INST737 - Introduction to Data Science

Milestone - 3 Report

-Srikanth Parvathala, Vijay Arni, Yating Yang

Research Question:

- ➔ Can future stock prices be predictable using historical stock prices of specific companies and macroeconomic indicators like the nominal GDP index, real GDP index, unemployment rate, and federal funds rate?

Data Preprocessing:

- ➔ Post Milestone 2, no further data collection and cleaning processes have been performed. However, we used data preprocessing in order to create dummy variables for one of our independent variables, "Source." The Source determines the companies that are incorporated in the dataset.
- ➔ For feature selection, we focus on a specific set of independent variables as outlined in our research question. These variables include the Monthly Nominal GDP Index, Monthly Real GDP Index, Unemployment Rate, Federal Funds Rate, and data from key tech companies such as Amazon, Apple, Facebook, Google, and Netflix.

Question 1 - SVM

In our analysis, we examined the use of Support Vector Machines to forecast stock market prices, particularly the Close Variable. Close determines the closing value of a company's stock for a particular month. Since 'Close' is a continuous characteristic, regression analysis is required instead of classification.

Linear SVM Model

First, we trained and evaluated a linear SVM model. The dataset underwent the necessary preparation, such as scaling values, before the SVM with a linear kernel. The Root Mean Square Error (RMSE), a common statistic for regression tasks, was used to assess the model's performance.

Non-Linear SVM

We repeated the analysis using non-linear kernels, such as the sigmoid, polynomial, and radial basis function (RBF), to investigate the possibilities of SVMs better. This allowed us to evaluate how various kernels affected the model's capacity to identify non-linear relationships in the data.

Data Preprocessing

- Dummy variables were created for categorical features once again.
- Features were scaled to ensure equal weighting in the SVM algorithm.

Model Training and Evaluation

Linear Kernel: The linear model provided a baseline understanding of the data.

RBF Kernel: Demonstrated significantly better performance, suggesting the presence of non-linear relationships in the data.

Polynomial Kernel: Also performed well, indicating the data's non-linear nature.

Sigmoid Kernel: Performed poorly, meaning it was unsuitable for this dataset.

Hyperparameter Tuning and Cross-Validation

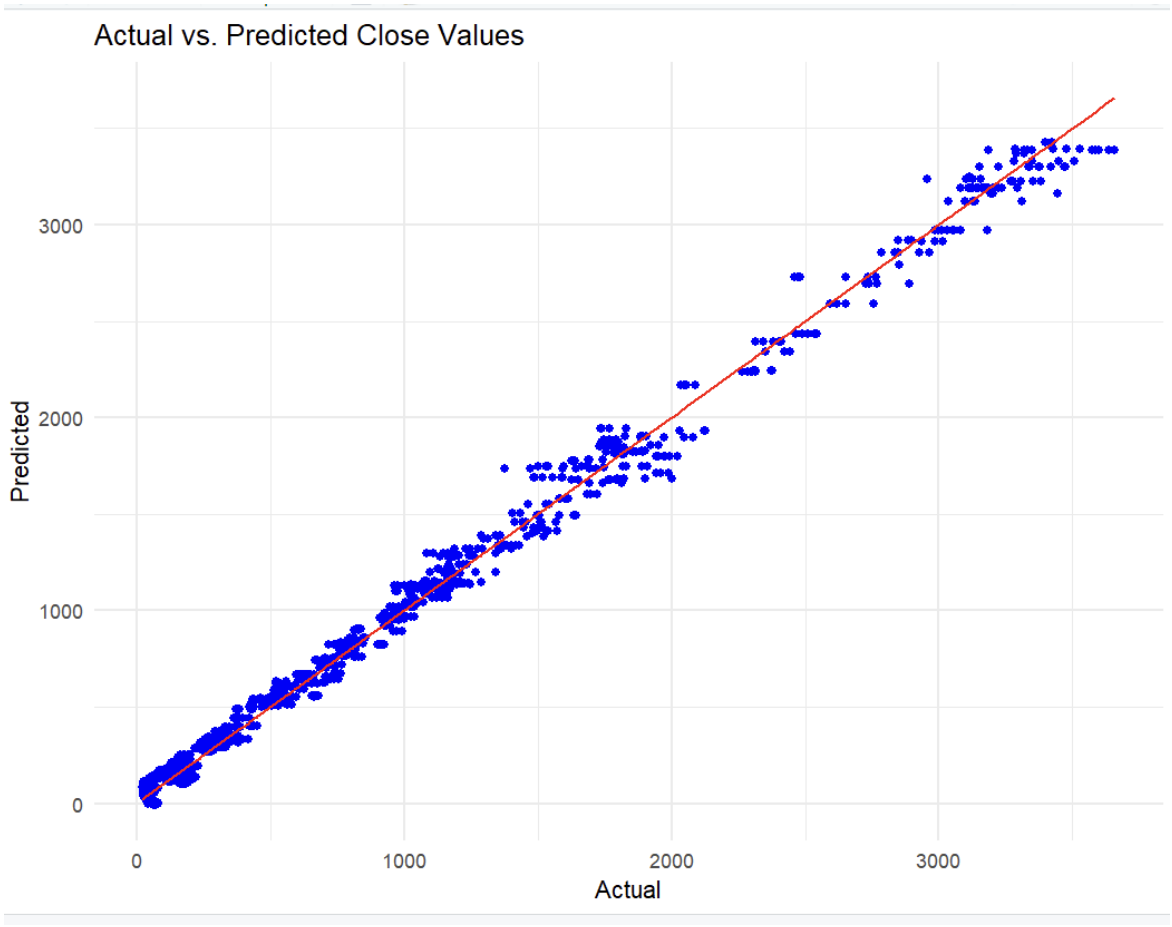
Hyperparameter tuning was conducted for the RBF kernel, optimizing the cost and gamma parameters. The best model from this tuning showed improved performance. Cross-validation with ten folds was used to ensure the model's generalizability.

Results

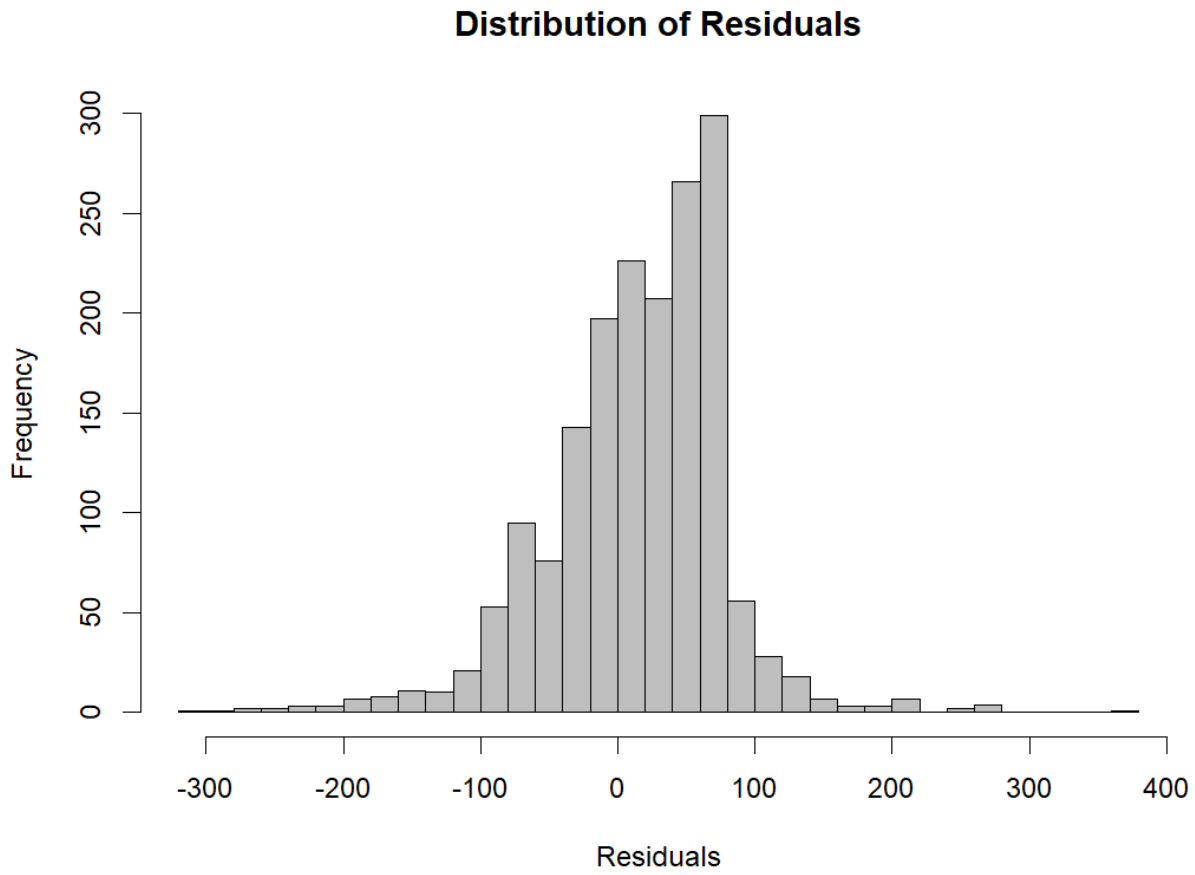
- ➔ RMSE for Linear Kernel: **433.97**
- ➔ RMSE for RBF Kernel: **66.39**
- ➔ RMSE for Polynomial Kernel: **74.26**
- ➔ RMSE for Sigmoid Kernel: **50,923.34**

The RBF kernel with optimized parameters (cost = 100, gamma = 0.1) was the most effective.

