INST737 - Introduction to Data Science

Milestone - 2 Report

Changes considered after Milestone 1:

After carefully reviewing the feedback for Milestone 1, We decided to add an additional variable Federal Funds Rate which might play a significant role in stock price prediction. Federal fund rates are nothing but the interest rates that banks lend money to each other's. So we're considering that as a predictable variable. We collected this data from the fred.stlouisfed.org website. We also, revised our research question based on the features that we're considering to predict the closing stock price.

Revised research question:

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

Data cleaning & Merging:

Three datasets containing GDP, Unemployment Rate, and Federal Funds Rate data were loaded and their date columns were standardized to a common format. These datasets were then merged based on their date information to form a single dataset (Merged_data.csv).

A separate stock market price dataset was loaded. To facilitate merging with the previously merged dataset, month and year information was extracted from the date columns of both datasets. The stock market price dataset was merged with the previously merged dataset based on this month and year information. Unnecessary columns were dropped post-merging, and the final dataset was saved for further analysis(StockPrediction_data.csv).

This data cleaning and merging datasets process ensures that the datasets are combined in a way that provides a comprehensive view of the independent variables (Nominal GDP, Real GDP,

Unemployment Rate, and Federal Funds Rate) along with the dependent variable (Stock Market Price) for predictive modelling.

Q1. a. Linear Regression

From our dataset, closing stock price is a predictable variable and Monthly Nominal GDP Index, Monthly Real GDP Index, Unemployment rate, Federal fund rates and Source are the independent variables.

We have divided the dataset into training and testing sets with 80 % of data for training and 20% of remaining data for testing set. Since the feature Source is categorical variable, firstly we pre-processed the dataset and created a dummy variables for each source (Sourceamazon, Sourceapple, Sourcefacebook, Sourcegoogle and Sourcenetflix) and considered as our data for further analysis. Below the image of the dummy variables.

Sourceamazon	Sourceapple	Sourcefacebook	Sourcegoogle	Sourcenetflix
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0

Intercept and Coefficient:

Intercept: The predicted value of the response variable (Close) when the predictor is 0.

Coefficient: Represents the change in the response variable for a one-unit change in the predictor. Positive values indicate a positive relationship, while negative values indicate a negative relationship.

Based on the univariate linear regression analyses, here are the results for Intercept, Coefficient values and p-value:

Feature	**Intercept**	**Coefficient**	**p-value**
Monthly_Nominal_GDP_Index	-3.530e+03	2.068e-01	< 2.2e-16

Feature	Intercept	Coefficient	p-value
Monthly_Real_GDP_Index	-6.683e+03	4.001e-01	< 2.2e-16
Unemployment_Rate	432.629	49.565	< 2.2e-16
Federal_Funds_Rate	736.28	-61.29	3.668e-07
Sourceamazon	443.038	1216.589	< 2.2e-16
Sourceapple	839.68	-778.38	< 2.2e-16
Sourcefacebook	808.63	-627.69	< 2.2e-16
Sourcegoogle	544.84	684.65	< 2.2e-16
Sourcenetflix	784.91	502.19	< 2.2e-16

Predictive Feature:

- P-value indicates the significance of the predictor's coefficient. A p-value less than 0.05 typically suggests that the predictor is statistically significant.
- All features have p-values less than 0.05, which indicates they are statistically significant predictors and predictive features.

To determine the "most predictive" features, we typically look at a combination of the p-value and the coefficient. Lower p-values and higher magnitudes of the coefficient (either positive or negative) indicate stronger predictive power. Below are the observations based on the results.

- Sourceamazon has a coefficient of 1216.589, which is the highest value. This indicates that for every unit increase in Sourceamazon, there's an estimated increase of approximately 1216.589 in the 'Close' value, holding all else constant.

- Sourceapple and Sourcefacebook also have high magnitude coefficients but in the negative direction, indicating that increases in these predictors are associated with decreases in the 'Close' value.
- The p-values for all these features are very close to zero, which further emphasizes their statistical significance.

In summary, based on the value of the coefficients and the significance of the p-values, Sourceamazon, Sourceapple, and Sourcefacebook appear to be the most predictive features in the training data. The features with the lowest p-values (closest to 0) are the most predictive. In this case, "Monthly_Nominal_GDP_Index" and "Monthly_Real_GDP_Index" have the strongest relationships with the dependent variable.

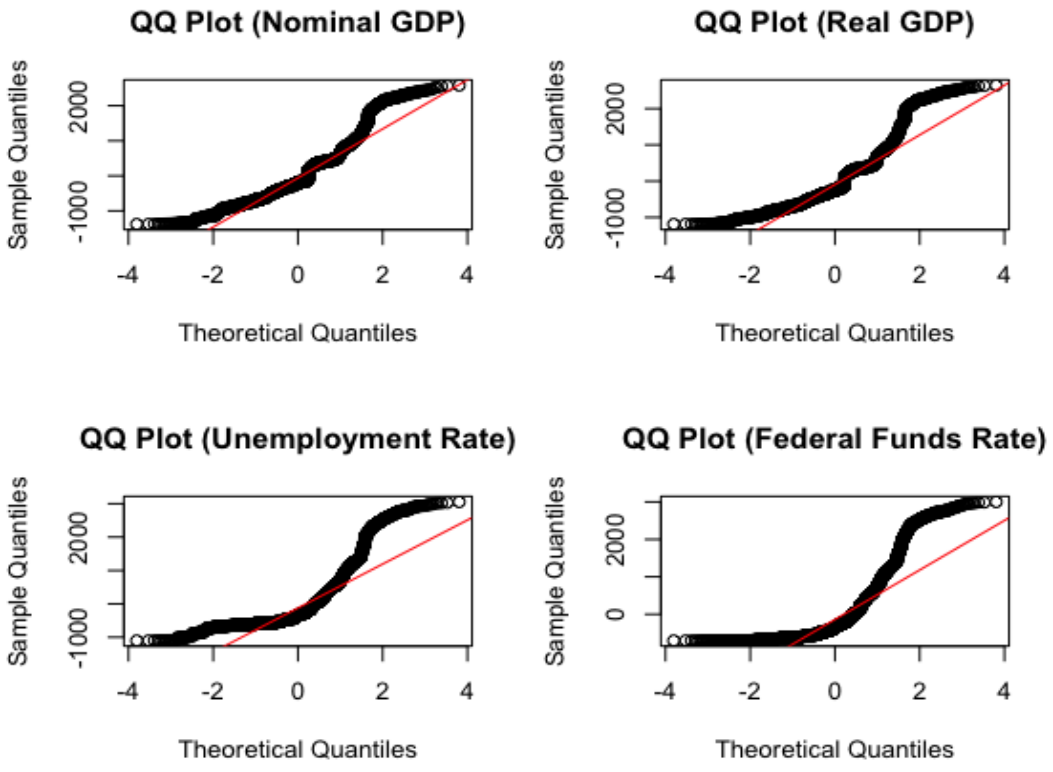
Residuals:

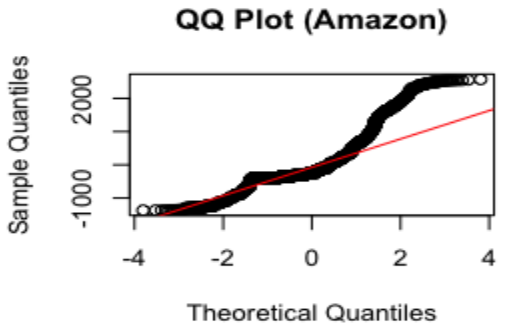
Residuals represent the differences between the observed values (actual 'Close' values in our dataset) and the values predicted by the model. They indicate the amount by which the predictions deviate from the actual data. Ideally, for a perfect model, all residuals would be zero, meaning every prediction matches the actual value.

