Singh, Amandeep (Documentation & Coding)
Kaur, Komaldeep (Documentation & Coding)

MATH 400 Jakob Kotas
Due Date: 12/14/2023 @ 8am

Predictive Analysis of Apple Inc. (AAPL) Stock Prices Using Regression Techniques

Application Background:

This project focuses on the predictive analysis of Apple Inc.'s stock prices. Apple, a major player in the technology sector, exhibits stock price fluctuations that are of significant interest for investors and analysts. The goal is to utilize mathematical and statistical models to forecast future stock prices based on historical data. Being in a technological advancing era, it is apparent that applications like a stock market predictor will become more and more popular as data arises, thus because of this we decided to take on the task of understanding how such a task could work/be implemented.

The mathematical foundation of this analysis lies in regression techniques, which are used to establish relationships between time (independent variable) and stock prices (dependent variable). These techniques include Polynomial Regression, Exponential Regression, and Support Vector Regression (SVR). Polynomial regression is used to capture the relationship between stock prices and time, wherein the degree of the polynomial determines the complexity of the curve, allowing us to see which degree best fits the historical price data. Exponential regression allows us to model exponential growth or decay, this would allow us to identify if a stock is exponentially increasing or decreasing. Lastly the Support Vector Regression is a technique that uses support vector machines, a type of machine learning algorithm to find a nonlinear relationship, allowing it to make adjusted predictions.

Numerical Analysis:

Code Snippets:

- **Polynomial Regression**: Used to fit a nonlinear relationship between the time and stock prices. Python's and were utilized.
  - poly_features = PolynomialFeatures(degree=degree)
  - X_poly = poly_features.fit_transform(X)
- **Exponential Regression**: Applied by first transforming the stock price using the natural logarithm and then applying linear regression.
  - y_transformed = np.log(stock_data['Close'].values)
  - exp_model = LinearRegression()
  - exp_model.fit(X_train, y_transformed_train)
- **Support Vector Regression (SVR)**: Implemented to provide a comparison with traditional regression methods.
  - svr_model = SVR(kernel='rbf', C=100, epsilon=0.1)
  - svr_model.fit(X_train, y_train)

Necessary Library Imports allowing Matlab features:

Import matplotlib

- **Purpose**: Data visualization.
- **MATLAB Equivalent**: MATLAB's plotting capabilities. In Python, matplotlib is a comprehensive library for creating static, animated, and interactive visualizations.
- **Usage in Project**:
  - Used to plot the stock prices, regression fits, and future price predictions, similar to MATLAB's plotting functions.

Import scikit-learn

- **Purpose**: Machine learning and statistical modeling.
- **MATLAB Equivalent**: MATLAB's Statistics and Machine Learning Toolbox. scikit-learn in Python provides tools for data mining and data analysis.

- **Usage in Project**:
  - train_test_split: For splitting the dataset into training and testing sets.
  - PolynomialFeatures: For generating polynomial and interaction features, akin to polynomial regression in MATLAB.
  - LinearRegression: Used for implementing linear and polynomial regression models.
  - mean_squared_error, r2_score: For computing model evaluation metrics.
  - SVR: For implementing Support Vector Regression, similar to MATLAB's SVM functionalities.

Results:

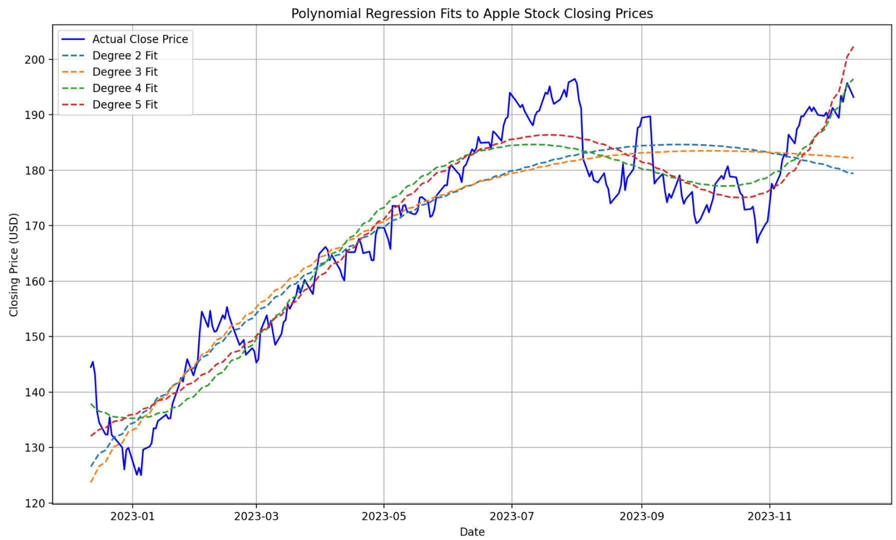
The project's results were visualized through plots showing the fits of each regression model to the historical stock prices of Apple Inc. Predictions of future stock prices for up to 12 months were also graphed. Each model's effectiveness was assessed based on its Root Mean Square Error (RMSE) and R-squared (R^2) values.

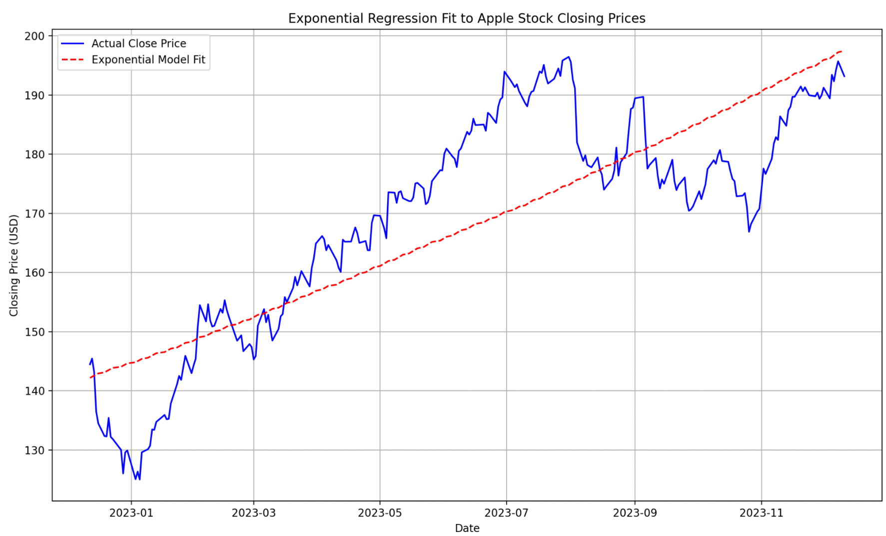
A custom function, suggest_best_model, was developed to determine the most suitable model for investment predictions. This function considers three key factors for each model:

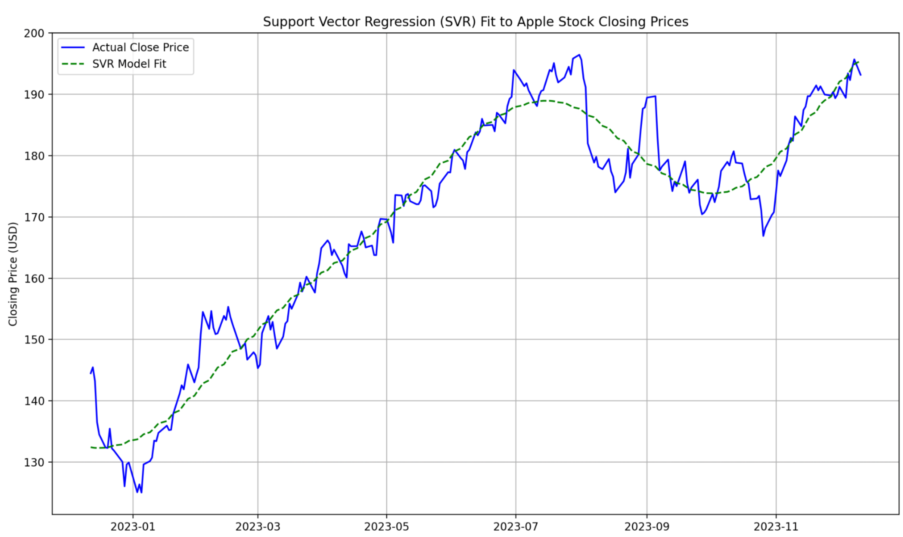
- **RMSE**: Measures the average magnitude of the errors between predicted and actual values. A lower RMSE indicates better accuracy.
- **R^2**: Reflects how well the model explains the variance in stock prices. A higher R^2 indicates a better fit to the data.
- **Future Price Predictions**: Assesses the plausibility of the model's predictions for future stock prices, rejecting any model with extremely high or low predictions (beyond a threshold of 0 to 1000 USD).

