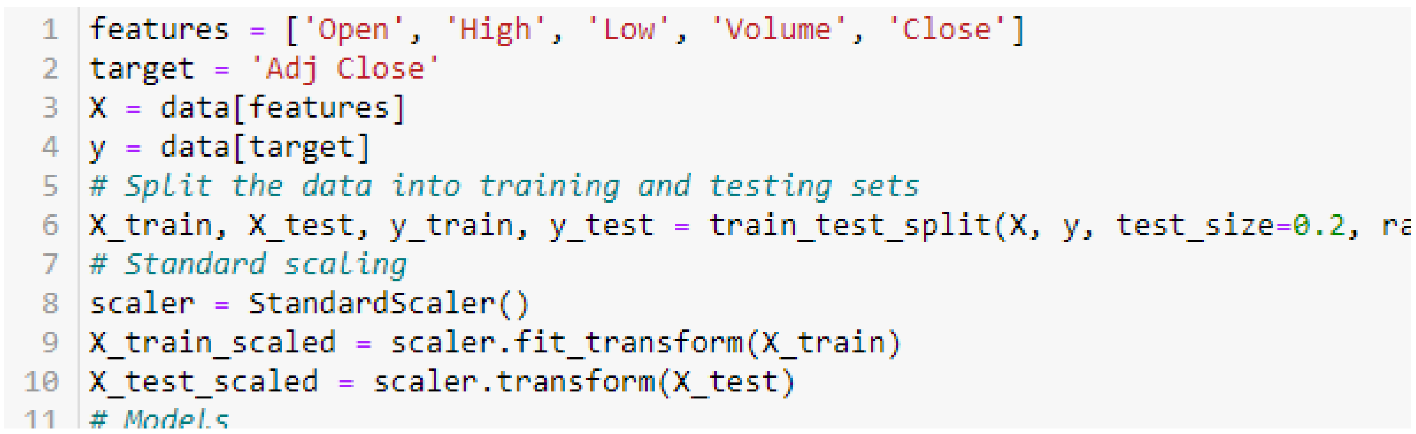


Regression

Importing the required libraries

```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, r2_score
import time
```

Using standard scalar to scale the each column to the same scale so as to not create bias towards columns with larger values.

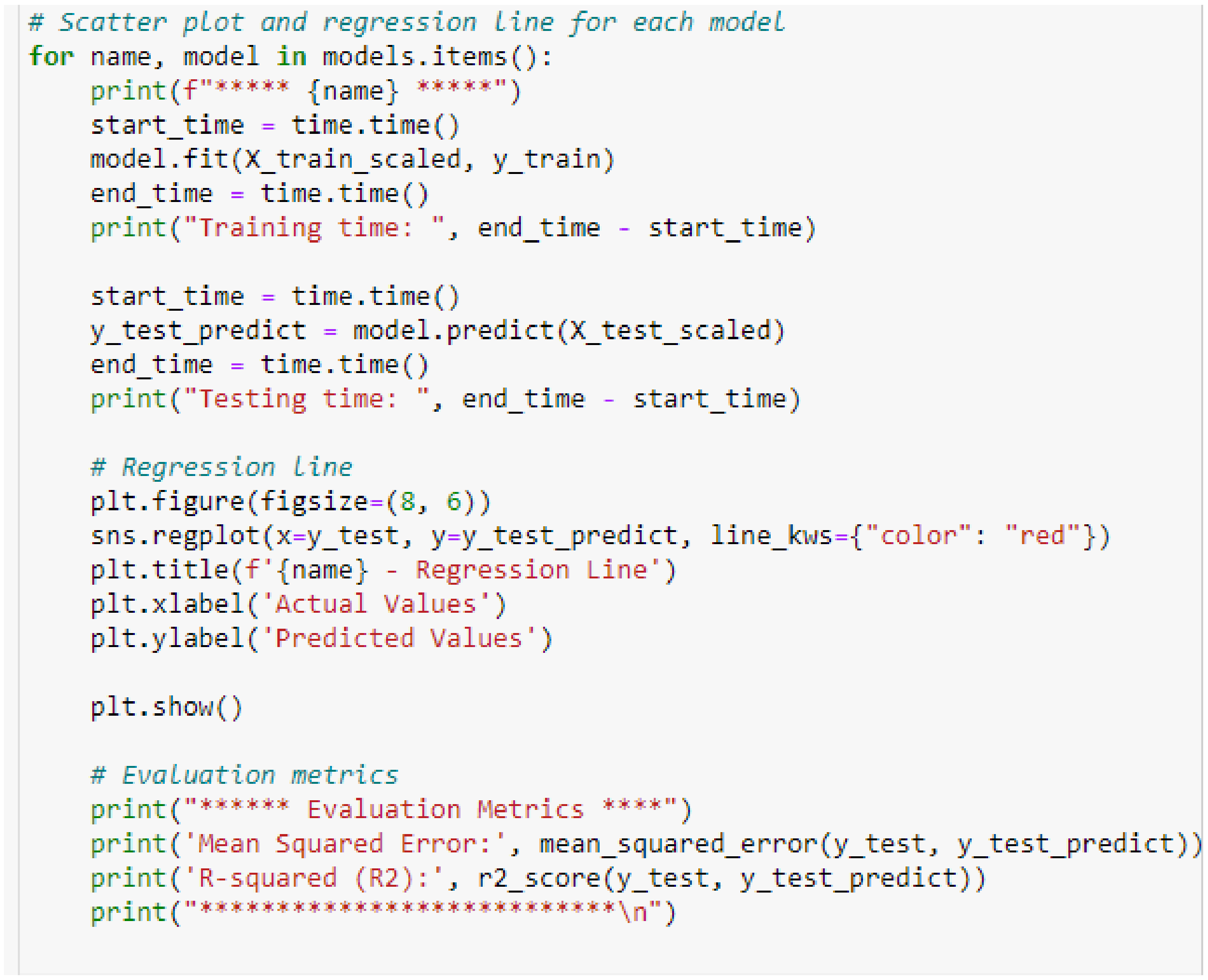


```python
features = ['Open', 'High', 'Low', 'Volume', 'Close']
target = 'Adj Close'
X = data[features]
y = data[target]
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, ra
# Standard scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
# Models
```

Four different regression models are employed: Linear Regression, Support Vector Regression (SVR), Decision Tree Regression, and Random Forest Regression. With the evaluation metrics Mean squared error and R2 squared error.

Each stock is loaded as a data frame and then indexed along with changing it to the date format.

