University of Hertfordshire UH

School of Physics,
Engineering and
Computer Science

MSc Data science Project

7PAM2002-0509-2023

Department of Physics, Astronomy and Mathematics

FINAL PROJECT REPORT Data Science

Title of Project:

Tesla Stock Price Prediction

Student Name and SRN:

Ubaid Ur Rehman / 22033731

Supervisor:	22033731
Date Submitted:	Aug 29th, 2024
Word Count:	4824

DECLARATION STATEMENT

This report is submitted in partial fulfilment of the University of Hertfordshire's Master of Science in Data Science degree requirements.

I have read the student guidance on academic integrity, misconduct, and plagiarism information at Assessment Offences and Academic Misconduct and understand the University's process for dealing with suspected cases of academic misconduct, as well as the potential penalties, which include failing the project module or course.

I certify that the work submitted is my own, and that any material derived or quoted from the published or unpublished work of others has been properly acknowledged. (See UPR AS/C/6.1, Section 7 and UPR AS/C/5, Section 3.6). I did not write the report or code using ChatGPT or any other generative AI tool (except as declared or referenced).

I did not use human participants or conduct a survey for my MSc Project.

I hereby grant permission for the report to be made available on module websites, provided that the source is acknowledged.

Student Name printed: Ubaid Ur Rehman

Student Name signature:

Student SRN number: 22033731

UNIVERSITY OF HERTFORDSHIRE

SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

Contents

Tesla Stock Price Prediction

1. Abstract:

To demonstrate it, this project seeks to analyze historical stock prices of Tesla Inc. from June 29th 2010 to February 3rd 2023. This dataset includes key financial values such as Open, High, Low and Close prices, and the Volume, which are all critical for carrying out a viable finance analysis. From these and other such ratios this particular research aims to find out the patterns, trends, and fluctuations in the TSLA stock over the years. The emphasis is put on evaluating not only the data on historical performance but also such parameters as the reasons for price volatility, increased trading turnover and changes in the sentiment in bulls and bears.

The insights derived from this analysis will benefit investors, financial analysts and other experts in the field of securities and stock exchange looking forward to forecasting the stock prices of Tesla. Tesla's stock can also be said to exhibit trend persistence, and comprehending its past behavior as well as influences on its price will enable stakeholders to develop better future predictions and necessary conformities. Besides enhancing the available body of knowledge within the financial analysis of Tesla, this project is useful in providing solutions to real-life investment problems, especially to those involved in the financial markets.

2. Introduction

Tesla Inc. has in recent decades become one of the most significant actors in financial markets worldwide mainly because of its innovations in the EV market. Tesla was founded in 2003, and within a short time became a disruptor in the automotive industry and the energy industry with its unique products and ideas. Tesla cars are rapidly gaining popularity, thanks to their leader Elon Musk and transformed from a small company to a market maker for change towards sustainable energy solutions. This spectacular expansion has attracted investors all over the globe, and has seen Tesla exchange handily on the stock markets in the last one and a half decades.

Indeed, Tesla's stock can be regarded as highly volatile which is beneficial sometimes but can be rather a problem in terms of stock trading. These fluctuations are usually as a result of several factors such as technological innovations by the firm, new production records, shift in regulatory policies, competition factors and the overall market trends. It's worth saying that stock prices show not only how efficiently the company operates but also involve psychic factors such as events happening in the world, speculations, and other macro factors.

Since Tesla is one of the most sought-after stocks in the market, this project proposes to concentrate on the examination of Tesla's historical stock price information from June 29, 2010 to February 3, 2023. The input features consist of fundamental elements of each financial record, namely the Open price, the High price, the Low price, the Close price and the Volume of the corresponding trades. These indicators give a more or less picture of the performance of the stock during each trading day and reflect the dynamics of the market and stock market investors.

More specifically, the goal of this project is to analyze the historical trends in Tesla’s shares and identify trends about them. By systematically analyzing these key financial metrics, we seek to answer the central research question: In what way could past data be employed in order to forecast future variations and volatilities of Tesla’s stock? This question is more interesting if looked at in the context of Tesla, in which the technological trend and market changes bring forth uncertainty in the stock price.

To this end, the analysis uses statistical tools and analyzes the information using models that capture financial magnitude. Analyzing the pattern and relationship of the data will be done through statistical techniques which will help make sense of Tesla’s stock from when it started trading to date. Financial modeling, on the other hand, will enable the preparation of future stock price changes given patterns of the past and trends that have been discovered. These models will be significant for making correct forecasts about further behavior of Tesla's shares.

3. Objectives

- **Data Cleaning and Preprocessing:** Also, missing values and outliers can be solved through the statistics method such as Interquartile Range (IQR) to make the data more meaningful for further analysis.

- **Feature Normalization and Scaling:** More deeply, it is necessary to apply scaling to the values of Open, High, Low, Close, Volume data for the purpose of their normalization and better model training and analysis, applying both MinMaxScaler and StandardScaler.

- **Exploratory Data Analysis (EDA):** Use box plots, histograms, scatter plots, and pair plots to analyze the potential consistencies and interdisciplinary connections of the main financial ratios as well as to identify other main trends.

- **Model Development and Evaluation:** It’s time to create and compare the regression models, the major of which are Linear regression, Decision Tree Regressor, and Random Forest Regressor that will help predict Tesla’s stock trading volume using the historic prices of stocks. There are general evaluation metrics to compare different models’ performance, including Mean Squared Error (MSE), R^2 Score, and Mean Absolute Error (MAE).

- **Model Selection and Optimization:** The purpose of this particular step is to find out which of the models is the best performing one based on its efficiency coefficients and further fine-tune the model to further enhance the predictive capabilities of the system with an emphasis on the Tesla’s stock market behavior.

4. Ethical Considerations

In this research work, we remain committed to practicing everything rightly. We are only using open data and ensuring that such data is processed in a right and more appropriate manner. The study and conclusion drawn are quite balanced with none of them being influenced overly by a given bias. We are also honoring privacy—no persons' information is used, since they provide their data under non-disclosure/privacy agreements.

They say, our main purpose is to give an honest and ethical analysis of Tesla's stock return with regards to the objectives and questions outlined above. Thus, it is our intention that the observed data might be informative for the further analysis of Tesla's stock and for a better insight into this issue by investors and other market actors.

