**Machine Learning for Predicting Options Price Movements Using Historical Volatility Patterns**

**Submitted by**

**Chukwuebuka Igbosonu**

[**Higher Diploma in Science in Data Analytics Information**](https://moodle2024.ncirl.ie/course/view.php?id=976)

**06-12-2024**

Table of Contents

[INTRODUCTION 4](#_Toc183714033)

[METHODOLOGY 4](#_Toc183714034)

[**Data Collection** 5](#_Toc183714035)

[**Historical Volatility Calculation** 5](#_Toc183714036)

[**Data Cleaning and Transformation** 5](#_Toc183714037)

[**Exploratory Data Analysis (EDA)** 5](#_Toc183714038)

[**Option Data Analysis** 5](#_Toc183714039)

[**Moving Averages and Trend Analysis** 6](#_Toc183714040)

[**Correlation and Relationship Analysis** 6](#_Toc183714041)

[**Volume and Implied Volatility Analysis** 6](#_Toc183714042)

[**Cumulative Returns and Risk Analysis** 6](#_Toc183714043)

[**Visualization and interpretation** 6](#_Toc183714044)

[**Machine Learning Model Training for Option Price Prediction** 6](#_Toc183714045)

[**LSTM Model Training for Option Price Prediction** 7](#_Toc183714046)

[**Backtesting the Predictive Models** 7](#_Toc183714047)

[**Live Trading Strategy Model Development** 7](#_Toc183714048)

[DATA ANALYSIS 8](#_Toc183714049)

[**Introduction** 8](#_Toc183714050)

[**Descriptive Statistics** 8](#_Toc183714051)

[Data Visualization 11](#_Toc183714052)

[**Time Series Plot for Key Variables** 11](#_Toc183714053)

[**Volatility Analysis** 13](#_Toc183714054)

[**Distribution Analysis** 14](#_Toc183714055)

[**Relationship Analysis** 17](#_Toc183714056)

[**Telsa Options Data Exploration** 17](#_Toc183714057)

[**Time-based Analysis** 19](#_Toc183714058)

[**Tesla Historical Volatility Over Different Periods** 22](#_Toc183714059)

[**Machine Learning and Random Forest Classifier** 29](#_Toc183714060)

[**Data Preprocessing and Target Definition** 30](#_Toc183714061)

[**Train-Test Split** 30](#_Toc183714062)

[**Feature Scaling** 30](#_Toc183714063)

[**Training the Random Forest Classifier** 30](#_Toc183714064)

[**Model Evaluation** 31](#_Toc183714065)

[**Hyperparameter Tuning** 32](#_Toc183714066)

[**GridSearchCV** 32](#_Toc183714067)

[**Learning Curve** 33](#_Toc183714068)

[**Validation Curve** 34](#_Toc183714069)

[**Analyzing Predicted Probabilities** 35](#_Toc183714070)

[**Long-Short-Term Memory (LSTM) Model** 37](#_Toc183714071)

[**Model Evaluation** 38](#_Toc183714072)

[**Model Comparison: RandomForestClassifier vs LSTM** 38](#_Toc183714073)

[**Model Back Testing** 39](#_Toc183714074)

[**Visualize the Backtesting Results** 40](#_Toc183714075)

# INTRODUCTION

This report provides comprehensive documentation for a predictive trading model to forecast stock price movements and automate trading decisions. The primary objective of this model is to leverage advanced machine learning techniques, specifically Long-Short-Term Memory (LSTM) networks, to analyze historical and real-time financial data to predict price movements. The model aims to optimize portfolio performance and manage risk by integrating these predictions with a robust decision-making framework.

The development of this model aligns with the growing need for data-driven approaches in financial markets, where accurate and timely predictions can significantly impact trading outcomes. The model's key features include preprocessing mechanisms for real-time data streams, predictive capabilities based on historical trends, and automated trade execution triggered by prediction thresholds. These features have been meticulously tested to ensure reliability and scalability, making the model suitable for diverse market conditions.

This report outlines the model’s core components, including its architecture, data preprocessing pipeline, feature engineering techniques, and decision-making logic. It also details the methodology adopted for model training, evaluation, and backtesting. Performance metrics such as accuracy, precision, recall, and Sharpe ratio have been used to assess the models' effectiveness. Additionally, the backtesting process simulates real-world trading scenarios to validate the model’s ability to perform under varying market dynamics.

This report is intended for stakeholders such as data scientists, financial analysts, and trading professionals. It provides a user manual for understanding the model’s functionality, interpreting its output, and integrating it into trading operations. It aims to bridge the gap between technical complexity and practical application, ensuring that users at all expertise levels can derive value from the model.

# METHODOLOGY

The analysis involves a detailed exploration of Tesla’s stock and options market data to understand trends, correlations, and key financial metrics. This methodology outlines the steps undertaken in data collection, cleaning, analysis, and visualization processes to generate actionable insights.

## **Data Collection**

Tesla stock data was fetched using the Yahoo Finance Python library (yfinance) from January 1, 2007, to the present day. Historical stock prices include opening, closing, high and low prices, and trading volume. Simultaneously, Tesla's options chain data, including calls and puts, were obtained for all available expiration dates. Additional market indicators, such as daily price changes, were also calculated to complement the analysis.

## **Historical Volatility Calculation**

Historical volatility, a key metric for understanding stock price fluctuations, was computed using a rolling standard deviation of daily returns, scaled annually.

## **Data Cleaning and Transformation**

The collected datasets were merged into a unified structure to facilitate analysis. Missing values were handled using backfilling to ensure completeness. The data types for critical columns, such as Expiration\_date and lastTradeDate for option data, were standardized to DateTime format. Duplicates were removed, and derived features like bid-ask spread and rolling volatility metrics were calculated for additional insights.

## **Exploratory Data Analysis (EDA)**

Time series plots were generated to visualize Tesla’s Closing prices and compare them with broader market trends. Historical volatility and daily returns were examined over time to identify patterns. The distribution of key features, such as stock prices, returns, implied volatility, and bid-ask spreads, was visualized using histograms and kernel density plots. Scatter plots and heatmaps explored relationships between features, such as the correlation between volatility, returns, and trading volumes.

## **Option Data Analysis**

Options data was analyzed to assess the distribution of strike prices, implied volatility, and open interest. The bid-ask spread, a measure of market liquidity, was computed and visualized. A time-based analysis of options volume and open interest across expiration dates revealed patterns in trader activity and sentiment.

## **Moving Averages and Trend Analysis**

Moving averages, specifically the 50-day and 200-day averages, were calculated and plotted against Tesla’s closing prices to identify long-term trends. Rolling volatility metrics over short- (10-day) and long-term (60-day) windows were also examined to understand the persistence of price fluctuations.

## **Correlation and Relationship Analysis**

Pairwise correlations between key financial metrics were calculated to identify statistically significant relationships. Pairplots and scatterplots were employed to visually examine relationships between daily returns, historical volatility, and other variables. Heatmaps were used to present the correlation matrix, highlighting significant positive and negative correlations.

## **Volume and Implied Volatility Analysis**

Relationships between trading volume and price metrics were explored through scatterplots. Implied volatility, a measure of market expectations of future volatility, was analyzed over time and in relation to strike prices, with insights on options moneyness (in-the-money vs. out-of-the-money).

## **Cumulative Returns and Risk Analysis**

Cumulative returns were calculated to understand Tesla stock's overall performance over time. The rolling historical volatility was plotted alongside the cumulative return to illustrate the interplay between risk and return in Tesla’s stock performance.

## **Visualization and interpretation**

Comprehensive visualizations, including line charts, scatterplots, histograms, and bar charts, were employed to make the insights accessible and interpretable. These visualizations aimed to provide a holistic view of Tesla’s stock and options data dynamics, enabling stakeholders to make informed decisions.

## **Machine Learning Model Training for Option Price Prediction**

To predict option price movements, we trained machine learning models using a supervised learning approach. The target variable was the direction of price movement (up or down) based on the bid and ask prices. At the same time, the features included ‘strike’, ‘bid’, ‘ask’, ‘volume’, ‘openInterest’, ‘impliedVolatility’, and ‘inTheMoney’. A Random Forest classifier was implemented as a baseline due to its rustiness to overfitting and ability to handle complex interactions between features. The dataset was split into training and test sets, ensuring temporal order to avoid data leakage. Feature importance analysis was also conducted to understand the contribution of each feature.

## **Backtesting the Predictive Models**

Backtesting was conducted to evaluate the predictive models in a simulated trading environment. Historical data was used to test the model's ability to forecast option price movements and generate profitable trades. A rolling-window approach ensured predictions were made using only past data, emulating a real-world trading scenario. Key performance metrics, including cumulative return, Sharpe ratio, and maximum drawdown, were strategy’s profitability and risk-adjusted returns. The results informed refinements in the predictive models and trading strategy.

## **Live Trading Strategy Model Development**

A live trading strategy model was developed to execute trades based on the selected predictive model’s outputs. The Alpaca Broker API was integrated to enable real-time data retrieval, order placement, and portfolio management. The strategy continuously monitored market data, generating buy and sell signals based on the model’s predictions. A risk management framework was implemented to minimize potential losses, including position sizing and stop-loss rules. The live strategy was tested in a paper trading environment to validate its functionality before deployment in live trading. This step bridges the gap between predictive modelling and practical application, demonstrating the approach's feasibility in real-world scenarios.

# DATA ANALYSIS

## **Introduction**

The data analysis chapter comprehensively explores the dataset to uncover insights, validate assumptions, and prepare the data for predictive modelling. This process involves descriptive statistics, exploratory data analysis (EDA), and feature engineering to enhance the dataset’s predictive power. Visualizations and summary statistics are employed to identify the data's patterns, correlations, and trends, offering a deeper understanding of the key variables influencing option price movements. Anomalies and missing values are addressed through appropriate preprocessing techniques to ensure the quality and reliability of the dataset. This chapter lays the groundwork for model training by transforming raw data into a structured, clean, and meaningful format, enabling the development of robust and effective trading strategies.

## **Descriptive Statistics**

***Table 1: Combined Data Description***

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Statistics | Open | High | Low | Close tesla | Volume | Dividends | Stock Splits | Daily Return | Historical Volatility | Close market | Daily Change |
| count | 3629.0 | 3629.0 | 3629.0 | 3629.0 | 3629.0 | 3629.0 | 3629.0 | 3629.0 | 3629.0 | 3629.0 | 3629.0 |
| mean | 79.46626827157736 | 81.2053061017454 | 77.61532335535624 | 79.44727232870763 | 96737180.84871866 | 0.0 | 0.002204464039680353 | 0.0021246279161578913 | 0.536710663768927 | 79.44727232870763 | 0.09276201269060956 |
| std | 104.76285171533345 | 107.08918078079253 | 102.23381963440879 | 104.6884763699842 | 77942267.97445093 | 0.0 | 0.09678167995201194 | 0.036112591637860535 | 0.19638053891744278 | 104.6884763699842 | 5.0317498999225005 |
| min | 1.0759999752044678 | 1.108667016029358 | 0.9986670017242432 | 1.053333044052124 | 1777500.0 | 0.0 | 0.0 | -0.2106282432039079 | 0.1834226728376559 | 1.053333044052124 | -46.480010986328125 |
| 25% | 11.990667343139648 | 12.315333366394045 | 11.69333267211914 | 11.981332778930664 | 48674600.0 | 0.0 | 0.0 | -0.0161356983874155 | 0.3908658871743029 | 11.981332778930664 | -0.2786674499511719 |
| 50% | 17.78266716003418 | 18.06666755676269 | 17.488000869750977 | 17.81333351135254 | 82035900.0 | 0.0 | 0.0 | 0.0012263011647262 | 0.5063817140623103 | 17.81333351135254 | 0.0093331336975097 |
| 75% | 175.1300048828125 | 179.3800048828125 | 172.5 | 175.66000366210938 | 122452500.0 | 0.0 | 0.0 | 0.0193540620736329 | 0.6301766109872069 | 175.66000366210938 | 0.3773345947265625 |
| max | 411.4700012207031 | 414.4966735839844 | 405.6666564941406 | 409.9700012207031 | 914082000.0 | 0.0 | 5.0 | 0.243950758246163 | 1.3935976092988096 | 409.9700012207031 | 47.66665649414063 |

The dataset consists of 3,629 observations, capturing various metrics related to Tesla’s stock and market performance. Key variables include opening, high, low, and closing prices for Tesla stock, trading volume, dividends, stock splits, daily returns, historical volatility, market-level closing prices, and daily changes. The average closing price for Tesla stock is approximately 79.45, with a standard deviation of 104.69, highlighting significant variability in stock prices. Trading volume shows an average of 96.74 million, with a peak of over 914 million, indicating periods of heightened activity. Daily returns exhibit a slight positive mean (0.0022) with a maximum of 24.39% and a minim 21.06%, reflecting the stock’s volatility. Historical volatility ranges from 18.34% to 139.36%, averaging 53.67%, suggesting consistent price movement fluctuations. Dividends remain constant at zero, while stock splits range up to a maximum value of 5. The descriptive analysis highlights the dynamic nature of Tesla’s stock, with notable extremes and variability in trading patterns, offering crucial insights for predictive modelling.