They are scaled using the standard scalar method along with dropping of null values and fit into the dictionary containing each of the different regression models one at a time both the execution time for training and testing is registered along with the evaluation metrics for each of the models.



```
# Scatter plot and regression line for each model
for name, model in models.items():
    print(f"***** {name} *****")
    start_time = time.time()
    model.fit(X_train_scaled, y_train)
    end_time = time.time()
    print("Training time: ", end_time - start_time)

    start_time = time.time()
    y_test_predict = model.predict(X_test_scaled)
    end_time = time.time()
    print("Testing time: ", end_time - start_time)

    # Regression line
    plt.figure(figsize=(8, 6))
    sns.regplot(x=y_test, y=y_test_predict, line_kws={"color": "red"})
    plt.title(f'{name} - Regression Line')
    plt.xlabel('Actual Values')
    plt.ylabel('Predicted Values')

    plt.show()

    # Evaluation metrics
    print("****** Evaluation Metrics ****")
    print('Mean Squared Error:', mean_squared_error(y_test, y_test_predict))
    print('R-squared (R2):', r2_score(y_test, y_test_predict))
    print("***************************\n")
```

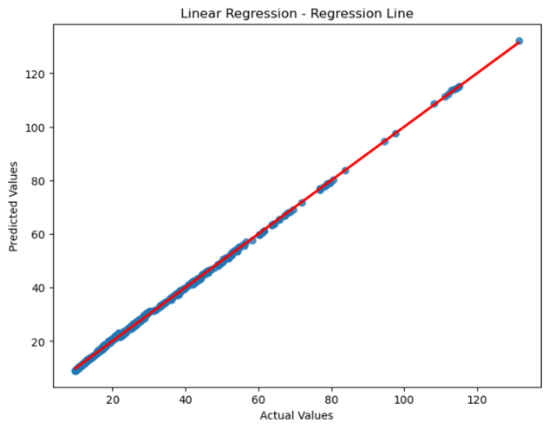
Evaluation results

For the technology sector (Apple stock)

Model Performance

Linear Regression

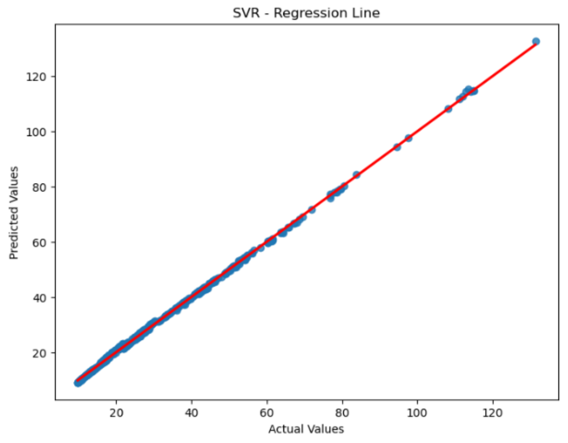
```
***** Linear Regression *****
Training time:  0.0030095577239990234
Testing time:  0.0
```



```
****** Evaluation Metrics ****
Mean Squared Error: 0.22310658974693703
R-squared (R2): 0.9994431825279091
***************************
```

Support vector machine regression with linear kernel

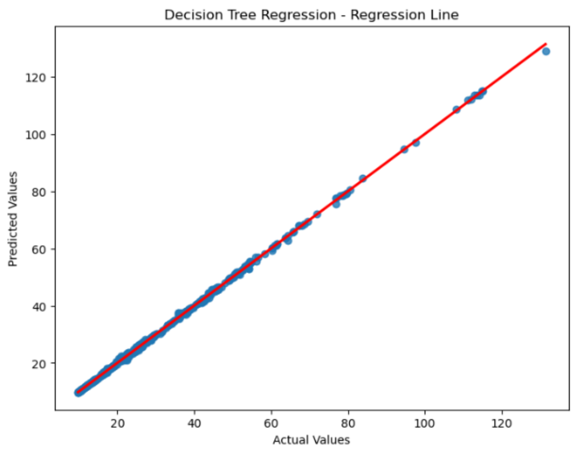
```
***** SVR *****
Training time:  0.23198390007019043
Testing time:  0.020281076431274414
```



```
****** Evaluation Metrics ****
Mean Squared Error: 0.26605460840412914
R-squared (R2): 0.9993359951642049
***************************
```

Decision Tree Regressor

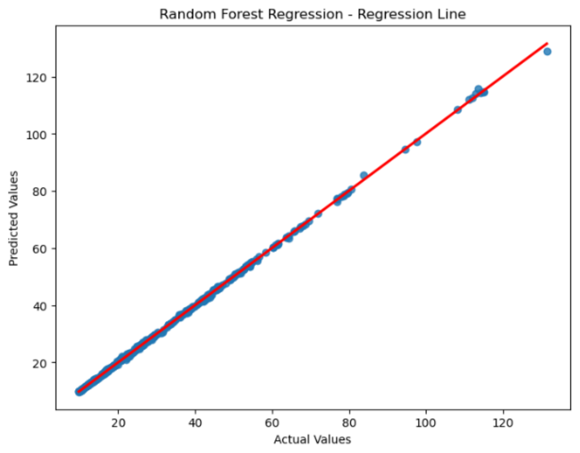
```
***** Decision Tree Regression *****
Training time:  0.022939920425415 04
Testing time:  0.0
```



```
****** Evaluation Metrics ****
Mean Squared Error: 0.17535145432636293
R-squared (R2): 0.9995623672360542
***************************
```

Random Forest Regressor

```
***** Random Forest Regression *****
Training time:  1.6835684776306152
Testing time:  0.013990163803100586
```



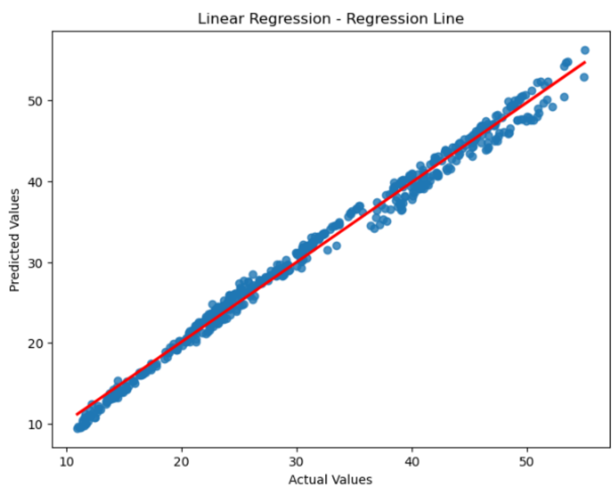
```
****** Evaluation Metrics ****
Mean Squared Error: 0.11235932410750697
R-squared (R2): 0.9997195796193813
***************************
```

For the Finance sector (Morgan Stanley stock)