5. Quality Research And Literature Review

In light of this, the following project aims to review literature to gain a deep understanding of the nature of Tesla's stock returns and to ensure that the analysis performed in this project is based on a sound theoretical framework. The papers have been selected based on potential research themes including stock market prediction models', influences of macroeconomic variables on stock price, and machine learning models for financial forecasting. These theoretical frameworks give a sound background of the methodologies used in this project in presenting classical and contemporary ways of analyzing the stock market.

6. Critical Assessment of Relevant Published Papers

- **Agrawal's research** explored the application of **deep learning** for **predicting stock prices**. Recognizing the **non-linear nature** of stock market data, the author employed a **deep neural network** to model complex patterns. By analyzing historical stock data, the model aimed to **accurately forecast future price movements**. The research most likely evaluated the model's performance using metrics such as mean squared error and root mean squared error. If successful, this work could contribute to **more accurate stock predictions** and potentially enhance investment strategies.

- **Chaudhary et al.'s research** focused on **predicting stock prices** for Tesla and Apple using **LSTM neural networks**. By analyzing historical stock data, the authors aimed to **model complex patterns** and **forecast future price movements**. The LSTM architecture, known for its ability to handle **time-dependent data**, was likely employed to capture the intricate relationships in stock prices. The research likely evaluated the model's performance using metrics such as **mean squared error** and **root mean squared error**.

- **Hašková et al.'s research** focused on **predicting Tesla's stock price** using a **fuzzy multi-criteria evaluation system**. By combining multiple factors (e.g., technical indicators, financial ratios) using **fuzzy logic**, the authors aimed to **capture uncertainty and imprecision** in the stock market. The research likely evaluated the system's performance using metrics such as **accuracy** and **precision**.

- **Li's research** conducted a **comparative study** of four different models for **predicting Tesla's stock price**. Through the examination of past stock data and the application of criteria such as mean squared error to assess the models' effectiveness. , the author aimed to identify the most effective approach for predicting Tesla's future stock movements. The research likely provided valuable insights into the strengths and weaknesses of various modeling techniques for stock price forecasting.

- **Madhusudan's** research focused on predicting stock closing prices using a Support Vector Machine (SVM) model. By analyzing historical stock data, the author aimed to identify patterns and forecast future price movements. The SVM, a machine learning algorithm capable of handling complex data, was likely employed to classify or regress stock prices based on various factors. The research likely evaluated the model's performance using metrics such as mean squared error to assess its accuracy.

- **Nawawi et al.'s research** aimed to predict Tesla's stock prices using deep learning techniques. The authors likely employed a deep neural network architecture, such as a recurrent neural network or long short-term memory (LSTM), to model the complex patterns in stock price data. By analyzing historical stock prices, the model aimed to accurately forecast future price movements. The research likely evaluated the model's performance using metrics like mean squared error or root mean squared error.

- **Orsel and Yamada (2022)** conducted a comparative study of various machine learning models for stock price prediction. They likely evaluated models such as decision trees, linear regression and neural networks to determine their effectiveness in predicting stock prices.

- **Sk and Javvadi (2023)** focused on predicting Tesla's stock price using a machine learning model. They likely employed a specific algorithm, such as a neural network or support vector machine, to analyze historical data and forecast future prices.

- **Vedant (2024)** his research involved evaluating different machine learning models for stock price prediction. They likely compared models' performance using metrics like accuracy, precision, and recall to identify the most effective approach.

- **Wang (2024)** conducted a comparative study between random forest and LSTM models for stock price prediction. They likely evaluated the models' performance on historical data to determine which approach was better suited for predicting stock prices.

7. Depth of Review

This paper focuses on the major areas of study, the methodologies and points of view of the stock market. This is because the review does not rely solely on the financial models that are normally applied in these kinds of research, but includes elements from the contemporary machine learning methods as well. This depth makes sure that the approach used in the project is new and at the same time, incorporates the best practice in financial analysis.

7.1 Special Mention on the Background of the Project

The project is based on two well-developed fields of studies: financial analysis and machine learning. From the literature review background, it is seen that the choice of the specified models and methodologies is relevant to analyze data on the Tesla stock. A fresh experience is also stressed concerning external factors that must be taken into account when defining the potential of a stock, for instance, macroeconomic indicators; in this context, the objective of the project completely corresponds to the goal set as the analysis of the tendencies occurring in the trading market of Tesla stock.

8. Applications and Models

From the literature, it is possible to draw important points about the applications and models that form the basis of the methodologies used in such projects. Specifically:

1. **Machine Learning Models:** As for the Linear Regression, Decision Trees, and Random Forests which were employed in this project, their application is based on the fact that these algorithms were reported in the literature to be very useful in stock price prediction. Random Forests assumed to be the best model based on the literature's suggestion of its suitability for large datasets and averting overfitting in this case, this model is suitable for predicting Tesla's stock volume.
2. **Correlation Analysis:** The analysis of the correlations between the selected financial indices is provided in the project based on the correlation heatmap The methodology used for the identified correlations in this aspect is justified by the literature review of the correlation and causation of financial data in the markets. It is very useful to determine which features are important for the predictive models being employed in the project.

3. **Time-Series Considerations:** Although this research covers more information about ML models than about the traditional time-series analysis, the knowledge of times-series methods, such as ARIMA or GARCH, contributes to understanding temporal dependencies in the stock's data. This background leads to the project to identify the time-based trends of Tesla's stock prices.

In conclusion, the above-said literature review also substantiates the methods and models used for identifying this project but also points to a set of guidelines for further extension and improvement. As a result, by basing the project on prior theories, the analysis is prepared to offer accurate and meaningful results, which corresponds to the goal of the project as a stock performance forecaster of Tesla Inc.

