Cloud Computing Final Project

**Title :** **Valuation and Forecasting Stock Prices of Publicly Traded Companies Using Spark Framework**

Members -

Group 2

[Joel Joseph Kinny](mailto:jk2112@scarletmail.rutgers.edu)

[Jay Panchal](mailto:jdp272@scarletmail.rutgers.edu)

Suhas Kollur

Prof. Maria Striki

| **Item** | **Status** | **Related files** | **Notes** |
| --- | --- | --- | --- |
| Environment and Tools Setup | Completed |  |  |
| Research | Completed |  |  |
| Data Cleaning and Analysis | Completed |  |  |
| Training and Testing | Completed | finalStockPrediction.ipynb  DDM.ipynb |  |
| Simulations | Completed |  |  |
| Analysis | Completed |  |  |
| Conclusion | Completed |  |  |

[**Environment Tools and Setup Guide 3**](#_oxvn9lalri2d)

[Valuation and Forecasting of Stock Prices 5](#_qx7iudt4p0dk)

[Methodology 6](#_6jwdmih3dk3m)

[Data Collection 6](#_4ftoebtr42fi)

[Data Modelling 6](#_ebwo92s6pfs)

[Project Architecture 7](#_yt1mr2bfg28v)

[Experimental Setup 8](#_5511b42nihnu)

[Experimental Observations 8](#_fn7wvexmcwhh)

[Transformation Steps 10](#_vcffj2ulueg7)

[Results 13](#_gf754rcovg4v)

[Conclusion 14](#_6wyffih0tym7)

[**References 15**](#_nkr11b5r06o)

## Environment Tools and Setup Guide

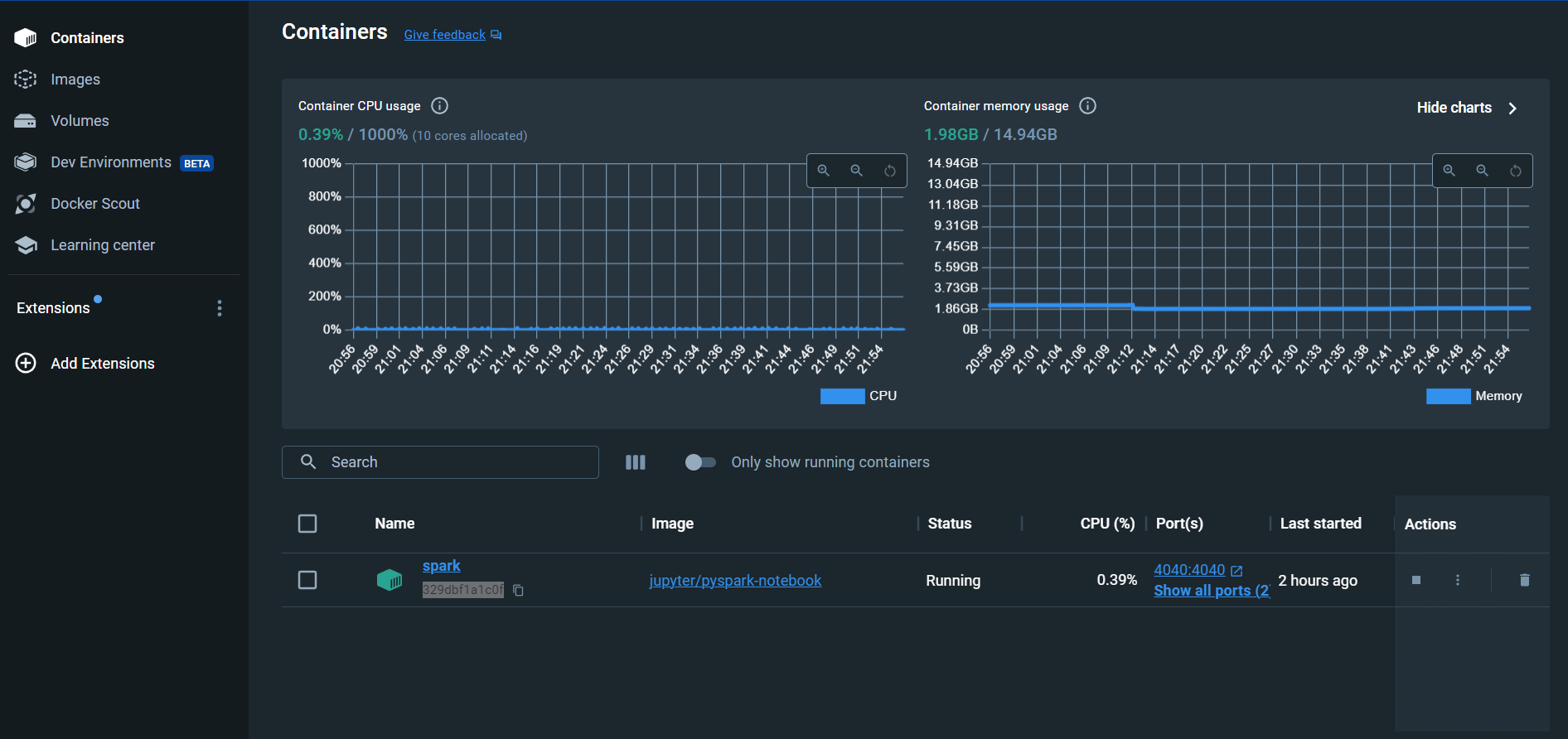
1. Install Docker Desktop
2. Fetch the Spark distribution for Local or Cluster mode
3. Add additional dependencies for Jupyter notebook for interactive login and run
4. Optionally, the Docker container can be used to spin up a Standalone cluster based on availability of machines to deploy Spark Worker nodes or can be run on a single machine with multiple daemons.
5. The below link is used for the image and instructions for setting up the environment