Model Performance

Linear Regression

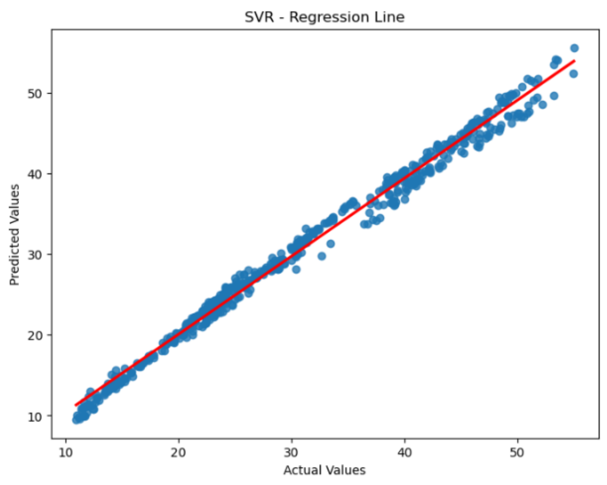
```
***** Linear Regression *****
Training time:  0.0010144710540771484
Testing time:  0.0010113716125488281
```



```
****** Evaluation Metrics ****
Mean Squared Error: 1.184081575633409
R-squared (R2): 0.9910128251217856
***************************
```

Support vector machine regression with linear kernel

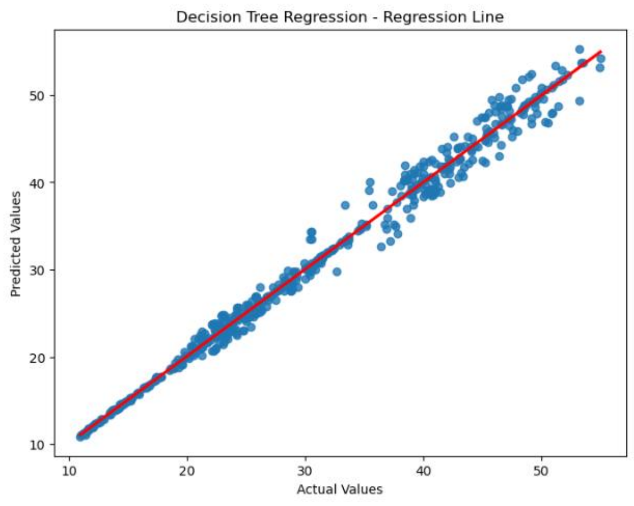
```
***** SVR *****
Training time:  0.2847576141357422
Testing time:  0.029831647872924805
```



```
****** Evaluation Metrics ****
Mean Squared Error: 1.3756066534419993
R-squared (R2): 0.9895591504736443
***************************
```

Decision Tree Regressor

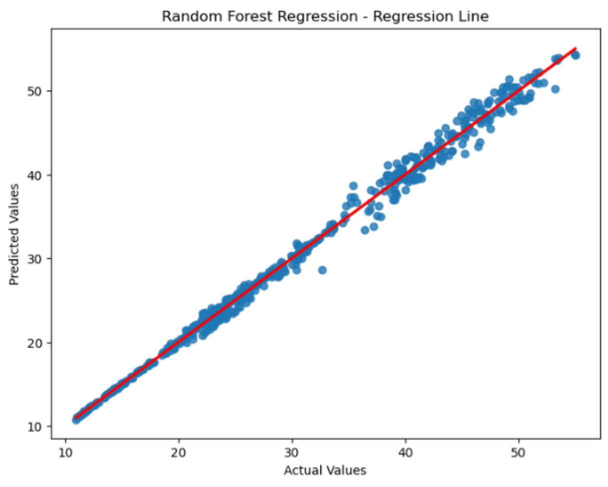
```
***** Decision Tree Regression *****
Training time:  0.027004718780517578
Testing time:  0.0010001659393310547
```



```
****** Evaluation Metrics ****
Mean Squared Error: 1.4660656350661867
R-squared (R2): 0.9888725671301556
***************************
```

Random forest Regressor

```
***** Random Forest Regression *****
Training time:  1.779094934463501
Testing time:  0.015005350112915039
```



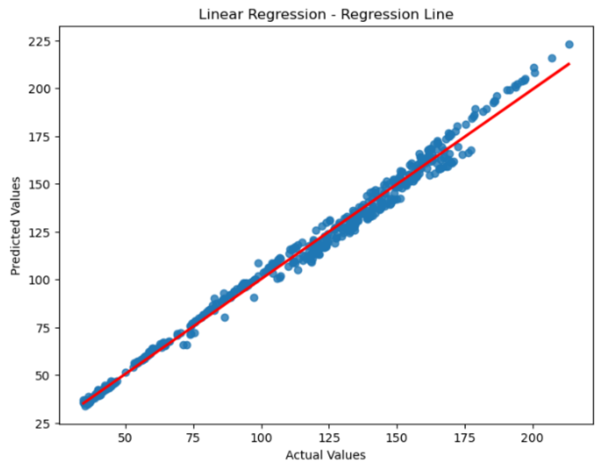
```
****** Evaluation Metrics ****
Mean Squared Error: 0.7935316288703328
R-squared (R2): 0.9939770978057512
***************************
```

For the Aviation sector (Allegiant Travel Company)

Model Performance

Linear Regression

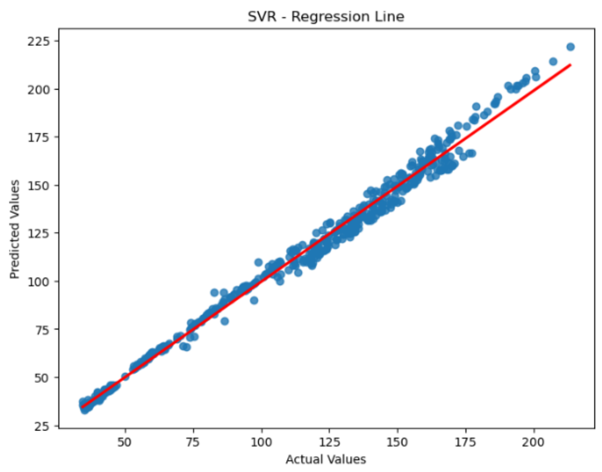
```
***** Linear Regression *****
Training time:  0.006001949310302734
Testing time:  0.000997781753540039
```



```
****** Evaluation Metrics ****
Mean Squared Error: 19.302255244699236
R-squared (R2): 0.9905172732189749
**************************
```

Support vector machine regression with linear kernel

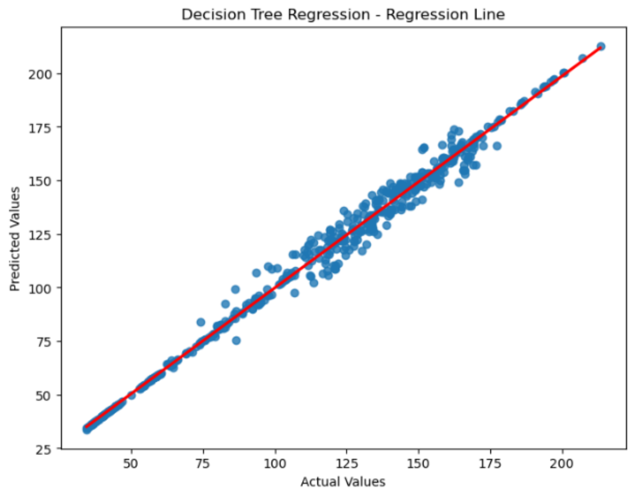
```
***** SVR *****
Training time:  0.20859622955322266
Testing time:  0.025189638137817383
```



```
****** Evaluation Metrics ****
Mean Squared Error: 20.81131204885221
R-squared (R2): 0.9897759104512869
**************************
```