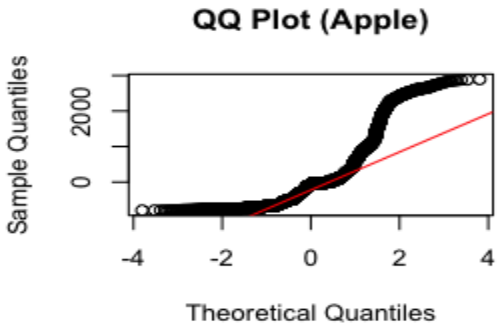
Feature	**Residual Min**	**Residual 1Q**	**Residual Median**	**Residual 3Q**	**Residual Max**
Monthly_Nominal_GDP_Index	-1378.4	-533.2	-210.7	410.9	2581.4
Monthly_Real_GDP_Index	-1186.2	-558.5	-245.9	365.2	2640.7
Unemployment_Rate	-1101.0	-569.4	-338.4	319.2	3055.3
Federal_Funds_Rate	-703.0	-584.6	-352.4	306.8	3001.3
Sourceamazon	-1368.9	-364.6	-254.7	211.2	2571.1
Sourceapple	-794.14	-586.01	-67.54	136.32	2891.72

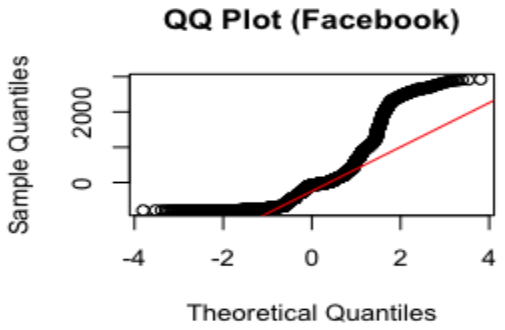
Feature	Residual Min	Residual 1Q	Residual Median	Residual 3Q	Residual Max
Sourcefacebook	-786.04	-666.64	-82.76	177.63	2922.78
Sourcegoogle	-734.7	-450.8	-341.5	13.4	3186.6
Sourcenetflix	-762.3	-646.1	-159.3	251.4	2946.5

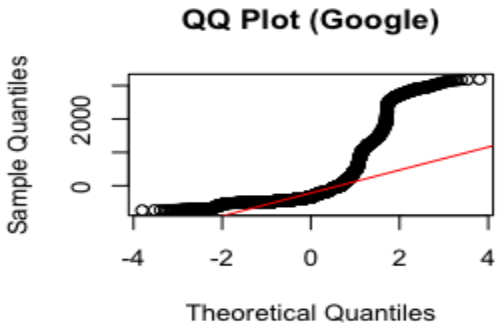
Below graphs are the visually inspect the residuals for the most predictive feature, Monthly Nominal GDP Index, using a QQ plot and residual plot.

