MATH 400- 01
FINAL PROJECT
Prof. Jacob Kotas

Project Team: Akshat Sohal
Gurpreet Singh

Application Background: This project delves into the predictive analysis of stock prices for two prominent technology giants, Apple and Google. Using linear, quadratic (polynomial of degree 2), and logarithmic regression models, the application forecasts the future stock prices of these companies and compares them with their historical stock performance. The fascinating aspect of this study is the focus on two major players in the tech industry, whose stock movements are keenly observed by investors and market analysts. This analysis not only provides a graphical representation of how well each model predicts the stock prices but also offers a quantitative comparison using R-squared (R^2) and Mean Squared Error (MSE) values. The project's intrigue lies in its ability to measure the accuracy of these mathematical models against real-world data, providing insights into the reliability of statistical methods in financial forecasting.

Numerical Analysis: At the core of this project lies the application of regression techniques, fundamental tools in statistical analysis and predictive modelling. The mathematical models employed are following:

<u>Linear Regression</u>: This model posits a linear relationship between the independent variable (time) and the dependent variable (stock price). Mathematically, it's represented as $y=\beta 0+\beta 1x+\epsilon y=\beta 0+\beta 1x+\epsilon$, where yy is the stock price, xx is time, $\beta 0\beta 0$ and $\beta 1\beta 1$ are the coefficients, and $\epsilon\epsilon$ is the error term.

Func: def linear_regression(X, y):

<u>Polynomial (Quadratic) Regression</u>: An extension of linear regression, it includes terms up to the square of the independent variable, fitting a parabolic curve to the data. The equation is $y=\beta 0+\beta 1x+\beta 2x2+\epsilon y=\beta 0+\beta 1x+\beta 2x2+\epsilon$. This model is particularly useful for capturing non-linear trends in stock prices.

Func: def polynomial_regression(X, y, degree):

<u>Logarithmic Regression:</u> This approach transforms the independent variable using a logarithmic scale, fitting the model $y=\beta 0+\beta 1\log(x)+\epsilon y=\beta 0+\beta 1\log(x)+\epsilon$. It's effective for data showing exponential growth or decay.

Func: def logarithmic_regression(X, y):

Computing natural log and transforming: X_transformed = np.log(X)

Each of these models was implemented using Python libraries, with the regression analysis conducted on historical stock data sourced from Yahoo Finance. The effectiveness of these models was quantitatively assessed using two key metrics: R^2, which indicates the proportion of variance in the stock price explained by the model, and MSE, a measure of the average squared difference between the predicted and actual values. Furthermore, it also compares and analyses which stock trend aligned better with predictions based on our metrics in bar graph form.

Major Python Libraries Used and their purpose:

- yfinance: Fetches historical stock data from Yahoo Finance.

pandas: Data manipulation and analysis.

numpy: Scientific computing with powerful n-dimensional array object and mathematical functions.

-sklearn.linear_model.LinearRegression: Implements linear regression modelling.

- sklearn.preprocessing.PolynomialFeatures: Generates polynomial and interaction features for polynomial regression.

