Analyzing Stock Price Predictions: A Comparative Study of Machine Learning Models

Problem Statement:

Apple Inc. is a renowned multinational technology company, and its stock price is influenced by various factors, including market trends, financial performance, global events, and macroeconomic indicators. Accurate prediction of Apple's stock prices is crucial for investors and financial analysts to make informed decisions.

Objective:

The goal of this project is to develop a predictive model that can forecast close price of the apple stock from the features selected in the data.

Description of the Dataset

The core dataset for this analysis, labeled 'AAPL.csv', encompasses a comprehensive historical record of Apple Inc.'s stock prices. This dataset documents daily stock activities, detailing key metrics such as the opening price ('Open'), the day's highest price ('High'), the lowest price ('Low'), the closing price ('Close'), the adjusted closing price ('Adj Close'), and the trading volume ('Volume'). The time frame covered in this dataset extends from December 12, 1980, through 2021. To enrich the dataset and enhance the analysis, additional features like 'DayOfWeek' and 'Month' can be derived from the 'Date' column, but after the exploratory data analysis we can decide to extract, or we can ignore feature engineering as it complicates the model by adding many features. For the purpose of model training and validation, the dataset is segmented into training and testing subsets in various formats for better accuracy and output.

Exploratory Data Analysis:

After loading the dataset, the first thing to do is to know what is the data and the type of the variables in it. By using info() and describe() functions we can get the statistics of the data and the below images show the information about the data and the statistics of the data for further analysis.

```
df
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	1980-12-12	0.128348	0.128906	0.128348	0.128348	0.100178	469033600
1	1980-12-15	0.122210	0.122210	0.121652	0.121652	0.094952	175884800
2	1980-12-16	0.113281	0.113281	0.112723	0.112723	0.087983	105728000
3	1980-12-17	0.115513	0.116071	0.115513	0.115513	0.090160	86441600
4	1980-12-18	0.118862	0.119420	0.118862	0.118862	0.092774	73449600
...	...	...	...	...	...	...	...
10463	2022-06-13	132.869995	135.199997	131.440002	131.880005	131.880005	122207100
10464	2022-06-14	133.130005	133.889999	131.479996	132.759995	132.759995	84784300
10465	2022-06-15	134.289993	137.339996	132.160004	135.429993	135.429993	91533000
10466	2022-06-16	132.080002	132.389999	129.039993	130.059998	130.059998	108123900
10467	2022-06-17	130.070007	133.080002	129.809998	131.559998	131.559998	134118500

10468 rows × 7 columns

```
# Check for missing values
print(df.isnull().sum())
```

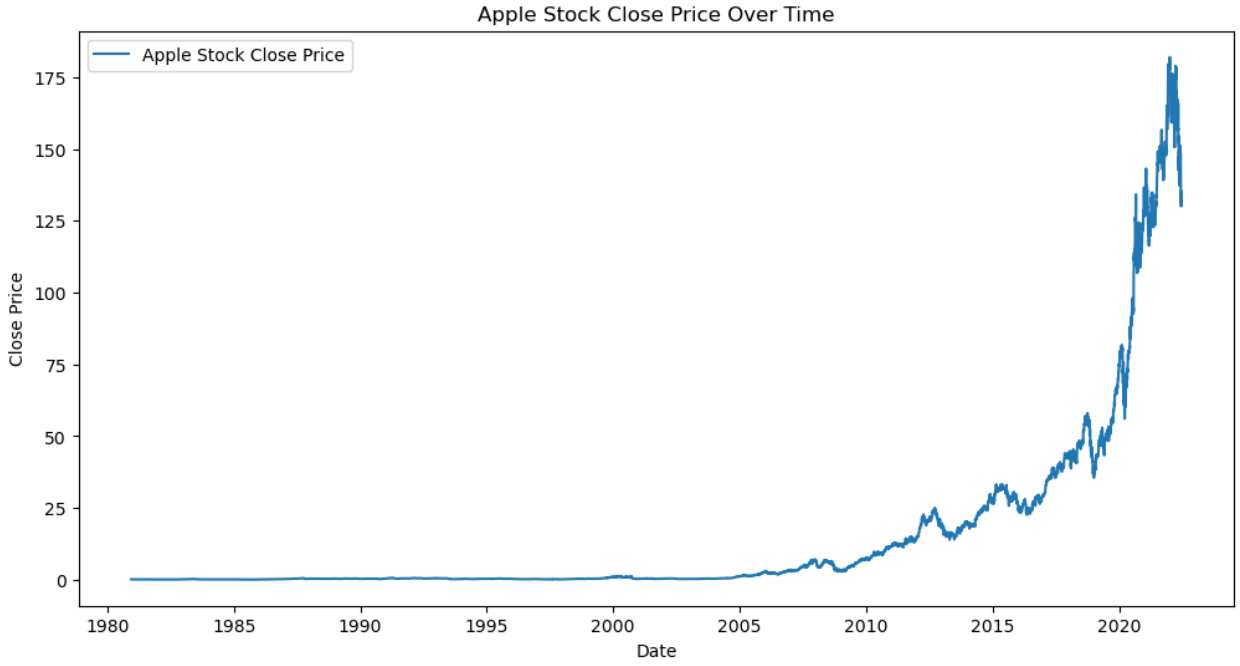
```
Date         0
Open         0
High         0
Low          0
Close        0
Adj Close    0
Volume       0
dtype: int64
```