***Table 2: Tesla Options Data Description***

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Statistics | strike | lastPrice | bid | ask | change | percentChange | volume | openInterest | impliedVolatility |
| count | 3850.0 | 3850.0 | 3850.0 | 3850.0 | 3850.0 | 3850.0 | 3850.0 | 3850.0 | 3850.0 |
| mean | 327.97857142857146 | 89.44875324675326 | 1.5316493506493507 | 1.5620649350649352 | -0.0013558441558441559 | -0.0006304207792207792 | 442.9311688311688 | 0.7467532467532467 | 0.11932046106404887 |
| min | 5.0 | 0.01 | 0.0 | 0.0 | -5.22 | -2.42712 | 1.0 | 0.0 | 1.0000000000000004e-05 |
| 25% | 175.0 | 3.8049999999999997 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 0.0 | 1.0000000000000004e-05 |
| 50% | 310.0 | 47.004999999999995 | 0.0 | 0.0 | 0.0 | 0.0 | 10.0 | 0.0 | 0.007822421875 |
| 75% | 460.0 | 156.03 | 0.0 | 0.0 | 0.0 | 0.0 | 72.0 | 0.0 | 0.12500875 |
| max | 710.0 | 385.2 | 328.2 | 336.15 | 0.0 | 0.0 | 67987.0 | 1153.0 | 15.031250605468747 |
| std | 182.9569516667688 | 100.20591285504777 | 20.130989019294102 | 20.534831867952896 | 0.08412791744424969 | 0.039116580648905615 | 3042.8736345813095 | 24.253661255276082 | 0.3099040119197749 |

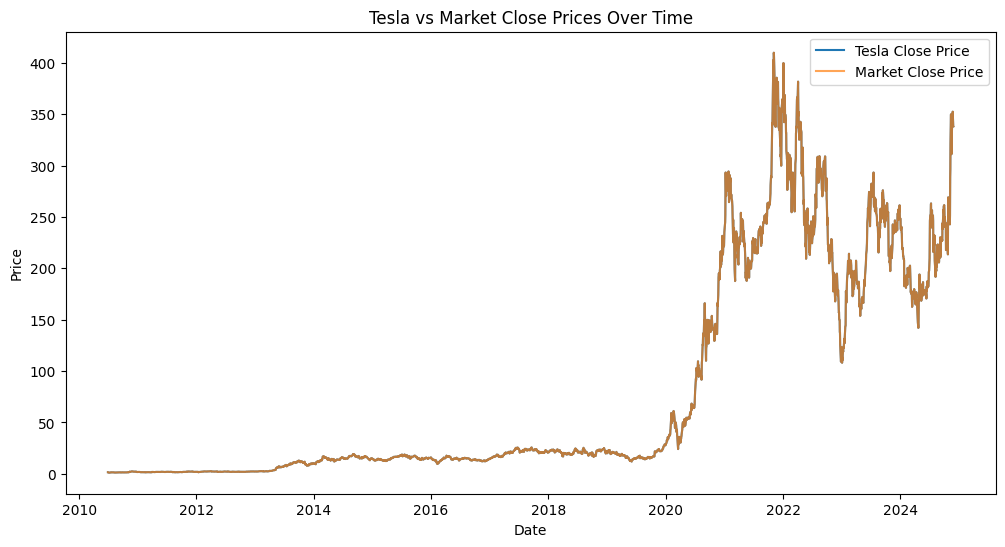
The Tesla options dataset consists of 3850 observations, providing various metrics related to options trading. Key variables include strike price, last price, bid, ask, price change, percentage change, volume, open interest, implied volatility, and expiration dates. The strike price represents the predetermined price at which the option can be exercised. The last price indicates the most recent price at which the option was traded. Bid and Ask are the highest price a buyer is willing to pay and the lowest price a seller is willing to accept, respectively, providing insights into market liquidity and spreads. Change measures the absolute price difference from the previous trading session, while percentage change expresses this variation as a percentage, indicating the option’s relative price movement. Volume reflects the total number of contracts traded during a session, showcasing market activity. Open Interest represents the total number of outstanding option contracts that have not been settled, offering a measure of market interest in a particular option. Implied Volatility is a forward-looking metric derived from option prices, representing the market’s expectations of future price volatility for Tesla stock. Finally, the Expiration Date marks the date when the option contract expires, defining the time frame for trading or exercising the option. The variables collectively provide a comprehensive view of the options market and are crucial for predictive modelling and trading strategy.

The mean strike price is 327.98, with large standard deviation of 182.96, suggesting considerably variability in strike price across the options. The last price of the options has an average of 89.45, with a maximum of 385.20, while bid and ask prices are both low on average (1.53 and 1.56, respectively), indicating that many options are lightly traded. Volume averages 442.93 contracts, with a maximum of 67,987, reflecting sporadic bursts of activity. Open Interest is also highly variable, with a standard deviation of 2042.87, suggesting that most options have limited trading activity. Implied volatility has a mean of 0.12 and a wide range, from near zero to a maximum of 15.03%, which is typical for options on volatile stocks like Tesla. Expiration dates range from late 2024 to early 2027, highlighting the datasets inclusion of both short-term and long-term options. The data’s variability in this metric is essential for analyzing option price movements and building predictive models.

## Data Visualization

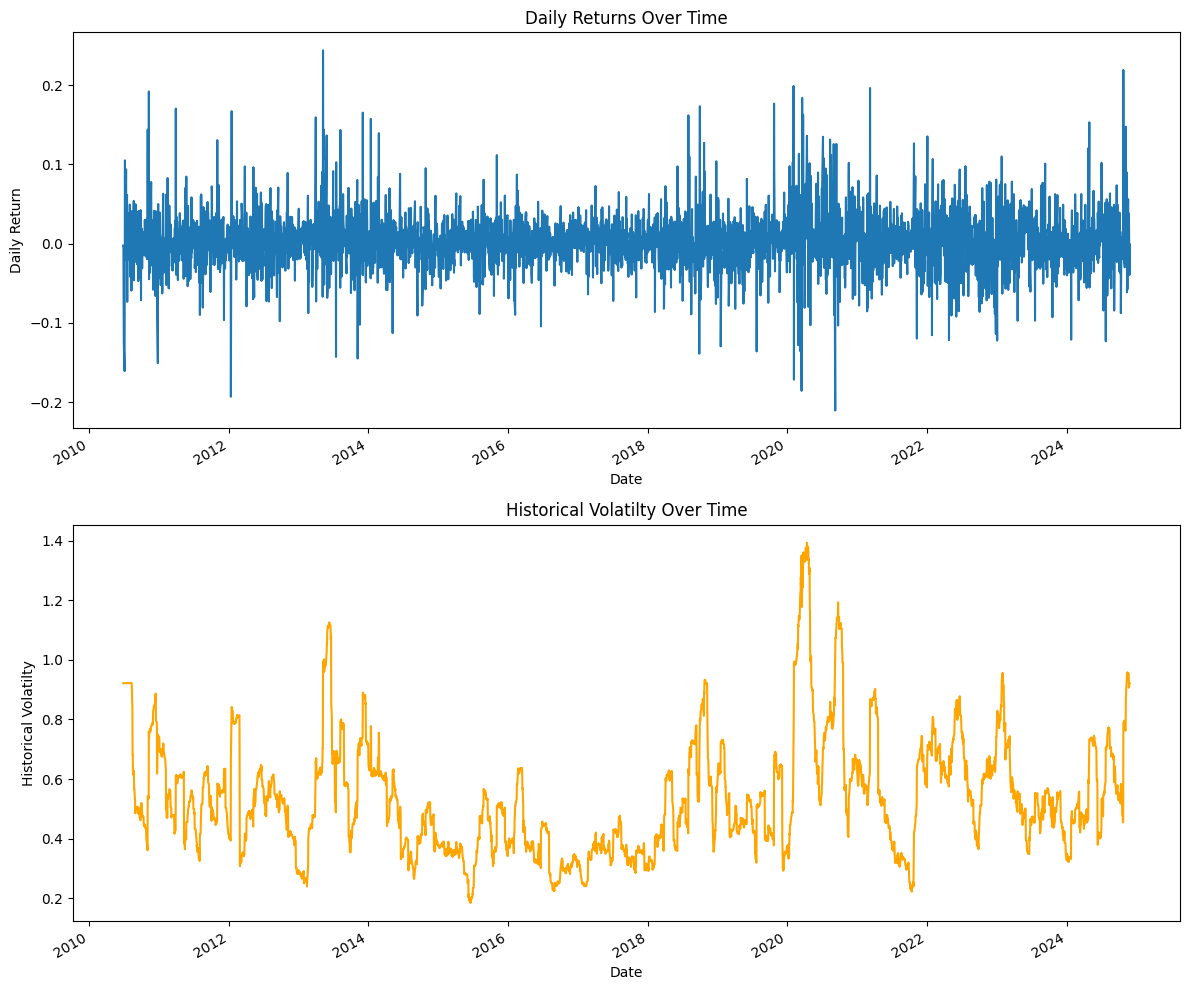
### **Time Series Plot for Key Variables**

**Figure 1: *Time series of Tesla stock closing prices over time***

****

The plot compares the closing prices of Tesla and the general market from 2010 to 2024. Initially, both Tesla and the market showed relatively stable prices from 2010 to around 2019. However, after 2019, there was a noticeable price surge, with Tesla’s price exhibiting significant volatility and higher peaks than the general market. The period post-2019 highlights Tesla’s remarkable growth and fluctuations, underscoring its performance distinction from the broader market.

**Figure 2: *Plotting daily Return and Historical Volatility***

******

The top plot, titled ‘Daily Returns Over Time’ from 2010 to 2024, showcases the daily returns on the y-axis, ranging from -0.2 to 0.2. This plot reveals high volatility in daily returns, with significant spikes around 2012, 2014, and 2020, indicating increased market activity and fluctuations.