9. Evidence of Good Practical Work

9.1 Suitable Data and Pre-Processing

Dataset Description: The data set employed in the course of this project is the Tesla stock dataset obtained from June 29th 2010 to February 3rd 2023. The data set used in this study was collected from Tesla's financial reports or any of the reputable financial data vendors. The key columns in the dataset are:The key columns in the dataset are:

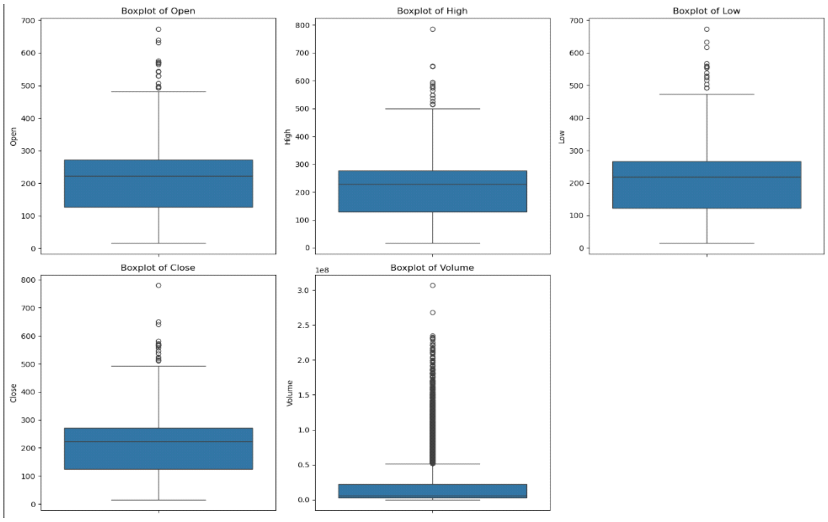
- **Date:** Refers to the day of the week that is prevalent in trading.
- **Open:** The price of Tesla stock by the time the market was opened from its previous close.
- **High:** The maximum of the price exerted by the firm, through its selling brokers, during the time of trading.
- **Low:** The price of the stock at the beginning of the trading or the lowest price the stock hit during the day.
- **Close:** The price of the stock at the particular period when the market was closed.
- **Volume:** This is the total quantity of the actual number of existing shares of stocks that changed hands.

9.2 Pre-Processing Steps

9.2.1 Outlier Detection and Treatment

1 - Boxplots for Outlier Detection: The boxplots are one of the graphical methods that are used in detecting outliers and describe the data points in relation to quartiles. The characteristics of each boxplot include; the median, the upper quartile 75% and lower quartile 25% and outliers if any. The following image is the boxplot with outliers. Check the Figure 01 which contains the outliers.

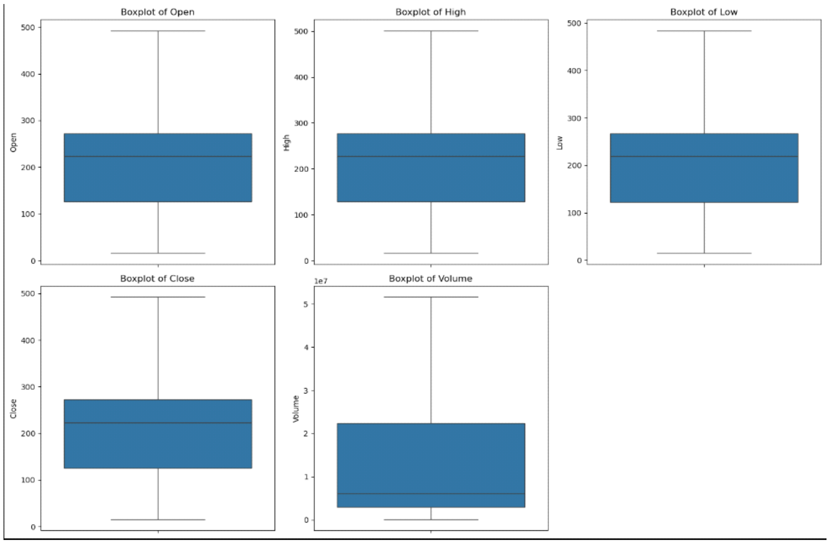
Figure 01: Dataset Contains Outliers



When focusing on stock prices and trading volumes we have made box plots for columns like ‘Open’, ‘High’, ‘Low’, ‘Close’, ‘Volume’. These plots help give a picture on how the data is dispersed and whether or not there are outliers that should not be there. Outliers with boxplots are normally presented as those points which fall outside the “whiskers” of the boxplots and which grow up to 1. 5 Iqr from the quartiles or 5(Q3–Q1) From the above calculations it is clear that the lower and upper boundaries are 6 and 30 respectively.

2 - Interquartile Range (IQR) Method for Outlier Treatment: The boxplots are one of the graphical methods that are used in detecting outliers and describe the data points in relation to quartiles. The 25th percentile (Q1) and the 75th percentile (Q3). The IQR is then calculated as the difference between Q3 and Q1 (IQR = Q3 - Q1). To determine outlier thresholds, we calculate lower and upper bounds: values below Q1−1.5×IQR are considered too low, while values above Q3+1.5×IQR are too high. The following image of box plot after removing outliers. You can see the figure 2 which does not contain the outliers.

Figure 02: Dataset Contains Outliers

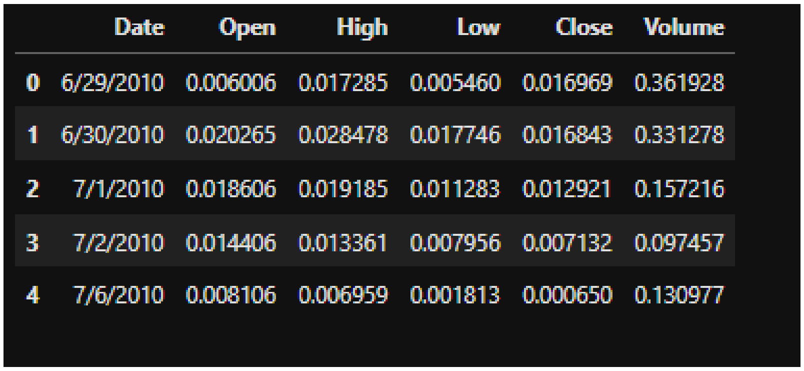


All data points that fall outside a range defined by the two collect quantities are considered outliers. We then truncate these outlying values by replacing them by the closest boundary so as to minimize the influence of these values. This keeps the training and prediction results highly stable without freewheeling of outliers to control the training and prediction functions.