References

- <https://github.com/jupyter/docker-stacks/tree/main/images/pyspark-notebook>

- <https://jupyter-docker-stacks.readthedocs.io/en/latest/using/specifics.html#apache-spark>

Docker Desktop -



Spark-Jupyter Container (Running on Local Mode)



## Valuation and Forecasting of Stock Prices

In the dynamic world of financial markets, the ability to accurately forecast stock prices and assess intrinsic values is paramount. This project aims to develop a robust framework for the valuation and forecasting of publicly traded companies. Leveraging technical and quantitative analysis techniques, we will employ the Apache Spark framework for efficient data processing and analysis. The model will provide forecasts for stock prices and intrinsic values, aiding investors in making informed decisions.

This project aims to develop or rather provide a platform to implement the Spark framework for the use of parallel and efficient computation of the given problem of valuation and stock price forecasting.

## Methodology

### Data Collection

Acquired historical and real-time financial data from reputable sources to form the foundation of the analysis.

Executed a meticulous data preprocessing phase to rectify data quality issues and render it suitable for subsequent analysis.

Transformed the data into a format conducive to comprehensive model training and analysis.

### Data Modelling

Employed technical analysis indicators and quantitative analysis techniques to establish a robust foundation for stock price forecasting and intrinsic value assessment.

Implemented sophisticated feature engineering mechanisms to capture key market dynamics and economic indicators.

Leveraged machine learning approaches to construct a highly predictive forecasting model.

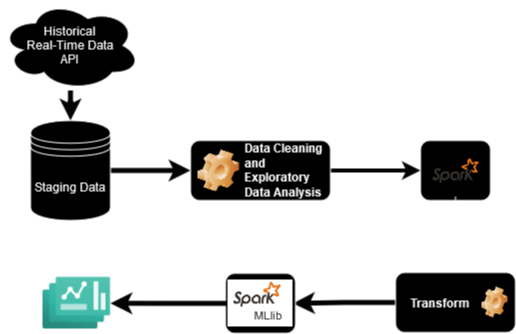
**. Data Collection Layer**

* Role:
  + Involves sourcing historical and real-time financial data.
* Components:
  + Various data sources integrated for comprehensive financial information.

**2. Data Processing and Analysis Layer**

* Role:
  + Utilizes Apache Spark for efficient data processing.
  + Houses the valuation and forecasting model.
* Components:
  + Apache Spark for distributed computing.
  + Valuation and forecasting model.