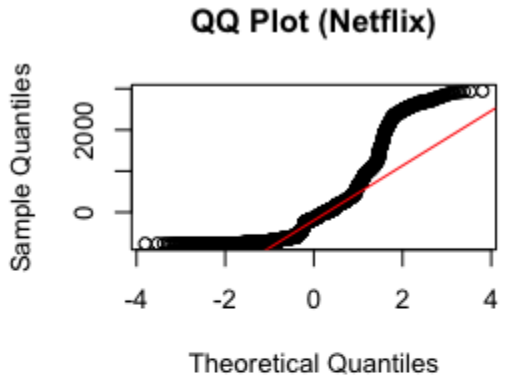






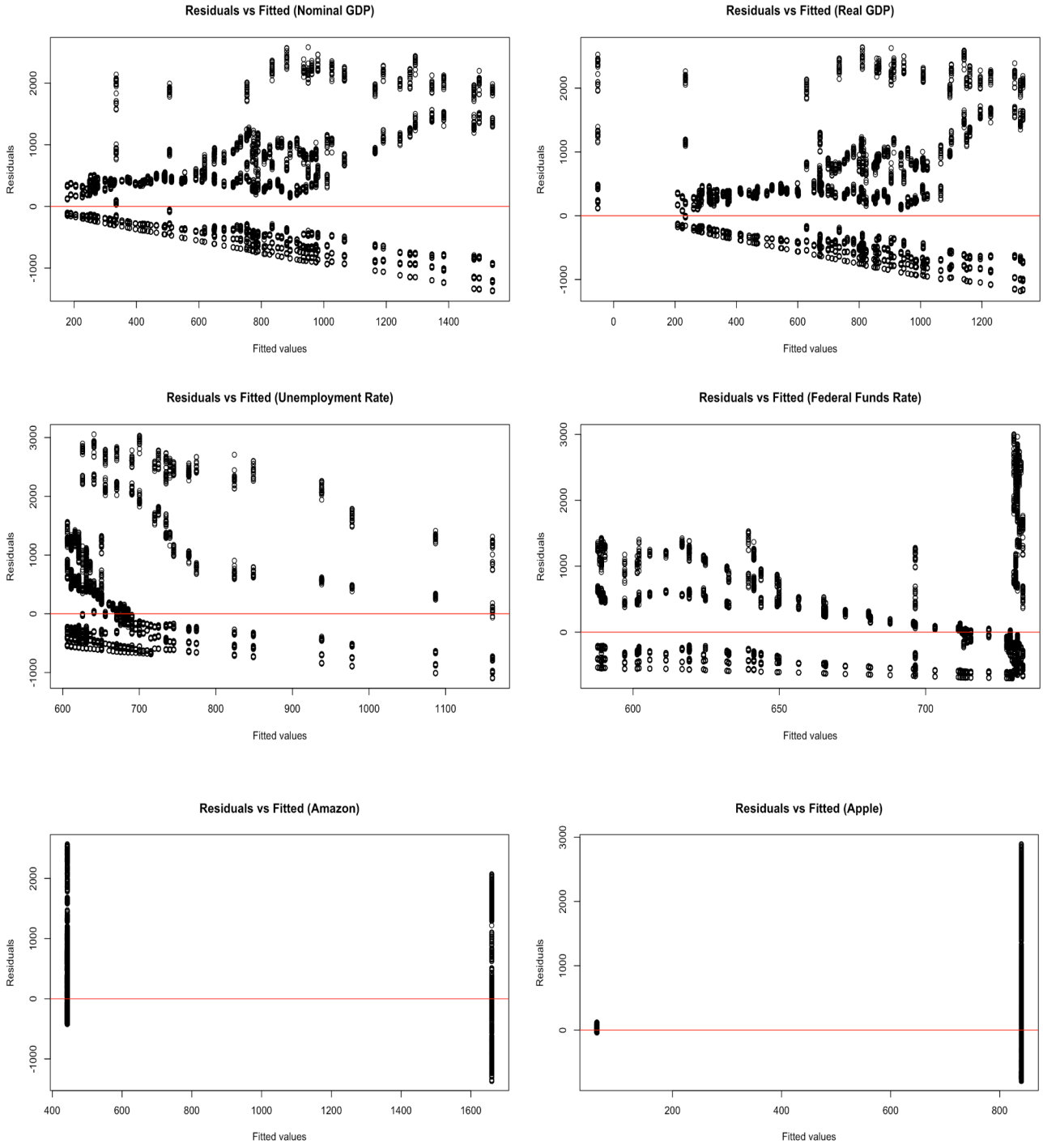


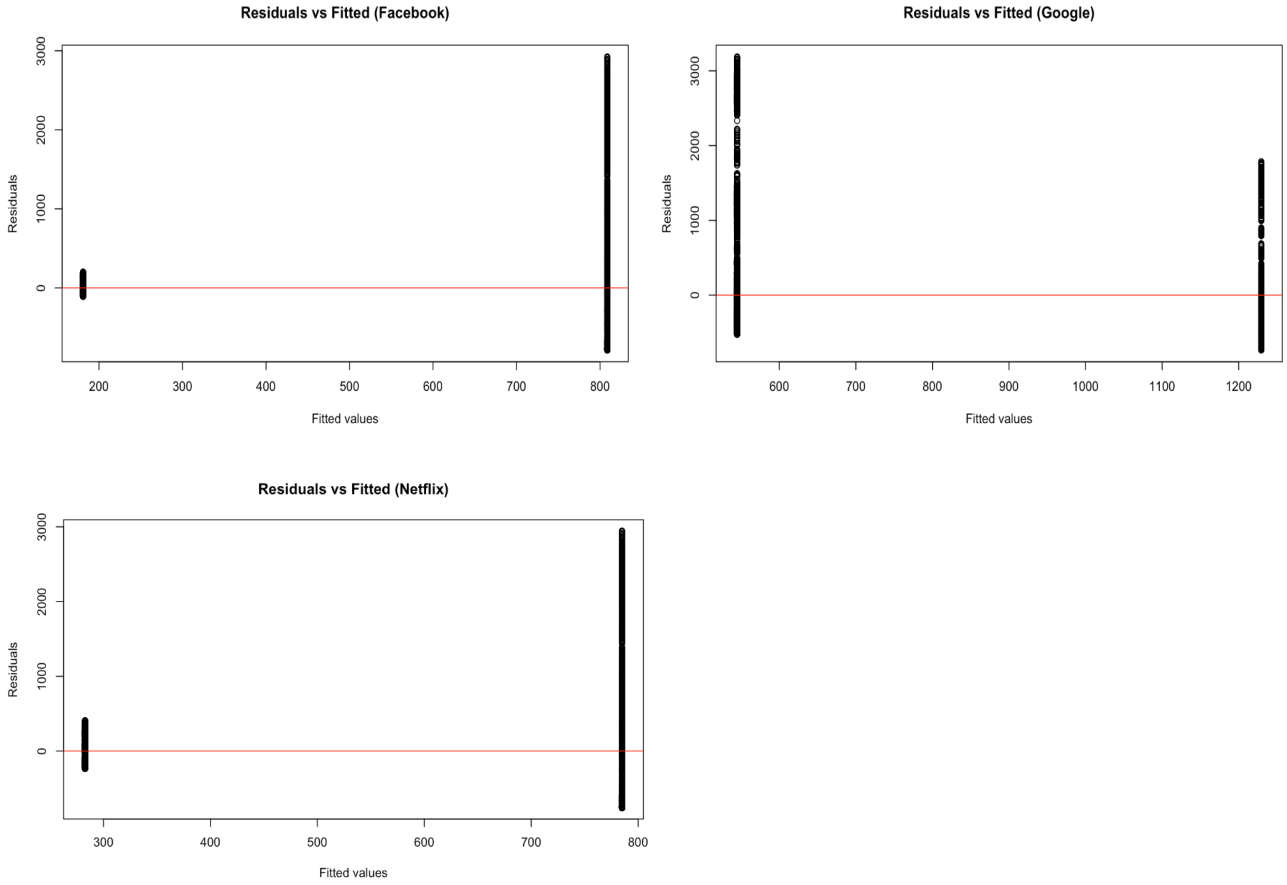




QQ Plot: This plot helps us determine if the residuals are normally distributed. If the residuals follow the red line closely, they are approximately normal. In our case, the points largely follow the straight line in the middle portion, suggesting that the central part of the data's distribution is close to normal.

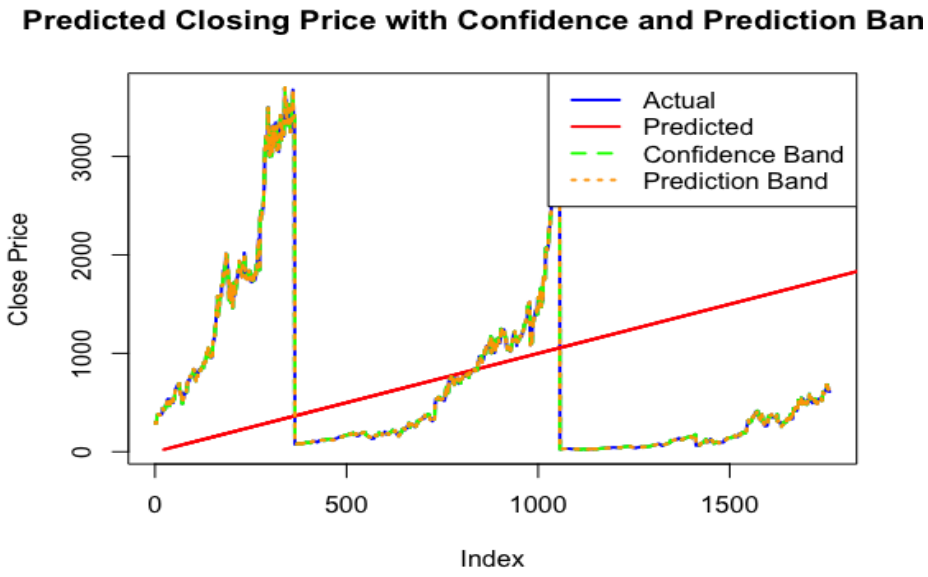
However, there are deviations at the tails, especially in the upper right corner, suggesting that the data may have heavier tails than a normal distribution. This indicates the presence of potential outliers or extreme values in the residuals. This non-normality, especially if it's substantial, can affect the reliability of some statistical tests that assume normally distributed residuals.





This residual plot shows the residuals on the vertical axis and the fitted values on the horizontal axis. If there's a clear pattern in this plot, it indicates potential non-linearity in the data. From the above residual plot, we see the spread of residuals is not constant in the data.
Given the non-linearity observed in the residual plot, deviations from normality in the QQ plot, imply that the variance of the residuals is not equal across all levels of the independent variables. This indicates a simple linear regression model might not be the best fit for this problem.

We used the trained data to predict prediction accuracy using the correlation between the predicted and real values and the mean square error for closing stock prices for the test data are calculated using confidence and prediction bands. The confidence bands give a range for the expected mean closing stock price, while the prediction bands provide a range in which an individual new stock closing price is expected to fall.



The correlation (R-squared) between the predicted and real values is 0.99999983

This indicating that the model can explain almost 100% of the variation in the closing stock prices with the provided features.

The mean squared error (MSE) between the predicted and real closing stock prices is 0.4030519

MSE value suggesting that the model's predictions are very close to the actual values, with minor deviations.

b. Multivariate Regressions

For the multivariate regression model, considering combinations of independent features indeed improves the prediction results. The best models for each number of predictors are based on the adjusted R^2 and AIC values. The higher the adjusted R^2 and the lower the AIC, the better the model is in terms of fit and prediction. From the results, it is evident from the increasing adjusted R^2 values that more predictors are added to the model. The adjusted R^2 value represents the proportion of the variance in the dependent variable that is predictable from the independent variables. A higher value indicates a better fit of the model to the data.

The model with just one predictor, Sourceamazon, has an adjusted R^2 of 0.3392 and has an AIC value of 14684.94. However, by the time we include eight predictors in the model, the adjusted

R^2 increases to 0.78584 and the AIC value decreases to 8. This shows that considering combinations of features improves the model's explanatory power.

Below are the co-efficients of each feature while performing Multivariate regression:

Intercept for multivariate regression is -2888

Predictor	**Coefficient (Estimate)**	**p-value**
Monthly_Nominal_GDP_Index	2.710e-01	< 2e-16
Monthly_Real_GDP_Index	-1.347e-01	0.00779
Unemployment_Rate	3.595e+01	5.88e-15
Federal_Funds_Rate	-5.453e+01	8.12e-14
Sourceamazon	1.368e+03	< 2e-16
Sourceapple	-2.295e+02	< 2e-16
Sourcefacebook	-1.075e+02	1.66e-13
Sourcegoogle	9.337e+02	< 2e-16
Sourcenetflix	NA	NA

Sourcenetflix was dropped due to multicollinearity, so it doesn't have a coefficient in this model.

The most predictive features can be identified by looking at both the magnitude and significance of their coefficients, as well as their individual contributions to the adjusted R^2.

Sourceamazon stands out as the most predictive feature, given its large coefficient (1.368e+03) and the fact that it alone explains about 33.92% of the variance in the stock Close price. Sourcegoogle is another key predictor, with a coefficient of 9.337e+02.

Monthly_Nominal_GDP_Index and Unemployment_Rate also play significant roles, as indicated by their coefficient values.

Other features, while still statistically significant, have a lesser impact on the model's predictive power when considered individually. However, their combined effect, especially in interaction with other predictors, might still be essential for the model's overall accuracy.

The dataset has been split into training and testing sets right after pre-processing the data. To predict the accuracy, the correlation value should be closer to 1 and the mean square error value less. Based on the test data, we calculated correlation and mean square error values for multivariate regression.

- Correlation value: 0.882386292602172
- Mean Square Error (Multivariate Regression): 150745.609776844

Since the correlation value is high which is close to 1, suggesting a strong positive linear relationship. The multivariate regression model seems to be performing well in predicting the stock closing prices, as indicated by the high correlation between predicted and actual values. However, the mean square error (MSE) gives us an idea about the magnitude of the prediction errors. When compared with individual features, multivariate regression has less MSE of 150,745.61 which makes this model prediction accuracy high and might be a better fit for the data.

c. Regularization

For the regularization techniques, we used Ridge (L2 Regularization) and Lasso (L1 Regularization) for both individual features and multivariate features. Here are the results:

Ridge Regression Results

Feature/Predictor	**Correlation**	**Mean Squared Error (MSE)**
Monthly_Nominal_GDP_Index	0.4016	570,809.3
Monthly_Real_GDP_Index	0.3641	590,253.3

Unemployment_Rate	0.0922	674,930.5
Federal_Funds_Rate	0.0508	678,433.9
Sourceamazon	0.6026	434,621.4
Sourceapple	0.3714	586,370.4
Sourcefacebook	0.3025	617,819.6
Sourcegoogle	0.2979	619,773.7
Sourcenetflix	0.2204	647,361.6
Multivariate	0.8827	151,123.1

Lasso Regression Results

Feature/Predictor	**Correlation**	**Mean Squared Error (MSE)**
Monthly_Nominal_GDP_Index	0.4016	571,107.8
Monthly_Real_GDP_Index	0.3641	590,437.7
Unemployment_Rate	0.0922	674,951.4
Federal_Funds_Rate	0.0508	678,477.1
Sourceamazon	0.6026	433,717
Sourceapple	0.3714	586,357.4
Sourcefacebook	0.3025	617,825
Sourcegoogle	0.2979	619,977.4