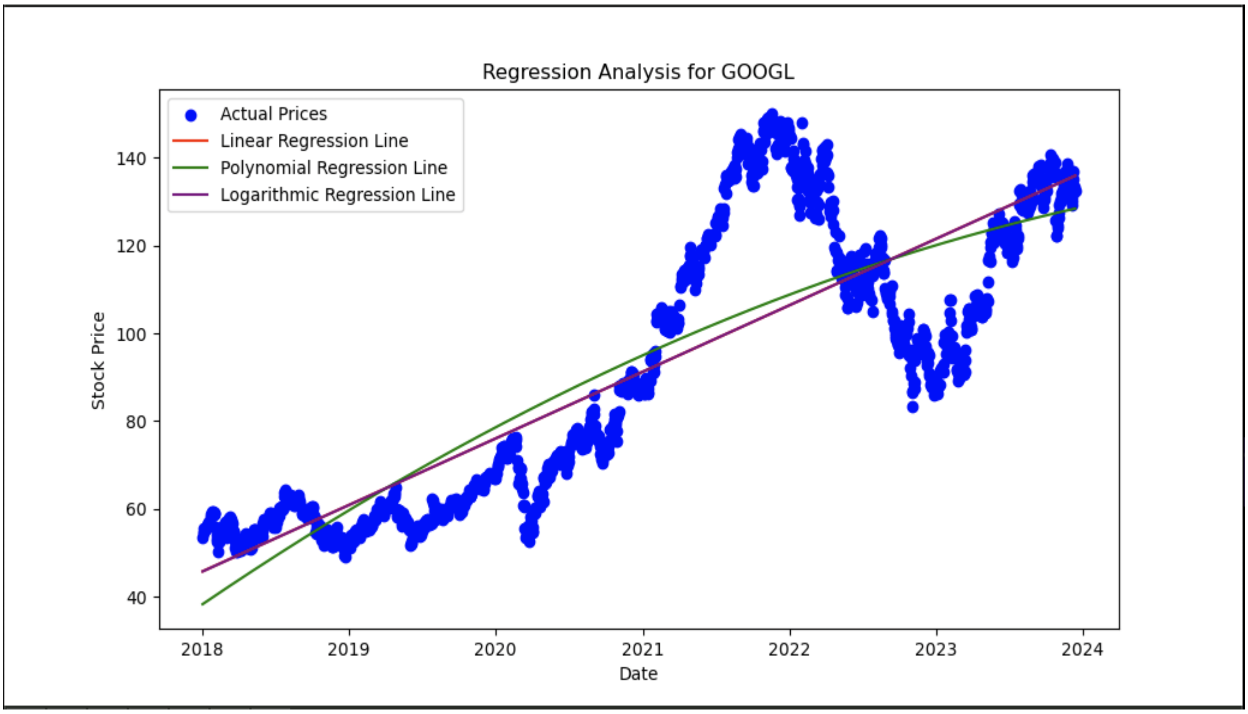
- sklearn.metrics.mean_squared_error: Computes the mean squared error for model evaluation.

-sklearn.metrics.r2_score: Computes the coefficient of determination (R^2) for model evaluation.

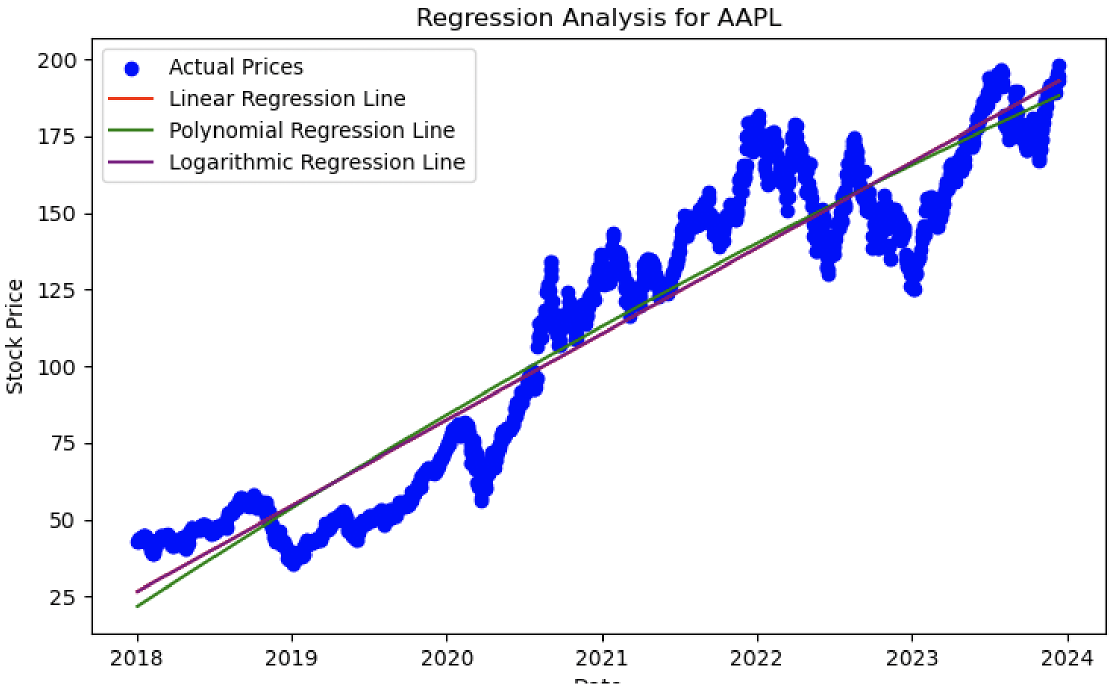
-matplotlib.pyplot: Data visualization for creating plots and charts.