## Project Architecture



These are the major components of the project that govern the end to end process of the stock prediction model -

1. The Data Sources - The Financial data needs to be sourced from the Internet for each company that needs to be analyzed. The primary data sources that were used for this project include
   1. AlphaVantage
      1. The AlphaVantage API is used to fetch the Time Series data which contains the Daily data of the stock open, close, high and low values.It is also used to get the Company Overview data, Company Income Statement, Company
   2. Yahoo Finance
      1. This API has historical data as well as current data and gets

## Experimental Setup

* Environment
  + Docker Container running PySpark on Standalone mode
  + Docker Container running PySpark on Single Cluster mode
* Setup
  + Setup of Docker Engine
  + Gathering of relevant dependencies
  + Container execution for PySpark Notebook and Linux image
  + Jupyter or Spark Submit
* Spark
  + Local (as many workers as threads on device)
  + Pseudo cluster with one Executor and Two Workers
  + Executor Memory -> 6GB, 8GB, 12GB, 16GB

## 

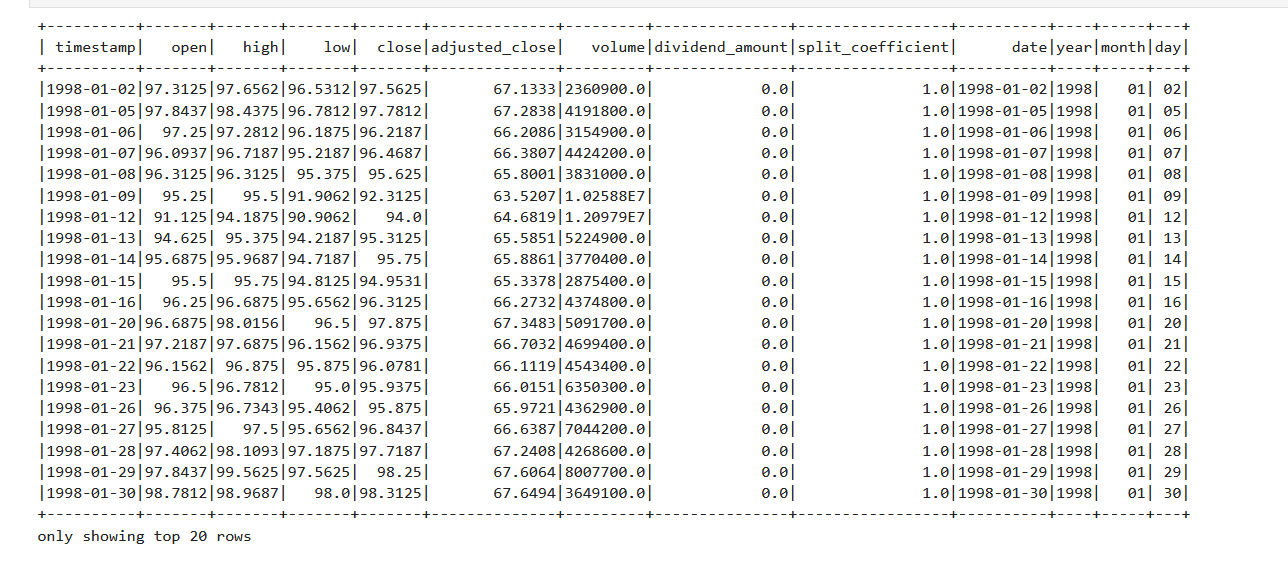
## 

## 

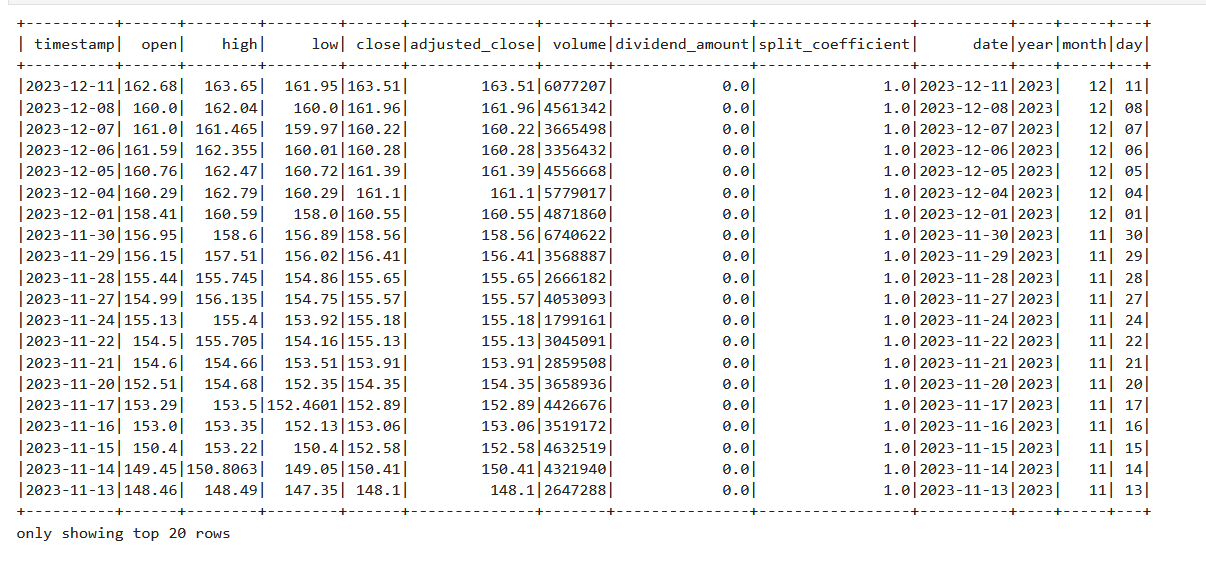
## Experimental Observations

The following analysis and training has been done on the Jupyter notebooks running on the Docker container in Local mode.

**Historical Stock Data of Apple Inc.**



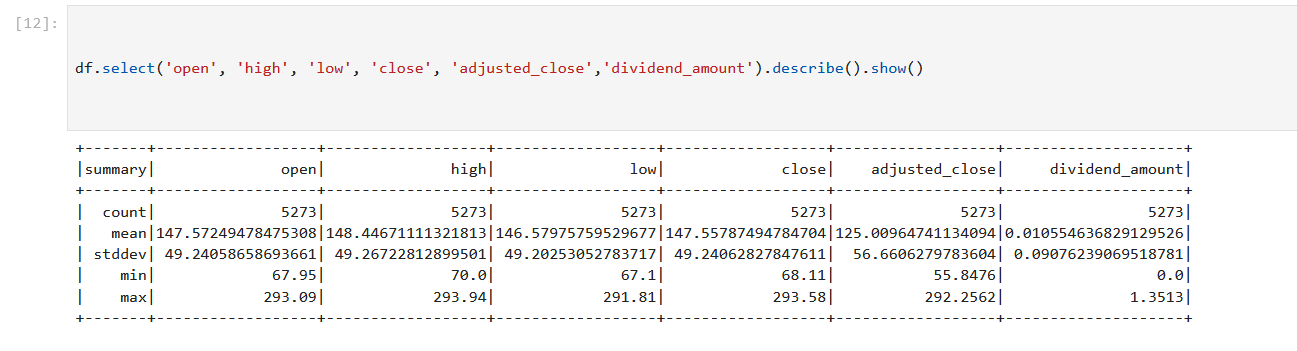
**Latest Stock Data of IBM**



**Year-wise distribution of Data**



Descriptive statistics of the Stock data



### Transformation Steps

The following steps were undertaken on the RDD and DataFrames that were used for this project

Windowing -

This creates a Window specification with no partitioning and orders the data by the "timestamp" column. The ordering is crucial for window functions that involve looking at previous or following rows

Lagged Close Price -

This adds a new column "lag\_close\_1" to the DataFrame, representing the close price of the previous day. The F.lag function is a window function that retrieves the value of the "close" column from the previous row within the specified window.

Daily Return -

Computes the daily return as the percentage change from the opening to closing prices. It creates a new column "daily\_return" with this calculated value.

Intra-day Volatility -

Calculates the intra-day volatility as the difference between the highest and lowest prices of the day. It adds a new column "intra\_day\_volatility" to the DataFrame.

Daily Volatility -

Computes the daily volatility as the change in closing price from the previous day. It adds a new column "daily\_volatility" to the DataFrame.

7-Day Moving Average -

Calculates a 7-day moving average for the closing prices. The F.avg function is a window function that computes the average over a specified window. In this case, it considers the current row and the previous 6 rows.