```
df=pd.read_csv('AAPL.csv')
#Basic Data information
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10468 entries, 0 to 10467
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Date       10468 non-null  object
 1   Open       10468 non-null  float64
 2   High       10468 non-null  float64
 3   Low        10468 non-null  float64
 4   Close      10468 non-null  float64
 5   Adj Close  10468 non-null  float64
 6   Volume     10468 non-null  int64
dtypes: float64(5), int64(1), object(1)
memory usage: 572.6+ KB
```

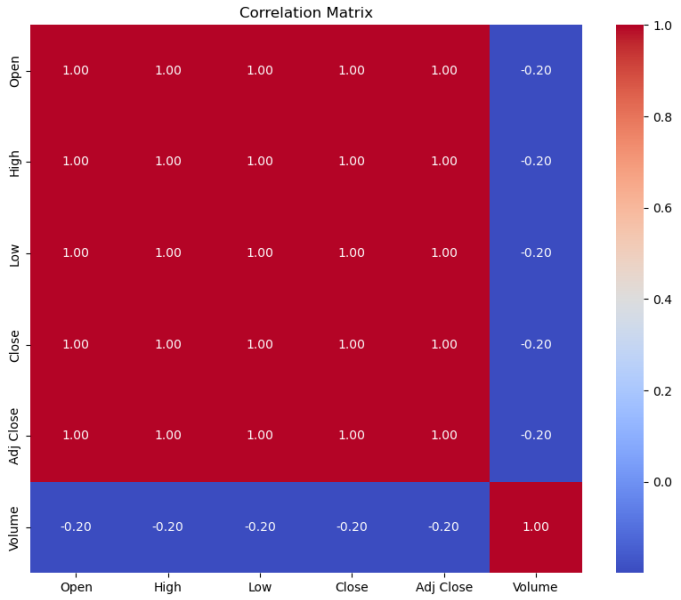
From the data we can say that there is no presence of null values, and the data types of the variables are float number which is obvious as they represent the price of the stock, a minor change in the cents also impact the growth of the stock

Performing a time-series analysis for the close variable in the dataset as we need to predict it and we need to know the variability change of the column over time. To do so we need to change the date column in the dataset should be set to data format as we need to do the analysis over time. Below graph shows the projection of the market over time.



As we can see that over the time the stock price has increased that is from 1980 to 2021 the price of the stock in the market has changed where the phenomenon is called as inflation.

Computing correlation for the variables with respect to the close price of the stock gives us the idea of how strong the variables are correlated and what to be considered and what to be eliminated.

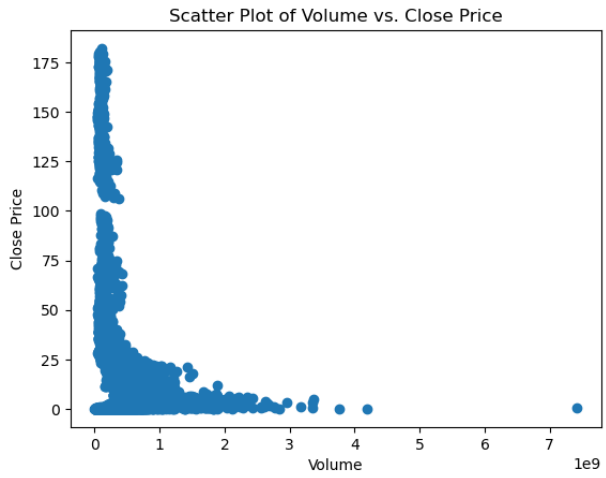


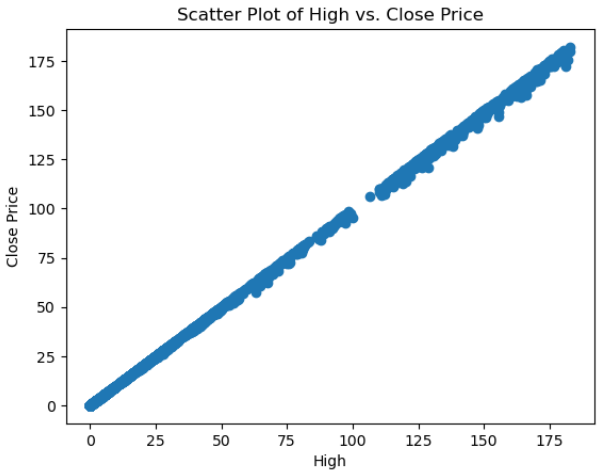