Decision Tree Regressor

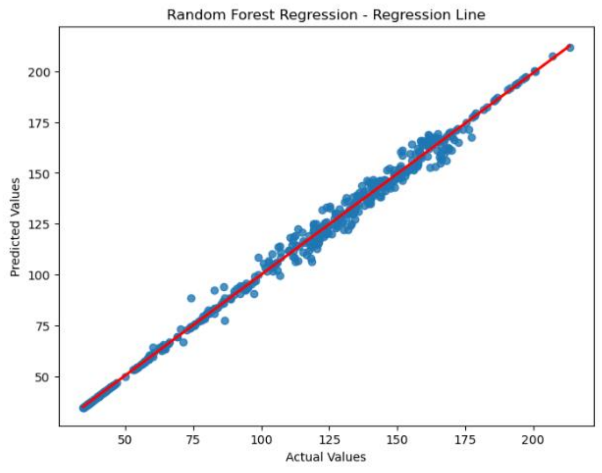
```
***** Decision Tree Regression *****
Training time:  0.02450728416442871
Testing time:  0.0010004043579101562
```



```
****** Evaluation Metrics ****
Mean Squared Error: 19.801391391984993
R-squared (R2): 0.990272059815088
**************************
```

Random forest Regressor

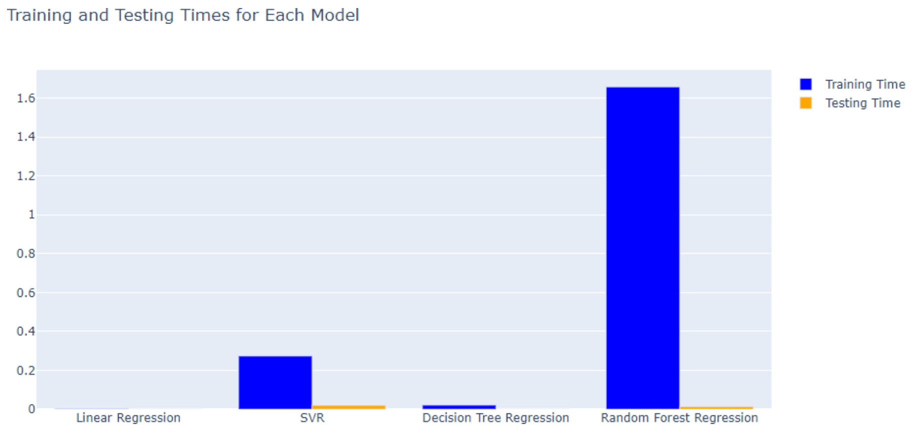
```
***** Random Forest Regression *****
Training time:  1.7210667133331299
Testing time:  0.015003204345703125
```



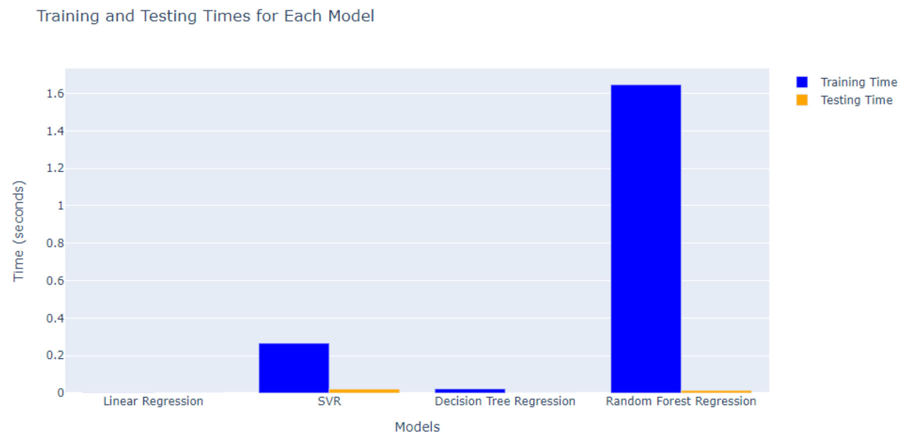
```
****** Evaluation Metrics ****
Mean Squared Error: 12.458731305032755
R-squared (R2): 0.9938793294614485
**************************
```

Training and Testing times for different stocks for each of the different stocks on various models