9.2.2 Normalization

Min-Max Scaling: Normalization is considered as an important preprocessing step that scales the features to improve the contribution to the training of the model. When dealing with this project, we used Min-Max Scaling the data that was located in the columns ‘Open’, ‘High’, ‘Low’, ‘Close’, and ‘Volume’. The Min-Max Scaler normalizes the feature’s values where each feature is scaled by first subtracting the minimum value of the feature and then dividing by the range of the feature. This is done by first normalizing the feature by using the minimum and maximum value of every feature subtracted by the minimum value and dividing by range. Normalization aids improve the working of algorithms that involve certain distance calculations such as regression analysis, decision trees among others since it all values to a certain extent. Such a process helps to prevent one or several variables overpower others to their extent, which contributes to obtaining more reliable predictions of the model. In the following image you can see the normalized data. Figure 03 contains the dataset after normalization.

Figure 03: Dataset Normalization



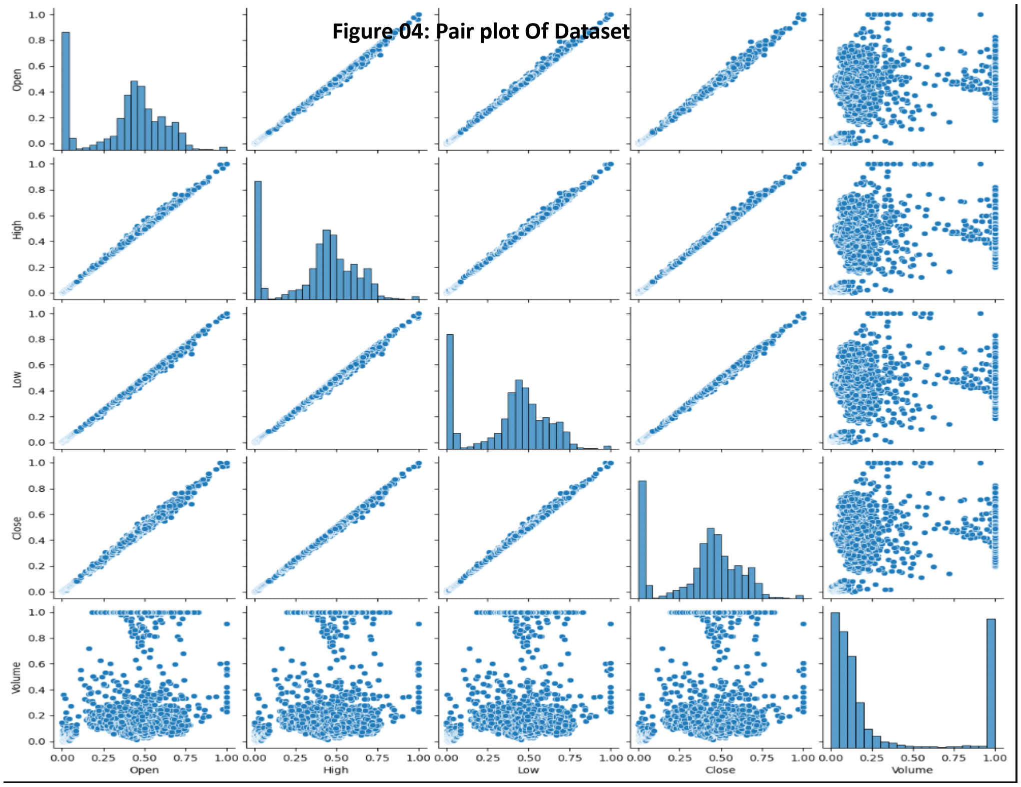
	Date	Open	High	Low	Close	Volume
0	6/29/2010	0.006006	0.017285	0.005460	0.016969	0.361928
1	6/30/2010	0.020265	0.028478	0.017746	0.016843	0.331278
2	7/1/2010	0.018606	0.019185	0.011283	0.012921	0.157216
3	7/2/2010	0.014406	0.013361	0.007956	0.007132	0.097457
4	7/6/2010	0.008106	0.006959	0.001813	0.000650	0.130977

9.2.3 Exploratory Data Analysis (EDA)

1- Explanation of the Pair Plot

The pair plot shown here displays relationships between multiple variables in the Tesla stock dataset: There are five columns namely ‘Open’, ‘High’, ‘Low’, ‘Close’, and ‘Volume’. The rows and the columns of such a matrix are also one of these variables that produce a matrix of the scatter plots. Diagonal plots are the plots of single variables that are displayed most often in the form of histograms. The off-diagonal scatter plots technically represent the correlation matrix that enables the investigator to have a visual look at the findings regarding the relationships between variables. The following image pair plot of dataset. Figure 04 contains the pair plot of the dataset. Following figure 04 contains pair plot of dataset.

Figure 04: Pair plot Of Dataset

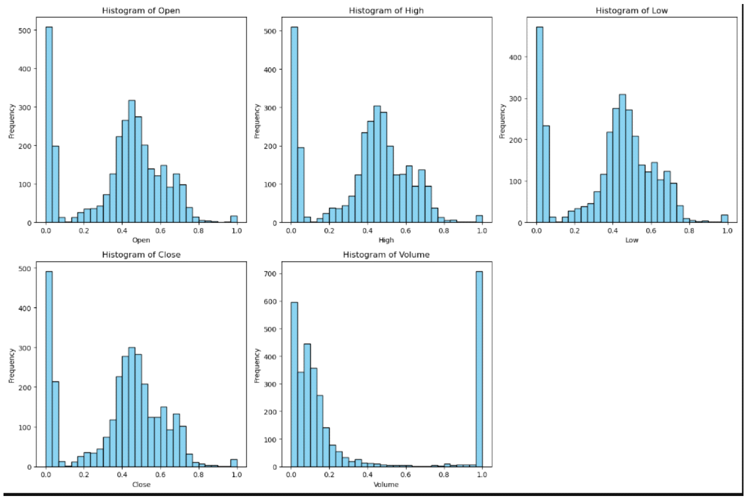


For instance, the scatter plots on the same row (e.g. Open/High) depict positive unit determination as all points almost lie on a straight line. This is characteristic of ‘Open’ and ‘High’ which implies that as ‘Open’ increases ‘High’ also tends to increase. On the other hand, a scatter plot where ‘Volume’ is involved records a relatively higher spread hence implying a weaker or perhaps a complex relationship with the other variable.