Binary Indicators -

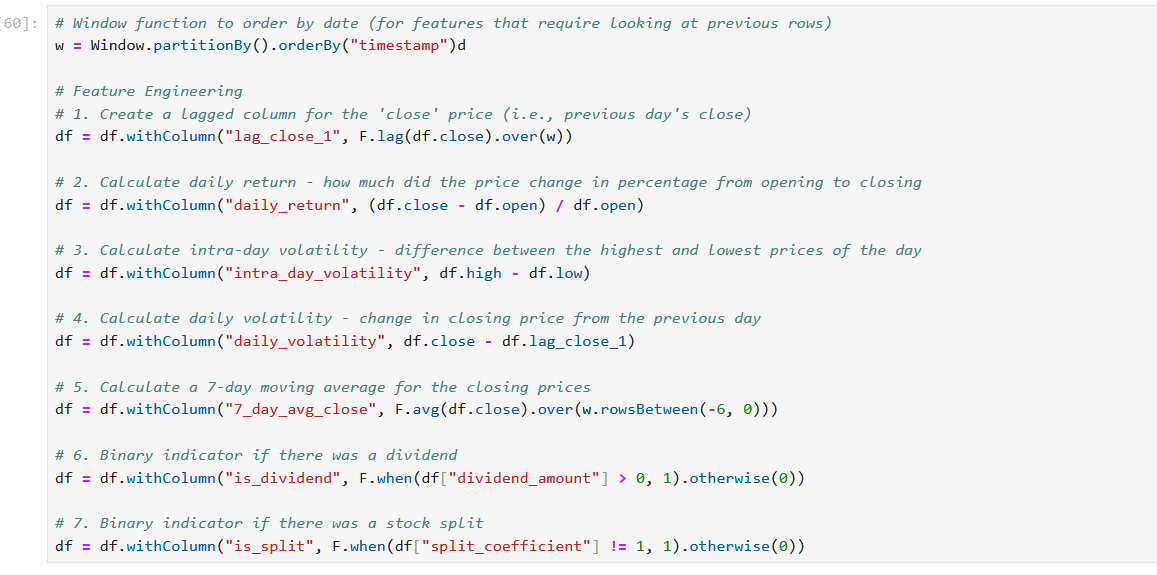
Creates a binary indicator column "is\_dividend" that equals 1 if there was a dividend (dividend\_amount > 0), and 0 otherwise

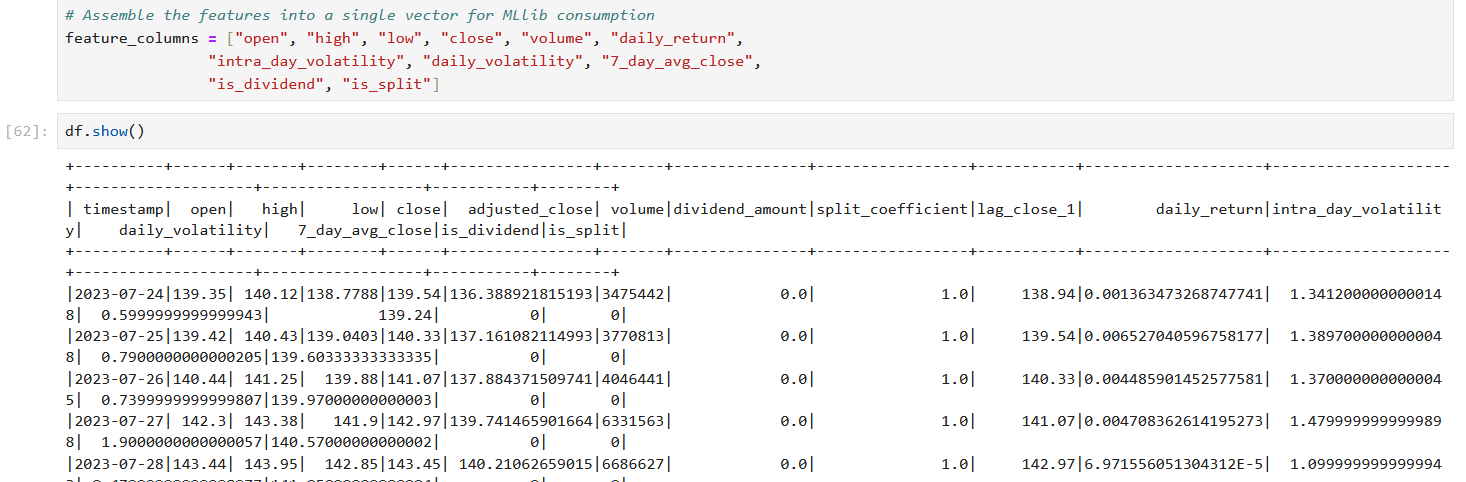
Creates a binary indicator column "is\_split" that equals 1 if there was a stock split (split\_coefficient != 1), and 0 otherwise.