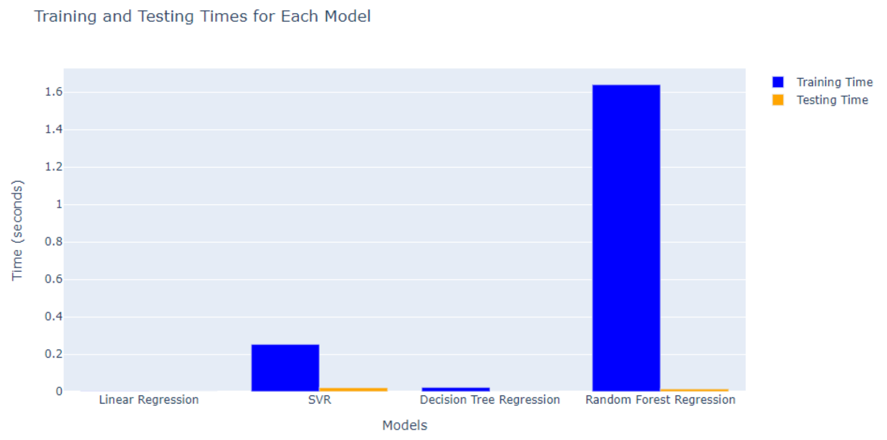
Apple



Morgan Stanley



Allegiant Travel company

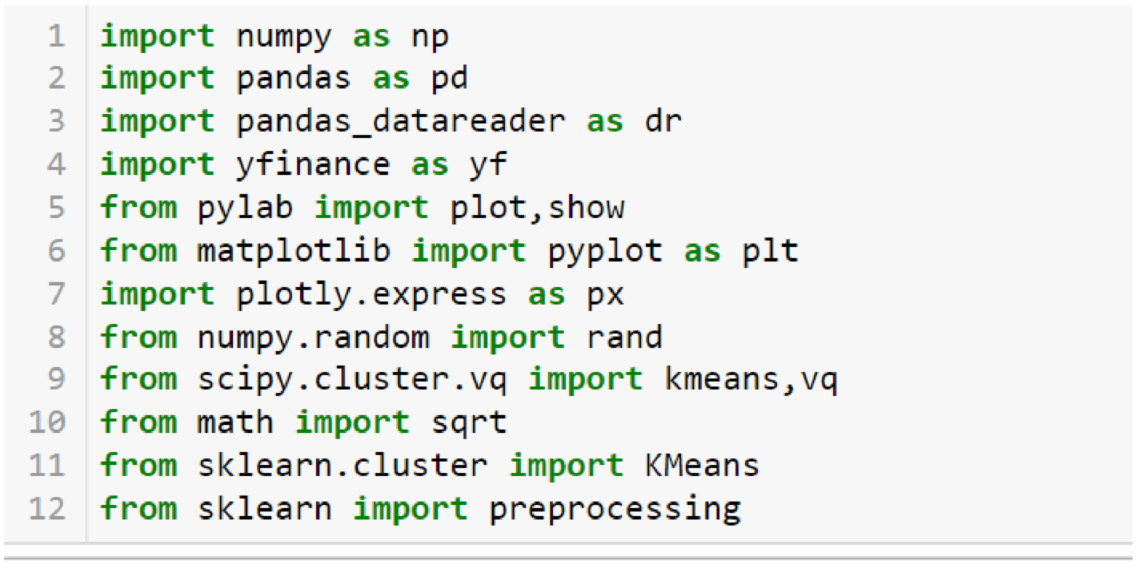


Clustering task

In this section our aim is to categorize diverse companies exhibiting varying characteristics into distinct clusters based on variability. From an investors stand point we want to create different risk profiles through cluster analysis. By grouping these companies together we would be able to identify similarities in risk profiles and analyze where our selected companies fall into and depending on the needs of the investor either aggressive or conservative we could decide to move on with the investment or reevaluation and further selection of stock is carried out. By grouping these companies together we can enhance our ability to discern the shared characteristics of risk profiles and make well-informed decisions

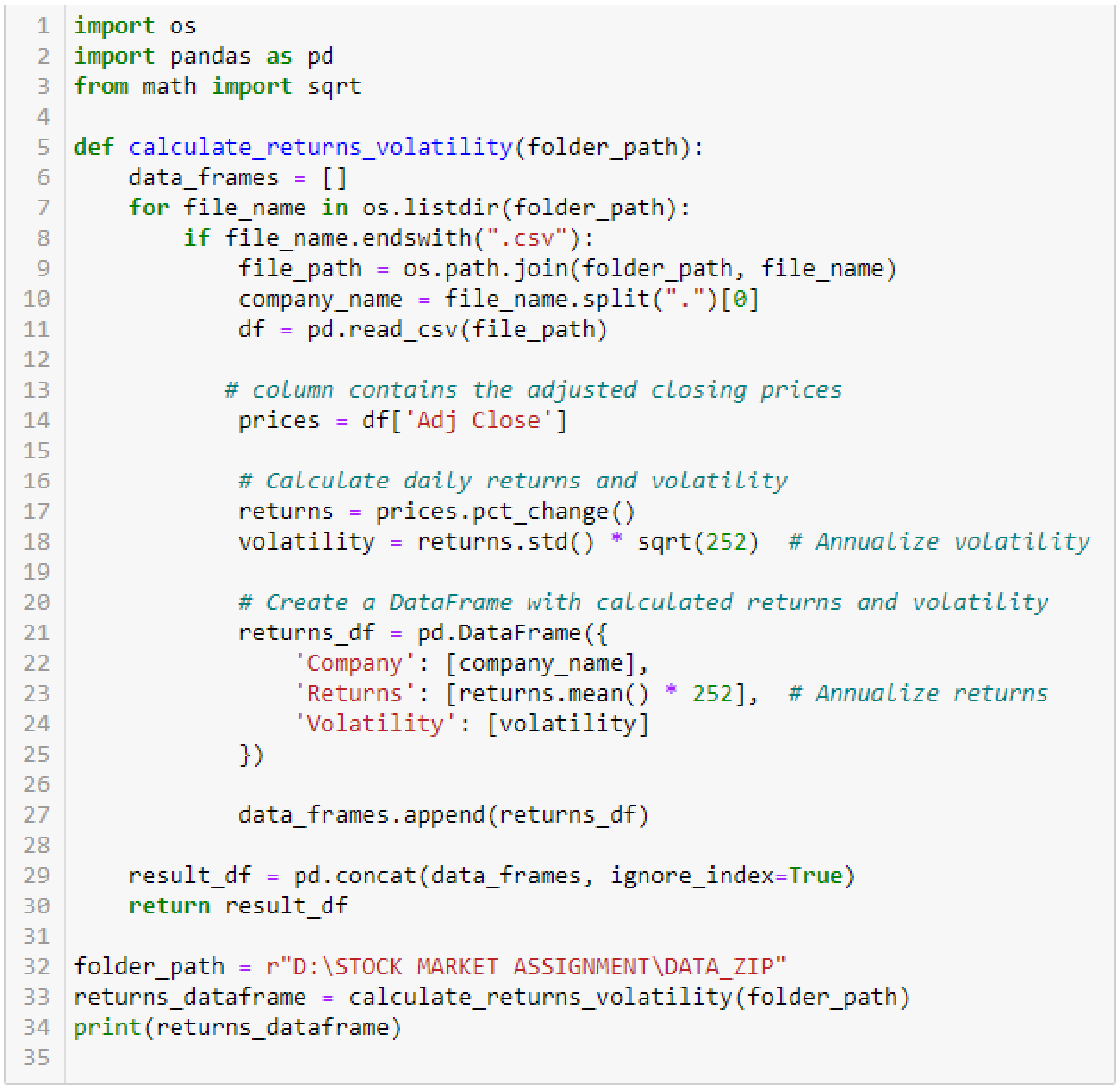
The procedure followed for the task is presented below

Importing of the required libraries



```python
import numpy as np
import pandas as pd
import pandas_datareader as dr
import yfinance as yf
from pylab import plot,show
from matplotlib import pyplot as plt
import plotly.express as px
from numpy.random import rand
from scipy.cluster.vq import kmeans,vq
from math import sqrt
from sklearn.cluster import KMeans
from sklearn import preprocessing
```