The second plot, titled ‘Historical Volatility Over Time,’ also spans from 2010 to 2024, with the y-axis representing historical volatility, ranging from 0.2 to 1.4. It highlights periods of increased volatility, particularly around 2012, 2014, and 2020, with notable pick in 2020. These plots provide a comprehensive view of the asset’s fluctuations and associated risk over time, emphasizing periods where market behavior was particularly unpredictable and volatile.

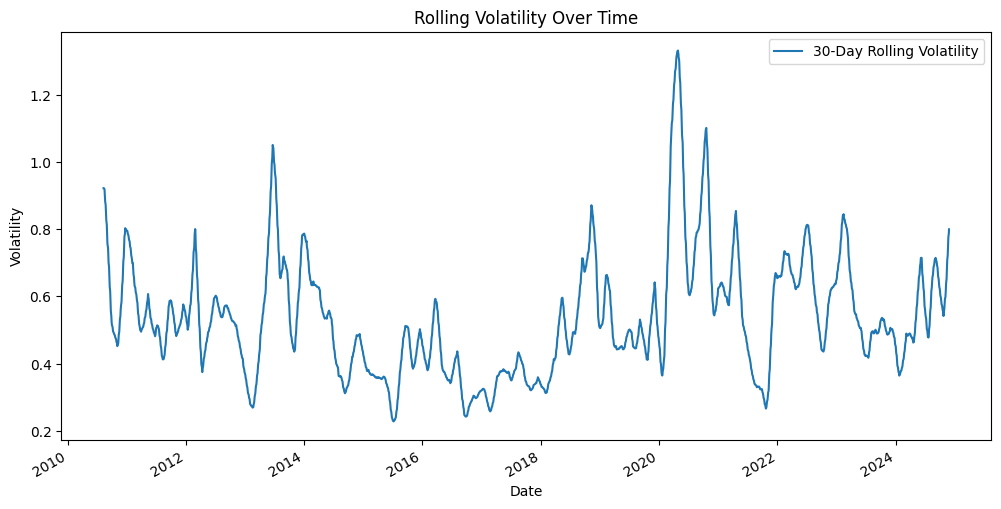
### **Volatility Analysis**

***Table 3: Correlation between volatility and returns***

|  |
| --- |
| Daily Return Historical Volatility Daily Change |
| Daily Return 1.000000 0.048276 0 .641951 |
| Historical Volatility 0.048276 1.000000 0.011556 |
| Daily Change 0.641951 0.011556 1.000000 |

The correlation matrix reveals the relationship between Daily Return, Historical Volatility, and Daily Change. Daily Return has a moderate positive correlation with Daily Change (0.641951), suggesting that daily changes also tend to increase as daily returns increase. However, Daily return has a very weak correlation with Historical Volatility (0.048276), indicating little to no direction relationship between the two variables. Historical volatility also exhibits a weak correlation with Daily change (0.011556), highlighting that changes in volatility do not significantly impact daily changes.

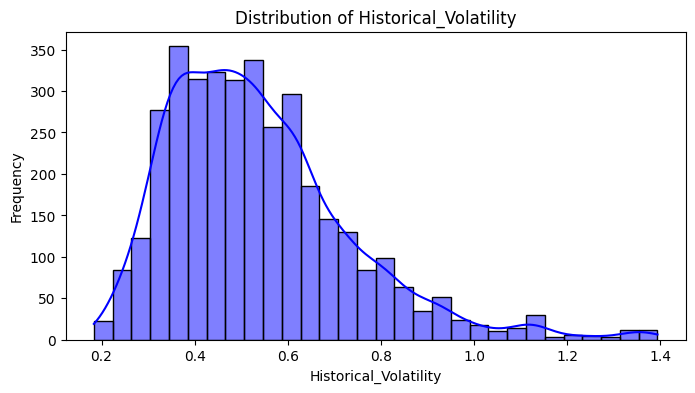
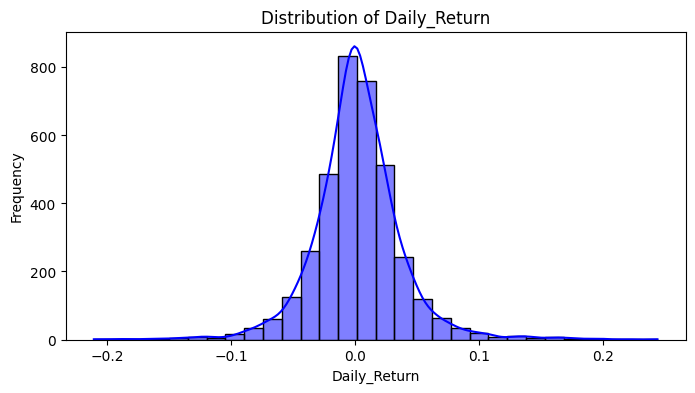
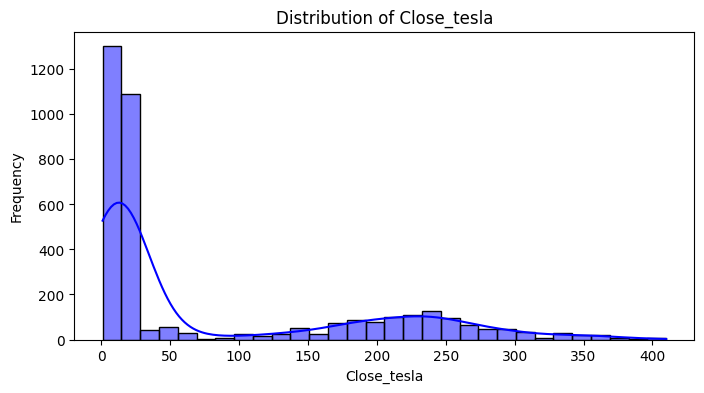
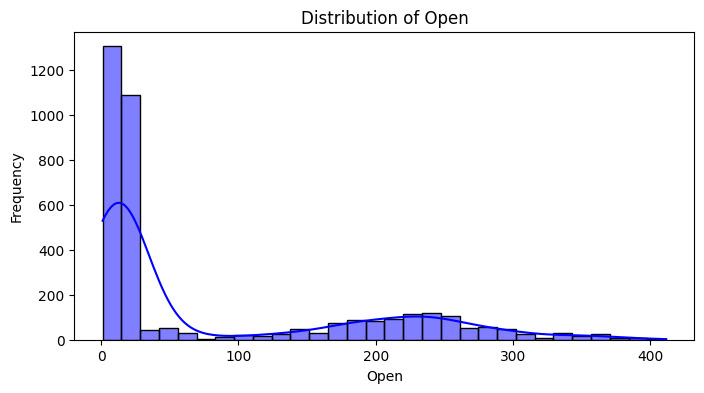
**Figure 3: *rolling volatility plot***

****

The plot illustrates the 30-day rolling volatility of a dataset spanning from 2010 to 2024. The x-axis marks a date range, while the y-axis measures volatility, which fluctuates between 0.2 and 1.4. The plot reveals distinct peaks in volatility around 2014, 2020, and 2022, with the most pronounced spike occurring in 2020, where volatility surpasses 1.2. These peaks indicate periods of heightened market instability or uncertainty.

### **Distribution Analysis**

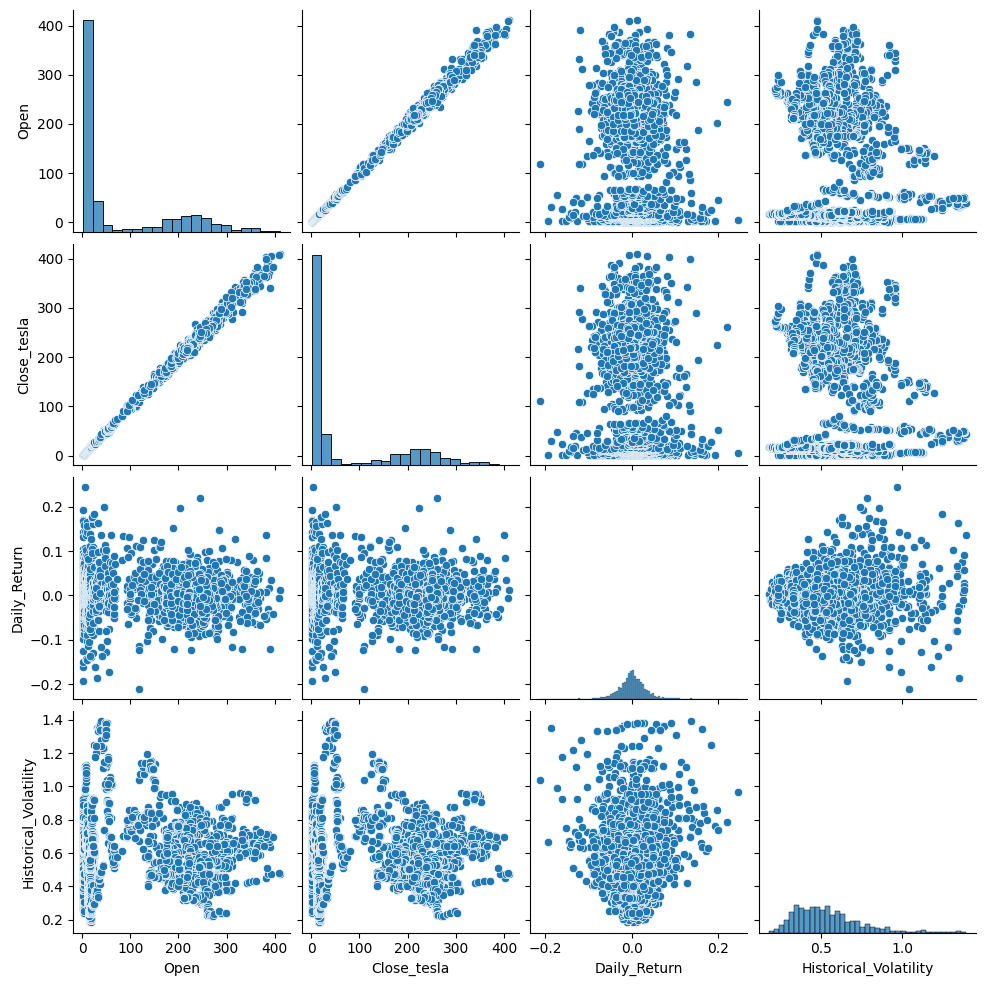
***Figure 4: Distribution Plots***

****

The plots show the distribution of different variables. The Open and Close Prices have relatively similar distributions, mostly between 0 and 30, representing the prices between 2010 and 2019. Another distribution between 180 and 300 represents the periods between 2020 and 2024. The distribution of Daily Returns lies between -0.1 and 0.1, following a normal distribution. Historical volatility also has a normal distribution, with most data points between 0.2 and 1.0.