Feature/Predictor	Correlation	Mean Squared Error (MSE)
Sourcenetflix	0.2204	647,468.4
Multivariate	0.8827	150,329.7

Based on the above results, both Ridge and Lasso regressions offer nearly identical performance in terms of correlation and mean square error. The performance metrics are also very close to those of the multivariate regression without regularization. Thus, for this dataset, adding regularization doesn't seem to substantially improve or deteriorate the prediction results compared to the standard multivariate regression.

d. Repeating Experiments with Different Random Splits

After repeating the experiments 10 times with different random splits of the data, here are the results:

Iteration	Linear Regression	Ridge Regression	Lasso Regression
1	0.8821327	0.882714	0.8827233
2	0.8838896	0.882714	0.8827233
3	0.8839128	0.882714	0.8827233
4	0.8875460	0.882714	0.8827233
5	0.8852347	0.882714	0.8827233
6	0.8912392	0.882714	0.8827233

Iteration	Linear Regression	Ridge Regression	Lasso Regression
7	0.8812801	0.882714	0.8827233
8	0.8842982	0.882714	0.8827233
9	0.8861767	0.882714	0.8827233
10	0.8904527	0.882714	0.8827233

Mean Squared Error (MSE):

Iteration	Linear Regression	Ridge Regression	Lasso Regression
1	141014.2	151123.1	150329.7
2	155334.4	151123.1	150329.7
3	141702.0	151123.1	150329.7
4	141646.7	151123.1	150329.7
5	154649.4	151123.1	150329.7
6	141112.8	151123.1	150329.7
7	153059.1	151123.1	150329.7
8	160421.6	151123.1	150329.7
9	155531.4	151123.1	150329.7
10	150485.0	151123.1	150329.7

Average Correlation values for:

- Linear Regression: 0.8856
- Ridge Regression: 0.8827
- Lasso Regression: 0.8827

Average Mean Squared Error (MSE) for:

- Linear Regression: 149,495.67
- Ridge Regression: 151,123.13
- Lasso Regression: 150,329.68

Across the 10 iterations with different random splits, the models' performance metrics are quite consistent. All three models (linear, ridge, and lasso) have very similar average correlation and mean square error values. This indicates the robustness of the models, and the random splitting of the data doesn't introduce a large variability in the performance metrics.

Q2. Logistic Regression and NB

In our analysis, we sought to predict stock price movements using logistic regression and Naive Bayes classifiers. We utilized various economic indicators as independent features and ensured they were categorized appropriately before running the classifiers.

a. Logistic Regression Analysis

Intercept:

The intercept of our logistic regression model is 0.33549. This value represents the log odds of the stock price increasing when all the independent variables are held constant at zero.

Coefficients and Their Significance:

Monthly_Real_GDP_Index: The coefficient is 0.08511, and it's statistically significant with a p-value of 0.012190.

Unemployment_Rate: The coefficient is 0.11022, and it's highly significant with a p-value of 0.000295.

Federal_Funds_Rate: The coefficient is 0.02404, but it's not statistically significant with a p-value of 0.415609.

Prev_Close: The coefficient is -0.12172, and it's statistically significant with a p-value of 0.016544.

Source: The coefficients for different companies varied. For instance, Apple, Facebook and Netflix were significant with p-values of 0.0121, 0.142, and 0.0012, while Google was not statistically significant with a 0.300 p-value.

Log-Odds and Odds Ratios:

The coefficients represent the change in the log-odds for a unit increase in the predictor. To interpret the coefficients in terms of odds ratios, one would exponentiate the coefficients. For instance, the odds ratio for Monthly_Real_GDP_Index would be exp(0.08511).

Most Predictive Features:

Based on the magnitude and significance of the coefficients, the most predictive features in our training data were Unemployment_Rate, Prev_Close, and the Source for specific companies.

Predictions on Testing Dataset:

We utilized our trained logistic regression model to predict the direction of stock movement on the testing dataset. The model's performance, as evaluated by the AUC, was 0.5014185, suggesting that the model's performance was better than random guessing.

b. Naive Bayes Analysis:

Training and Testing:

We divided our dataset into training and testing sets and trained the Naive Bayes classifier. The confusion matrix for the predictions without the Laplace estimator was:

	Predicted Down	Predicted Up
Actual Down (Laplace)	354	460
Actual Up (Laplace)	412	535

Laplace Estimator:

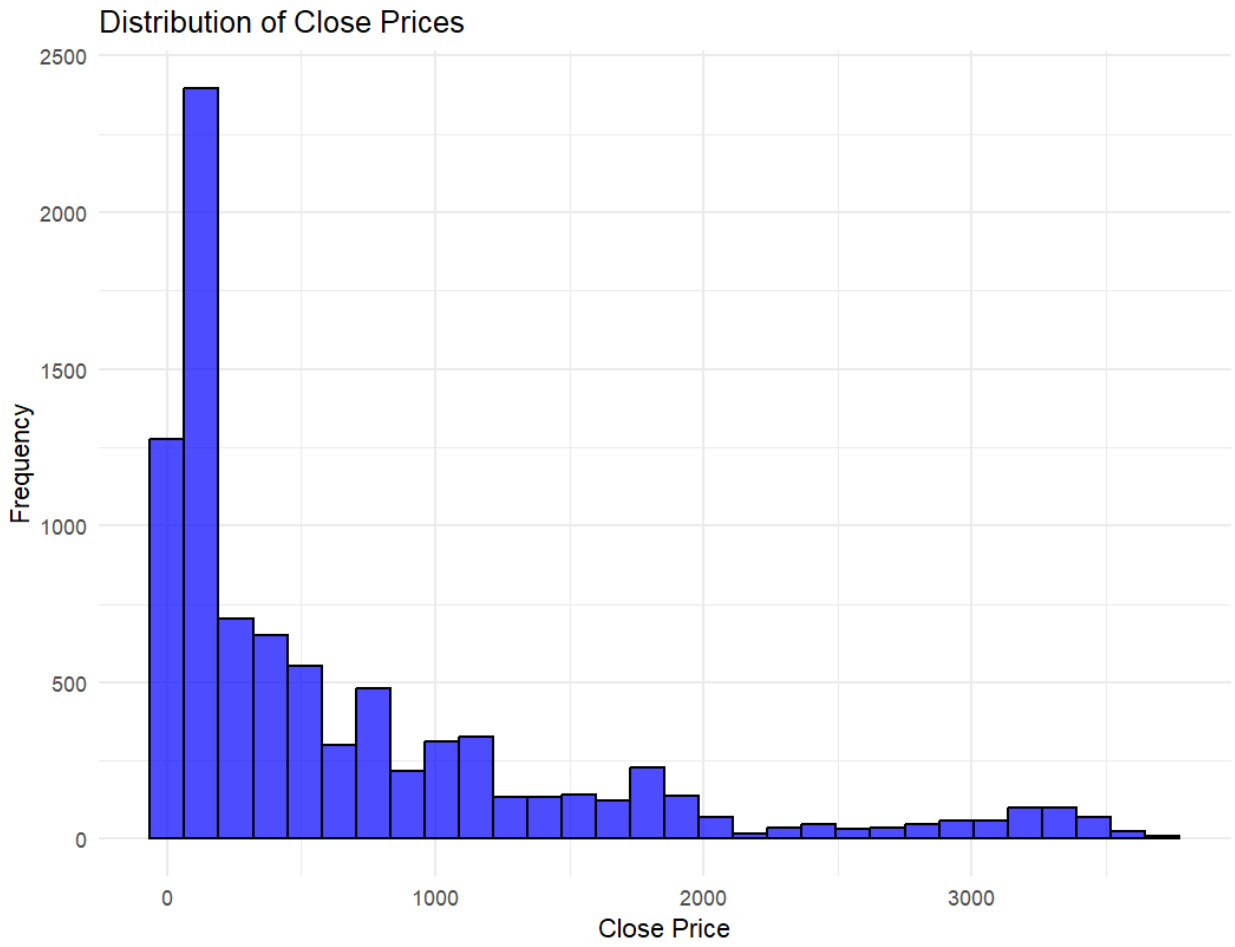
We repeated the process using the Laplace estimator. The confusion matrix for these predictions was nearly the same as without using Laplace Estimator:

	Predicted Down	Predicted Up
Actual Down	354	460
Actual Up	413	534

The results with and without the Laplace estimator were nearly identical, suggesting that the Laplace estimator did not significantly improve the classifier's performance in this instance.

Visualized Data Report:

- 1. Histogram of Close Prices: This histogram showcases the distribution of close prices for a certain stock or financial instrument.
- Most of the close prices are clustered in the lower range, specifically around the 0-500 range, indicating that the stock typically closes within this price range. As the price increases beyond 500, the frequency drops significantly, suggesting that higher closing prices are less common.