Feature Engineering -

The following features have been selected from the dataset -

Open, High, Low, Close, Volume, Adjusted Close, Daily Return, Daily Volatility, Intra Day Volatility, 7 day average close and Is Dividend and Is Split indicators for the training set.





Intrinsic Value Computation

The value of the actual market price differs from the intrinsic value of the company and is useful in determining whether the company stock price is undervalued or overvalued. It is a useful indicator of the company fundamentals and the current stock valuation.

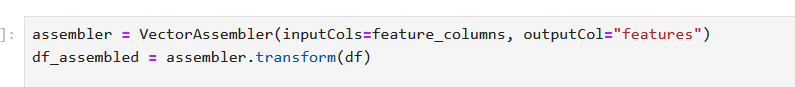
There are multiple models that are used for intrinsic value calculation. THe popular used models are Residual Income Model, Dividend Distribution Model and Dividend Cash Flow Model. THe Multi Stage Dividend Model has been implemented for the calculation of DDM. The DDM has been calculated using RDDs for efficient processing.

## 

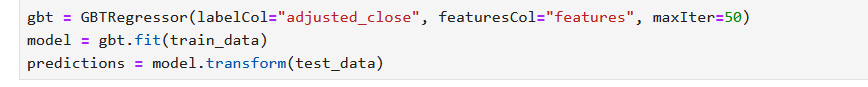
## Results

The feature

The VectorAssembler is a feature transformer that combines a given list of columns into a single vector column. This is often used as a preparation step before feeding the data into machine learning algorithms that expect input features as a single vector column.



Gradient Boosted Trees (GBT) is a powerful ensemble machine learning algorithm that can be used for stock market prediction, including stock price regression. The GBTRegressor in PySpark's MLlib library is an implementation of the GBT algorithm for regression tasks.