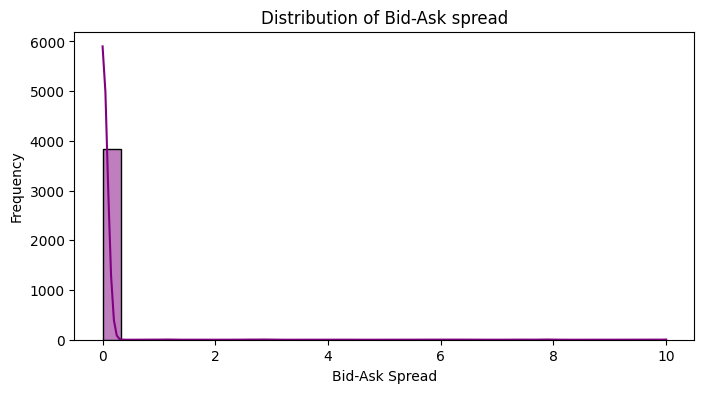
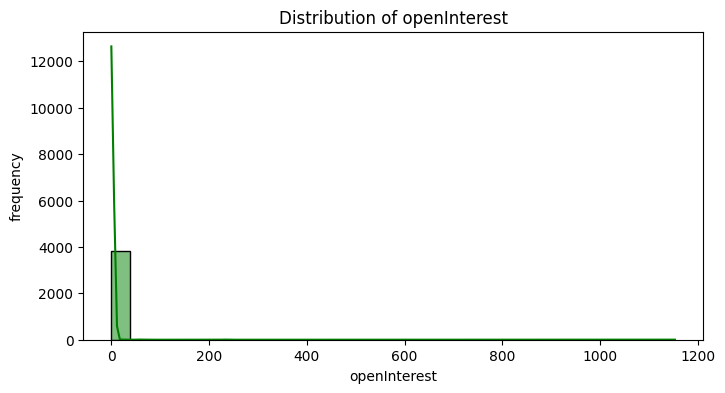
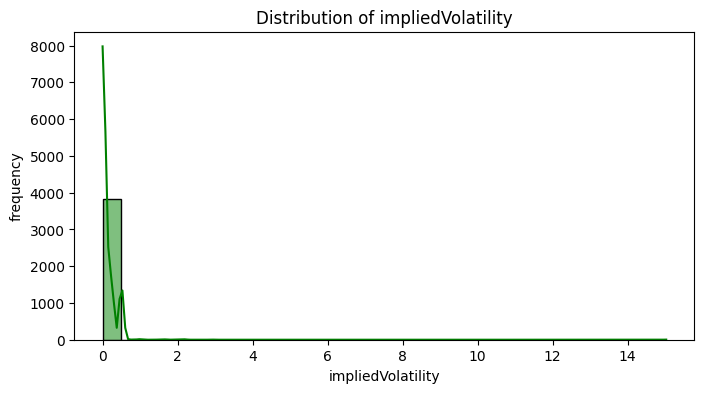
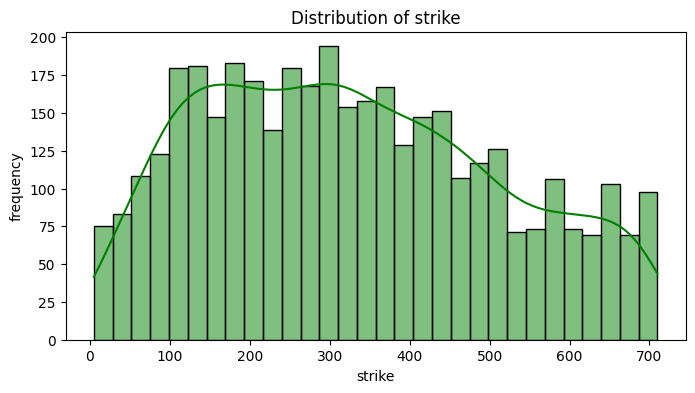
### **Relationship Analysis**

***Figure 5: Pairplot for relationships between key columns***

******

## **Telsa Options Data Exploration**

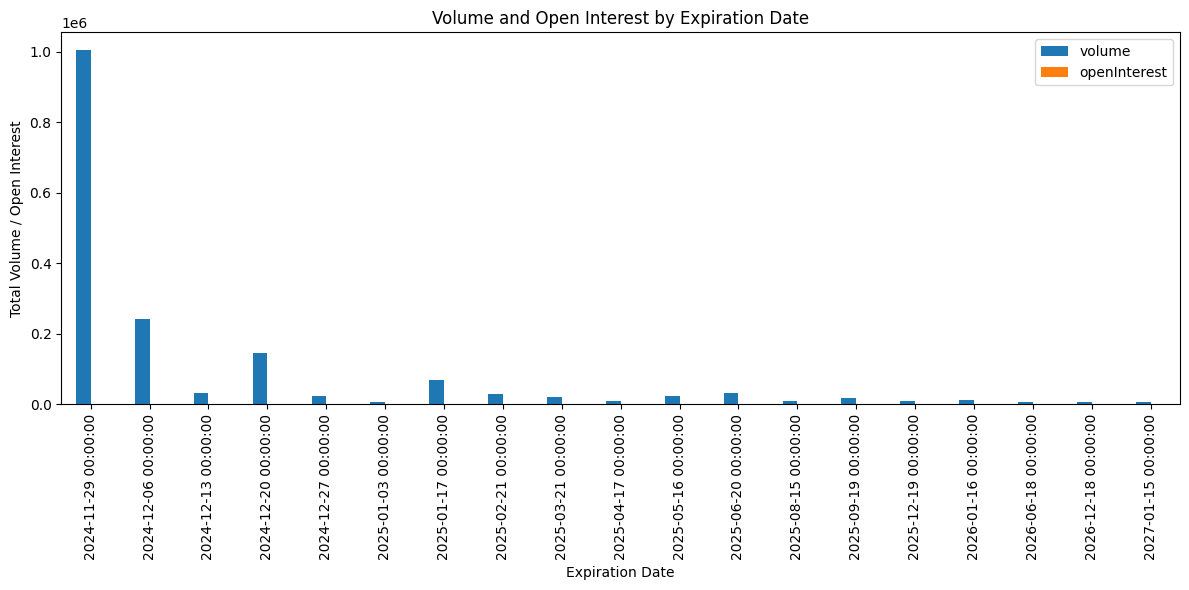
***Figure 6: Distribution Plot***

****

The plots show the distribution of data variables for different options. The strike price is relatively uniform, with its data points uniformly spread within the range. Open interest and the Bid-Ask spread have values mostly between 0 and 0.5, as shown by the distribution plots.

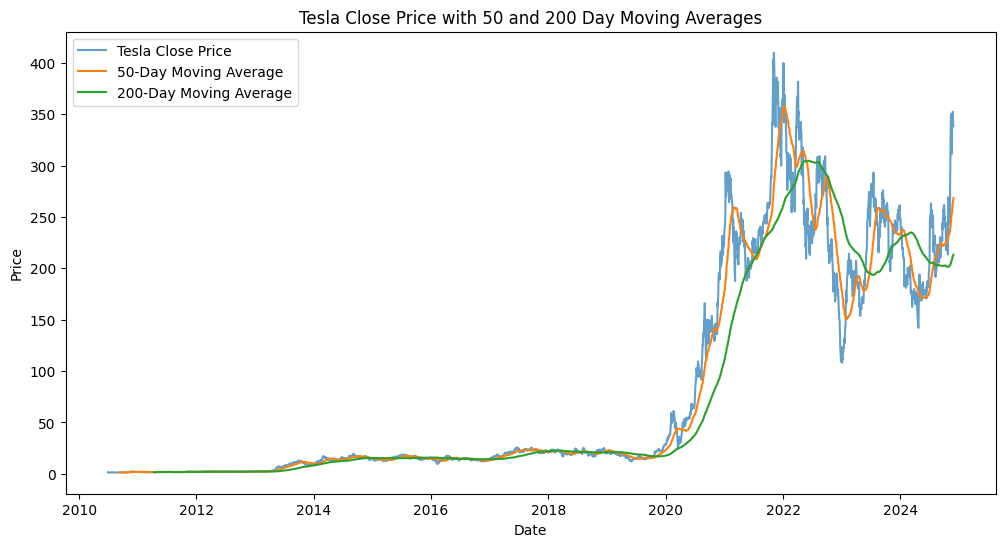
### **Time-based Analysis**

***Figure 7: Aggregate volume and open interest by Expiration Date***

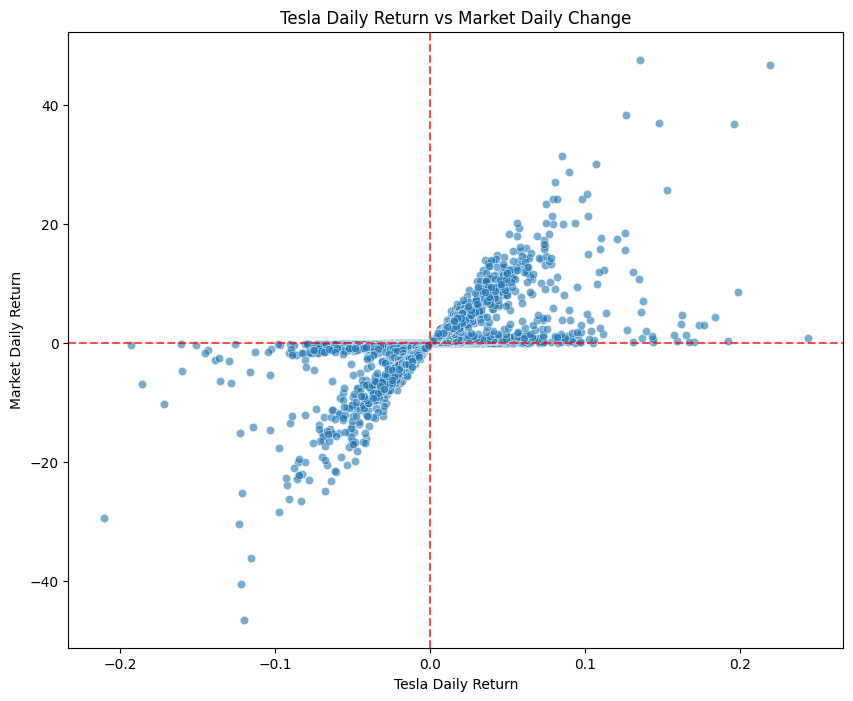


The plot shows a significant concentration of trading activity and open interest as the options expiration date approaches, with less trading activity for expiration dates further from the current date. This pattern indicates that traders focus on near-term rather than long-term options.

***Figure 8: Moving Average of Tesla Close Price***

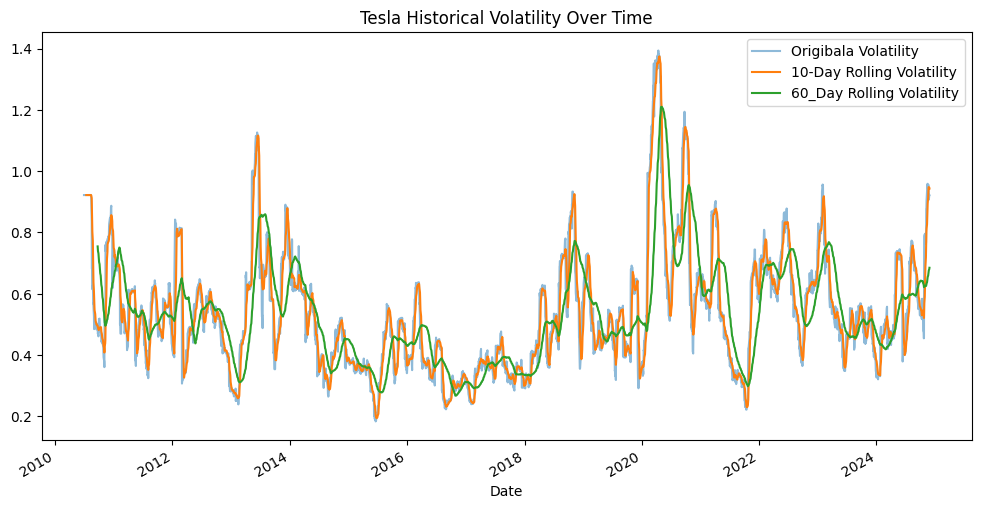
The plot illustrates Tesla's closing stock price and its 50-day and 200-day moving averages from 2010 to 2024. The closing stock price, represented by the blue line, shows a significant rise, particularly after 2020, indicating substantial growth and increased investor interest. The 50-day moving average, shown in orange, is a short-term trend indicator that smoothes out daily price fluctuations and reacts more quickly to price changes. The 200-day moving average, represented by the green line, is a long-term trend indicator, offering a more stable view of the stock's overall trend over a longer period. The crossover points between the closing price and the moving averages are noteworthy and critical for traders. For instance, a "Golden Cross," where the 50-day moving average crosses above the 200-day moving average, suggests a strong upward trend. In contrast, a "Death Cross," where the 50-day moving average crosses below the 200-day moving average, indicates a strong downward trend.