2. Correlation Matrix:

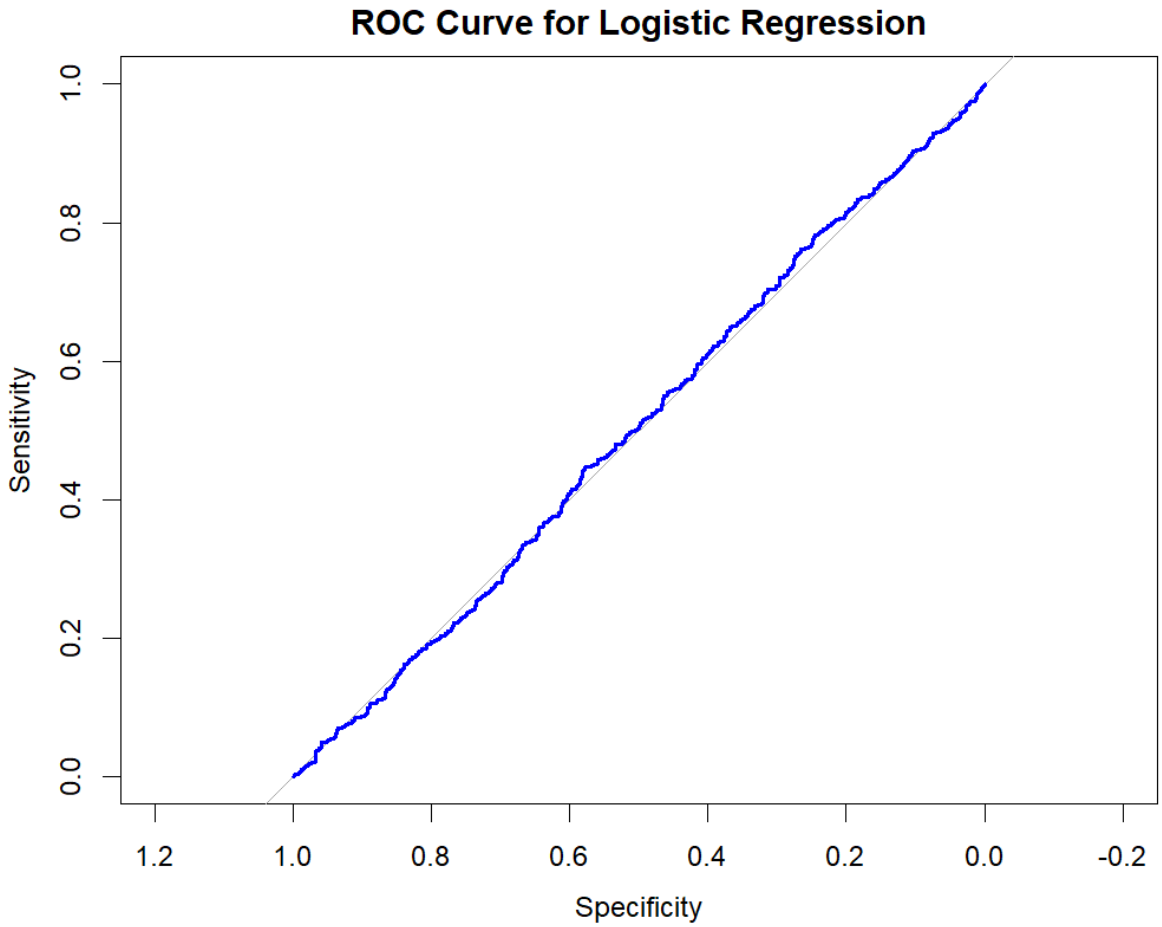
- This matrix represents the correlation between different economic indicators and some other factors.
- The size and color of each circle represent the magnitude and direction of the correlation respectively. For example:
- Monthly_Real_GDP_Index and Unemployment_Rate appear to have a negative correlation, as indicated by the red color.
- Federal_Funds_Rate seems to have a very slight positive correlation with "Prev_Close", as indicated by the blue shade.

3. ROC Curve for Logistic Regression:

- The below image is the Receiver Operating Characteristic curve. The ROC curve is a graphical representation used to assess the performance of a binary classifier system.
- The 45-degree diagonal line represents a random classifier (no discriminative power). A good model will have its ROC curve located above this line.

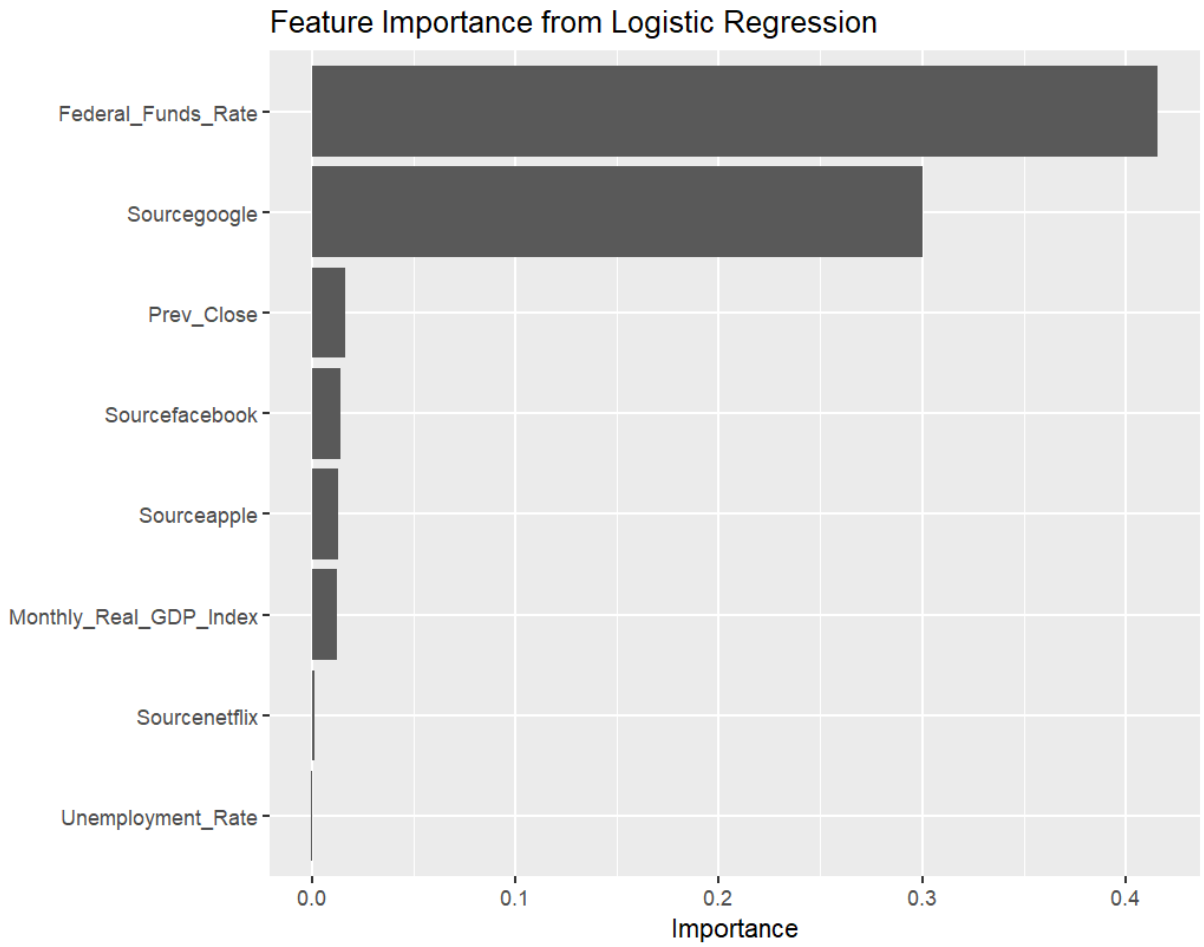
The curve is relatively close to the 45-degree line, suggesting that the logistic regression model's predictive ability might be close to random. However, for a comprehensive evaluation, the Area

Under the Curve AUC value should also be considered.



4. Feature Importance from Logistic Regression:

- This bar graph represents the importance of various features in the logistic regression model.
- "Federal_Funds_Rate" seems to have the highest importance followed by "Sourcegoogle" and "Prev_Close". This means that changes in the federal funds rate have the highest impact on the logistic regression model's predictions.
- On the other hand, factors like "Sourcenetflix" and "Unemployment_Rate" have comparatively low importance in the model.



Conclusions:

The logistic regression model suggests that among the predictors, Unemployment_Rate is the most significant in predicting the direction of stock movement.