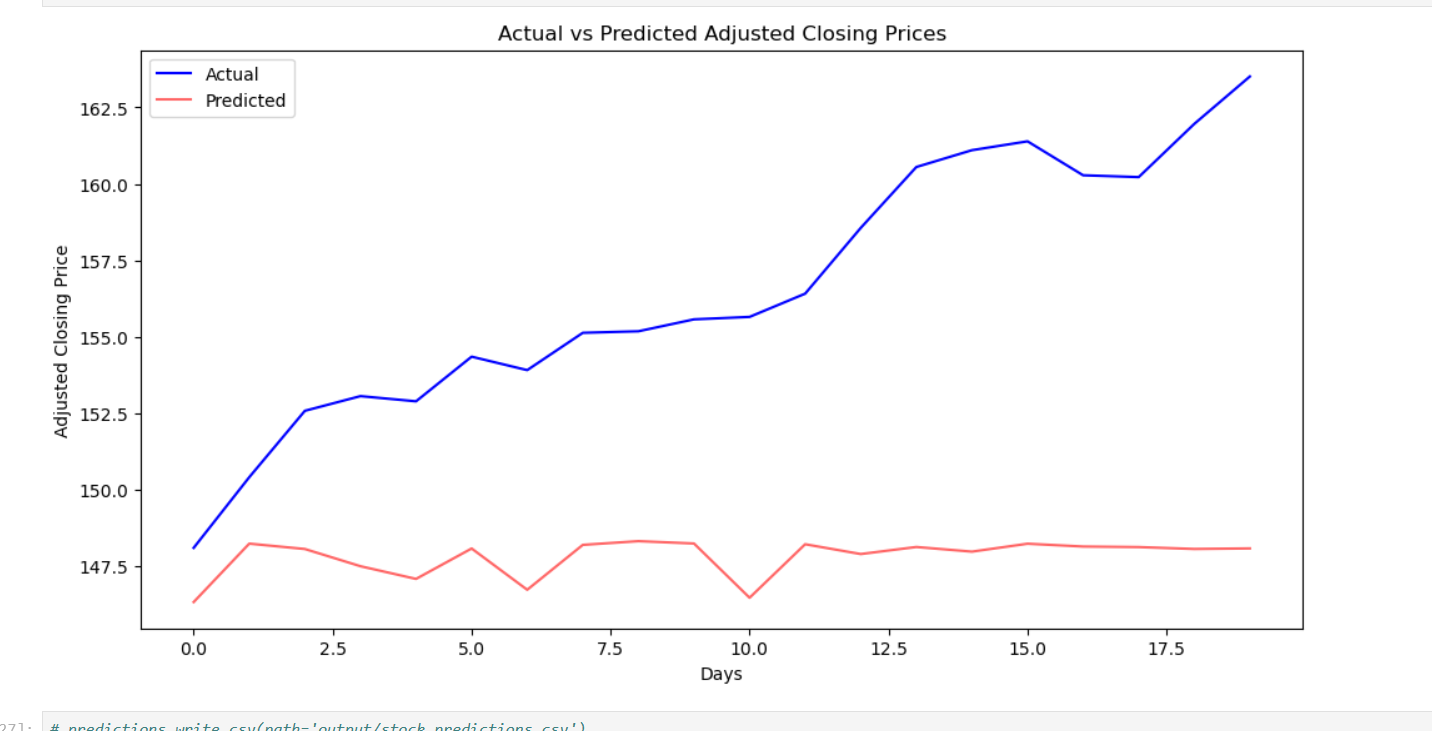


Metrics for Accuracy :

Root Mean Squared Error (RMSE) on test data = 9.544595132509585

MAE: 8.734564548282654

R2: -4.40344806670358



## Conclusion

The model for the Stock Market prediction is capable of predicting stock prices with an MSE of only around 9

This value can be further lowered when the training data has been expanded into multiple companies training over multiple epochs

There can be finer adjustments made to the Model Training which has been deferred to future scope

The Spark Distributed framework was used to handle the data processing and coordination of parallel tasks for faster and more efficient computations

## References

1. Menon, V.K., Chekravarthi Vasireddy, N., Jami, S.A., Pedamallu, V.T.N., Sureshkumar, V., Soman, K.P. (2016). Bulk Price Forecasting Using Spark over NSE Data Set. In: Tan, Y., Shi, Y. (eds) Data Mining and Big Data. DMBD 2016. Lecture Notes in Computer Science(), vol 9714. Springer, Cham. <https://doi.org/10.1007/978-3-319-40973-3_13>
2. Seif, M.M., Ramzy Hamed, E.M., Abdel Ghfar Hegazy, A.E.F. (2018). Stock Market Real Time Recommender Model Using Apache Spark Framework. In: Hassanien, A., Tolba, M., Elhoseny, M., Mostafa, M. (eds) The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018). AMLTA 2018. Advances in Intelligent Systems and Computing, vol 723. Springer, Cham. https://doi.org/10.1007/978-3-319-74690-6\_66
3. Ahmed. S. Wafi, Hassan Hassan, Adel Mabrouk, Fundamental Analysis Models in Financial Markets – Review Study, Procedia Economics and Finance, Volume 30, 2015,Pages 939-947,ISSN 2212-5671, https://doi.org/10.1016/S2212-5671(15)01344-1.
4. Javed Awan, Mazhar and Mohd Rahim, Mohd Shafry and Nobanee, Haitham and Nobanee, Haitham and Munawar, Ashna and Yasin, Awais and Zain, Azlan Mohd, Social Media and Stock Market Prediction: A Big Data Approach (2021). M. J. Awan, M. Shafry, H. Nobanee, A. Munawar, A. Yasin et al., "Social media and stock market prediction: a big data approach," Computers, Materials & Continua, vol. 67, no.2, pp. 2569–2583, 2021, Available at SSRN: https://ssrn.com/abstract=3827106