***Figure 9: Tesla Daily Return vs. Market Daily Change***

The plot reveals the relationship between Tesla's daily stock returns and the overall market's daily returns. The x-axis represents Tesla's daily return, ranging from -0.2 to 0.2, while the y-axis indicates the market's daily return, ranging from around -40 to 40. Numerous blue dots populate the plot, each representing a data point of Tesla's performance against the market. The dense cluster of points near the origin (0,0), divided by red dashed lines into four quadrants, indicates a correlation between Tesla's stock returns and market performance. The spread of points demonstrates variability, with some outliers deviating significantly from the central cluster.

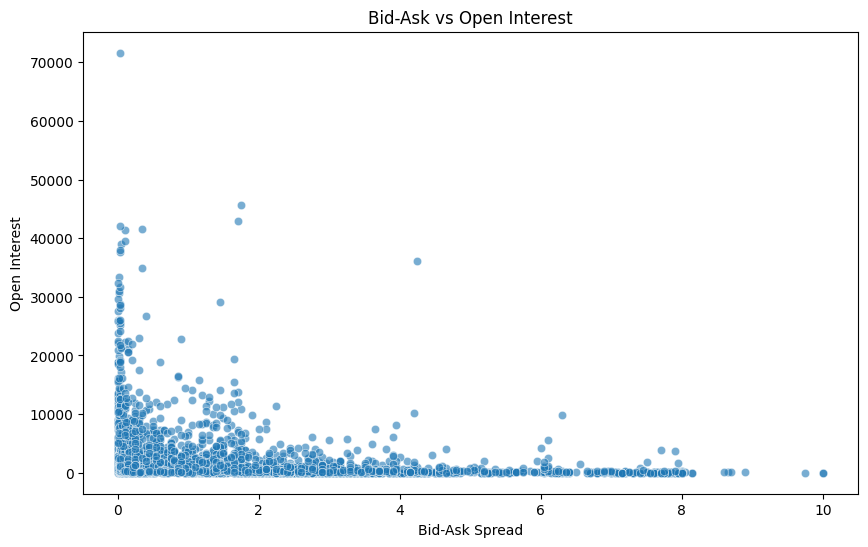
### **Tesla Historical Volatility Over Different Periods**

***Figure 10: Short-term and long-term rolling volatility***



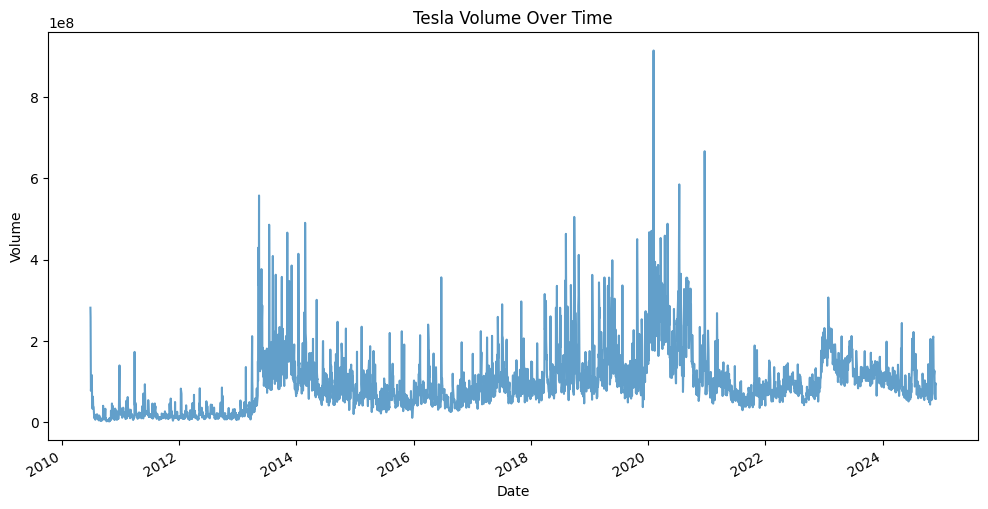
The plot depicts the historical volatility of Tesla’s stock from 2010 to 2025. The x-axis represents the dates, while the y-axis measures volatility, ranging from 0.2 to 1.4. It features three lines: the original volatility (light blue), the 10-rolling volatility (orange), and the 60-day rolling volatility (green). The original volatility shows raw, erratic fluctuations, while the 10-day and 60-day rolling volatilities smooth these out, revealing short-term and long-term trends, respectively. Notable peaks around 2013-2014, 2018-2019, and 2020-2021 indicate periods of higher market uncertainty or significant events impacting Tesla’s stock.

***Figure 11: Relationship Between Bid-Ask Spread and Open Interest***

******

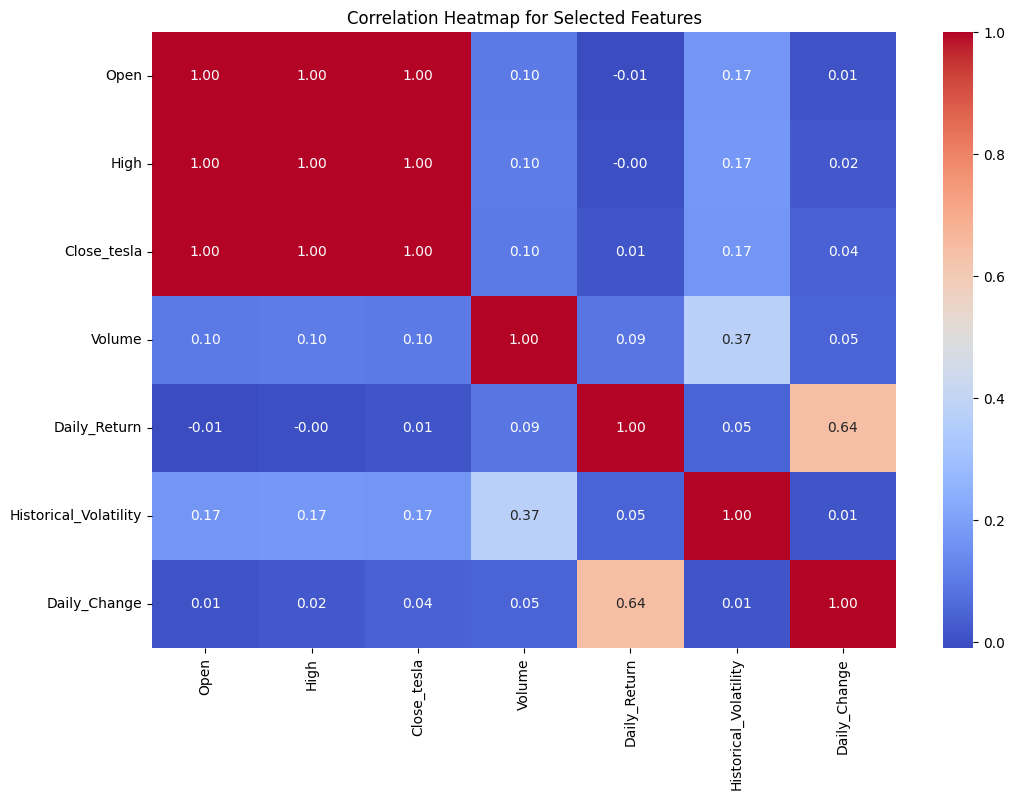
The scatter plot illustrates the relationship between bid-ask spread (x-axis) and open interest (y-axis) in the market. The bid-ask spread ranges from 0 to 10, while open interest ranges from 0 to 70000. The plot reveals a high density of data points at lower bid-ask spreads, particularly between 0 and 2, with most open interest values under 10000. A few outliers have significantly higher open interest values even at lower bid-ask spreads. As the bid-ask spread increases, the number of data points decreases, and open interest values generally tend to be lower. This suggests that markets with higher liquidity (indicated by higher open interest) typically experience bid-ask spreads.

***Figure 12: Tesla Volume Over Time***

******

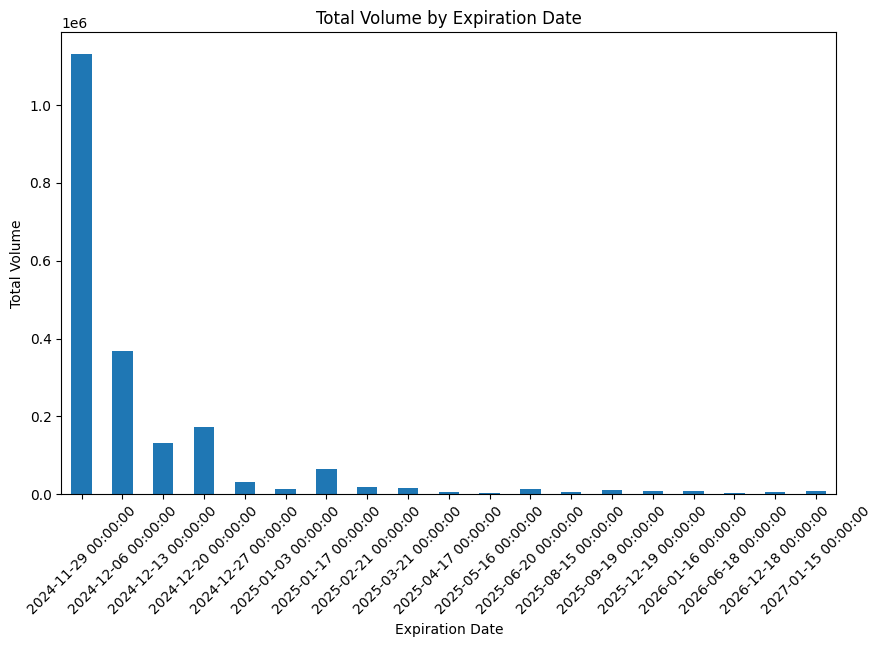
The line plot illustrates the trading volume of Tesla’s stock from 2010 to 2024. The x-axis represents the trading volume, reaching up to 100 million. The plot displays significant fluctuations, with notable peaks from 2013-2014, 2018-2019, and a major spike in 2020. Following 2020, the volume seems to decrease but still exhibits periodic spikes. These fluctuations reflect the varying levels of investor interest and trading activity over the years, often correlating with major events or market conditions affecting Tesla. This visual representation helps understand the trends in trading volume, indicating periods of heightened activity and investor focus on the stock.