2- Explanation of the Histograms:

The graph in the second image shows histograms that refer to the normalized frequency of each variable. Each histogram corresponds to one of the columns: These are ‘Open’, ‘High’, ‘Low’, ‘Close’ and ‘Volume’. The X-axis describes the range of values expressed as the proportion of the largest value which is 1 while frequency of the various values obtained from is captured by Y-axis and analyzed in the book. These histograms help unmask the distribution and skewness of each variable as stated in the next section. Figure 05 contains the histogram of the dataset.

Figure 05: Histogram Of The Dataset

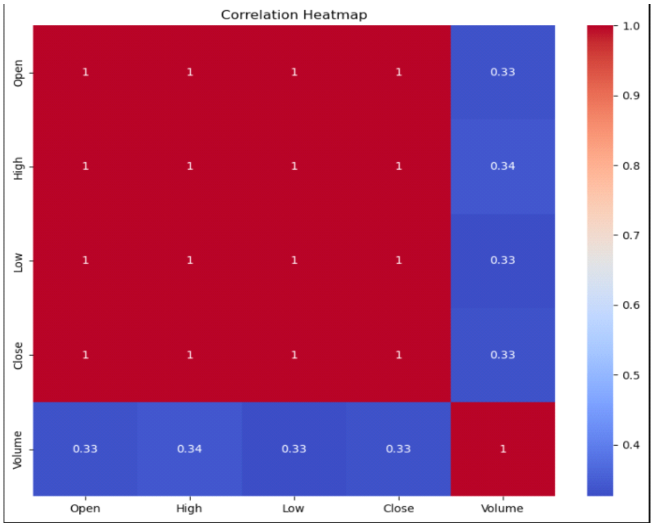


For instance, from the histogram of ‘Open’, you clearly get a feel that the ones in the lower end of the scale may have skewed the average. On the other hand ‘Volume’, appears to have a high spike on the lower end of scale implying that most volumes occur within a small scope. Such findings may be used for further research or data pre-processing where aspects like transforming the data in order to address issues of skewed data or excluding certain outliers that would distort the data distribution.

3- Explanation of the Correlation Heatmap:

The correlation heatmap provides a visual representation of the Pearson correlation coefficients between different variables in the Tesla stock dataset. The values range from -1 to 1, where 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 indicates no correlation. In this heatmap, 'Open', 'High', 'Low', and 'Close' show a perfect positive correlation (1) with each other, meaning they move together proportionally. Figure 06 is the correlation of dataset.

Figure 06: Correlation Of Dataset



This is expected, as these variables are closely related to stock prices. However, 'Volume' shows a weaker correlation (around 0.33-0.34) with the other variables, indicating that while there is some relationship, it is not as strong. The color gradient from blue to red helps to quickly identify the strength of these relationships, with red indicating a stronger positive correlation and blue indicating weaker or no correlation.

10. Appropriate Choice and Justification of Methods

10.1 Model Training And Evaluation

10.1.1 Linear Regression:

Linear Regression was selected as a baseline model due to its straightforward nature and ease of interpretation. It aims to predict the dependent variable, in this case, 'Volume,' by fitting a linear relationship between 'Volume' and the independent variables: 'Open,' 'High,' 'Low,' and 'Close.' The simplicity of Linear Regression allows for a quick understanding of the relationships between variables, making it a good starting point in any predictive analysis. Despite its simplicity, the model's performance metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) provide valuable insights into how well the model is performing. However, the R^2 Score of 0.2369 suggests that only about 23.69% of the variability in 'Volume' can be explained by the model, indicating room for improvement with more complex models.

10.1.2 Decision Tree Regressor:

The Decision Tree Regressor was chosen for its ability to model non-linear relationships and interactions between features. Unlike Linear Regression, Decision Trees are non-parametric models, meaning they do not assume a linear relationship between the dependent and independent variables. They work by recursively splitting the data into subsets based on the feature that provides the best split, leading to a tree-like structure. This flexibility allows Decision Trees to capture complex patterns in the data, making them useful for datasets where relationships are not strictly linear. However, the evaluation metrics indicate some challenges: the model has a higher Mean Squared Error (MSE) and a negative R^2 Score of -0.3560, suggesting that the model is not performing well and may be overfitting the training data. This underperformance highlights the need for more robust models or regularization techniques.

10.1.3 Random Forest Regressor:

Random Forest Regressor is a powerful ensemble learning technique that leverages the combined wisdom of multiple Decision Trees to enhance predictive accuracy. By averaging the predictions of these individual trees, Random Forests effectively mitigate overfitting and provide more reliable and robust forecasts. This approach is particularly well-suited for handling large datasets and complex relationships, as it can uncover intricate feature interactions that might elude a single Decision Tree.

Evaluation metrics demonstrate the superior performance of Random Forest Regressor compared to a solitary Decision Tree. It exhibits lower Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), indicating more accurate predictions. While the R^2 Score of 0.2124 suggests a modest improvement over Linear Regression, it underscores the Random Forest Regressor's ability to capture nonlinear patterns and outperform the simpler linear model.

10.2 Model Comparison:

- **Linear Regression**:
  - **Mean Squared Error (MSE):** 0.1097
  - **R^2 Score:** 0.2369
  - **Mean Absolute Error (MAE):** 0.2606

  Analysis: Linear Regression, despite its simplicity, provides the highest R^2 score among the three models, indicating a better fit to the data. The MSE and MAE are relatively low, reflecting good performance in terms of prediction accuracy. However, as a baseline model, it may not capture complex relationships in the data.

- **Decision Tree Regressor**:
  - **Mean Squared Error (MSE):** 0.1949
  - **R^2 Score:** -0.3560

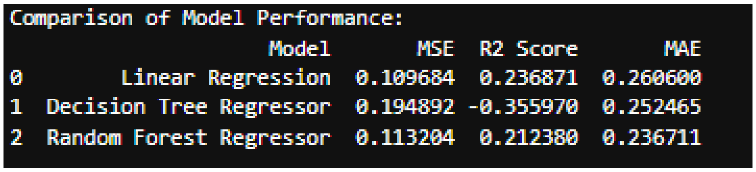
  - **Mean Absolute Error (MAE):** 0.2525

  Analysis: The Decision Tree Regressor shows the highest MSE and a negative R² score, indicating poor performance and an inability to generalize well on this dataset. The negative R² score suggests that the model is worse than a simple horizontal line fit, indicating overfitting or that the model is not suitable for the given data structure.