-datetime: Provides classes for manipulating dates and times.

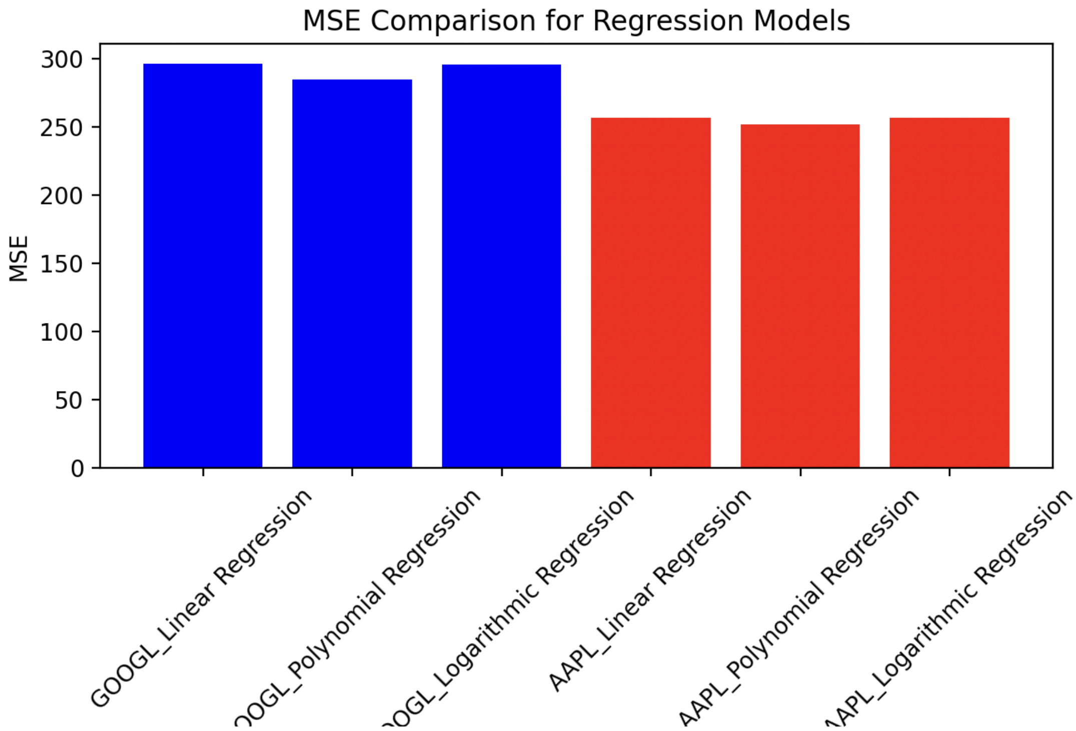
Results: The results were presented in graphical form to plotting historical data from past 5 years and regression lines with different colors.The x-axis represents time (dates), and the y-axis represents stock prices.