***Figure 13: Correlation Heatmap***



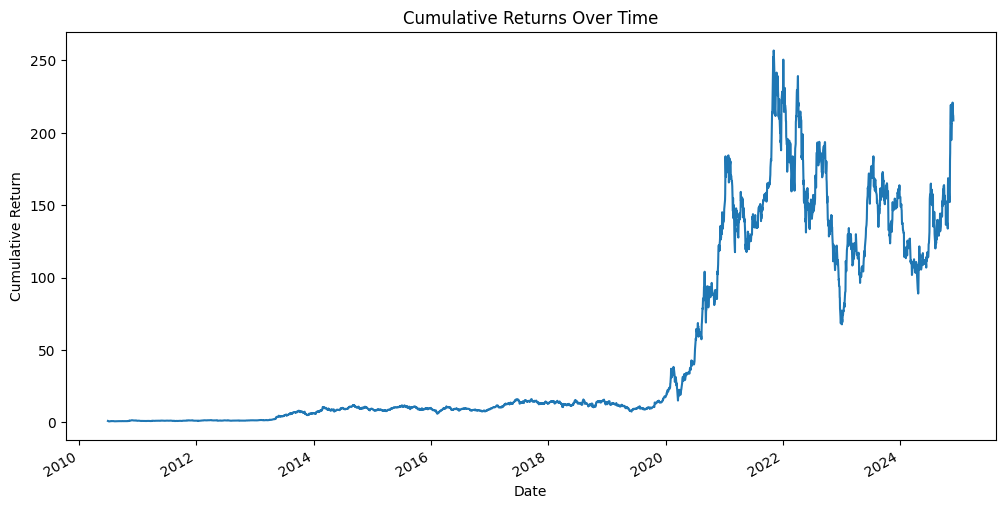
The correlation heatmap for Tesla stock data features offers a clear visual representation of the relationship between variables such as Open, High, Close, Volume, Daily Return, Historical volatility, and daily change. Each cell’s color indicates the strength of correlation, from blue for low or negative correlation to red for high positive correlations. Key observations include perfect correlations among Open, High, and Close, meaning these metrics move together. Volume shows weak correlations with other features, the highest being 0.37 with Historical Volatility, indicating it’s less influenced by price movements or volatility. Daily Return correlates moderately with Daily Change (0.64), suggesting a direct relationship.

***Figure 14: Expiration Date vs. Volume***

******

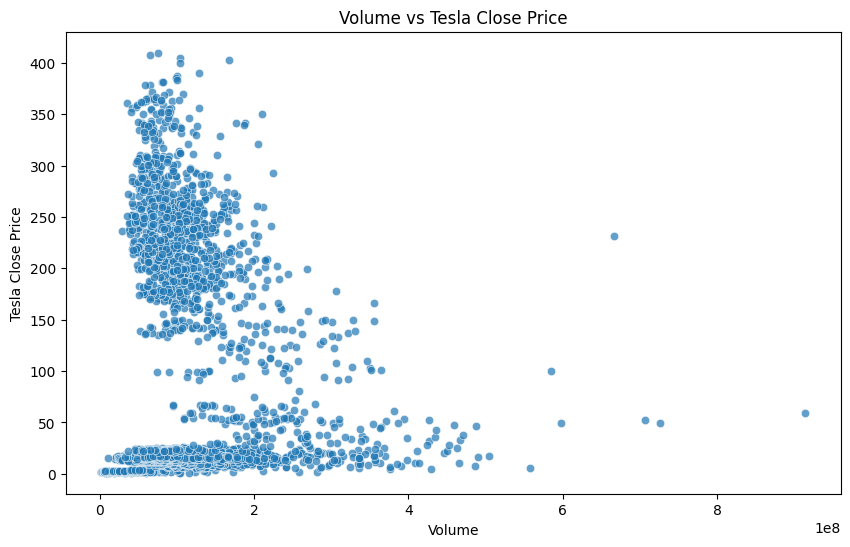
The plot displays the total volume of a certain entity (likely financial contracts or trades) categorized by their expiration dates. The x-axis represents the expiration dates, ranging from November 29, 2024, to January 15, 2027, while the y-axis shows the total volume, with the highest value reaching over 1 million units. The data reveals a significant volume concentration around the earliest expiration dates, particularly on November 29, 2024, with the highest volume by a large margin. Subsequent dates show a steep decline in volume, with a few minor peaks, but overall, the volume diminishes considerably as the expiration dates extend further into the future. This suggests that most of the activity or interest is focused on the near-term expirations, with much less activity for longer-term expirations. This pattern could be relevant for understanding market behavior, investor preferences, or the liquidity of the instruments being analysed.

***Figure 15: Cumulative Returns Over Time***

******

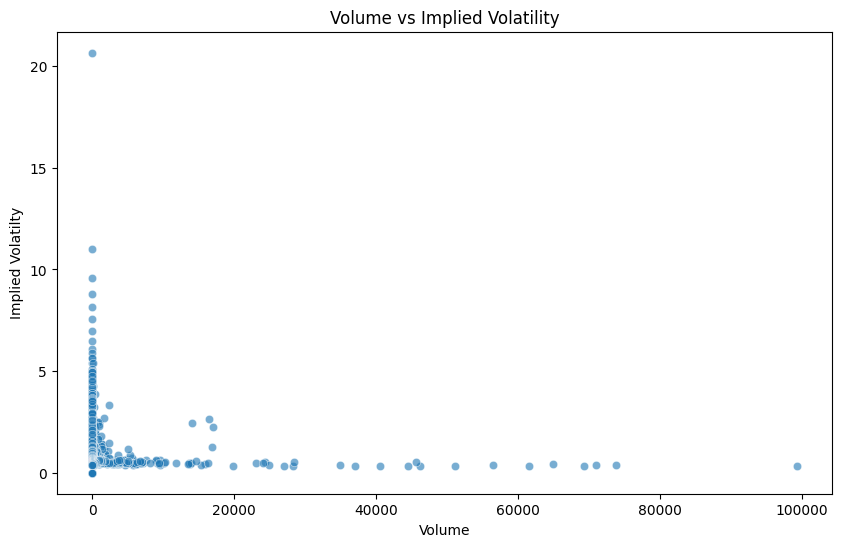
The plot shows the cumulative returns over time from 2010 to 2024. The x-axis represents the date, ranging from 2010 to 2024, while the y-axis represents the cumulative return, which ranges from 0 to 250. The plot indicates that the cumulative returns remained relatively flat from 2010 to around 2018, with minor fluctuations. However, starting in late 2018 or early 2019, there is a significant upward trend in cumulative returns, peaking around 2021. After reaching the peak, the returns exhibit high volatility with several sharp declines and recoveries. The most recent data point shows another upward trend, approaching the previous peak levels. This plot is interesting as it highlights periods of significant growth and volatility in cumulative returns, which could be relevant for understanding market behavior or investment performance over the specified period.

***Figure 16: Volume vs. Tesla Close Price***

******

The scatter plot explores the relationship between Tesla’s trading volume and its closing stock price. The x-axis represents the trading volume, which ranges from 0 to approximately 900 million, while the y-axis represents the closing price, ranging from 0 to 400. The dense cluster of data points at lower volumes and prices indicates that most trading activity happens at lower volumes and prices. As the trading volume increases, the closing price tends to be lower, with a few outliers present at higher volumes. This trend suggests that higher trading volumes are often associated with lower closing prices for Tesla’s stock. The plot is insightful for investors and analysts as it highlights Tesla's stock's trading behavior and price dynamics.

***Figure 17: Volume vs. Implied Volatility***

******

The scatter plot explores the relationship between trade volume and the implied volatility of an asset. On the x-axis, we have the volume of trades ranging from 0 to 100,000, and on the y-axis, we have the implied volatility, which ranges from 0 to 20. Most data points are concentrated near the origin, indicating that most trades occur at low volumes and exhibit low implied volatility. There are a few notable outliers with higher implied volatility at low volumes and some scattered points showing higher volumes but low implied volatility. This distribution suggests that higher volumes generally correspond with lower implied volatility, while instances of high volatility tend to occur at lower trade volumes. This plot can provide valuable insights for traders and analysts when assessing market behavior and risk.

## **Machine Learning and Random Forest Classifier**

Machine learning (ML) is a subset of artificial intelligence that enables computers to learn from data and make predictions or decisions without being explicitly programmed. Machine learning models can uncover patterns and relationships by leveraging historical datasets, providing actionable insights for various applications. A prominent type of machine learning model is the Random Forest Classifier, an ensemble-based algorithm that operates by constructing multiple decision trees during training. Each tree contributes to the model’s final prediction through a majority voting mechanism. This approach enhances the accuracy of predictions and mitigates overfitting, making Random Forest robust and effective for classification tasks.

## **Data Preprocessing and Target Definition**

Effective machine learning begins with preprocessing the dataset to ensure it is clean and relevant. In this analysis, the target variable, **price movement**, was defined as a binary classification task, indicating whether the option’s last price increased from the previous session. This was achieved using the. diff() method and converting the result into binary values (1 for an increase, 0 otherwise). Missing values for the target variable were dropped to maintain data integrity. Additionally, the **inTheMoney** feature was converted into binary format (1 for true, 0 for false), ensuring consistency in categorical data representation. Key features such as **strike, bid, ask, volume, openInterest, impliedVolatility**, and **inTheMoney** were selected for model training based on their relevance to options pricing and price movement.

### **Train-Test Split**

The dataset was split into training and testing sets using an 80-20 split ratio to validate the model’s performance. This ensures that the model is trained on a majority of the data while being evaluated on unseen data to assess its generalization capability. The training set comprised 3,080 records, while the test set contained 770 records. The consistency in feature dimensions across training and testing sets (7feature) further validated the data preparation process.

### **Feature Scaling**

Before training the Random Forest model, feature scaling was applied using the **StandardScaler**. Scaling standardizes the features by removing the mean and scaling them to unit variance, ensuring that no feature disproportionately influences the model due to its scale. The training set was fitted to the scaler, and the test was transformed to ensure consistent scaling both datasets.

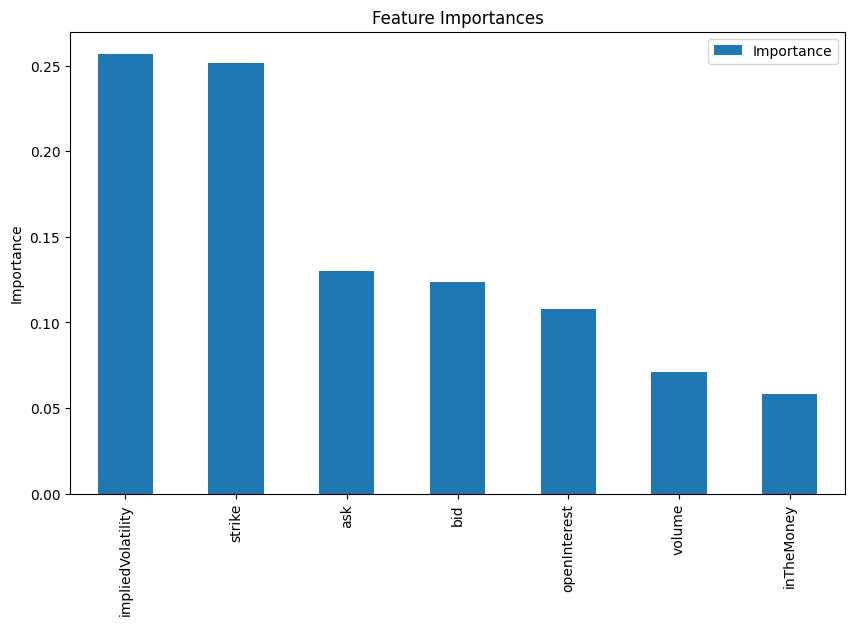
### **Training the Random Forest Classifier**

The Random Classifier was trained using the preprocessed and scaled training data. Parameters such as 100 decision trees, a maximum tree depth of 10, and a balanced class weight were chosen to optimize the model. The model achieved an accuracy of 85% on the test set, indicating reliable predictive capability.

## **Model Evaluation**

The model's performance was evaluated using various metrics. The classification report showed precision, recall, and F1 scores for both classes, highlighting a balanced performance. The confusion matrix revealed that the model correctly classified 372 price decreases (class 0) and 284 price increases (class 1). While some misclassifications occurred (52 false positives and 62 false negatives), the overall metrics, including an F1-score of 0.85 for both classes, demonstrated the model’s reliability in predicting option price movements.