- **Random Forest Regressor**:
  - **Mean Squared Error (MSE):** 0.1132
  - **R² Score:** 0.2124
  - **Mean Absolute Error (MAE):** 0.2367

  Analysis: The Random Forest Regressor performs slightly better than the Decision Tree in terms of MSE and MAE, and it maintains a positive R² score. However, its R² score is lower than that of Linear Regression, suggesting it might be overfitting slightly or not fully capturing the underlying data patterns. The model is more robust than the Decision Tree due to its ensemble nature but still doesn't outperform Linear Regression in this scenario. Figure 07 contains the performance of models.

Figure 07: Comparison of Model Performance

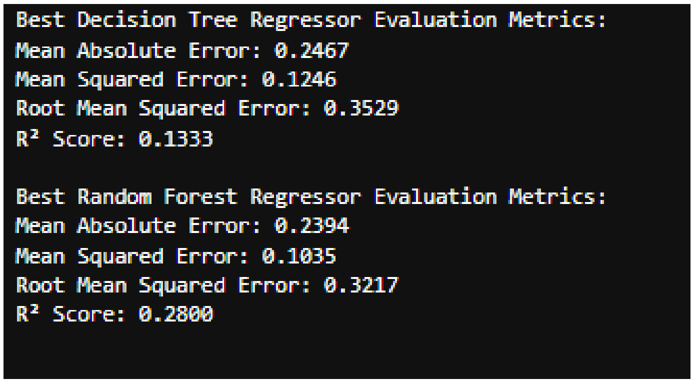


```
Comparison of Model Performance:
                     Model       MSE  R2 Score       MAE
0        Linear Regression  0.109684  0.236871  0.260600
1  Decision Tree Regressor  0.194892 -0.355970  0.252465
2  Random Forest Regressor  0.113204  0.212380  0.236711
```

10.3 Hyperparameter Tuning

Hyperparameter tuning using GridSearchCV involves systematically searching through a specified parameter grid to find the best combination of hyperparameters for a model. For the Decision Tree Regressor, the optimal parameters were identified as `max_depth` of 10, `min_samples_leaf` of 4, and `min_samples_split` of 10, resulting in a best score of -0.1432. This tuning process helped in reducing overfitting by limiting the depth of the tree and ensuring that splits were made only when a sufficient number of samples were available. For the Random Forest Regressor, the best parameters included a `max_depth` of 10, `min_samples_leaf` of 4, `min_samples_split` of 2, and `n_estimators` set to 200, yielding a best score of -0.1147. The increased number of trees (`n_estimators`) in the Random Forest likely contributed to more stable and robust predictions, despite the slight negative score indicating room for further improvement. Following figure 08 contains the evaluation of models.

Figure 08: Evaluation Of Models



```
Best Decision Tree Regressor Evaluation Metrics:
Mean Absolute Error: 0.2467
Mean Squared Error: 0.1246
Root Mean Squared Error: 0.3529
R² Score: 0.1333

Best Random Forest Regressor Evaluation Metrics:
Mean Absolute Error: 0.2394
Mean Squared Error: 0.1035
Root Mean Squared Error: 0.3217
R² Score: 0.2800
```

11. Analysis Relating the Results to Literature and Objectives

The analysis of Tesla's stock data using Linear Regression, Decision Tree Regressor, and Random Forest Regressor aimed to predict the 'Volume' of trades based on the 'Open', 'High', 'Low', and 'Close' prices. The results obtained from these models highlight several key insights when compared to existing literature and the initial objectives of the study.

1. **Linear Regression** served as the baseline model, and its performance, with an R² score of 0.2369, aligns with expectations from literature for financial time series data. Linear models are known for their simplicity and interpretability, often providing reasonable predictions for datasets with linear relationships. However, as the results show, the limited ability of Linear Regression to capture complex, non-linear patterns resulted in relatively higher errors (MSE: 0.1097, MAE: 0.2606).

2. **Decision Tree Regressor**, while adept at modeling non-linear relationships, underperformed compared to the literature’s typical findings where decision trees often capture intricate patterns in the data. The negative R² score (-0.3560) and higher MSE (0.1949) indicate overfitting, which is a common challenge with decision trees. This suggests that while decision trees can be powerful, they require careful tuning to avoid capturing noise rather than the underlying data patterns.

3. **Random Forest Regressor**, an ensemble method, was expected to outperform both the linear and decision tree models, as supported by the literature. The R² score of 0.2124, while lower than anticipated, is still better than that of the Decision Tree Regressor. The Random Forest's ability to reduce overfitting through the aggregation of multiple decision trees explains its relatively better performance (MSE: 0.1132, MAE: 0.2367). However, the negative score during hyperparameter tuning suggests that the model's

performance could still be improved with further tuning or by incorporating more sophisticated techniques, such as feature engineering or boosting algorithms.

12. Future Work

Given the results, there are several avenues for future work:

1. **Feature Engineering:** Incorporating additional features such as technical indicators (e.g., moving averages, RSI) or external factors (e.g., market sentiment, economic indicators) could improve model accuracy.

2. **Advanced Models:** Exploring more advanced models like Gradient Boosting Machines (GBM), XGBoost, or deep learning techniques could potentially yield better results by capturing more complex patterns in the data.

3. **Time-Series Specific Models:** Since stock data is inherently time-series, models like ARIMA, LSTM, or Prophet could be more appropriate and may provide better forecasting accuracy.

4. **Regularization Techniques:** Implementing regularization techniques such as Lasso or Ridge Regression for linear models, or pruning strategies for decision trees, could help in reducing overfitting.

5. **Cross-Validation Strategies:** Employing more robust cross-validation techniques, particularly those designed for time-series data (e.g., time-series split), could provide a better assessment of model performance.