Visualizations:



Actual vs. Predicted Close Values Scatter Plot: This picture illustrates how the model's predictions compare to the closing values that occurred. The red line on the plot represents the line of perfect prediction. Accurate predictions are indicated by observations that closely match this line. The red line's dense clustering of dots shows a good correlation between the actual and predicted values, indicating that the model is successful in capturing the trend of the stock closing prices. On the other hand, departures from the line, especially at larger value ranges, highlight the limitations of the model and the regions where prediction accuracy declines.



This residual analysis is represented by this histogram, which shows a distribution that resembles a normal curve with a zero center. This implies that the predictions made by the model are generally objective. The distribution is notably skewed to the right, suggesting that the model may underpredict the closing values in a particular subset of the dataset. When evaluating the predictive power of the model, this skewness should be taken into account as it could indicate possible areas for model improvement.

To answer the research question, the analysis demonstrated that the RBF kernel, after careful hyperparameter tuning and cross-validation, yielded a significantly lower Root Mean Square Error (RMSE) compared to other models, indicating a substantial predictive accuracy. While the results are promising, they also reflect the inherent complexity and volatility of the financial

markets, suggesting that while SVM can be a powerful tool for prediction, it should be used as an investment strategy that accounts for potential market shifts and unforeseen events.

Question 2 - Neural Networks

In this section, we trained our stock prediction model using a neural network. Several steps were implemented in our code before initiating the training process. Firstly, we utilized a normalization function to scale numeric vectors to a [0, 1] range. This step is crucial, as our input variables, including the stock's opening and closing prices, the monthly nominal GDP index, the monthly real GDP index, the unemployment rate, and the federal funds rate, are all continuous values. This process ensured that all values were proportionally scaled so that the smallest value became 0, the largest became 1, and all other values fell somewhere in between on a proportional scale. Additionally, to guarantee a random split into training and testing datasets, we employed the sample function in R for random selection. During the model training phase, we tuned the model using different numbers of hidden layers, neurons per layer, and activation functions. The learning rate was set at 0.01, and the model underwent 100 training epochs.

Building the Neural Network Model with Four Macroeconomic Variables

Our research question aims to utilize macroeconomic indicators to predict stock prices. Accordingly, we trained the model with these variables: Monthly_Nominal_GDP_Index, Monthly_Real_GDP_Index, Unemployment_Rate, and Federal_Funds_Rate. It's important to note that our dataset includes stock prices from Amazon, Google, Facebook, Netflix, and Apple. For this training phase, we consolidated all data from these five companies without training them separately.

Training and Tuning

Initially, we trained the model using only one hidden layer, varying the number of neurons from 1 to 30 and employing two different activation functions. The overall results were not ideal, with the correlation between the training and testing data hovering around 0.41 – a value we hoped would be higher. The results indicated that the configuration of 3 neurons with a logistic activation function performed the best.

In an effort to fine-tune our neural network model, we then increased the number of hidden layers, training models with two hidden layers containing 3, 5, and 10 neurons, respectively. Among these, the configuration of 3 neurons in each layer, trained with the tanh activation function, yielded the best correlation of 0.4168. However, this result did not surpass the performance achieved by the one hidden layer, three neuron configuration.

We continued increasing the number of hidden layers to see if it would improve the model's performance. The models with three hidden layers, using 1 and 5 neurons per layer, also did not show outstanding results, with correlations remaining around 0.413. However, when we tried the combination where the model was trained with three neurons at the first, second layer, and five neurons at the third layers, the model performed a lot better with a correlation rate of 0.4175613, which is the second-best performance among all the results.

The table below shows the performance of the model with different configurations, with the best-performing model highlighted:

Hidden Layers	# of Neurons Per Layer	Activation Function	Performance Result (correlation)
1	1	logistic	0.4133735
1	1	tanh	0.4132109
1	**3**	**logistic**	**0.4178648**
1	3	tanh	0.4174407
1	5	logistic	0.4164240
1	5	tanh	0.4171570
1	10	logistic	0.4149247
1	10	tanh	0.4132938
1	15	logistic	0.4154507
1	15	tanh	0.4130074
1	30	logistic	0.4163106
1	30	tanh	0.4147200
2	3, 3	logistic	0.4136843

Hidden Layers	# of Neurons Per Layer	Activation Function	Performance Result (correlation)
2	3, 3	tanh	0.4168739
2	5, 5	logistic	0.4163078
2	5, 5	tanh	0.4102345
2	10, 10	logistic	0.4122378
2	10, 10	tanh	0.4084850
3	1, 1, 1	logistic	0.4131900
3	1, 1, 1	tanh	0.4131724
3	5, 5, 5	logistic	0.4138913
3	5, 5, 5	tanh	0.4137921
3	**3, 3, 5**	**logistic**	**0.4175613**
3	3, 3, 5	tanh	0.4122615
4	3, 3, 5, 5	logistic	0.4155830
4	3, 3, 5, 5	tanh	0.4129365
4	3, 3, 3, 5	logistic	0.4167061
4	3, 3, 3, 5	tanh	0.4159741

Our observations from the performance results include:

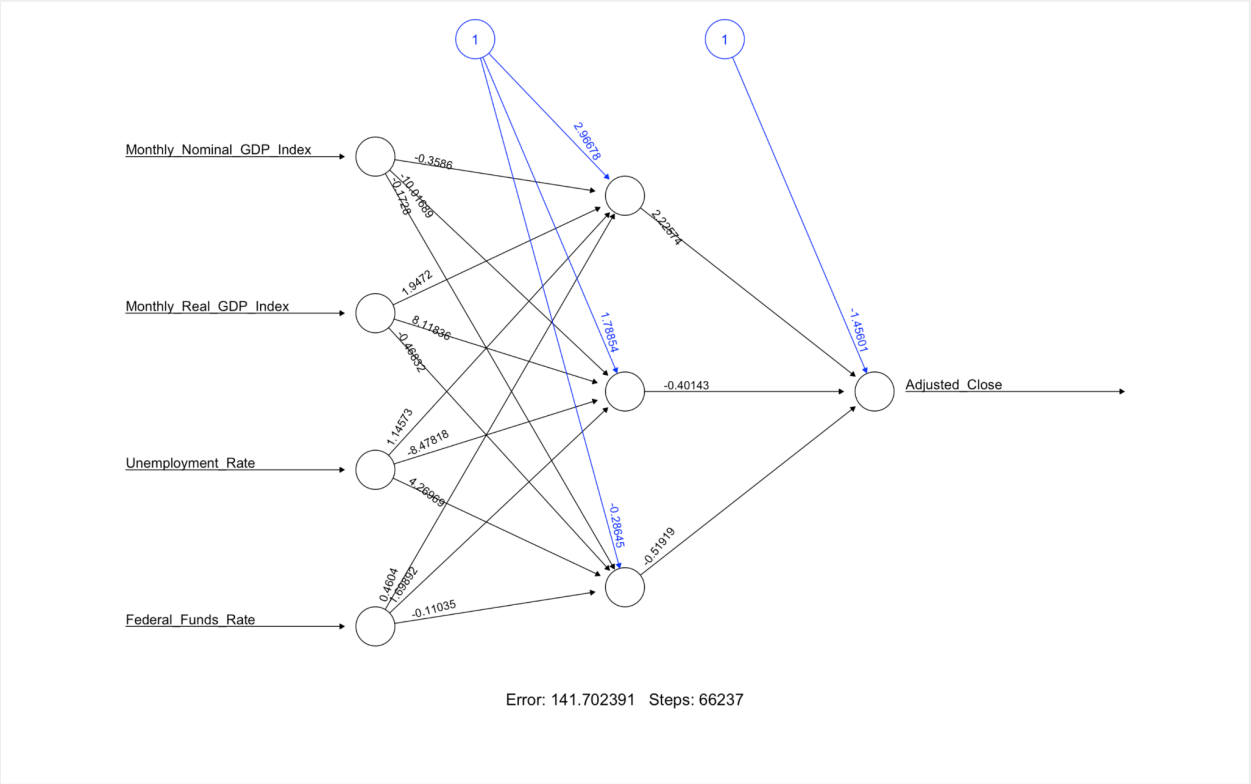
1. **Best Performance Model:** The most effective model, as highlighted, had one hidden layer with three neurons using the logistic activation function, achieving a correlation of approximately 0.4178648.
2. **Moderate Correlation Performance:** The correlation coefficients across all configurations range around 0.41 to 0.42.
3. **Complexity vs. Performance:**
    a. Models with a single hidden layer and varying neuron counts (from 1 to 30) exhibited slight variations in performance. The highest correlation in this group was 0.4178648, achieved with the aforementioned best-performing model configuration.
    b. Increasing the model's complexity — either through more neurons or additional layers — did not consistently enhance performance. This suggests that simpler

models might be sufficient to capture the underlying patterns in the data. For instance, models with more complex architectures, such as three layers in a 3, 3, 5 configuration, did not significantly outperform simpler models.

4. **Activation Function**: There was no consistent pattern indicating that one activation function consistently outperformed the other across different network architectures.