- The performance of the logistic regression model, as indicated by the AUC, is slightly better than a random guess but might benefit from further refinement or inclusion of additional relevant predictors.
- The Federal_Funds_Rate, despite having a coefficient, is not statistically significant.
- The Naive Bayes classifier, with or without the Laplace estimator, produced the same classification results on the testing dataset. This suggests that the Laplace estimator did not have an impact on the classification for this particular dataset.

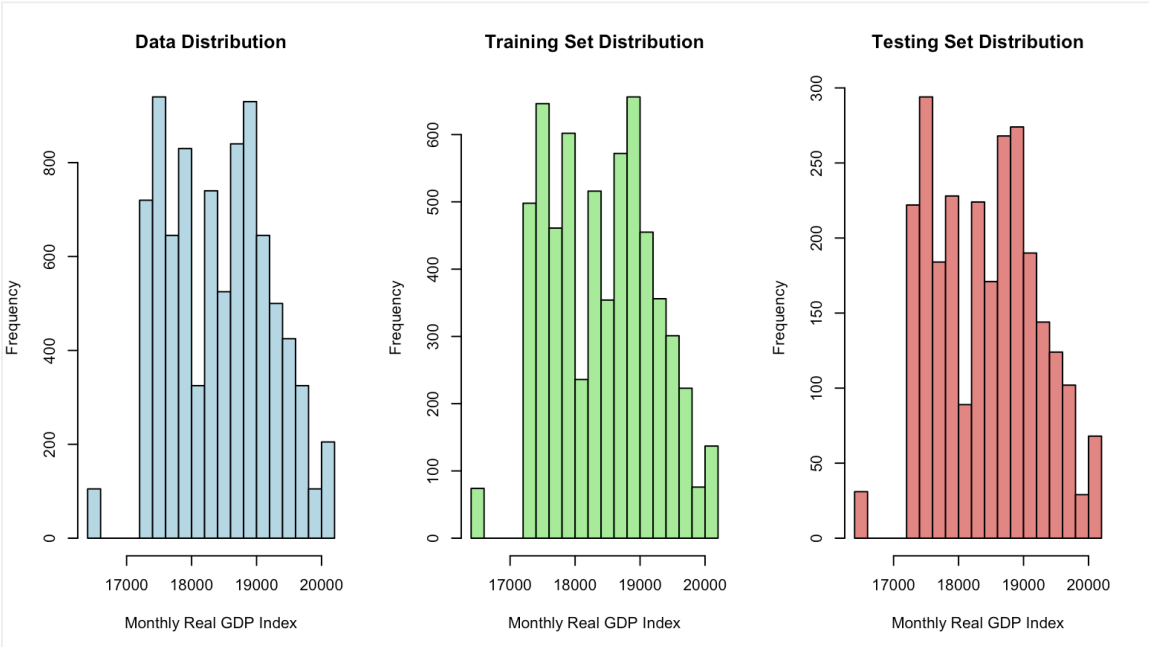
Q3. Decision Trees and Random Forests.

a. Split Dataset Into Training and Testing Sets

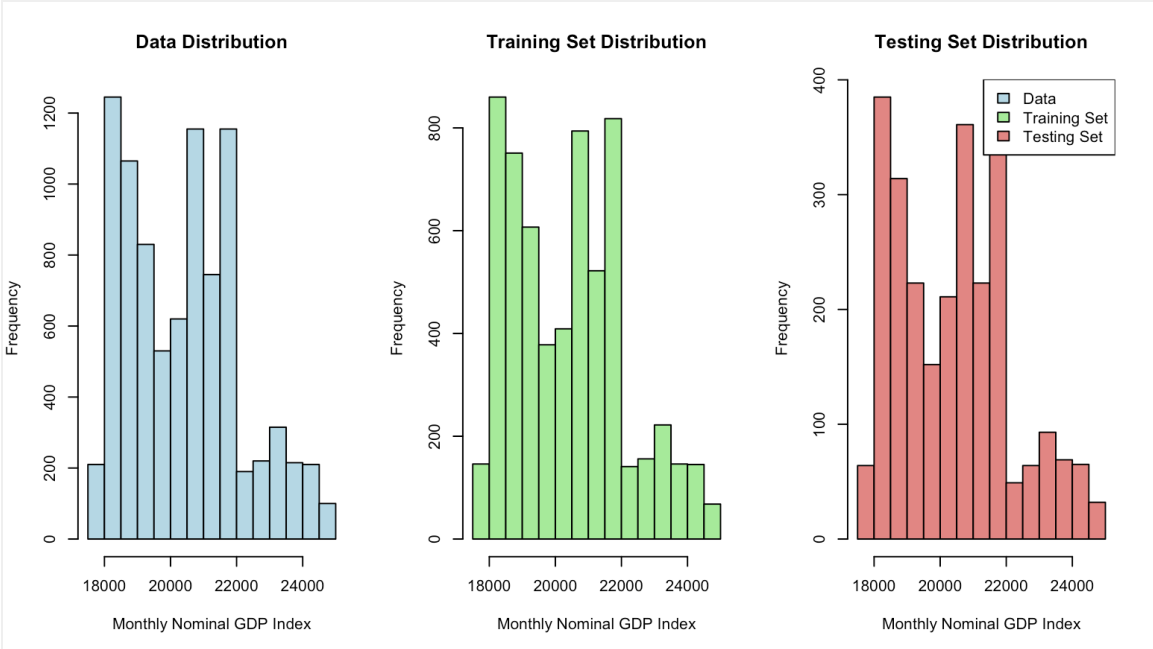
We adopted different data splitting approaches as we recognized their significant impact on prediction accuracy. Initially, we randomized the data and split it into an 80-20 ratio. However, when we used these datasets for training the decision tree, this yielded low accuracy (0.0005678592 on testing data, 0.060477 on training data) with a tree size of 373.

To enhance accuracy, we used a method of categorizing data based on the "Close" column. This involved determining the minimum and maximum values and calculating the range width to divide data into distinct classes. We further applied K-fold cross-validation to improve data division. As a result, accuracy greatly improved, with accuracy on the training data increasing from 0.060477 to 0.2881317 and on the testing data from 0.005678592 to 0.2556818.

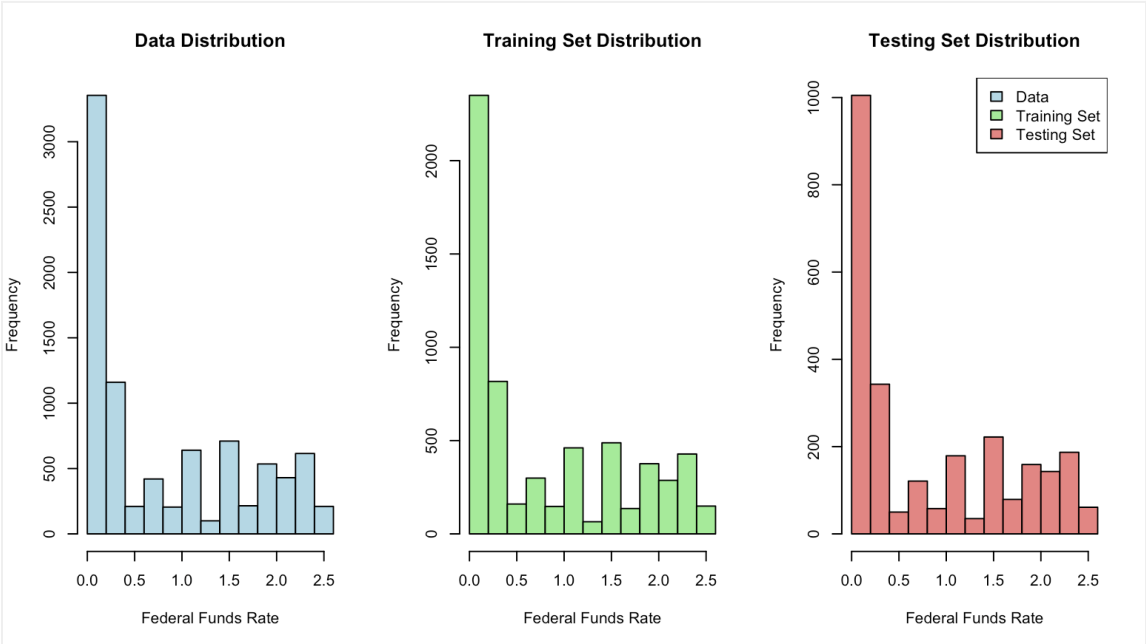
Finally, In order to make sure that the distribution of training and testing data sets are similar to the original datasets, we chose four independent variables for illustration, including Monthly Real GDP Index, Monthly Nominal GDP Index, Federal Funds Rate, and Unemployment Rate. The resulting distribution plots of these four variables are presented below.