**Feature Importance**



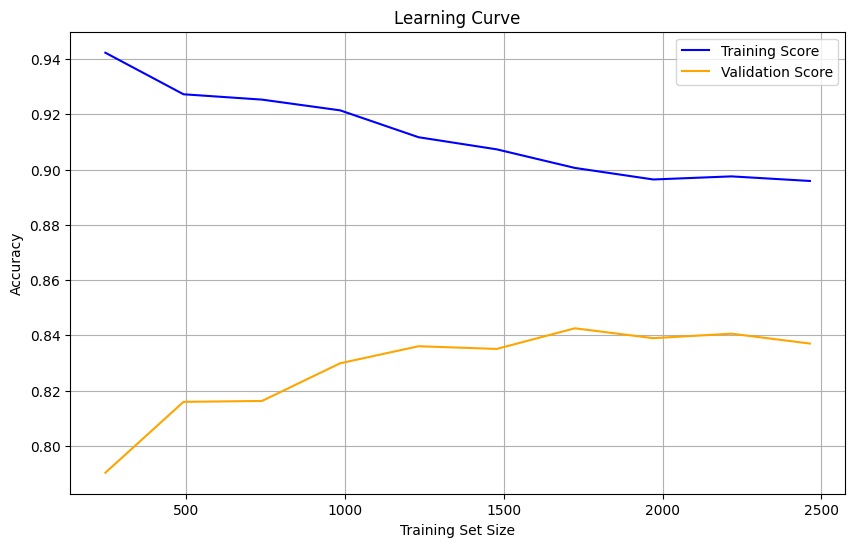
The bar chart visually demonstrates which features significantly impact the model’s predictions. The y-axis denotes the importance score, ranging from 0.00 to 0.25, while the x-axis lists the features: **implied Volatility, strike, ask, bid, open Interest, volume**, and **inTheMoney**. Notably, **implied volatility** and **strike** emerge as the most critical features, each with an importance score slightly exceeding 0.25. Following these, **ask** and **bid** are important, with scores around 0.12 and 0.11, respectively. **OpenInterest** has a slightly lower significance, just under 0.10. Meanwhile**, volume** and **inTheMoney** appear to be the least influential, with scores approximately 0.05 and 0.03. This analysis aids in feature selection, emphasizing the features that most enhance the model’s predictive accuracy.

## **Hyperparameter Tuning**

### **GridSearchCV**

Hyperparameter tuning is a crucial step in optimizing machine learning models, as it involves identifying the best combination of parameters to enhance model performance. In this analysis, GridSearchCV was employed to systematically explore different configurations of hyperparameters for the Random Forest Classifier. The search focused on parameters such as the number of trees *(n-estimators),* the maximum depth of trees *(max-depth*), the minimum number of samples required to split a node (*min\_samples\_split*), and the minimum number of samples required to be at a leaf node (*min\_samples\_leaf*). GridSearchCV iteratively trained and evaluated the model using 3-fold cross-validation for each combination by defining a grid of potential values for these parameters. This exhaustive search identified the optimal configuration as *max\_depth=10*, *min\_samples\_split=10*, *min\_sample\_leaf=1,* and *n\_estimators=100*, which resulted in the highest cross-validated accuracy. To further ensure the robustness of the tuned model, a separate 5-fold cross-validation was performed using the optimized parameters. The results indicated a mean accuracy of 83.70% with a standard deviation of ±1.24%%, demonstrating that the model generalizes well across different subsets of the training data.

### **Learning Curve**

****

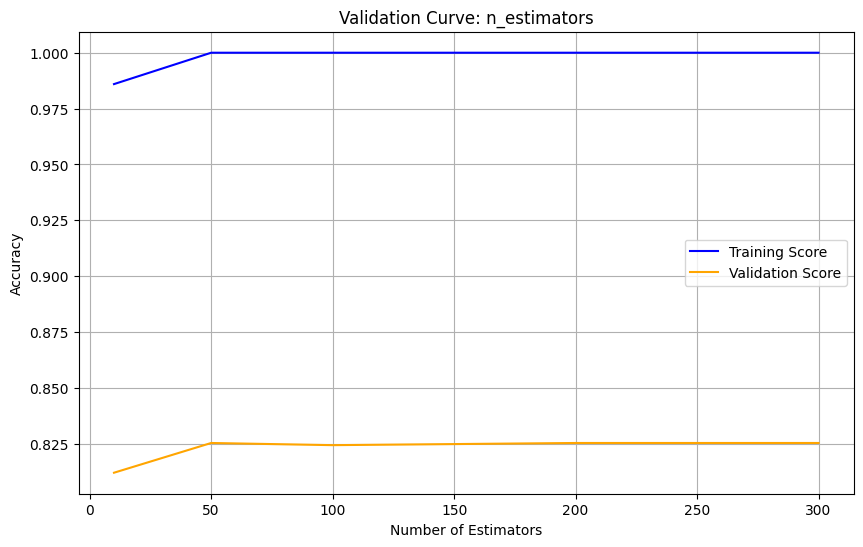
The image shows a learning curve plot for a machine learning model, specifically a Random Forest model, as indicated by the variables *rf\_model*. The plot displays the relationship between the training set size and the accuracy of the model on both the training and validation datasets. The x-axis represents the training set size, ranging from approximately 200 to 2500 samples, while the y-axis represents the accuracy, ranging from 0.80 to 0.94.

The blue line represents the training score, which shows a decreasing trend as the training set size increases. Initially, the training accuracy is very high (around 0.94) for small training set sizes, but it gradually decreases and stabilizes to around 0.90 as the training set sizes increase. This indicates that the model initially overfits the training data when the training set is small.

The orange line represents the validation score, which initially shows an increasing trend and stabilizes. The validation accuracy starts at around 0.80 and increases to approximately 0.84 as the training set size increases. This suggests that the model’s performance on unseen data improves as more training data is provided. Still, it eventually plateaus, indicating that adding more training data beyond a certain point does not significantly improve the model’s performance.

Overall, the plot indicates that the model benefits from more training data up to a certain point, after which the gains in validation accuracy diminish. The gap between the training and validation scores suggests that there is still some overfitting, but it decreases as the training set size increases.

### **Validation Curve**

****

The validation curve plot illustrates the impact of the number of estimators (*n\_estimators*) on the accuracy of a RandomForestClassifier. The x-axis represents the number of estimators, ranging from 10 to 300, and the y-axis shows the accuracy, which varies between 0.8 and 1.0. The blue line depicts the training score, while the orange line represents the validation score.

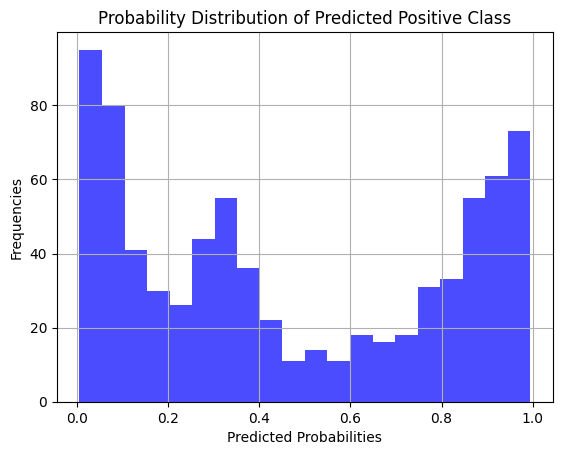
As the number of estimators increases, the training accuracy (blue line) starts at a high value of around 0.99. It quickly reaches 1.0, indicating that the model perfectly fits the training data with more estimators. However, the validation accuracy (orange line) begins at approximately 0.82 and remains relatively flat as the number of estimators increases, suggesting that the model’s performance on unseen data does significantly improve with more estimators. This pattern indicates potential overfitting, where the model performs exceptionally well on the training set but does not generalize as well to the validation set. Thus, increasing the number of estimators beyond a certain point may not yield substantial benefits in validation accuracy.

**RandomSearchCV**

Feature selection, model evaluation, and hyperparameter tuning are vital to optimizing machine learning performance. Using SelectFromModel, a feature selection method was applied to reduce the dimensionality of the dataset by retaining only the most important features as determined by the trained Random Forest model. While this process simplifies the model, the reduced feature set decreased accuracy to 73%, suggesting that some eliminated features may have contributed valuable information. The Out-of-Bag (OOB) score of 82.60% also provided an unbiased estimate of the model’s generalization performance, leveraging the built-in OOB sampling of Random Forests. RandomizedSearchCV was used for faster hyperparameter tuning to enhance the model's efficiency. Unlike GridSearchCV, this method randomly samples from a defined parameter distribution, significantly reducing computational costs while maintaining effectiveness. The optimal parameters identified through this search included a maximum tree depth of 13, a minimum sample split ratio of 0.033, a minimum leaf sample ratio of 0.11, and 141 trees. This tuning approach streamlined the model configuration while preserving robust performance.

## 

## **Analyzing Predicted Probabilities**



The histogram shows the probability distribution of a random forest model's predicted positive class probabilities. The x-axis represents the predicted probabilities, which range from 0 to 1, and the y-axis represents the frequencies of these probabilities. The histogram is divided into 20 bins, showing the count of predictions within each probability range. The plot reveals high frequencies of predictions near 0 to 1, indicating that the model is confident in its predictions for many instances. However, there are also notable frequencies in the mid-range probabilities, suggesting some uncertainty in the model’s predictions for certain instances. This distribution provides insights into the model’s confidence levels and highlights areas where the model may need further improvement.

Evaluating the model’s reliability and performance across different datasets is crucial to ensure its robustness and prevent overfitting. A confidence interval of the model’s accuracy was computed to quantify the uncertainty around its performance. The Random Forest model’s cross-validation accuracy was 83.70% ± 1.73% at a 95% confidence level, indicating consistent and reliable predictions within this range. Furthermore, a comparison of the model’s performance on the training and test datasets revealed a training accuracy of 88.77% and a test accuracy of 84.55%. While the training accuracy slightly exceeds the test accuracy, the difference is minimal, suggesting the models generalizes well without significant overfitting. The mean squared error (MSE) further supported this conclusion, with training MSE at 0.112 and the test MSE at 0.155. These results demonstrate that the model effectively balances fitting the training data and maintaining performance on unseen data, indicating strong predictive capabilities.

## **Long-Short-Term Memory (LSTM) Model**

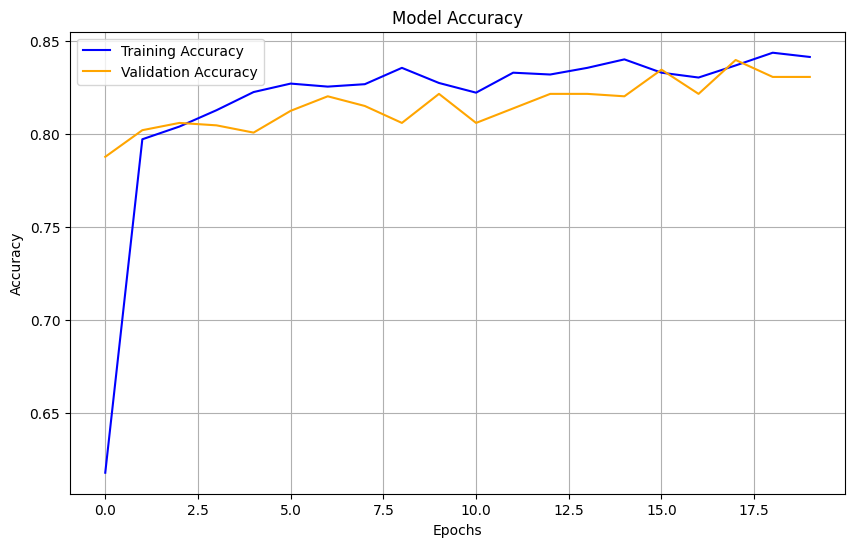
Deep learning models, particularly Long-Short Term Memory (LSTM) networks, are well-suited for sequential data tasks because they capture long-term dependencies. LSTM is a recurrent neural network (RNN) designed to overcome the vanishing gradient problem, allowing it to learn and retain patterns across longer sequence data. In this project, an LSTM model was implemented to predict binary outcomes based on sequential data, leveraging the temporal nature of the features. This approach is ideal for datasets such as time series data, stock prices, and options data, where the order of observations significantly influences the predictions.