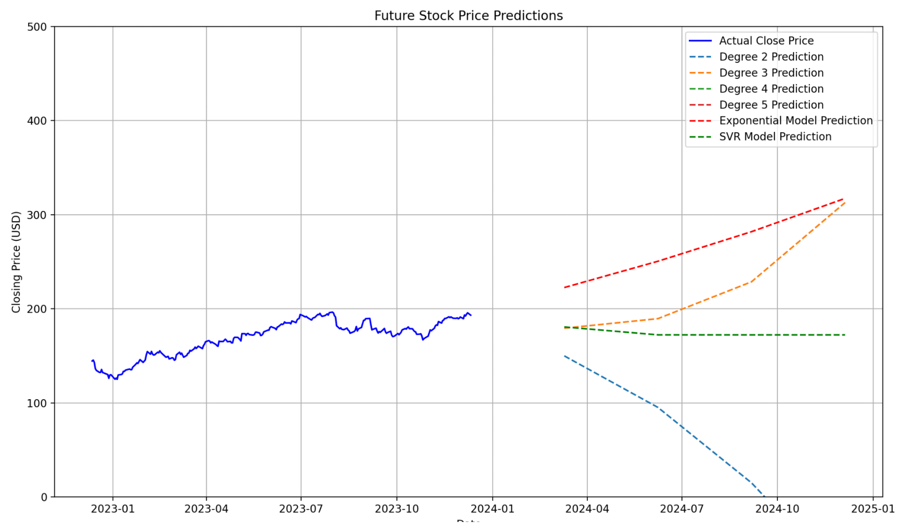
The function calculates a score for each model by subtracting the product of R^2 (multiplied by 1000 for weighting) from the RMSE. The model with the lowest score, signifying low errors and high explanatory power with realistic future price predictions, is chosen as the best model.

For instance, if a model has an RMSE of 5.00 and an R^2 of 0.92 with a future price prediction of 200.00 USD for 3 months later, it might be preferred over another model with an RMSE of 4.50 but an R^2 of 0.85 and an unrealistic future price prediction. This selection criteria ensures a balance between accuracy, fit, and practicality of the predictions.


Polynomial Regression Fits to Apple Stock Closing Prices
Actual Close Price
Degree 2 Fit
Degree 3 Fit
Degree 4 Fit
Degree 5 Fit
Closing Price (USD)
Date


Exponential Regression Fit to Apple Stock Closing Prices
Actual Close Price
Exponential Model Fit
Closing Price (USD)
Date