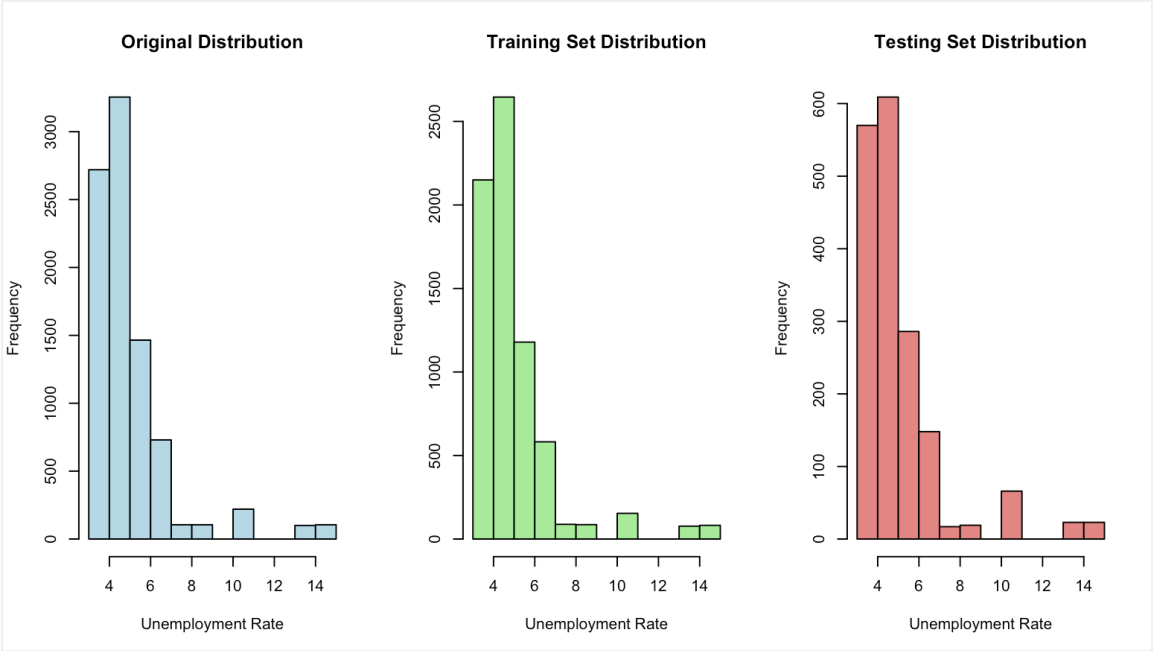
Distribution of Monthly Real GDP Index



Distribution of Monthly Nominal GDP Index



Distribution of Federal Funds Rate



Distribution of Unemployment Rate

b. Decision Tree and the Confusion Matrices

The Decision Tree Incorporating 4 Variables

We employed the C50 algorithm to train a decision tree with a testing dataset representing 20% of the original data, while the remaining 80% served as the training dataset. Our target variable was the "Close" column, which we aimed to predict.

Interpretation of The Main Resulting If-Then Rules

In our initial approach, we included variables such as Monthly_Nominal_GDP_Index, Monthly_Real_GDP_Index, Unemployment_Rate, and Federal_Funds_Rate. Below, we present a portion of the decision tree. The resulting tree had a size of 33 nodes and provided key decision points. Factors such as "Federal_Funds_Rate" and “Monthly_Real_GDP_Index” indices play an important role in determining outcomes.

```
Federal_Funds_Rate <= 0.9:
:...Monthly_Real_GDP_Index > 18274.32:
:   :...Unemployment_Rate <= 5.6:
:   :   :...Unemployment_Rate > 4.6: 10 (286/254)
:   :   :   Unemployment_Rate <= 4.6:
:   :   :   :...Federal_Funds_Rate <= 0.36: 23 (281/252)
:   :   :       Federal_Funds_Rate > 0.36: 3 (100/90)
:   :   Unemployment_Rate > 5.6:
:   :   :...Unemployment_Rate <= 6.7: 8 (758/654)
:   :       Unemployment_Rate > 6.7:
:   :       :...Unemployment_Rate <= 8.4: 7 (294/249)
:   :           Unemployment_Rate > 8.4: 6 (94/79)
:   Monthly_Real_GDP_Index <= 18274.32:
:   :...Unemployment_Rate <= 5.2:
:       :...Monthly_Real_GDP_Index > 17644.14:
:       :   :...Unemployment_Rate <= 4.6: 9 (273/179)
:       :   :   Unemployment_Rate > 4.6:
:       :   :   :...Monthly_Nominal_GDP_Index <= 18774.47: 55 (103/77)
:       :   :       Monthly_Nominal_GDP_Index > 18774.47: 8 (465/357)
:       :   Monthly_Real_GDP_Index <= 17644.14:
:       :   :...Monthly_Real_GDP_Index <= 17520.27:
:       :       :...Monthly_Nominal_GDP_Index <= 18346.29: 1 (378/299)
:       :       :   Monthly_Nominal_GDP_Index > 18346.29: 6 (284/212)
:       :       Monthly_Real_GDP_Index > 17520.27:
:       :       :...Monthly_Nominal_GDP_Index <= 18512.66: 6 (189/149)
:       :           Monthly_Nominal_GDP_Index > 18512.66:
:       :           :...Unemployment_Rate <= 5: 1 (281/226)
:       :               Unemployment_Rate > 5: 7 (93/69)
:       Unemployment_Rate > 5.2:
:       :...Unemployment_Rate > 7.9:
:           :...Monthly_Nominal_GDP_Index <= 19095.95: 4 (93/77)
:           :   Monthly_Nominal_GDP_Index > 19095.95: 5 (188/158)
:           Unemployment_Rate <= 7.9:
:           :...Monthly_Nominal_GDP_Index > 18161.75: 5 (187/125)
:               Monthly_Nominal_GDP_Index <= 18161.75:
:               :...Unemployment_Rate > 5.6: 1 (91/71)
:                   Unemployment_Rate <= 5.6:
:                   :...Unemployment_Rate <= 5.4: 5 (197/149)
:                       Unemployment_Rate > 5.4: 4 (86/61)
```

Decision tree incorporating the variables Monthly_Nominal_GDP_Index, Monthly_Real_GDP_Index, Unemployment_Rate, and Federal_Funds_Rate

Accuracy and Confusion Matrices for Training and Testing Data

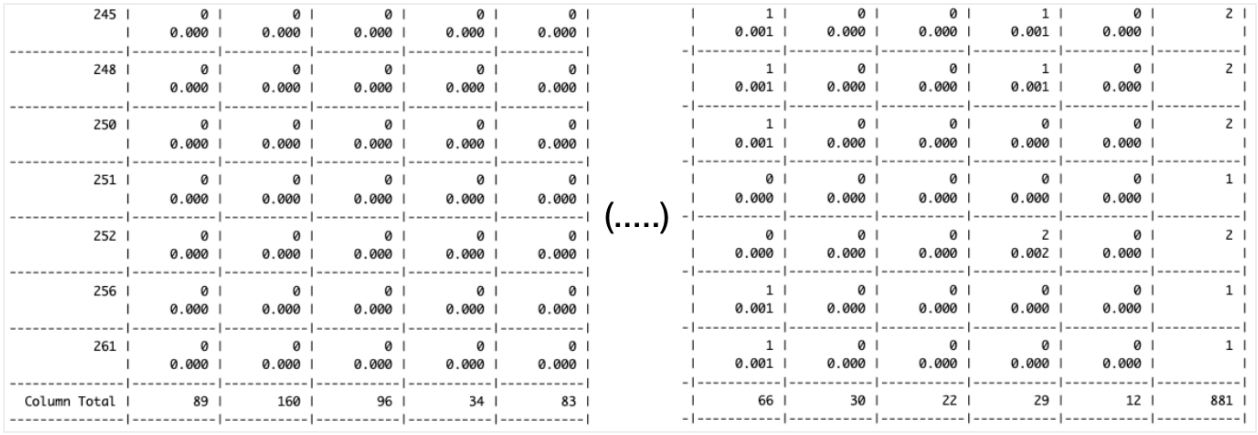
Unfortunately, the accuracy of the model, reflecting the percentage of correctly and incorrectly classified samples, did not meet the expectation. On the training dataset, the accuracy was 20.04% (0.2004291), and on the testing dataset, it was 18.72% (0.1872872). Additionally, the error rate was 79.95% (0.7995709) on the training data and 81.27% (0.8127128) on the testing data.

```
Evaluation on training data (7923 cases):

          Decision Tree
        ----------------
       Size      Errors

         33 6335(80.0%)   <<
```