Best model visualization

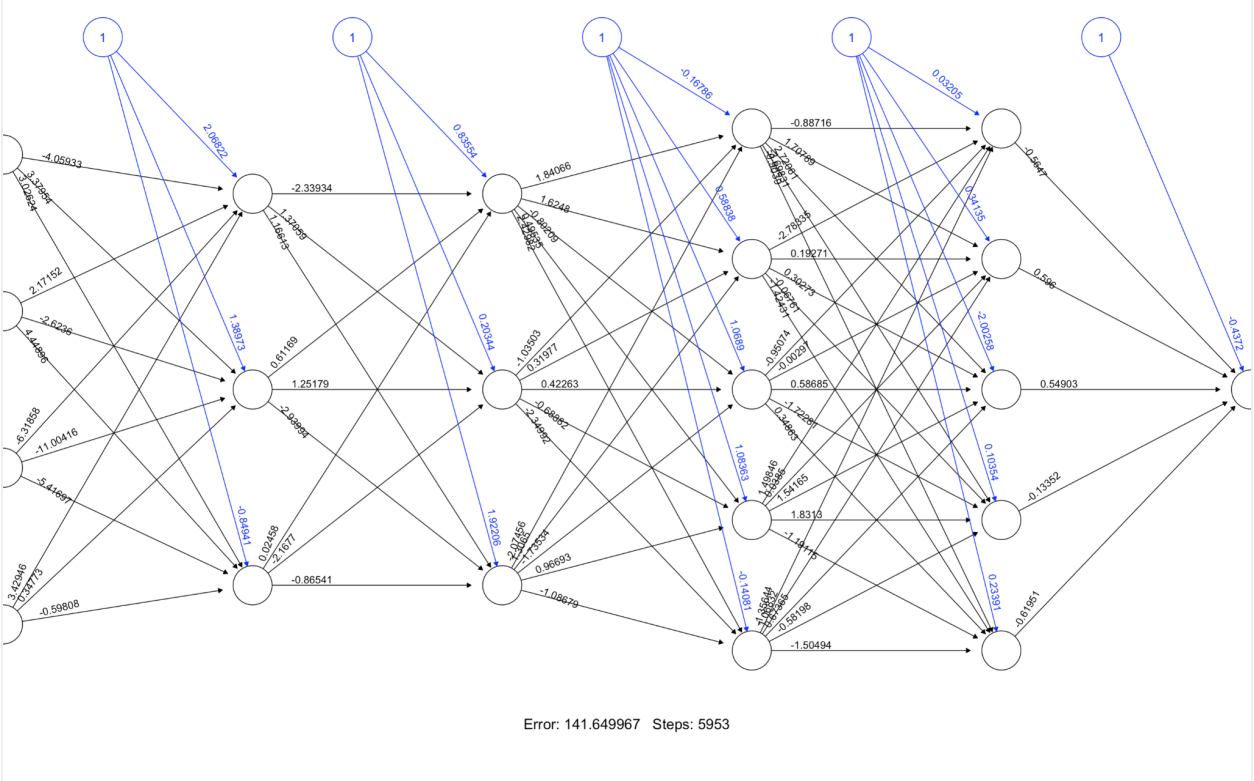
The top-performing model featured 1 hidden layer with 3 neurons, utilizing the logistic activation function.



Best performance model

Second-best model visualization

The model with the second-highest performance incorporated 3 hidden layers, with 3 neurons in the first and second layers and 5 neurons in the third layer, all employing the logistic activation function.



Second-best performance mode

Training Models Individually for Each Company

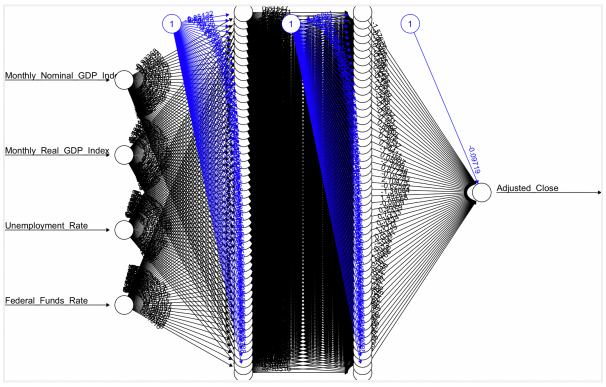
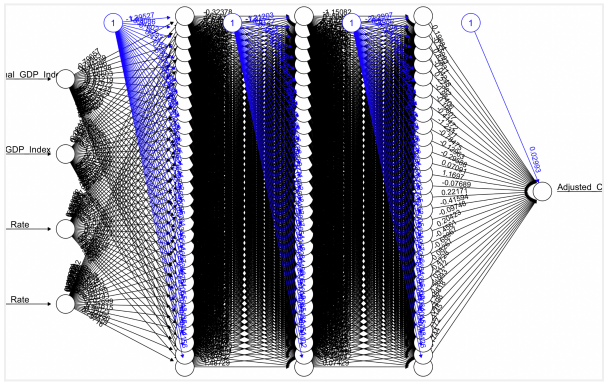
Given that our dataset comprises stock prices from Amazon, Google, Facebook, Netflix, and Apple, we explored the prospect of training distinct models for each company to assess potential improvements in performance.

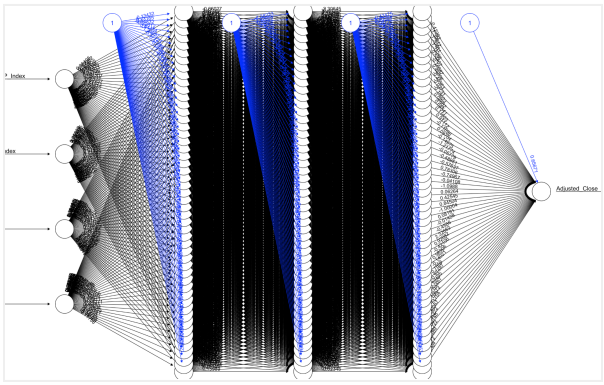
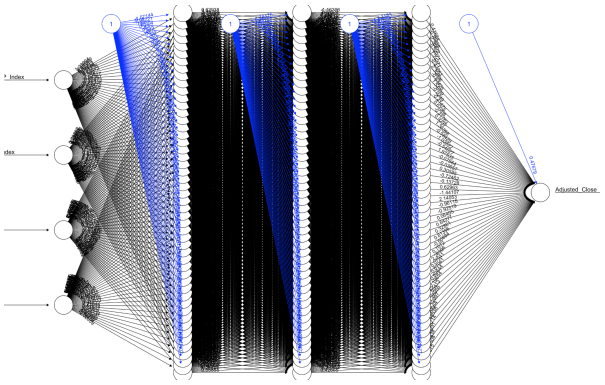
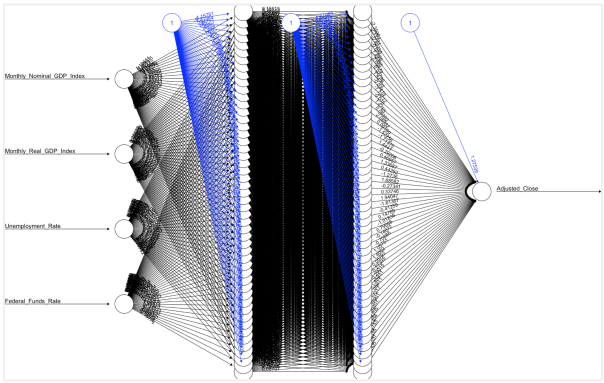
Hidden Layers	# of Neurons Per Layer	Activation Function	Amazon	Google	Apple	Netflix	Facebook
1	1	logistic	0.9833081	-	0.987043	0.9718635	0.9685004
1	1	tanh	0.9831858	-	0.985476	0.9718714	0.9687677
1	3	logistic	0.9951430	0.9956423	0.992538	0.9876198	0.9863857
1	3	tanh	0.9955818	0.9955393	0.995159	0.9867118	0.9869333
1	5	logistic	0.9948362	0.9947407	0.995491	0.9884241	0.9911105
1	5	tanh	0.9957692	0.9959869	0.996025	0.99182	0.9898509
1	10	logistic	0.9966307	0.9959497	0.996694	0.995314	0.9904691
1	10	tanh	0.9970744	0.9963602	0.996716	0.9950227	0.9931016
1	30	logistic	0.9961772	0.9963748	0.996486	0.995216	0.9927873
1	30	tanh	0.9973239	0.9968399	0.996789	0.9958637	0.993324
2	3, 3	logistic	0.9954005	-	-	-	-
2	3, 3	tanh	0.9955640	-	-	-	-
2	30, 30	logistic	0.9972147	0.9962358	0.9969322	0.9967001	0.9952526
2	30, 30	tanh	0.9982242	0.9976439	0.9979694	0.9970763	0.9959945
2	50, 50	logistic	0.9973301	0.9966927	0.9972486	0.9966444	0.9939522
2	50, 50	tanh	**0.9983923**	0.9980465	0.9979023	0.9971092	**0.9961078**
3	10, 10, 10	logistic	0.9969031	-	0.9969084	0.9967448	0.9939765
3	10, 10, 10	tanh	0.9974783	-	0.9976471	0.9962165	0.9950188
3	30, 30, 30	logistic	0.9977986	0.9970140	0.9969402	0.996653	0.9952092
3	30, 30, 30	tanh	0.9982883	0.9978821	0.9978999	0.9970962	0.9956999
3	50, 50, 50	logistic	0.9977016	0.9967903	0.9972104	0.9960267	0.9939821
3	50, 50, 50	tanh	0.9983663	**0.9980605**	**0.9979631**	**0.997264**	0.9960779

The highlighted cells indicate the best-performing models for each company. Our observations from the performance results include:

1. **High Overall Correlation Performance:** The correlation coefficients for these 5 companies are notably high across various configurations, ranging from around 0.983 to 0.998.