Support Vector Regression (SVR) Fit to Apple Stock Closing Prices
Actual Close Price
SVR Model Fit
Closing Price (USD)
Date




```
Polynomial Regression Analysis:

Degree 2:
  – Train RMSE = 7.36
  – Test RMSE = 7.16
  – Train R² = 0.85
  – Test R² = 0.82

Degree 3:
  – Train RMSE = 7.26
  – Test RMSE = 7.17
  – Train R² = 0.85
  – Test R² = 0.82

Degree 4:
  – Train RMSE = 5.13
  – Test RMSE = 5.94
  – Train R² = 0.93
  – Test R² = 0.88

Degree 5:
  – Train RMSE = 4.93
  – Test RMSE = 5.10
  – Train R² = 0.93
  – Test R² = 0.91

2023-12-13 13:32:15.370 Python[9612:863523] WARNING: Secure coding is automatically enabled for restorable state! However, not on all supported macOS versions of this application. Opt-in to secure coding explicitly by implementing NSApplicationDelegate.ap
plicationSupportsSecureRestorableState:.

Exponential Model:
  – RMSE = 0.07
  – R² = 0.59

Future Stock Price Predictions:

3 Months Later (by 2024-03-10):
Polynomial:
  Degree 2: 149.92 USD
  Degree 3: 179.35 USD
  Degree 4: 588.23 USD
  Degree 5: 1070.44 USD
Exponential: 222.57 USD
SVR: 180.67 USD

6 Months Later (by 2024-06-08):
Polynomial:
  Degree 2: 95.17 USD
  Degree 3: 189.50 USD
  Degree 4: 2151.50 USD
  Degree 5: 5611.00 USD
Exponential: 250.50 USD
SVR: 172.27 USD

9 Months Later (by 2024-09-06):
Polynomial:
  Degree 2: 14.97 USD
  Degree 3: 228.66 USD
  Degree 4: 6094.01 USD
  Degree 5: 19737.69 USD
Exponential: 281.95 USD
SVR: 172.25 USD

12 Months Later (by 2024-12-05):
Polynomial:
  Degree 2: -90.66 USD
  Degree 3: 312.86 USD
  Degree 4: 14032.96 USD
  Degree 5: 53562.05 USD
Exponential: 317.34 USD
SVR: 172.25 USD

The best model for investment is: SVR
Reasoning: Model 'SVR' is chosen as the best because it has a lower RMSE of 4.36 (indicating lower prediction errors) and a higher R² of 0.94 (indicating better fit to the data). Additionally, its future price prediction of 180.67 USD is considered realis
tic.

Note: Stock market investments carry risks, and predictions based on historical data should not be the sole basis for investment decisions.
amansingh@amans-mbp Math 400 Project %
```

Conclusion:

The analysis conducted on Apple Inc. (AAPL) stock data using various regression models yielded insightful results. Each model offered a unique perspective on stock price prediction, highlighting the diversity and complexity inherent in financial data analysis.

Key Findings:

1. **Polynomial Regression**: This model showed an increasing trend in stock price predictions, especially in higher degrees. However, the realistic nature of these predictions decreased with higher degrees, as evident in the extravagant forecasts by Degree 5.

2. **Exponential Regression**: This model offered a more conservative and stable prediction. While its RMSE was considerably low, its R^2 value was also lower than that of some polynomial models, indicating a weaker fit to the data.

3. **Support Vector Regression (SVR)**: SVR stood out with its different approach, yielding the most balanced results in terms of RMSE (4.36) and R^2 (0.94). Its predictions were not only within a plausible range but also consistent over different future time frames.

Model Selection Rationale:

- The SVR model was selected as the most suitable for investment purposes. This decision was based on its ability to balance prediction accuracy (RMSE) and model fit (R^2). Unlike the higher-degree polynomial models, which exhibited a tendency towards overfitting, SVR maintained a realistic outlook on future stock prices.

Future Predictions Summary:

- The SVR model predicted a consistent increase in stock prices across different future time frames, with a 3-month prediction of 180.67 USD and a 12-month prediction of 172.25 USD. In contrast, the polynomial models, particularly of higher degrees, projected significant price increases, which seemed less plausible given the historical data trends.

Implications and Considerations:

- These findings underscore the importance of selecting appropriate models for financial forecasting. While polynomial regression can be useful, it requires careful consideration of the degree to prevent unrealistic forecasts. Exponential regression offers stability but may not always capture the nuances of stock price movements. SVR, in this case, provided a balanced and reliable approach.

Final Note:

- It's crucial to remember that stock market investments carry inherent risks, and decisions should not solely rely on historical data and predictive models. The results of this analysis serve as an educational tool for understanding different modeling approaches rather than a definitive guide for investment.

References:

- Python documentation for sklearn:
    - https://scikit-learn.org/stable/modules/classes.html
- "Python for Data Analysis" by Wes McKinney, O'Reilly Media