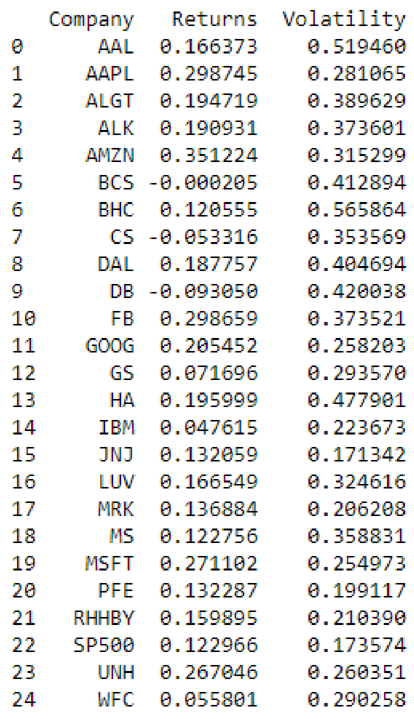
Calculating the volatility and the returns for each respective company for the adjusted closing price.



```
import os
import pandas as pd
from math import sqrt

def calculate_returns_volatility(folder_path):
    data_frames = []
    for file_name in os.listdir(folder_path):
        if file_name.endswith(".csv"):
            file_path = os.path.join(folder_path, file_name)
            company_name = file_name.split(".")[0]
            df = pd.read_csv(file_path)

           # column contains the adjusted closing prices
            prices = df['Adj Close']

            # Calculate daily returns and volatility
            returns = prices.pct_change()
            volatility = returns.std() * sqrt(252)  # Annualize volatility

            # Create a DataFrame with calculated returns and volatility
            returns_df = pd.DataFrame({
                'Company': [company_name],
                'Returns': [returns.mean() * 252],  # Annualize returns
                'Volatility': [volatility]
            })

            data_frames.append(returns_df)

    result_df = pd.concat(data_frames, ignore_index=True)
    return result_df

folder_path = r"D:\STOCK MARKET ASSIGNMENT\DATA_ZIP"
returns_dataframe = calculate_returns_volatility(folder_path)
print(returns_dataframe)
```

Calculate_returns_volatility function processes each of the data frame present in the folder reads through the csv files and extracts each of the companies data individually and calculate the annual volatility and annual returns.

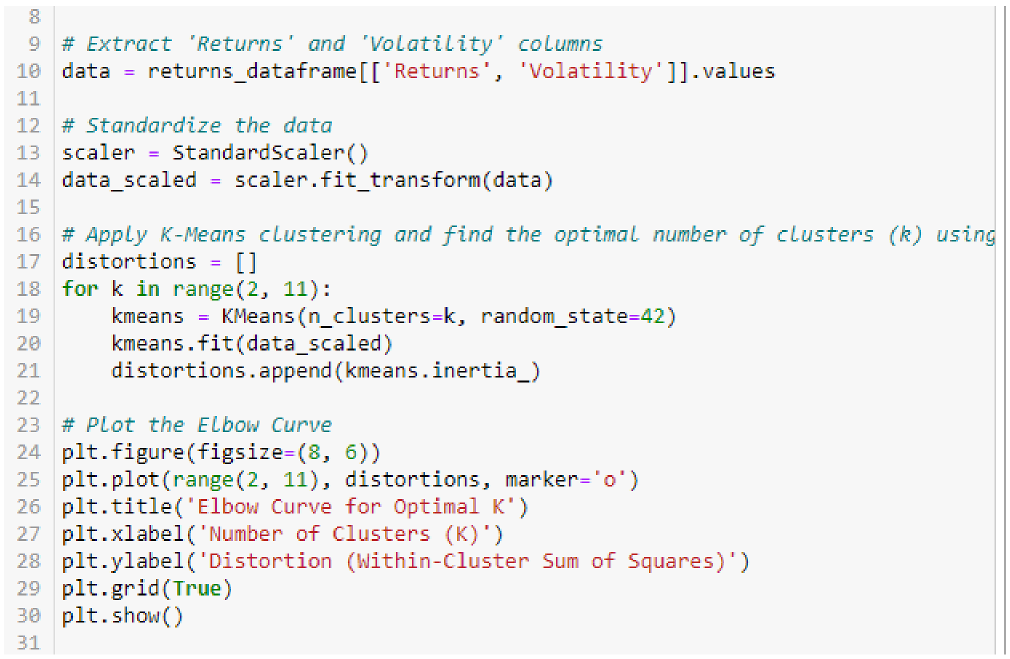
The annual returns and the amount of volatility is extracted as returns_dataframe



```
   Company   Returns  Volatility
0      AAL  0.166373    0.519460
1     AAPL  0.298745    0.281065
2     ALGT  0.194719    0.389629
3      ALK  0.190931    0.373601
4     AMZN  0.351224    0.315299
5      BCS -0.000205    0.412894
6      BHC  0.120555    0.565864
7       CS -0.053316    0.353569
8      DAL  0.187757    0.404694
9       DB -0.093050    0.420038
10      FB  0.298659    0.373521
11    GOOG  0.205452    0.258203
12      GS  0.071696    0.293570
13      HA  0.195999    0.477901
14     IBM  0.047615    0.223673
15     JNJ  0.132059    0.171342
16     LUV  0.166549    0.324616
17     MRK  0.136884    0.206208
18      MS  0.122756    0.358831
19    MSFT  0.271102    0.254973
20     PFE  0.132287    0.199117
21   RHHBY  0.159895    0.210390
22   SP500  0.122966    0.173574
23     UNH  0.267046    0.260351
24     WFC  0.055801    0.290258
```