The data preparation process included scaling features using MinMaxScaler to ensure they were normalized between 0 and 1, a crucial step for improving convergence in neural networks. Sequential data inputs were created using a sliding window approach, where a sequence length of 10 was used to construct training samples and corresponding target labels. The data was then split into training and testing datasets, with the training set used to learn model parameters and the testing set reserved for evaluating generalization.

The LSTM model architecture comprised two LSTM layers with 50 units each, the first configured return sequences to feed the second layer. Dropout layers were added after each LSTM to mitigate overfitting by randomly deactivating a portion of neurons during training. Finally, a dense layer with a sigmoid activation function outputs a probability for the binary classification task. The model was compiled using the Adam optimizer and binary cross-entropy loss, optimized for classification accuracy.

The model was trained over 20 epochs with a batch size of 32, achieving steady improvements in accuracy and loss. The initial epoch yielded an accuracy of 61.78% on the training set and 78.78% on the validation set, demonstrating the model’s capacity to learn quickly. As training progressed, accuracy improved significantly, reaching a final validation accuracy of 83.07%. The model exhibited consistent loss reduction, with the validation loss decreasing from 0.58 in the first epoch to 0.38 in later epochs, indicative of effective learning.

### **Model Evaluation**

****

The plot shows the accuracy of a machine learning model over several training epochs, with the training accuracy represented by the blue line and the validation accuracy by the orange line. Both lines demonstrate significant improvements in accuracy over time, indicating effective learning. The training accuracy increases rapidly and stabilizes near 0.85, while the validation accuracy, despite some fluctuations, stabilizes near 0.80. This close alignment between training and validation accuracy suggests that the model is learning well from the training data, and generalizing accuracy indicates that the model is learning well from the training data and generalizing effectively to unseen data without significant overfitting.

## **Model Comparison: RandomForestClassifier vs LSTM**

The comparison between the RandomForestClassifier and the LSTM model reveals distinct advantages and trade-offs based on the dataset and the problem’s sequential nature. The RandomForestClassifier, a traditional machine learning model, achieved a training accuracy of 88.77% and a test accuracy of 84.55%, showing strong generalization and robustness to overfitting. However, closer inspection indicated slight overfitting, as the gap between training and testing accuracy was notable, and the model’s performance plateaued when more features were introduced. On the other hand, the LSTM model excelled in capturing the sequential dependencies in the data, achieving a validation accuracy of 83.07% with consistent improvement across epochs. While the initial performance of the LSTM was lower than that of the RandomForestClassifier, it demonstrated superior generalization and adaptability to the problem’s time-series nature. Additionally, the inclusion of dropout layers in the LSTM architecture effectively minimized overfitting, ensuring the model's robustness on unseen data. These findings highlight the LSTM model as the better-performing approach for this task, particularly given the sequential dependencies in the dataset that traditional machine learning models like RandomForestClassifier could not adequately capture.

## **Model Back Testing**

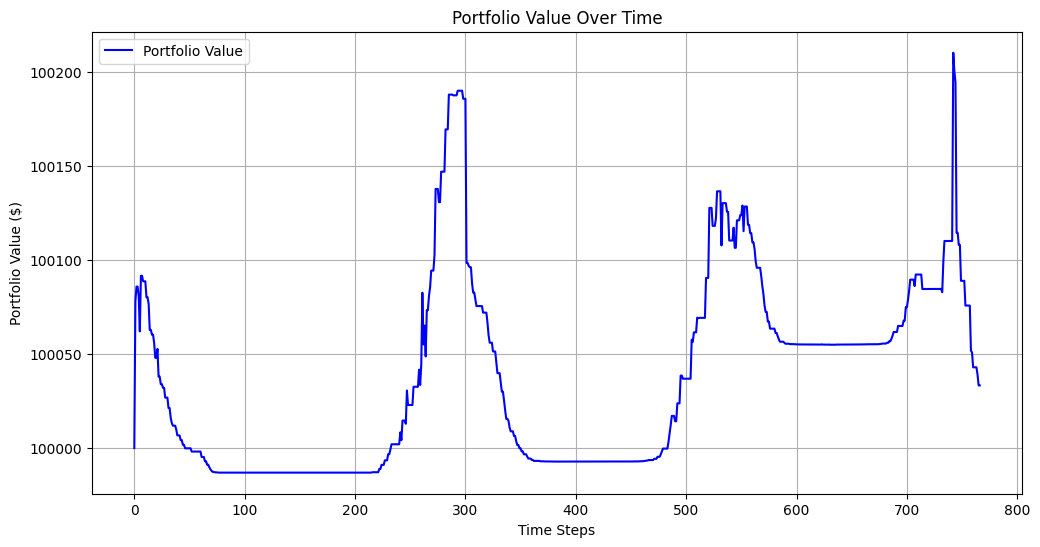
The predictive model's backtesting provides a practical evaluation of its performance in a simulated trading scenario, offering insights into its real-world application potential. Starting with an initial portfolio balance of $100,000, the strategy employed predictions from the model to determine entry and exit points for trades, mimicking a simple buy-and-sell trading approach based on forecasted price movements.

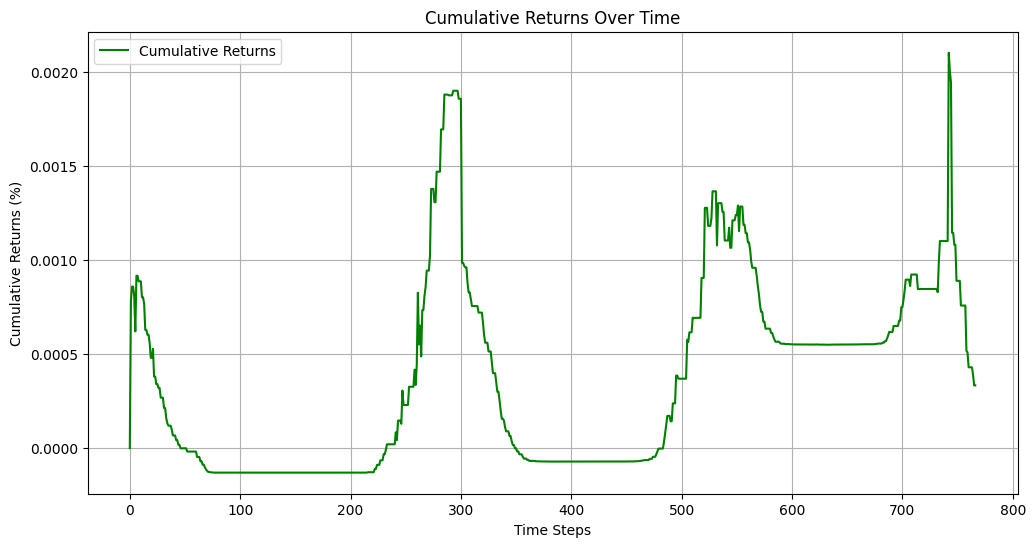
The results indicate a final portfolio value of $100,033.32, reflecting a marginal total return of 0.03%. This modest performance suggests that the model's predictive power, while consistent in avoiding significant losses, is limited in its ability to capture substantial profit opportunities. The maximum drawdown was negligible at -0.00, demonstrating that the model's strategy avoided significant declines in portfolio value, which is a positive indicator of risk management.

The Sharpe Ratio, a measure of risk-adjusted returns, was calculated to be 0.08. A low Sharpe Ratio implies that the model's excess returns over the risk-free rate are not substantial relative to those returns' variability (risk). This suggests that while the model minimizes risk, it struggles to generate significant profitability from its predictions.

The backtesting results highlight the model’s reliability in preserving capital and its limitations in producing meaningful returns. Enhancements such as refining the predictive model, incorporating transaction costs, and exploring more advanced trading strategies could improve the strategy's viability in real-world applications.

### **Visualize the Backtesting Results**





The plot illustrates the fluctuations in the value of a portfolio across different time steps. The x-axis represents the time steps, while the y-axis shows the portfolio value in dollars. The blue line indicates the portfolio's value, which starts slightly above $100,000 and experiences several notable fluctuations. Initially, the value decreases, hitting a low point around the 100th time step. It then rises sharply, peaking around the 300th time step at just over $10,200. Following this peak, the value drops again around the 400th time step before another rise and fall occur, with another peak around the 700th time step. The plot effectively demonstrates the dynamic and volatile nature of the portfolio's value over time, highlighting the investment performance and associated risks.

The plot illustrates the cumulative returns of a trading strategy over a series of time steps, represented by the green line. The x-axis denotes the time steps, while the y-axis indicates the cumulative returns in percentage. Initially, the cumulative returns show minor fluctuations around zero, suggesting a stable performance. However, around the 200th time step, there is a notable increase, peaking around the 300th time step, followed by a sharp decline. Another significant growth period starts around the 500th time step, with some fluctuations and a peak around the 700th time step, followed by another decline. This plot visually represents the trading strategy's performance over time, highlighting periods of profitability and drawdowns, thus providing insights into the model's ability to generate returns and manage risk.  
**Live Trading Strategy/Trading BOT**

The live trading model is designed to automatically execute trades based on predictions made by a machine learning model, specifically an LSTM, which forecasts the movement of stock prices for Tesla (TSLA). The system operates in real-time, continuously fetching the latest market data at regular intervals. It preprocesses this data to make it suitable for the model, which then predicts whether the price will rise or fall. Based on these predictions, the model executes trades with a predefined risk management strategy. If the model predicts a price increase and no position is currently held, it triggers a buy order, and similarly, it sells when the model predicts a price decrease.

The system manages the portfolio balance by determining the appropriate position size for each trade based on the available capital and the risk per trade. A key feature of the model is its risk management, incorporating stop-loss and take-profit mechanisms. If the trade moves unfavorably by a set percentage, the stop-loss triggers a sell order to minimize losses. Conversely, if the price reaches a predetermined profit level, the take-profit condition is met, locking in gains by selling the position. This automated approach allows the model to continuously monitor and adjust the portfolio, making decisions based on real-time data and pre-programmed trading logic while adhering to a specific risk tolerance.

Conclusion

In conclusion, the work presented outlines the development, evaluation, and deployment of a machine learning-driven trading strategy utilizing LSTM models for real-time market predictions. Through backtesting, the model demonstrated a modest return with effective risk management, showing its potential for live trading. The backtesting results, while indicating a slight positive return and minimal drawdown, suggest that the model's predictive accuracy and decision-making processes can be improved further for better profitability. The live trading system integrates real-time data fetching, trade execution, and a robust risk management framework, allowing for automated trading based on real-time predictions. Despite initial challenges, such as overfitting observed in traditional models like Random Forest, the LSTM model proved to be a more effective choice for this problem. Overall, the work illustrates the potential of machine learning in financial markets, while also highlighting the importance of continuous refinement and real-time risk management for successful trading strategies.