```
df.corr()['Close']
```

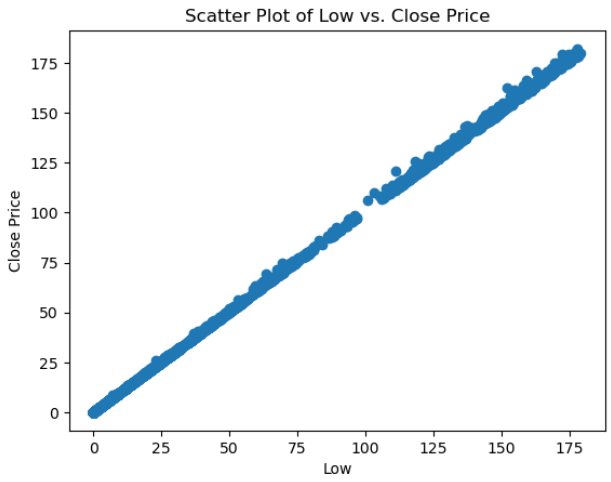
```
Open         0.999850
High         0.999924
Low          0.999928
Close        1.000000
Adj Close    0.999671
Volume      -0.196411
Name: Close, dtype: float64
```

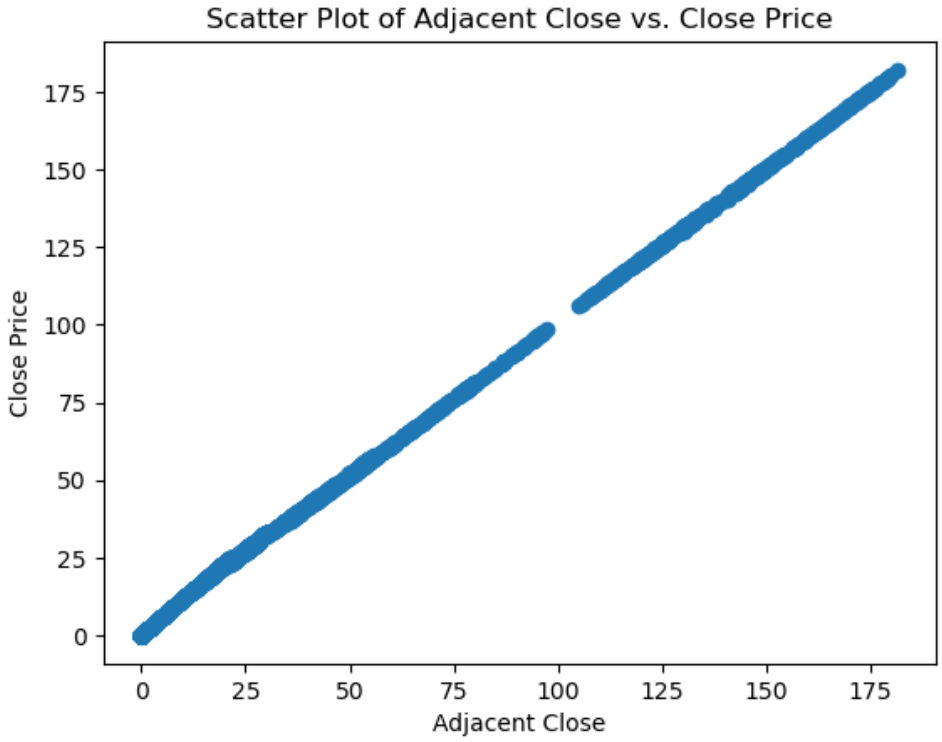
From the above values with respect to close variable the correlation coefficient is nearly equal to 1 stating that the variables are highly correlated and possess a linear relationship. From the above values Volume variable doesn't seem to have a linear relationship with the close variable, suggesting to drop the variable for the better performance of model and obtaining a high accuracy.

Scatter plots with respect to close variable can tell about the relationship between the variables visually which can help in determining the whether the data has any outliers or any other specific information.









The above scatter plots with respect to all variables show a linear relationship which is our initial conclusion with the correlation matrix. If we closely look into data there is no data with respect to prices near 100 which stands to be missing data. This plot can say that the price range is missing in between 100-120 where this could be a special case as there is sudden increase in the stock price with in a day which is rarely to happen. Getting a deeper knowledge for this the movement of stock was rapid in just 1 day which rarely happens in the market, which indicates a high inflation rate with respect to all variables.

From this above graphs and correlation and looking at the linearity of the data we would like to proceed with linear regression model to predict the close price of the variable

Machine Learning Model – Linear regression model :

Linear regression is a fundamental and widely used statistical technique for modeling the relationship between a dependent variable and one or more independent variables. It assumes that there is a linear relationship between the input variables and the output variable.

Main Objectives

The primary objective is to forecast the closing stock prices of Apple Inc. by employing a machine learning model. The study is driven by several key research questions, aimed at exploring and understanding the dynamics of stock price prediction:

- Effectiveness of Linear Regression: The first question delves into the capability of linear regression models in predicting stock prices. It investigates whether a linear regression approach, based on historical data, can accurately forecast the closing stock prices of Apple Inc.

Can linear regression, with its simplicity and interpretability, effectively predict the closing stock price of Apple Inc. using historical data?

Linear Regression with Look-Back Feature