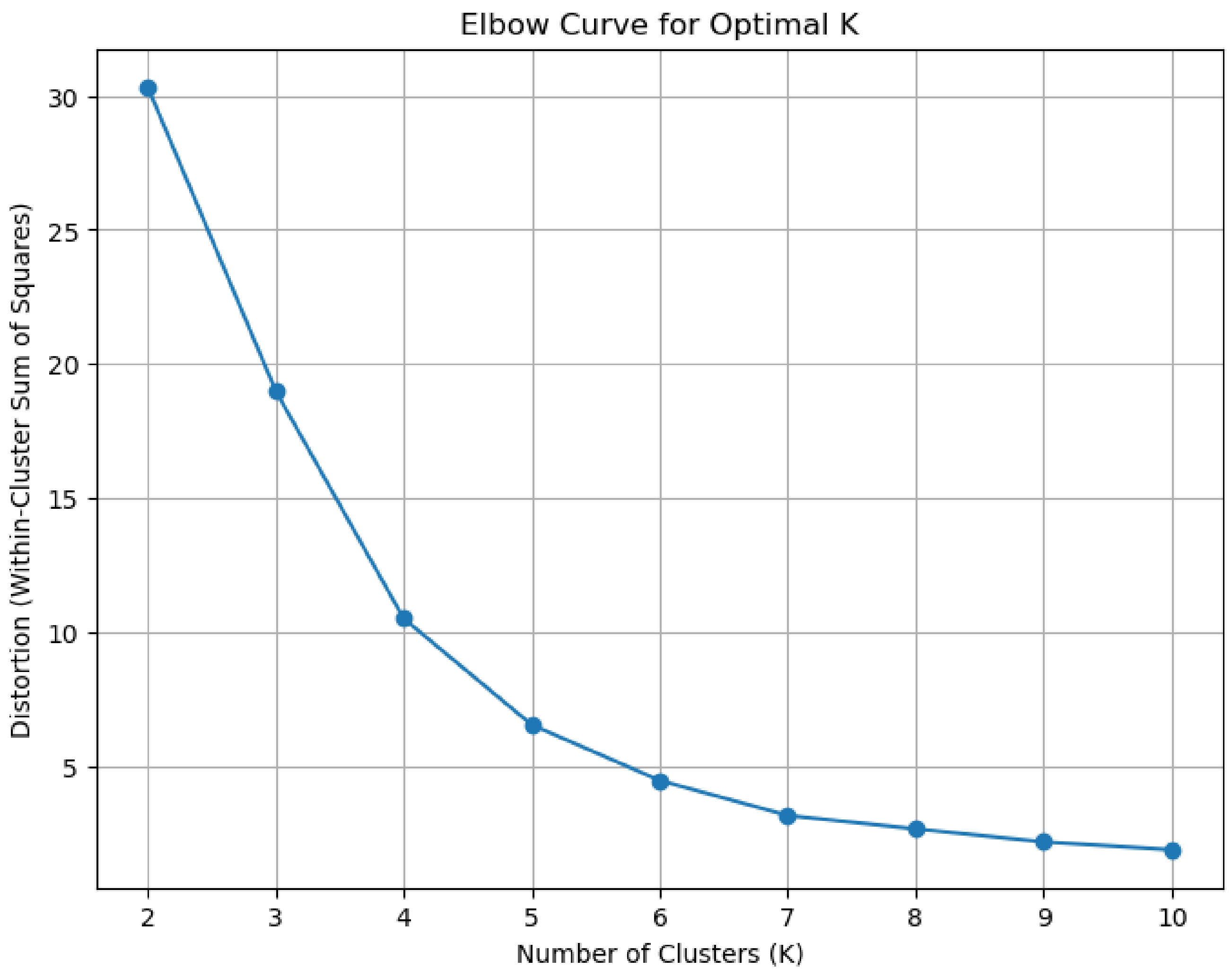
And the extracted data is used for unsupervised learning to create clusters among the companies which shows the variability in each individual company in essence we don't want all of our stocks picks to be placed in the same cluster that could potentially show same risk level we at least want one of the company to belong to different cluster.

We create a empty list distortions and apply k means over the range 2, 11 as the number of companies are around 26.



```python

# Extract 'Returns' and 'Volatility' columns
data = returns_dataframe[['Returns', 'Volatility']].values

# Standardize the data
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)

# Apply K-Means clustering and find the optimal number of clusters (k) using
distortions = []
for k in range(2, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(data_scaled)
    distortions.append(kmeans.inertia_)

# Plot the Elbow Curve
plt.figure(figsize=(8, 6))
plt.plot(range(2, 11), distortions, marker='o')
plt.title('Elbow Curve for Optimal K')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('Distortion (Within-Cluster Sum of Squares)')
plt.grid(True)
plt.show()

```

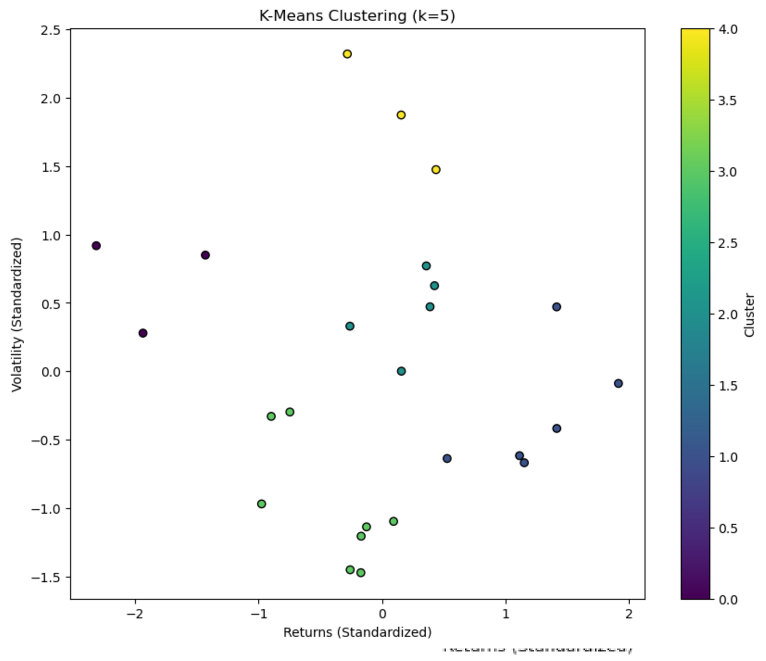
From the elbow method we can infer that the optimal number of clusters turned out to be 5 as evident from the plot below

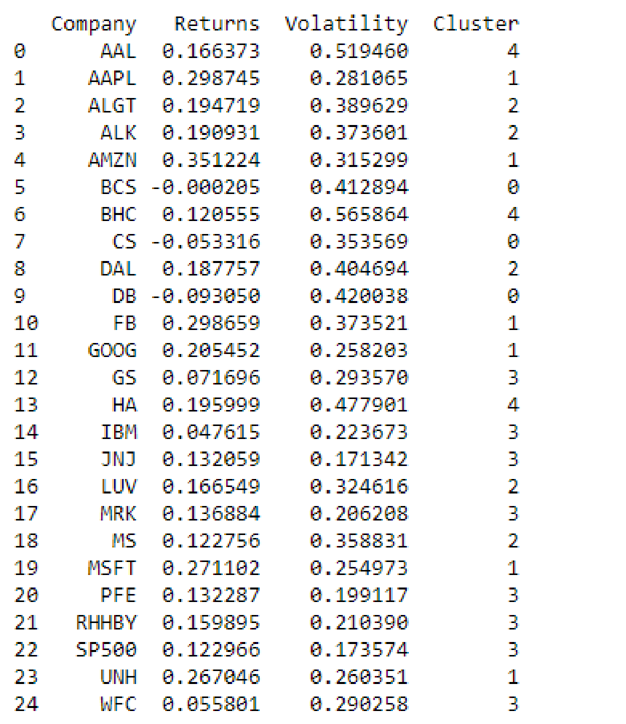


Now using the optimal number of cluster values we identify the clusters the companies belong to and verify that the picked companies