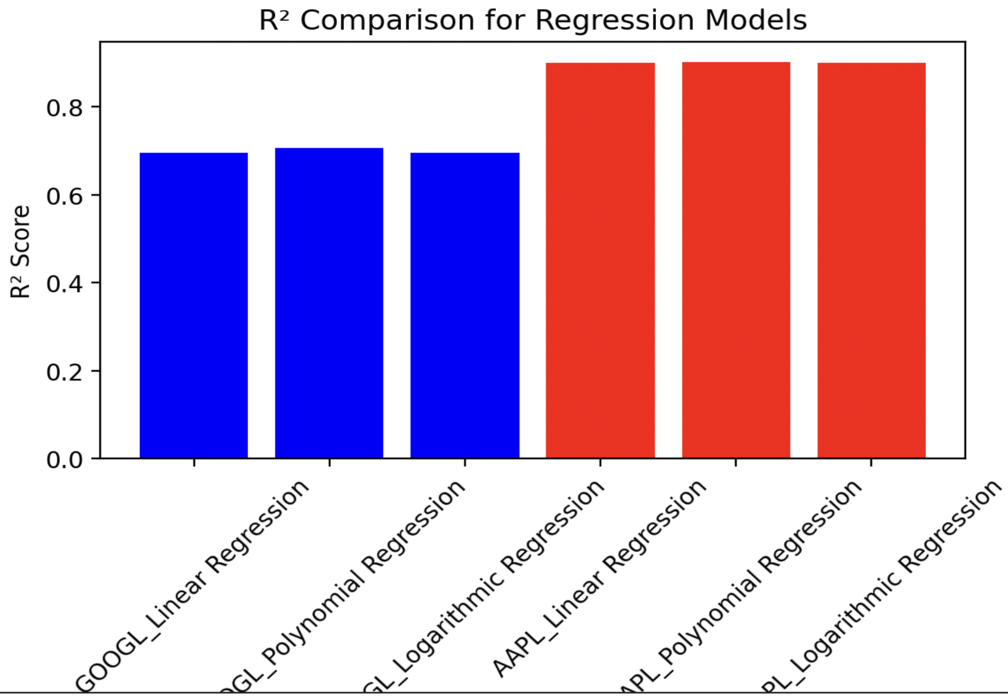


Note: Values for Linear and logarithmic were so close that they were overlapping due to close values. The reason here is these stocks exhibit slow rate change as they are big corporations.



Following graphs compare how our predictions compare for Google and Apple stock based on R^2 and MSE values:





Conclusion: The analysis conducted on the stock prices of Google (GOOGL) and Apple (AAPL) using linear, polynomial, and logarithmic regression models has yielded some insightful findings, which have both practical and theoretical implications.

<u>Model Performance Comparison</u>:

- For Google (GOOGL), polynomial regression showed a slightly better fit than linear and logarithmic models, as evidenced by its lower Mean Squared Error (MSE) and higher R-squared (R^2) values. This suggests that a quadratic relationship might slightly better capture the trends in Google's stock price over the analyzed period.
- For Apple (AAPL), both polynomial and logarithmic models performed better than the linear model, with polynomial regression having a slight edge. This is indicated by the lowest MSE and highest R^2 values among the three models. It implies that the stock price movement of Apple over the selected timeframe was best captured by a model that accounts for non-linear trends.

<u>Interpretation of R^2 and MSE Values:</u>

- The R^2 values, being closer to 1, especially for Apple, indicate a strong explanatory power of the models. A higher R^2 value means that the model explains a significant portion of the variance in the stock prices.
- The MSE values provide a measure of the average prediction error squared. Lower MSE values for Apple across all models suggest that predictions were more accurate and closer to the actual stock prices compared to Google.

In conclusion, this study enhances our understanding of how different regression models perform in the context of stock market data. It also provides an important insight that predictions vary for different stocks as we found that AAPL stock regressions more accurately follow the real trends.