2. **Effectiveness of Individualized Models:** Training separate models for individual companies significantly enhances the predictive accuracy. This implies that the stock price movements can be better modeled with a company-specific approach.
3. **Complexity vs. Performance:**
    a. Increasing the complexity of the model generally leads to improved performance. This is evident from the higher correlation coefficients observed in configurations with more neurons (10, 30, 50) and multiple hidden layers (2 and 3).
    b. The best-performing models for each company typically involve configurations with higher neuron counts and multiple layers, particularly with the tanh activation function.
4. **Activation Function:** Both logistic and tanh activation functions performed well, with tanh slightly outperforming logistic in most configurations.
5. **Absence of Overfitting:** Initially, we were concerned about potential overfitting, particularly as model performance significantly improved from 0.41 to 0.99. However, subsequent testing showed no indications of overfitting.

Best model visualization for Amazon The top-performance model featured 2 hidden layers with 50 neurons, utilizing the logistic activation function.	**Best model visualization for Google** The top-performance model featured 3 hidden layers with 50 neurons each, utilizing the tanh activation function.
	
Best model visualization for Apple	**Best model visualization for Netflix**

The top-performance model featured 3 hidden layers with 50 neurons, utilizing the logistic activation function.	The top-performance model featured 3 hidden layers with 50 neurons, utilizing the logistic activation function.
	
Best model visualization for Facebook The top-performance model featured 2 hidden layers with 50 neurons, utilizing the logistic activation function.	
	

Conclusion

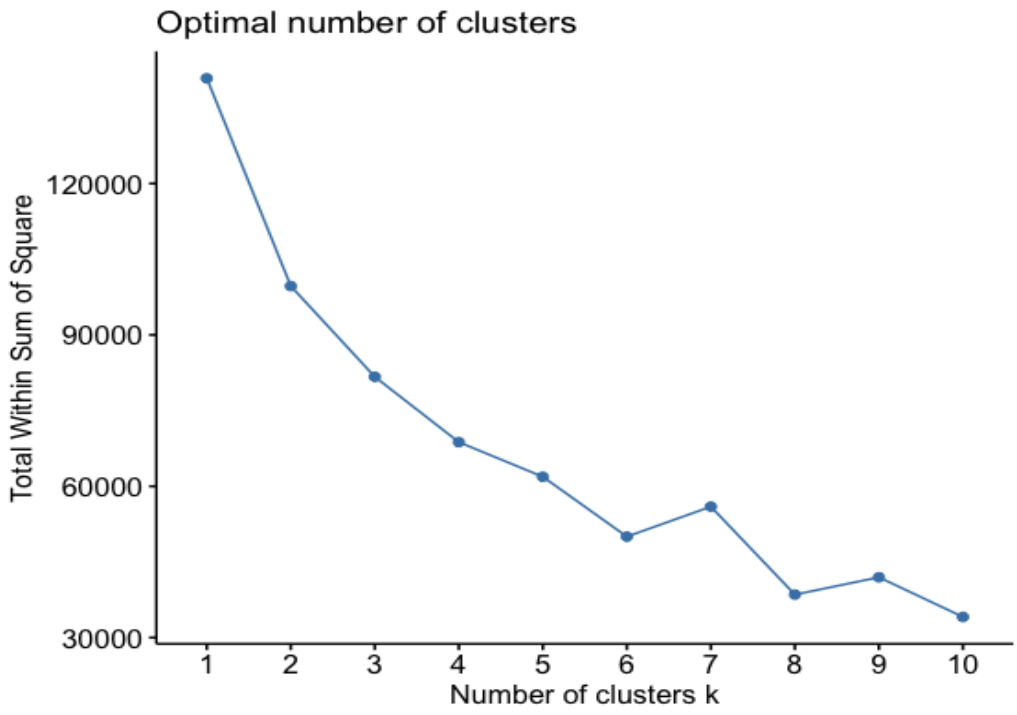
In our comprehensive study, we successfully applied neural network models to predict stock prices, focusing on both a generalized approach using key macroeconomic indicators and individualized models for specific companies. Our methodology involved scaling input variables to enhance model accuracy and experimenting with various configurations. In the generalized approach, a relatively simple model, trained with one hidden layer and three neurons, exhibited the best performance. However, the overall performance was not ideal, achieving only around

0.41 in correlation coefficients. In contrast, the individualized models demonstrated impressive results. Models tailored for specific companies, such as Amazon, Google, Apple, Netflix, and Facebook, showed superior predictive accuracy, particularly in more complex configurations with higher neuron counts and multiple hidden layers.

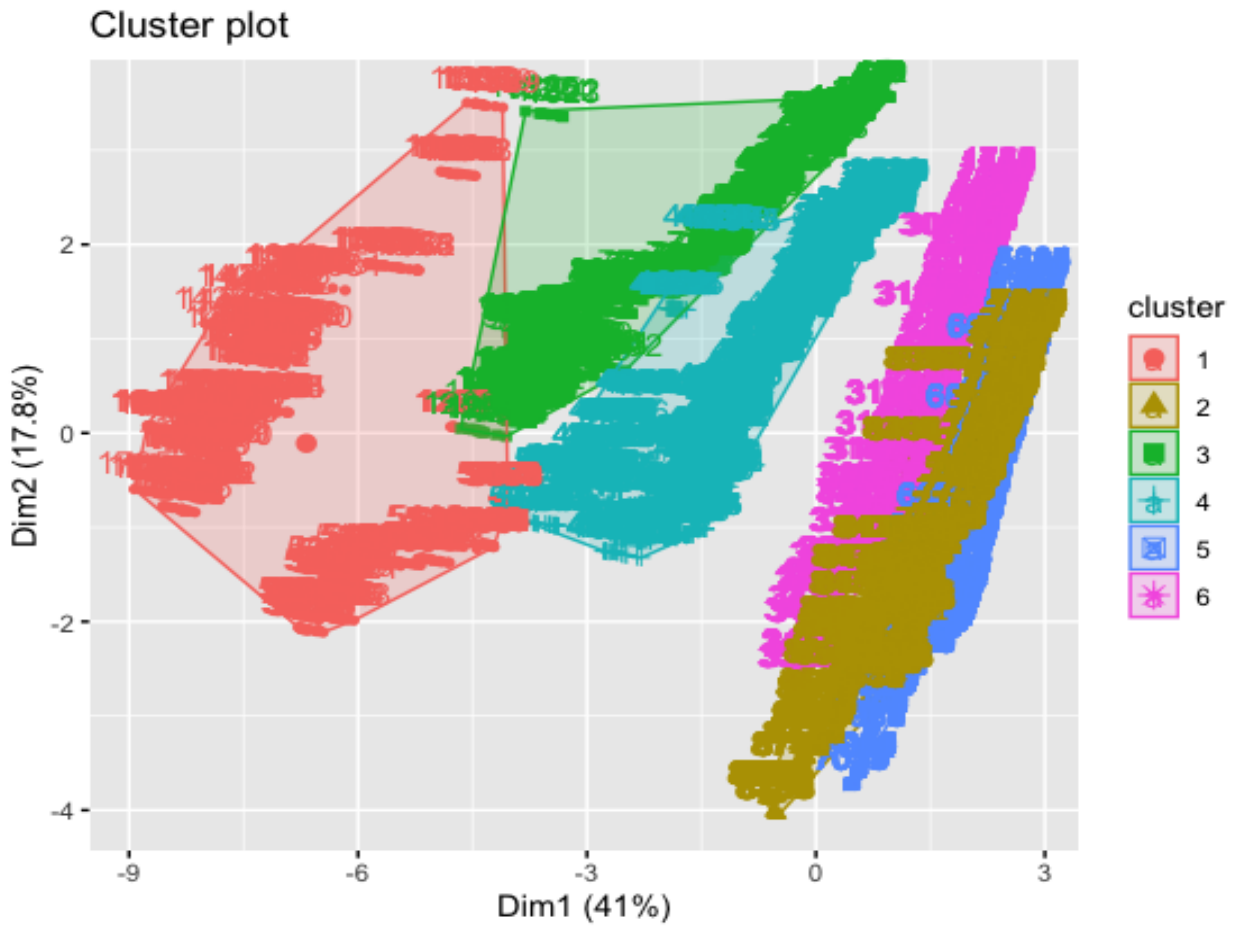
Question 3 - Clustering

In our Clustering analysis, initially we preprocessed the data to convert the date column into numerical by taking the first date (oldest date) as a reference point and calculating the number of days, and normalizing it. Later, we utilized three prominent clustering algorithms, each with its unique approach to discovering data patterns. K-Means clustering seeks to partition the data into K distinct clusters centered around the mean of the dataset's features. Hierarchical clustering constructs a tree-like structure of the data, revealing its nested hierarchies. Lastly, Density-Based clustering (DBSCAN) stands out by identifying clusters as areas of high density separated by regions of low density, capable of detecting outliers in the process.

K-Means Clustering



Based on the above plot, the best number of clusters is **6** as this is the point where the it has largest distance change with respect to the sum of squares.