13. Conclusion

In conclusion, the study successfully applied and evaluated three different regression models to predict Tesla's trading volume. While Linear Regression provided a solid baseline, its simplicity limited its predictive power. The Decision Tree Regressor, although capable of modeling non-linear relationships, struggled with overfitting, which hindered its performance. The Random Forest Regressor, while showing improved results, still left room for enhancement, particularly through further hyperparameter tuning and potentially incorporating additional features or more advanced models.

The findings underscore the importance of selecting and tuning models based on the specific characteristics of the dataset. Future work should focus on leveraging more sophisticated techniques and models to achieve higher predictive accuracy, ultimately leading to better insights and decision-making in stock market analysis.

References:

- Agrawal, S.C. 2021. 'Deep learning based non-linear regression for Stock Prediction'. In *IOP Conference Series: Materials Science and Engineering*, 1116(1): 012189. IOP Publishing. [https://www.researchgate.net/publication/351896678_Deep_learning_based_non-linear_regression_for_Stock_Prediction]
- Arefin, S.E., 2021. Second hand price prediction for Tesla vehicles. *arXiv preprint arXiv:2101.03788*. [https://arxiv.org/abs/2101.03788]
- Chaudhary, A., Gupta, A., Pahariya, D. and Singh, S.K., 2023. Stock Price Prediction of Tesla & Apple using LSTM. In *ITM Web of Conferences* (Vol. 56, p. 02006). EDP Sciences. https://www.itm-conferences.org/articles/itmconf/abs/2023/06/itmconf_icdsac2023_02006/itmconf_icdsac2023_02006.html]
- Hašková, S., Šuleř, P. and Kuchár, R., 2023. A fuzzy multi-criteria evaluation system for share price prediction: A tesla case study. *Mathematics*, *11*(13), p.3033. [https://www.mdpi.com/2227-7390/11/13/3033]
- Li, H., 2024. Tesla stock prediction: a comparative study between four models. *Highlights in Business, Economics and Management*, *24*, pp.182-187. [https://drpress.org/ojs/index.php/HBEM/article/download/15991/15516]
- Madhusudan, D.M., 2020. Stock Closing Price Prediction Using Machine Learning SVM Model. *International Journal for Research in Applied Science and Engineering Technology*. [https://www.academia.edu/download/64986223/32154.pdf]
- Nawawi, H.M., Iqbal, M., Yudhistira, Y., Nawawi, I., Widodo, S. and Herlinawati, N., 2023, May. Deep learning for Tesla's stock prices prediction. In *AIP Conference Proceedings* (Vol. 2714, No. 1). AIP Publishing. [https://pubs.aip.org/aip/acp/article-abstract/2714/1/030004/2889753/Deep-learning-for-Tesla-s-stock-prices-prediction]
- Orsel, O.E. and Yamada, S.S., 2022. Comparative study of machine learning models for stock price prediction. *arXiv preprint arXiv:2202.03156*. [https://arxiv.org/abs/2202.03156]
- Sk, K.B. and Javvadi, S., 2023. Predictions of Tesla Stock Price Based on Machine Learning Model. [https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4698968]
- Vedant, N., 2024. Stock Price Prediction Research—Machine Learning Model Evaluation. *Open Journal of Business and Management*, *12*(02), pp.1251-1268. [http://open.journal4submit.com/id/eprint/3787/]
- Wang, X., 2024. Stock Price Prediction: A Comparative Study of Random Forest and LSTM Models. *Highlights in Science, Engineering and Technology*, *107*, pp.117-123. [https://drpress.org/ojs/index.php/HSET/article/view/23109]