Apple and Morgan Stanley do not fall in the same cluster

The plot below shows the scatter plot of the companies plotted in a scatter plot color coded according to the cluster they belong to and the table below presents the companies with respect to each of there cluster





```
   Company   Returns  Volatility  Cluster
0      AAL  0.166373    0.519460        4
1     AAPL  0.298745    0.281065        1
2     ALGT  0.194719    0.389629        2
3      ALK  0.190931    0.373601        2
4     AMZN  0.351224    0.315299        1
5      BCS -0.000205    0.412894        0
6      BHC  0.120555    0.565864        4
7       CS -0.053316    0.353569        0
8      DAL  0.187757    0.404694        2
9       DB -0.093050    0.420038        0
10      FB  0.298659    0.373521        1
11    GOOG  0.205452    0.258203        1
12      GS  0.071696    0.293570        3
13      HA  0.195999    0.477901        4
14     IBM  0.047615    0.223673        3
15     JNJ  0.132059    0.171342        3
16     LUV  0.166549    0.324616        2
17     MRK  0.136884    0.206208        3
18      MS  0.122756    0.358831        2
19    MSFT  0.271102    0.254973        1
20     PFE  0.132287    0.199117        3
21   RHHBY  0.159895    0.210390        3
22   SP500  0.122966    0.173574        3
23     UNH  0.267046    0.260351        1
24     WFC  0.055801    0.290258        3
```

As evident from the cluster values extracted

Apple belongs to cluster 1

Morgan Stanley belongs to cluster 2

Allegiant Travel Company belongs to cluster 2.

Interpretation and conclusion

In conclusion, we have navigated through the intricate landscape of portfolio management, bridging traditional strategies along with a healthy balance of cutting-edge data analysis and machine-learning techniques within the evolving financial landscape.

The projects three fold objective of solving problems in portfolio management using Regression, Clustering and classification methods to pick ideal stock selection for a investor has successfully been achieved.

Ideal models for each of the companies from 3 different sectors for the classification of trend of the stocks are identified and tested the results show that the Ada boost model seems to work well for apple and simple logistic regression worked best for Morgan Stanley simple whereas support vector machine worked very well for Allegiant travel company.

Ideal models for each of the companies for 3 different sectors for the regression task of determining the adjusted closing prices seem to show that the Random forest model seems to works well for Apple, Morgan Stanley and but has a very long training time when compared to the training of other models and Allegiant travel company.

The ideal number of clusters based on the volatility and returns calculated annual shows that there are 5 different clusters that the companies belong and they are clustered using k means the clusters in 1 has relatively high variability and shows a steady increase rates while cluster 2 has a relatively stable rate of growth and variability the kind of portfolio that could be selected from the mix of these 3 companies is an ideal for investors who wish to moderate risk with slight risk taking.

Some of the ethical considerations of using machine learning techniques include heavy reliance on historical data leading to exacerbating of existing biases.

Many ensemble methods that are used and give good performance do tend to operate as black box models making it challenging to interpret results creating a lack of transparency.

There needs to be vigilance to ensure that machine learning applications comply with financial regulations and security laws present in the country.

Also, there are inherent risks associated with several different models making similar predictions resulting in a herding behavioral patterns in the market destabilizing the market.

Addressing these ethical considerations requires a collaborative effort involving technology experts, regulators and ethicists to structure guidelines, regular reviews and constant ongoing monitoring of the data and reports being generated should be a first priority.

References

Johnson, William H. A., and Diane H. Parente. *Project Strategy and Strategic Portfolio Management : A Primer*, Business Expert Press, 2013. *ProQuest Ebook Central*,

http://ebookcentral.proquest.com/lib/bcu/detail.action?docID=1097921.
Created from bcu on 2024-01-10 11:11:28.

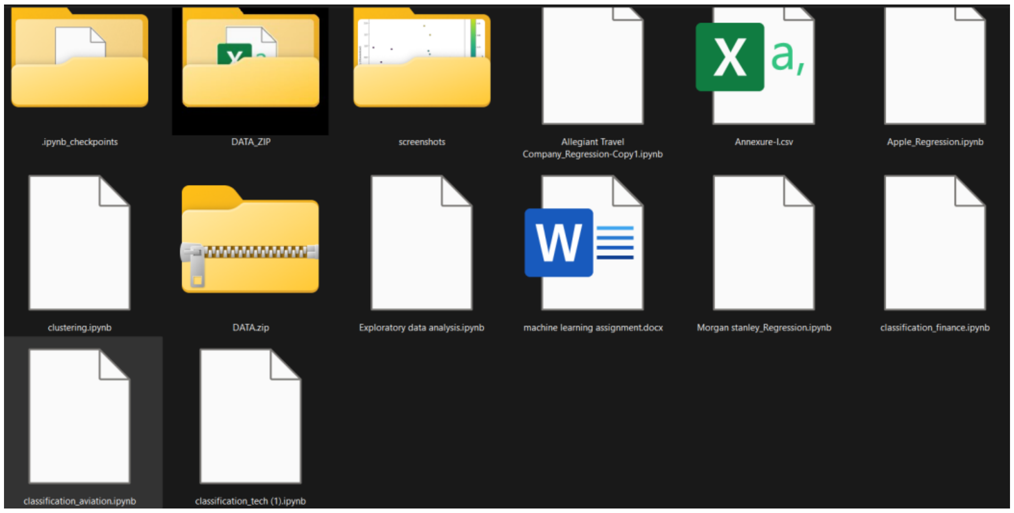
Johnson, R. Stafford. *Equity Markets and Portfolio Analysis*, John Wiley & Sons, Incorporated, 2014. *ProQuest Ebook Central*, http://ebookcentral.proquest.com/lib/bcu/detail.action?docID=827137.
Created from bcu on 2024-01-08 10:28:06.

Lopez, de Prado, Marcos. *Advances in Financial Machine Learning*, John Wiley & Sons, Incorporated, 2018. *ProQuest Ebook Central*, http://ebookcentral.proquest.com/lib/bcu/detail.action?docID=5240570.
Created from bcu on 2024-01-08 11:23:44.

S. Ravikumar and P. Saraf, "Prediction of Stock Prices using Machine Learning (Regression, Classification) Algorithms," *2020 International Conference for Emerging Technology (INCET)*, Belgaum, India, 2020, pp. 1-5, doi: 10.1109/INCET49848.2020.9154061.

Appendixes

https://drive.google.com/drive/folders/1AGYS9kLxliBPgkVb10LSmYWjYU9EbmMQ?usp=drive_link


.ipynb_checkpoints
DATA_ZIP
screenshots
Allegiant Travel Company_Regression-Copy1.ipynb
Annexure-I.csv
Apple_Regression.ipynb
clustering.ipynb
DATA.zip
Exploratory data analysis.ipynb
machine learning assignment.docx
Morgan stanley_Regression.ipynb
classification_finance.ipynb
classification_aviation.ipynb
classification_tech (1).ipynb

The google drive link consists of pdf file of the submitted assignment along with 3 files for regression for each companies and 3 files for classification and 1 file for clustering and a word document and PowerPoint presentation describing the project along with video link

The user should try and setup that path according to the system requirements to avoid system errors.