The number of elements per cluster are:

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
669	1761	1321	1532	1761	1761

669 1761 1321 1532 1761 1761

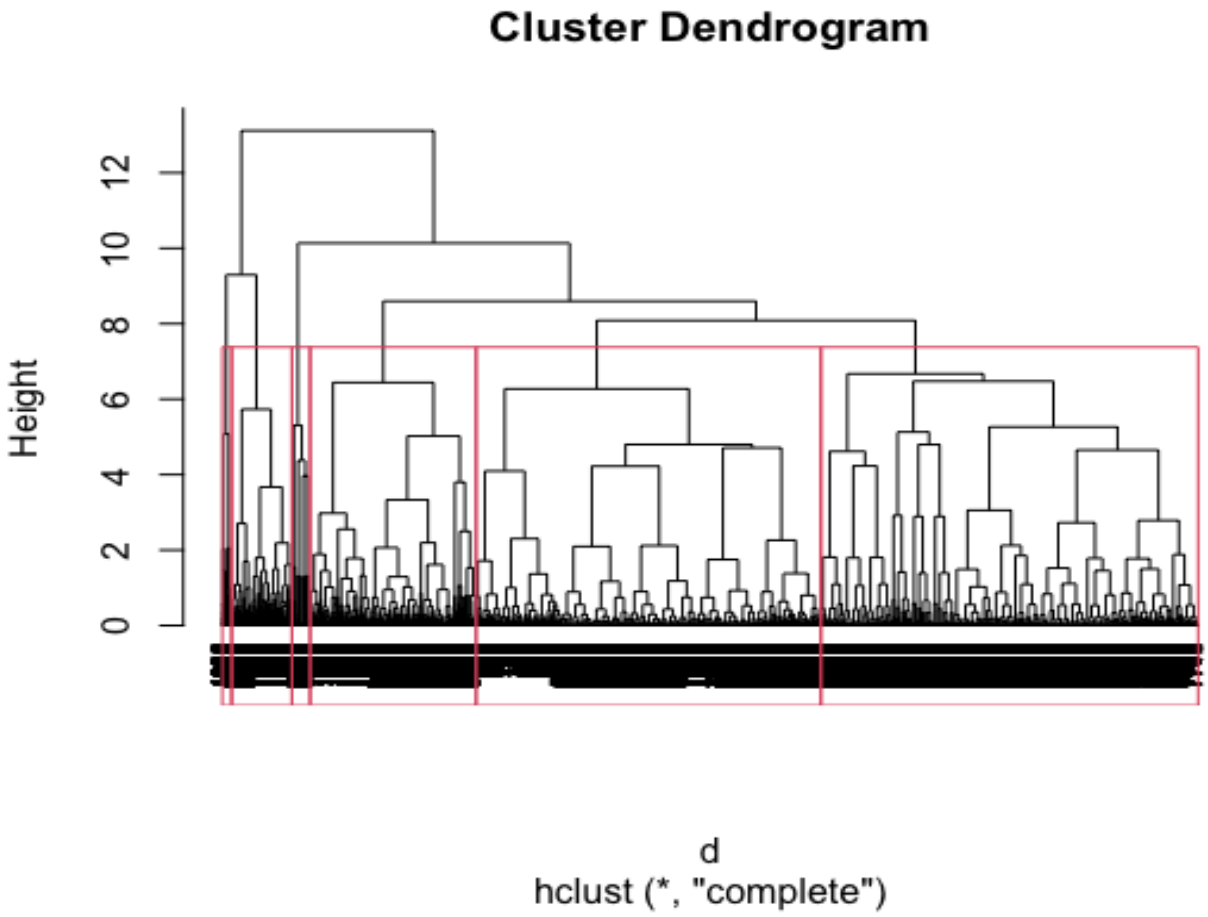
We began with K-Means clustering, where the elbow method guided us to a solution of six distinct clusters. The sizes of these clusters provide an intriguing view into the dataset's landscape. Each cluster has been found to contain a balanced number of data points, with the second, fifth and sixth clusters each holding 1,761 members. This balance suggests a dataset with shared commonalities yet distinct enough to form separate enclaves. The fourth cluster, slightly

less populous with 1,532 points, and the third, with 1,321 points, hint at more specific traits that are less widespread but still significant within the data. The first cluster, the smallest gathering of 669 points, may represent a niche pattern or anomaly within the dataset, deserving of special attention for its unique characteristics.

The equal distribution of data points across most clusters, each with exactly 1,761 points, is a remarkable phenomenon. It suggests that the data may contain symmetrical or evenly spread features, leading to an equitable division. This balance provides an opportunity for us to delve into each cluster and understand the nuances that make each group unique.

As we interpret the centroids of each cluster, we embark on a journey to understand the 'average' member of each group. Centroid 1, this cluster represents days with high stock prices (high values in Open, High, Low, Close, Adjusted_Close) and a strong economy (high Monthly_Nominal_GDP_Index and Monthly_Real_GDP_Index, low Unemployment_Rate, negative Federal_Funds_Rate). The sources are positively associated with Amazon and negatively with the other companies. Centroid 2 seems to be a baseline or average condition, as most values are around zero. It could represent typical market conditions without significant outliers. The source is strongly associated with Netflix. Centroid 3 could indicate days with moderately high stock prices and slightly negative economic indicators (lower GDP indexes, higher unemployment rate). The source is strongly associated with Amazon. Centroid 4 represents days with slightly higher than average stock prices and slightly negative economic indicators. It's strongly associated with Google. Centroid 5 might represent days with lower stock prices and typical economic conditions. Apple is the strongly associated source here and finally, centroid 6 indicates days with low stock prices and average economic conditions. Facebook is a significant source in this cluster.

Hierarchical Clustering



Based on the above dendrogram,

- The best number of clusters is **6**.
- The number of elements per cluster are:

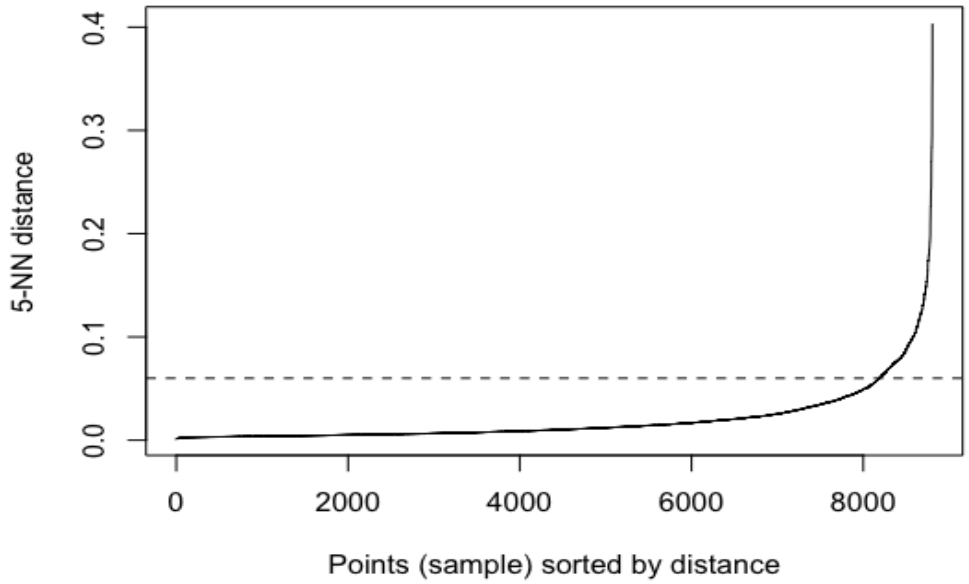
Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**	**Cluster 6**
3111	**1498**	**85**	**544**	**3403**	**164**

Hierarchical clustering provided a different perspective, unveiling a complex structure reminiscent of a family tree. With six main branches, this method revealed a hierarchy of clusters ranging from broad to narrow scopes. The largest branches, one with 3,111 points and another with 3,403, could be seen as the main families that dominate the dataset. In contrast, the smaller

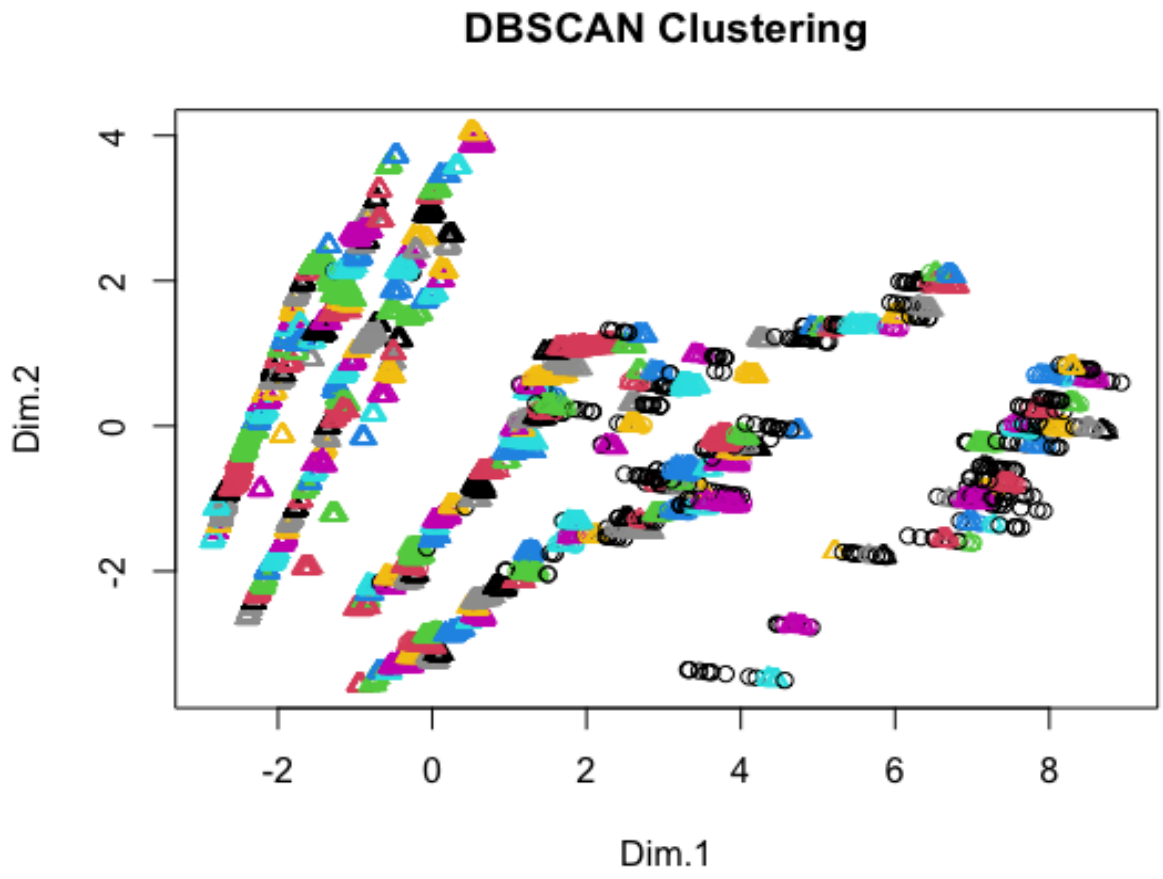
branches, particularly one with only 85 points, appeared to capture the more unique and nuanced patterns within the data.