'look_back**'** is a hyperparameter that determines how many past time steps the model should use to predict the next time step. In this case, it is set to 20, meaning the model considers the previous 20 time steps to predict the next time step. In time series forecasting, each data point is associated with a specific time, and the order of the data points matters. The look_back parameter defines how far back in time the model "looks" to gather information for making a prediction.

The process of our machine learning model, the first and foremost as the data is not normalized and we need to normalize the data in between to 0 and 1 using minmax scalar. MinMaxScaler is a data preprocessing technique used to scale and normalize the features of a dataset. It transforms the features by scaling them to a specified range, usually between 0 and 1. The formula for Min-Max scaling is as follows:

$$X\_\text{scaled} = ((X - \min(X)) / (\max(X) - \min(X))$$

MinMaxScaler is commonly used to ensure that all features contribute equally to the model, especially when features have different scales. It helps prevent features with larger magnitudes from dominating the learning process.

After normalizing the data we create dataset required for linear regression with look_back period included in it which it specifies to consider previous observations to predict the present. We have set the look_back period as 20 as a generalized format as it would

contain information of previous higher highs and lower lows which would help to predict the present close.

Considering K-Fold cross validation with 5 folds we run a for loop to train the model and parallelly calculate the MSE and R2 scores which help to determine the accuracy of the model.

The following are the parameters used to find the accuracy of the model if it seems to be good or bad.

Mean MSE (Mean Squared Error):

Value: 0.5366

Explanation: The Mean Squared Error is a measure of the average squared difference between the predicted and actual values. A lower MSE indicates better model performance. In our case, the average squared difference across the cross-validation folds is approximately 0.5366.

Standard Deviation MSE:

Value: 0.0755

Explanation: The Standard Deviation of the Mean Squared Error gives us an idea of the variability or spread of the MSE values across the different folds. A lower standard deviation suggests more consistency in model performance across folds.

Mean R-squared:

Value: 0.9995

Explanation: R-squared (coefficient of determination) is a measure of how well the model explains the variance in the target variable. It ranges from 0 to 1, where 1 indicates a perfect fit. Our mean R-squared of 0.9995 suggests that our model explains approximately 99.95% of the variance in the target variable on average across the cross-validation folds.

Standard Deviation R-squared:

Value: 6.34e-05 (close to zero)

Explanation: Like the standard deviation of MSE, the Standard Deviation of R-squared gives us an idea of the variability or spread of R-squared values across different folds. A small standard deviation indicates consistent performance.

Mean Squared Error (MSE) on Test Set:

Value: 2.1335

Explanation: This is the Mean Squared Error calculated on a separate test set that was not used during cross-validation. It provides an estimate of how well our model generalizes to new, unseen data. A lower MSE on the test set suggests good generalization performance.

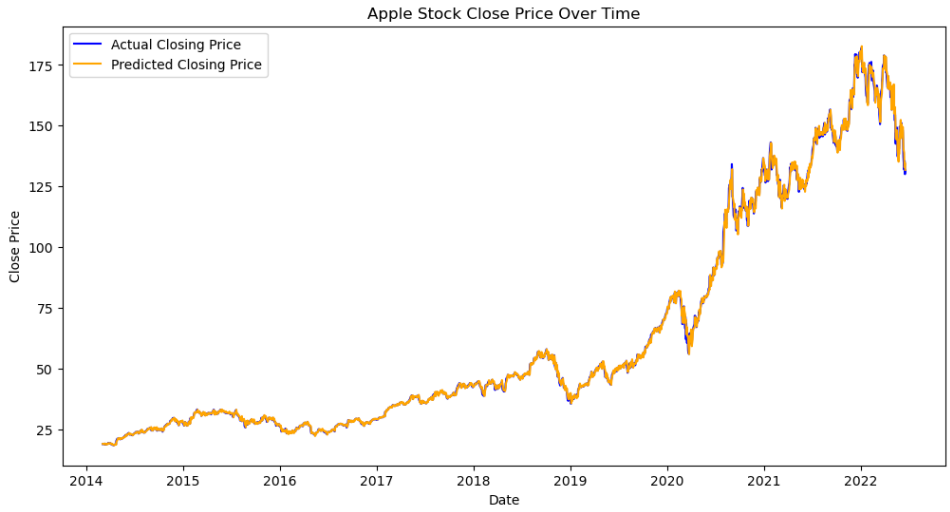
R-squared on Test Set:

Value: 0.99897

Explanation: Like the mean R-squared, this value (0.99897) indicates the proportion of variance in the target variable explained by our model on the test set. A value close to 1 suggests a good fit.

In summary, the metrics indicate that our linear regression model performs very well both in cross-validation and on the separate test set. The low MSE values, high R-squared values, and small standard deviations suggest that the model is accurately capturing the patterns in the data and generalizing well to new observations.

Following is the graph for actual values and predictions made by the model:



In the graph, the actual closing prices and the predicted closing prices are plotted over time, with the actual prices in blue and the predicted prices in orange. The graph shows a good alignment between the actual and predicted prices, suggesting that the model is effectively capturing the trend of the stock prices. There is a close tracking between the two lines throughout the displayed time period, which indicates that the model, with its look-back feature, is able to understand and follow the underlying pattern in the stock price movements.

Conclusion

The analysis set out to determine the most effective machine learning model for predicting the closing stock prices of Apple Inc. Through the application of several models, each with unique configurations and evaluation metrics, the study aimed to understand the capabilities of these models in forecasting within the volatile domain of financial time-series. The study reveals the complexities and challenges associated with predicting stock prices using machine learning models. It emphasizes the importance of choosing the right model, tuning parameters, and preprocessing data to capture the underlying patterns in stock price movements effectively. The Linear Regression with Look-Back Feature stands out for its potential, but further validation and testing would be necessary to confirm its superiority conclusively. This research contributes valuable insights into financial time-series prediction and underscores the need for a careful, nuanced approach to model selection and evaluation.