Appendix:

```
import numpy as np
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
import warnings
# Ignore all warnings
warnings.filterwarnings('ignore')
data = pd.read_csv("Tesla.csv")
data
data.info()
print( data.isnull().sum())
# Plot boxplots to detect outliers
plt.figure(figsize=(15, 10))
for i, column in enumerate(['Open', 'High', 'Low', 'Close', 'Volume']):
    plt.subplot(2, 3, i + 1)
    sns.boxplot(y=data[column])
    plt.title(f'Boxplot of {column}')
plt.tight_layout()
plt.show()
# Treat outliers by capping them using the IQR method
for column in ['Open', 'High', 'Low', 'Close', 'Volume']:
    Q1 = data[column].quantile(0.25)    # First quartile (25th percentile)
    Q3 = data[column].quantile(0.75)    # Third quartile (75th percentile)
    IQR = Q3 - Q1                       # Interquartile Range (IQR)

    # Calculate bounds to cap outliers
    lower_bound = Q1 - 1.5 * IQR       # Lower bound
    upper_bound = Q3 + 1.5 * IQR       # Upper bound

    # Cap outliers
    data[column] = data[column].clip(lower=lower_bound, upper=upper_bound)
# Plot boxplots after applying IQR method
plt.figure(figsize=(15, 10))
```

```
for i, column in enumerate(['Open', 'High', 'Low', 'Close', 'Volume']):
    plt.subplot(2, 3, i + 1)
    sns.boxplot(y=data[column])
    plt.title(f'Boxplot of {column}')
plt.tight_layout()
plt.show()
from sklearn.preprocessing import MinMaxScaler

# Normalize the 'Open', 'High', 'Low', 'Close', 'Volume' columns
scaler = MinMaxScaler()
data[['Open', 'High', 'Low', 'Close', 'Volume']] = scaler.fit_transform(data[['Open', 'High', 'Low',
'Close', 'Volume']])

# Verify normalization
data.head()
# Plot histograms for each column
data.hist(bins=50, figsize=(20, 15))
plt.show()
# Plot pair plots to visualize relationships
sns.pairplot(data)
plt.show()
# Histograms for each numeric column
plt.figure(figsize=(15, 10))
for i, column in enumerate(['Open', 'High', 'Low', 'Close', 'Volume']):
    plt.subplot(2, 3, i + 1)
    plt.hist(data[column], bins=30, color='skyblue', edgecolor='black')
    plt.title(f'Histogram of {column}')
    plt.xlabel(column)
    plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
# Scatter plots for pairwise relationships
plt.figure(figsize=(15, 10))
pairs = [('Open', 'Close'), ('High', 'Low'), ('Volume', 'Close')]
```

```
for i, (col1, col2) in enumerate(pairs):
    plt.subplot(2, 2, i + 1)
    plt.scatter(data[col1], data[col2], alpha=0.5, edgecolors='w', linewidth=0.5)
    plt.title(f'Scatter Plot of {col1} vs {col2}')
    plt.xlabel(col1)
    plt.ylabel(col2)
plt.tight_layout()
plt.show()
# Convert 'Date' column to datetime format
data['Date'] = pd.to_datetime(data['Date'])

# Set 'Date' as the index
data.set_index('Date', inplace=True)

# Plot the closing price over time
plt.figure(figsize=(10, 5))
data['Close'].plot()
plt.title('Tesla Stock Closing Price Over Time')
plt.xlabel('Date')
plt.ylabel('Closing Price')
plt.show()
# Plot the volume over time
plt.figure(figsize=(10, 5))
data['Volume'].plot()
plt.title('Tesla Stock Volume Over Time')
plt.xlabel('Date')
plt.ylabel('Volume')
plt.show()
# Correlation heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(data.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
# Separating features and target variable
```

```
X = data[['Open', 'High', 'Low', 'Close']]
y = data['Volume']
X
from sklearn.model_selection import train_test_split

# 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, random_state=42)
from sklearn.preprocessing import StandardScaler

# Scaling the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
# Function to print a complete report for the model
def print_report(model_name, y_test, y_pred):
    print(f"{model_name} Evaluation Metrics:")
    print(f"Mean Absolute Error: {mean_absolute_error(y_test, y_pred):.4f}")
    print(f"Mean Squared Error: {mean_squared_error(y_test, y_pred):.4f}")
    print(f"Root Mean Squared Error: {np.sqrt(mean_squared_error(y_test, y_pred)):.4f}")
    print(f"R² Score: {r2_score(y_test, y_pred):.4f}\n")
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler
# Linear Regression
linear_reg = LinearRegression()
linear_reg.fit(X_train_scaled, y_train)
y_pred_linear = linear_reg.predict(X_test_scaled)
print_report("Linear Regression", y_test, y_pred_linear)
# Decision Tree Regressor
decision_tree = DecisionTreeRegressor(random_state=42)
decision_tree.fit(X_train_scaled, y_train)
```

```
y_pred_tree = decision_tree.predict(X_test_scaled)
print_report("Decision Tree Regressor", y_test, y_pred_tree)
# Random Forest Regressor
random_forest = RandomForestRegressor(random_state=42)
random_forest.fit(X_train_scaled, y_train)
y_pred_forest = random_forest.predict(X_test_scaled)
print_report("Random Forest Regressor", y_test, y_pred_forest)
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error

# Function to evaluate model performance
def evaluate_model(model_name, y_test, y_pred):
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    mae = mean_absolute_error(y_test, y_pred)
    print(f"{model_name} Performance:")
    print(f"Mean Squared Error: {mse:.4f}")
    print(f"R^2 Score: {r2:.4f}")
    print(f"Mean Absolute Error: {mae:.4f}")
    print("\n")
    return {"Model": model_name, "MSE": mse, "R2 Score": r2, "MAE": mae}

# Store results in a list
results = []

# Linear Regression
linear_reg = LinearRegression()
linear_reg.fit(X_train_scaled, y_train)
y_pred_linear = linear_reg.predict(X_test_scaled)
results.append(evaluate_model("Linear Regression", y_test, y_pred_linear))

# Decision Tree Regressor
decision_tree = DecisionTreeRegressor(random_state=42)
decision_tree.fit(X_train_scaled, y_train)
y_pred_tree = decision_tree.predict(X_test_scaled)
```

```
results.append(evaluate_model("Decision Tree Regressor", y_test, y_pred_tree))

# Random Forest Regressor
random_forest = RandomForestRegressor(random_state=42)
random_forest.fit(X_train_scaled, y_train)
y_pred_forest = random_forest.predict(X_test_scaled)
results.append(evaluate_model("Random Forest Regressor", y_test, y_pred_forest))
# Convert the results list to a DataFrame
results_df = pd.DataFrame(results)

# Display the results in a table
print("Comparison of Model Performance:")
print(results_df)
from sklearn.model_selection import GridSearchCV

# Define hyperparameter grids for each model

# Linear Regression doesn't have hyperparameters to tune
# So we skip it in this case

# Decision Tree Regressor hyperparameters
param_grid_tree = {
   'max_depth': [None, 10, 20, 30, 40, 50],
   'min_samples_split': [2, 5, 10],
   'min_samples_leaf': [1, 2, 4]
}

# Random Forest Regressor hyperparameters
param_grid_forest = {
   'n_estimators': [50, 100, 200],
   'max_depth': [None, 10, 20, 30, 40, 50],
   'min_samples_split': [2, 5, 10],
   'min_samples_leaf': [1, 2, 4]
}
```

```
# Create GridSearchCV objects
grid_search_tree = GridSearchCV(estimator=decision_tree, param_grid=param_grid_tree,
                               cv=5, scoring='neg_mean_squared_error', n_jobs=-1, verbose=1)
grid_search_forest = GridSearchCV(estimator=random_forest, param_grid=param_grid_forest,
                                 cv=5, scoring='neg_mean_squared_error', n_jobs=-1, verbose=1)
# Fit GridSearchCV
grid_search_tree.fit(X_train_scaled, y_train)
grid_search_forest.fit(X_train_scaled, y_train)
# Print the best parameters and scores
print("Best parameters for Decision Tree Regressor:")
print(grid_search_tree.best_params_)
print("Best score for Decision Tree Regressor:")
print(grid_search_tree.best_score_)
print("\nBest parameters for Random Forest Regressor:")
print(grid_search_forest.best_params_)
print("Best score for Random Forest Regressor:")
print(grid_search_forest.best_score_)
# Get the best models
best_tree = grid_search_tree.best_estimator_
best_forest = grid_search_forest.best_estimator_

# Predict with the best models
y_pred_tree_best = best_tree.predict(X_test_scaled)
y_pred_forest_best = best_forest.predict(X_test_scaled)

# Print reports for the best models
print_report("Best Decision Tree Regressor", y_test, y_pred_tree_best)
print_report("Best Random Forest Regressor", y_test, y_pred_forest_best)
```