These branches are not just mere constructs; they represent clusters with distinct average features, much like the common traits shared within a family. The significant differences in these average values across clusters underscore the diversity contained within the dataset and highlight the potential for unique subgroup discoveries.

Density-Based clustering (DBSCAN)



From the above plot, we can see that at 0.06 epsilon value, there is a change in the distance between points. So we consider eps value of 0.06.



After analyzing the DBSCAN, we found total number of outliers is **289**.

Unlike the other two methods, DBSCAN's approach does not yield a central point for each cluster. Instead, it offers a raw, unfiltered look at the data's natural tendency to group, providing a more organic understanding of the dataset's structure. This method is particularly adept at revealing the outlying data points that do not conform to the general patterns, emphasizing the diversity and complexity of the dataset.

DBSCAN approached the data landscape with an eye for density, searching for areas where data points congregate. This method illuminated a variety of clusters, some as populous as over 200 points, which we could view as common social gatherings in the data realm. In stark contrast, a considerable number of points did not belong to any cluster and were deemed noise, representing the outliers or the solitary figures of the dataset.

Overall, The K-Means clustering suggests a division of the dataset into periods characterized by different levels of stock prices and economic indicators. The largest clusters (1, 2, and 6 in K-Means; 1 and 5 in Hierarchical) likely represent common market conditions, while the smaller clusters (Cluster 5 in K-Means; Cluster 3 and 6 in Hierarchical) might indicate more unique or rare economic situations. Hierarchical clustering adds depth by showing how these periods are connected or nested within each other, offering insights into the evolution of market conditions over time. DBSCAN's results underscore the complexity and irregularity inherent in the dataset. The 289 outliers identified may not fit neatly into the patterns observed in the other clusters, suggesting unique behaviors or exceptions to the general trends.

Question 4 - Comparative Analysis

The comparative analysis of our dataset was performed in Python. This analysis evaluates the performance of various machine learning classifiers in predicting categorical outcomes of stock prices. By transforming the continuous 'Close' price into discrete categories, the task is reformulated as a classification problem, enabling the use of a broader range of analytical techniques. The dataset consists of 8,805 records, each including economic indicators such as GDP nominal, unemployment rate, and federal funds rate, alongside the stock 'Close' price. The data was divided into quintiles to convert the continuous 'Close' price into a categorical variable, creating five classes representing different levels of closing price ranges.

Methodology

Data preprocessing was conducted, with numerical features standardized and categorical variables one-hot encoded. Missing values were imputed using the mean for continuous variables, preserving the dataset's integrity. Independent features Monthly_Nominal_GDP_Index, Monthly_Real_GDP_Index, Sourceapple, Sourcefacebook, Sourceamazon, Sourcegoogle, and Sourcenetflix were selected based on their correlation value is better with the target variable Close for the SVM model.

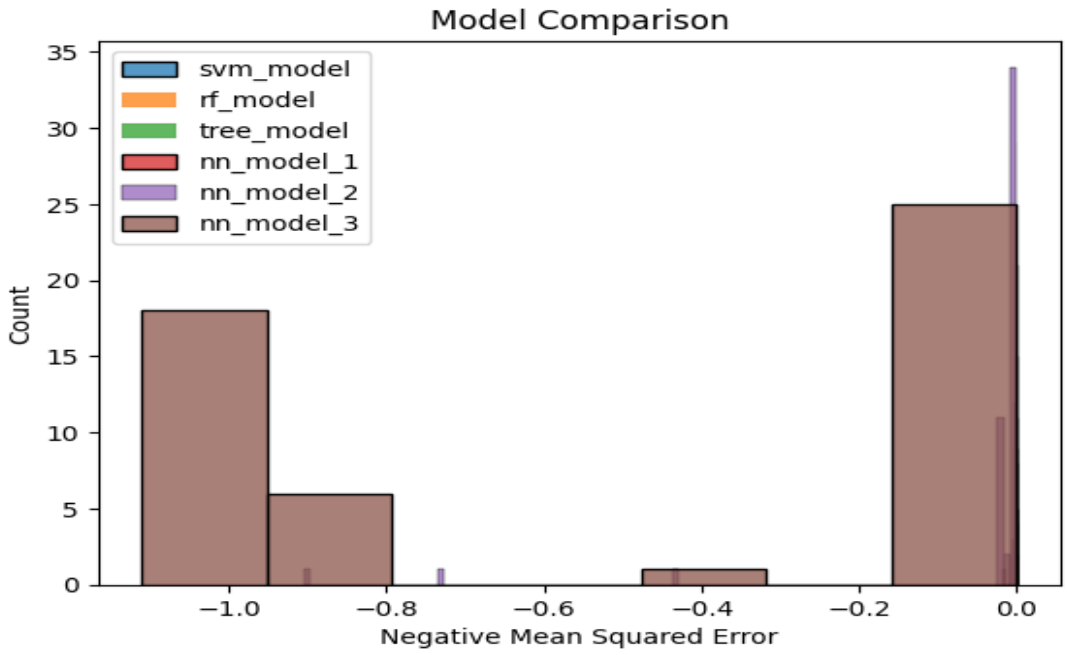
Four classifiers were selected for comparison: Support Vector Machines (SVM), Random Forests (RF), Neural Networks (NN) with various hidden layer configurations, and Decision Trees (DT) using the C5.0 algorithm. The CARET package in R was utilized for model training and validation, employing three distinct cross-validation techniques: K-Fold, Repeated Random Subsampling, and Bootstrapping. These methods not only allow for a thorough evaluation of model performance but also aid in examining the generalizability and stability of each classifier.

Each classifier was subjected to training and evaluation processes. The SVM was implemented with a linear kernel, chosen for its effectiveness in high-dimensional spaces. RF was configured with 100 decision trees to reduce variance. NN models were tested with different architectures to identify the optimal network complexity for the given task. The DT model was constructed using the C5.0 algorithm, known for its interpretability and ease of use.

Performance metrics were calculated for each model across all cross-validation folds. These metrics provided a comprehensive assessment of each classifier's predictive capabilities.

Findings

1. Model Comparison:

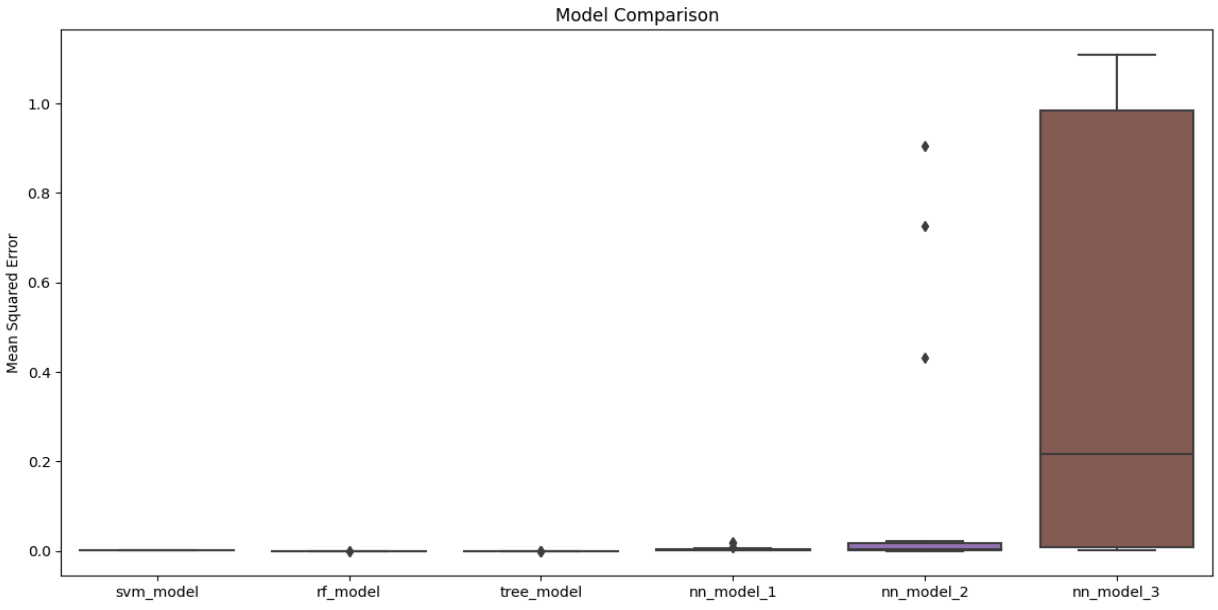


- SVM Model: The SVM model demonstrates a distribution of errors with a moderate range, indicating some variability in its performance across the dataset. The NMSE values are generally concentrated around -0.8, suggesting that while the model has a decent predictive capability, there is potential for improvement through hyperparameter tuning and optimization.

- RF Model: The histogram for the RF model shows a highly concentrated distribution of errors near zero. This tight clustering of low NMSE values is indicative of strong and consistent performance, making it the standout model in terms of both accuracy and reliability.

- NN Models: The performance of the three NN models (nn_model_1, nn_model_2, nn_model_3) is variable. Notably, nn_model_1 and nn_model_2 appear to have a limited number of counts, which may suggest a narrow evaluation or issues with the model training. nn_model_3, however, shows a broader range of NMSE values, with a significant number of counts close to zero but also a long tail towards larger errors. This implies that while the model can achieve high accuracy, it may lack consistency across different folds of the cross-validation.

2. Model Comparison and Mean Squared Error



Model Performance Indicator: Each boxplot represents the spread of MSE scores across cross-validation folds for each model. The central line in the box represents the median MSE, the box's edges represent the interquartile range (IQR, which is the middle 50% of the data), and the whiskers extend to the rest of the distribution, excluding outliers represented by diamonds.

SVM Model (svm_model): The SVM model has the lowest median MSE, indicating high predictive accuracy. Additionally, the IQR is very narrow, suggesting consistent performance across different cross-validation folds.

Random Forest Model (rf_model): Similar to the SVM model, the RF model also shows a low median MSE and a small IQR, indicating both accuracy and consistency in performance. The lack of outliers suggests that the model's performance is robust across different data splits.

Decision Tree Model (tree_model): The Decision Tree has a slightly higher median MSE than the SVM and RF models and a similar IQR, indicating a reasonable level of consistency but less predictive accuracy.

Neural Network Models (nn_model_1, nn_model_2, nn_model_3): The first two neural network models (nn_model_1 and nn_model_2) have very low median MSEs and extremely narrow IQRs, suggesting that they perform very well consistently across the cross-validation folds. However, nn_model_3 shows a significantly higher median MSE and a much larger IQR, indicating less predictive accuracy and greater variability in performance.

Outliers: The presence of outliers in nn_model_3 indicates that in some folds, the model performed much worse than in others, which may point to overfitting or instability in this particular neural network configuration.

Interpretation:

- The SVM and RF models appear to be the top performers, with SVM having a slight edge in terms of lower median MSE and less variability.

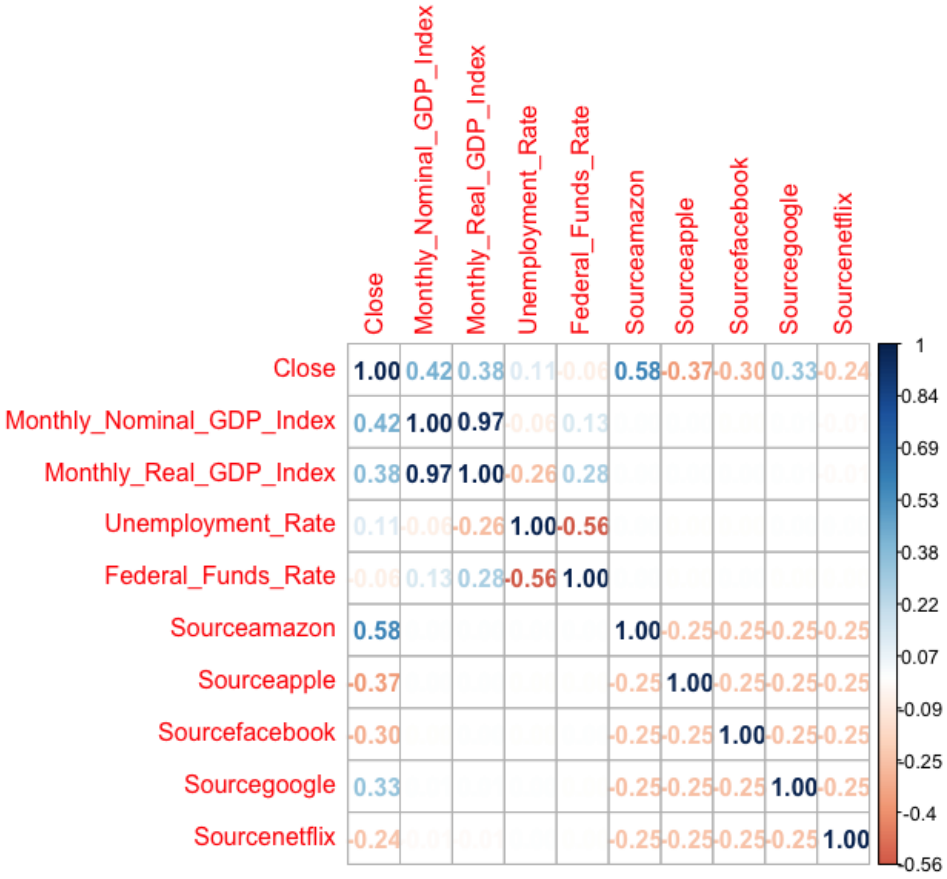
- The Decision Tree model is slightly less accurate than SVM and RF but does not show extreme variance in performance.
- The first two neural network configurations (nn_model_1 and nn_model_2) are competitive with very low MSEs, indicating that they might be well-tuned for the dataset.
- The third neural network configuration (nn_model_3) appears to be the least reliable, with a high median MSE and a large spread in errors, which could be due to overfitting or an inappropriate model architecture for the dataset.

Question 5 - Feature Selection

We perform three different feature selections on three different models; the Filter Method with SVM (Support Vector Machine), the Recursive Feature Elimination (RFE) with RandomForest, and the Wrapper Method with Linear Regression. For this analysis, we consider the subset of independent variables that we mentioned in our research question (Monthly_Nominal_GDP_Index, Monthly_Real_GDP_Index, Unemployment_Rate, Federal_Funds_Rate, Sourceamazon, Sourceapple, Sourcefacebook, Sourcegoogle and Sourcenetflix).

1. Filter Method on SVM model:

For this method, we visualize the correlation matrix as shown. Based on the matrix, Unemployment_Rate and Federal_Funds_Rate have correlation values near to 0 with respect to the dependent feature Close. Consequently, these two features were eliminated, and Models 1, 2, and 3 were trained. With each model iteration, the R-squared decreased by a very small margin. However, the removal of these variables will simplify the model.



Selected Features: The predictor variable Close has a strong positive correlation with Sourceamazon (0.58), and moderate positive correlations with Monthly_Nominal_GDP_Index (0.42), Monthly_Real_GDP_Index (0.38) and Sourcegoogle (0.33). This suggests that the stock's closing price is more likely to increase with the rising values of these four variables. Features such as , Sourceapple, Sourcefacebook, and Sourcenetflix exhibited a negative correlation with Close feature.

Independent features Monthly_Nominal_GDP_Index, Monthly_Real_GDP_Index, Sourceapple, Sourcefacebook, Sourceamazon, Sourcegoogle, and Sourcenetflix were selected based on their correlation value is better with the target variable Close for the SVM model.

SVM Model Performance:

- **Mean Squared Error (MSE)**: The model yielded an MSE of 0.01210371. This value, being close to zero, indicates a high level of accuracy in the model's predictions. Lower

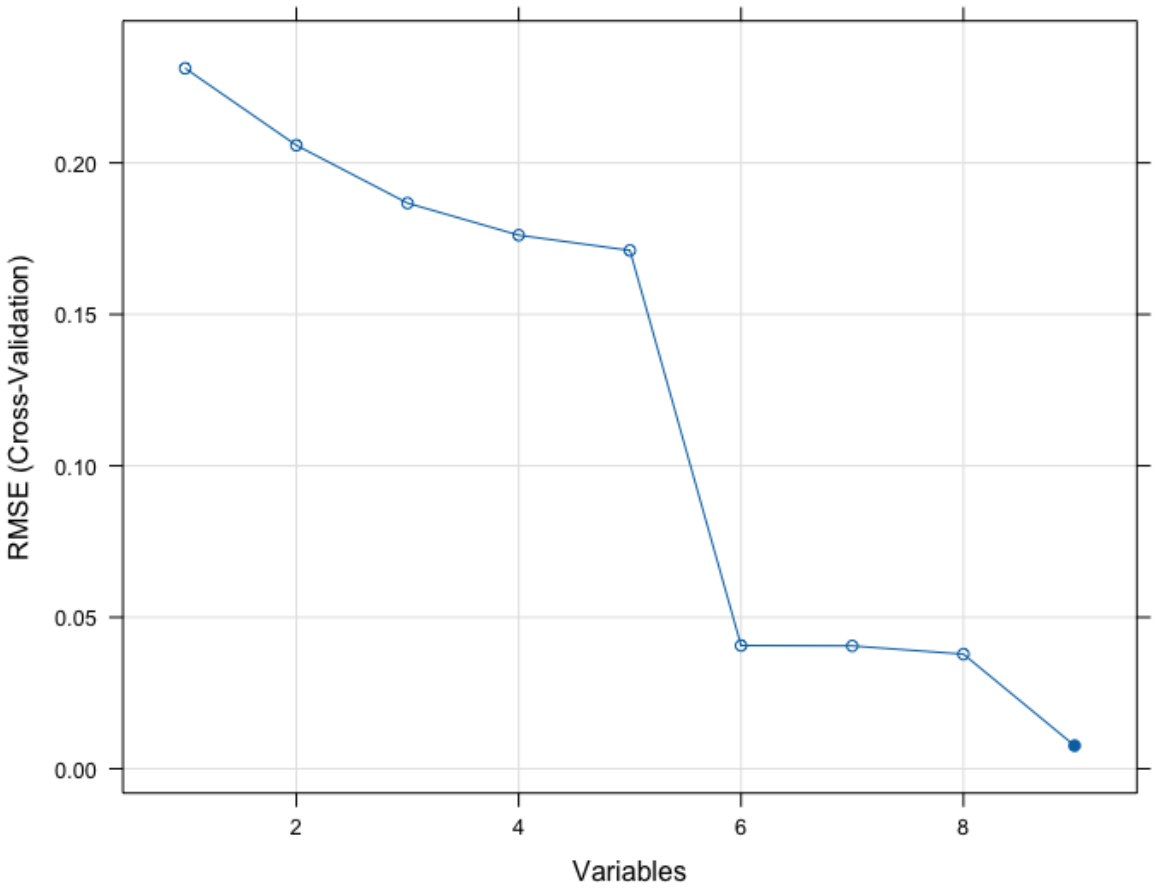
MSE values are desirable as they indicate minimal deviation between the predicted and actual values.

- **R-squared**: The R-squared value was 0.7013283. This means that approximately 70.13% of the variability in the Close prices can be explained by the model. An R-squared value closer to 1 indicates a better fit of the model to the data.

Findings:

The SVM model, guided by the Filter Method, demonstrates a strong ability to predict the Close prices, as evidenced by a low MSE and a high R-squared value. The model's reliance on economic indicators (GDP indices) and specific tech companies' stocks (like Amazon and Google) highlights these factors' significant impact on stock price movements.

2. Recursive Feature Elimination (RFE) on RandomForest model:



For RFE method, we generated the cross-validation plot as shown above. From the plot, we can see the number of features increases from 2 to around 5, there's a noticeable decrease in the RMSE. This suggests that adding more features up to this point improves the model's predictive accuracy. Based on this plot, the optimal number of features for the model might be around 5, where the RMSE is relatively low. Selecting more than 5 features does not appear to provide a substantial decrease in RMSE, implying that a model with 5 features may offer the best balance between model complexity and predictive power.

Selected Features: Features identified as most important include are Sourcenetflix, Sourcefacebook, Federal_Funds_Rate, Monthly_Nominal_GDP_Index and Sourceamazon

The inclusion of specific sources like Sourcenetflix and Sourcefacebook indicates their predictive power in the model. Similarly, economic indicators like the GDP indices and Federal_Funds_Rate suggest their strong influence on stock prices.

Findings:

The RFE method with RandomForest model identified a mix of stock sources and economic indicators as key drivers of stock price. The diversity in the types of selected features (from tech company stocks to macroeconomic variables) underscores the multifaceted nature of factors influencing stock prices.

3. Wrapper Method on Linear Regression model:

In this feature selection process, we began with a basic model including only an intercept and then constructed a comprehensive model with all predictors. Using a forward stepwise algorithm, we progressively added predictors to enhance model accuracy while balancing complexity. We end up in fitting a refined model, which identifying the most significant predictors for the stock closing price.

Selected Features: The final model included Sourceamazon, Sourcegoogle, Monthly_Nominal_GDP_Index, Unemployment_Rate, Sourceapple, Federal_Funds_Rate, Sourcefacebook, and Monthly_Real_GDP_Index.

Model Performance

- **Residual Standard Error**: 385.4, suggesting the average distance between the observed values and the model's predicted values is approximately 385.4 units.
- **Multiple R-squared**: 0.7861, indicating that about 78.61% of the variation in the Close price is explained by the model, signifying a strong fit.
- **F-Statistic**: A very high F-statistic (3231) on 8 and 7035 degrees of freedom implies the model is statistically significant.

Findings

The Linear Regression model using the Wrapper Method demonstrates a high degree of predictive accuracy for stock prices, as indicated by a high R-squared value. The selection of features is consistent with the other methods, reinforcing the importance of both company-specific and macroeconomic indicators in determining stock market movements.

Feature Selection Summary

Across different methodologies, the models consistently identify key factors influencing stock prices, including economic indicators like GDP indices and the Federal Funds Rate, as well as specific technology companies' stock movements. This consistency across models strengthens the reliability of these findings. The models show a strong fit and predictive power, suggesting their effectiveness in understanding and predicting stock market behavior.

Question 6 - Extra Credit

Our research question focuses on predicting the future stock prices using historical stock price of specific companies and macroeconomic indicators like the nominal GDP index, real GDP index, unemployment rate, and federal funds rate. Though our datasets don’t involve data gathered from individual-based research, there are several ethical issues worth considering during the phases of data collection and model evaluation.

Data Collection

In the data collection phase, the project utilizes three primary data sources: historical stock prices from Kaggle, the GDP index from S&P Global Market Intelligence, and the unemployment rate from the Federal Reserve Economic Data (FRED), originally sourced from the U.S. Bureau of Labor Statistics.

The historical stock price data, as mentioned by its author, was gathered through web scraping from the Yahoo Finance website, covering the period from 2015 to 2021. While this approach provides extensive coverage, ethical considerations arise regarding the use of web scraping techniques. It is important to ensure that this method complies with the terms of service of the website and respects data usage policies. Additionally, the method of data extraction can sometimes lead to incomplete or skewed data, depending on the scraping algorithm's design and the website's structure. This could affect the dataset's accuracy and representativeness. Also, dependence on secondary sources like Kaggle limits our control over data quality and independent verification of its accuracy.

The influence of human emotions and isolated events, such as the COVID-19 pandemic, on stock prices is a significant consideration. The pandemic not only affected economic indicators but also significantly influenced human emotions and behavior, factors that are difficult to quantify. Panic, optimism, and other emotional responses play a crucial role in financial decision-making. However, these abstract factors are often not directly represented in traditional financial datasets. Our project does not include such abstract factors as investor sentiment or market panic, which can significantly impact stock market behavior. Excluding these variables could result in a gap in comprehensively understanding the factors driving stock market trends. To ensure transparency, it's our duty to inform users and acknowledge our model’s limitations.

Model Evaluation

Another ethical concern arises from the limited diversity of companies included in our study. The companies we selected are not diverse enough in terms of industry representation. This lack of diversity could lead to models that are biased towards stable, large-cap companies and may not accurately represent the broader stock market, including smaller or stocks from different

industries. Such a limitation might result in predictions that are less applicable to the entire market, potentially misleading users who deal with a more diverse situation. Additionally, focusing on a narrow range of companies may amplify the model's biases, as it reflects the trends and behaviors of only a specific segment of the market.

Evaluating the model also involves considering how the model’s predictions might influence investor behavior, possibly leading to unintended consequences like investor losses. It's important to recognize the limitations of our models in capturing the complex interplay of quantitative and qualitative factors that affect stock prices. This acknowledgment is crucial for users of our research, as it informs them about the potential limitations of our predictions. Our responsibility includes being transparent about the model's limitations in accounting for unpredictable elements and the associated risks in basing investment decisions on these predictions.

Contribution Section

Question 1: developed by Vijay Arni;

Question 2: developed by Ya-Ting Yang;

Question 3: developed by Srikanth Parvathala;

Question 4: developed by Vijay Arni

Question 5: developed by Srikanth Parvathala

Question 6: developed by Ya-Ting Yang;

Vijay Arni prepared 100% of the R code for question 1 and 4;

Ya-Ting Yang prepared 100% of the R code for question 2;

Srikanth Parvathala prepared 100% of the R code for question 3 and 5;

All members contributed equally to the preparation and recording of the presentation;

References:

3.1. Cross-validation: evaluating estimator performance. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/cross_validation.html

Alexius, A. (2018). *Stock prices and GDP in the long run*. https://econpapers.repec.org/article/sptapfiba/v_3a8_3ay_3a2018_3ai_3a4_3af_3a8_5f4_5f7.htm

Farsio, F., & Fazel, S. (2013). *The Stock Market/Unemployment Relationship in USA, China and Japan. International Journal of Economics and Finance, 5(3)*. https://doi.org/10.5539/ijef.v5n3p24

Federal funds effective rate. (2023, October 2). https://fred.stlouisfed.org/series/FEDFUNDS

Historical Stock Price of (FAANG + 5) companies. (2021, December 30). Kaggle. https://www.kaggle.com/datasets/suddharshan/historical-stock-price-of-10-popular-companies/code?select=Microsoft.csv

Hwang, J. (2021, December 13). How to manipulate & visualize data for comparative analysis—Crunch time in the NBA. *Medium*. https://towardsdatascience.com/how-to-manipulate-visualize-data-for-comparative-analysis-crunch-time-in-the-nba-f20540e23b54

Kumbure, M. M., Lohrmann, C., Luukka, P., & Porras, J. (2022). Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems With Applications*, *197*, 116659. https://doi.org/10.1016/j.eswa.2022.116659

Lateef, Z. (2020, May 15). *Support Vector Machine in R: using SVM to predict heart diseases*. Edureka. https://www.edureka.co/blog/support-vector-machine-in-r/

Le, J. (2018, August 22). *Support vector machines in R*. https://www.datacamp.com/tutorial/support-vector-machines-r

Shiblee, L. S. (2009). The Impact of Inflation, GDP, Unemployment, and Money Supply On Stock Prices. *Social Science Research Network*. https://doi.org/10.2139/ssrn.1529254

Unemployment rate. (2023, September 1). https://fred.stlouisfed.org/series/UNRATE