The error rate on the training data

Below is the cross-tabulation for the tree model, involving 881 observations, visually illustrates the correct and incorrect classifications, revealing the assigned classes. For example, data originally labeled as "1" was correctly predicted as "1" 13 times, while it was incorrectly categorized as "5" four times.

	predicted default											
actual default	1	2	3	4	5	10	11	12	23	55	Row Total
1	13	0	0	3	4	0	0	0	0	3	56
	0.015	0.000	0.000	0.003	0.005	0.000	0.000	0.000	0.000	0.003	
2	1	26	1	0	0	7	7	6	0	0	48
	0.001	0.030	0.001	0.000	0.000	0.008	0.008	0.007	0.000	0.000	
3	1	4	16	1	1	0	0	0	0	0	23
	0.001	0.005	0.018	0.001	0.001	0.000	0.000	0.000	0.000	0.000	
4	2	0	2	7	3	0	0	0	0	0	14
	0.002	0.000	0.002	0.008	0.003	0.000	0.000	0.000	0.000	0.000	
5	4	0	0	0	18	(.....) 0	0	0	0	0	24
	0.005	0.000	0.000	0.000	0.020	0.000	0.000	0.000	0.000	0.000	
6	11	0	0	0	2	0	0	0	0	4	36
	0.012	0.000	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.005	
7	10	0	0	0	0	0	0	0	0	0	29
	0.011	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
8	7	0	0	0	0	0	0	0	0	1	35
	0.008	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	
9	0	2	0	0	0	4	0	0	1	0	21
	0.000	0.002	0.000	0.000	0.000	0.005	0.000	0.000	0.001	0.000	



245	0	0	0	0	0		1	0	0	1	0	2
	0.000	0.000	0.000	0.000	0.000		0.001	0.000	0.000	0.001	0.000	
248	0	0	0	0	0		1	0	0	1	0	2
	0.000	0.000	0.000	0.000	0.000		0.001	0.000	0.000	0.001	0.000	
250	0	0	0	0	0		1	0	0	0	0	2
	0.000	0.000	0.000	0.000	0.000		0.001	0.000	0.000	0.000	0.000	
251	0	0	0	0	0		0	0	0	0	0	1
	0.000	0.000	0.000	0.000	0.000	(.....)	0.000	0.000	0.000	0.000	0.000	
252	0	0	0	0	0		0	0	0	2	0	2
	0.000	0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.002	0.000	
256	0	0	0	0	0		1	0	0	0	0	1
	0.000	0.000	0.000	0.000	0.000		0.001	0.000	0.000	0.000	0.000	
261	0	0	0	0	0		1	0	0	0	0	1
	0.000	0.000	0.000	0.000	0.000		0.001	0.000	0.000	0.000	0.000	
Column Total	89	160	96	34	83		66	30	22	29	12	881

The cross-tabulation of the tree model

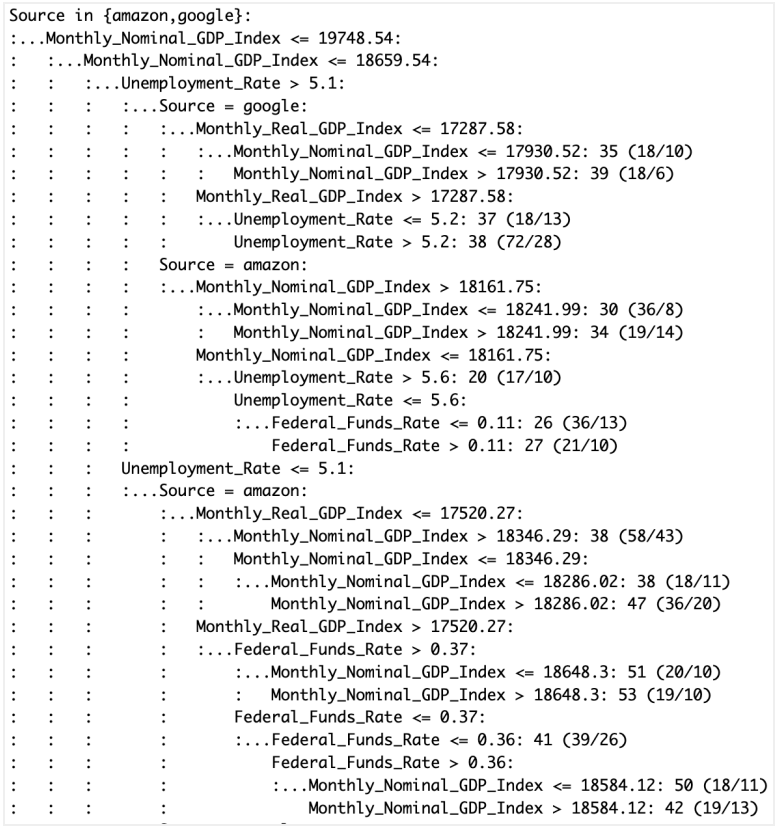
Based on the results obtained from the decision tree, including the accuracy rate, error rate, and cross-tabulation, it is evident that the decision tree model incorporating Monthly_Nominal_GDP_Index, Monthly_Real_GDP_Index, Unemployment_Rate, and Federal_Funds_Rate did not deliver satisfactory performance.

The Decision Tree Incorporating 5 Variables

In our second approach, we added “Source” as another variable to assess its impact on the decision tree and accuracy.

Key Findings from the Resulting If-Then Rules

First, the resulting tree is larger, with 236 nodes, than the 33-node tree using only four variables. The tree starts with a root node that splits the data based on the condition “Source in {amazon, google}” and “Source in {facebook,apple,netflix}”. If the data is associated with Amazon or Google, further divisions are based on "Monthly_Nominal_GDP_Index." Meanwhile, data associated with Facebook, Apple, or Netflix branches at the nodes "Source = apple" and "Source in {facebook, netflix}" before the further splits based on "Monthly_Nominal_GDP_Index."



First part of the decision tree: data associated with Amazon and Google

```
Source in {facebook,apple,netflix}:
:...Source = apple:
    :...Monthly_Nominal_GDP_Index <= 19221.89:
    :   :...Unemployment_Rate <= 10.2: 1 (528)
    :   :   Unemployment_Rate > 10.2: 4 (18/2)
    :   Monthly_Nominal_GDP_Index > 19221.89:
    :   :...Federal_Funds_Rate > 0.37:
    :       :...Monthly_Real_GDP_Index > 19167.46:
    :       :   :...Unemployment_Rate <= 3.5: 5 (35/13)
    :       :   :   Unemployment_Rate > 3.5: 4 (39)
    :       :   Monthly_Real_GDP_Index <= 19167.46:
    :       :   :...Monthly_Nominal_GDP_Index <= 20612.72: 2 (285/12)
    :       :       Monthly_Nominal_GDP_Index > 20612.72:
    :       :       :...Federal_Funds_Rate <= 2.19: 3 (135/10)
    :       :           Federal_Funds_Rate > 2.19:
    :       :           :...Monthly_Nominal_GDP_Index <= 21378.84: 2 (145/21)
    :       :               Monthly_Nominal_GDP_Index > 21378.84: 3 (20/1)
    :       Federal_Funds_Rate <= 0.37:
    :       :...Monthly_Real_GDP_Index <= 18606.12:
    :           :...Monthly_Nominal_GDP_Index <= 20881.38: 5 (38/8)
    :           :   Monthly_Nominal_GDP_Index > 20881.38: 6 (17/2)
    :           Monthly_Real_GDP_Index > 18606.12:
    :           :...Unemployment_Rate <= 5.6:
    :               :...Monthly_Real_GDP_Index <= 20015.41: 10 (89/39)
    :               :   Monthly_Real_GDP_Index > 20015.41: 12 (19/8)
    :               Unemployment_Rate > 5.6:
    :               :...Unemployment_Rate <= 6.3: 8 (116/36)
    :                   Unemployment_Rate > 6.3:
    :                   :...Unemployment_Rate > 6.7: 7 (60/15)
    :                       Unemployment_Rate <= 6.7:
    :                       :...Monthly_Nominal_GDP_Index <= 21625.2: 7 (17/6)
    :                           Monthly_Nominal_GDP_Index > 21625.2: 8 (22/4)
    Source in {facebook,netflix}:
    :...Monthly_Nominal_GDP_Index <= 19221.89:
        :...Unemployment_Rate > 5.2:
        :   :...Unemployment_Rate > 10.2:
        :   :   :...Source = facebook: 12 (17/8)
        :   :   :   Source = netflix: 30 (20/13)
        :   :   Unemployment_Rate <= 10.2:
        :   :   :...Monthly_Nominal_GDP_Index > 18161.75:
        :   :       :...Monthly_Nominal_GDP_Index <= 18177.39: 5 (38)
        :   :       :   Monthly_Nominal_GDP_Index > 18177.39:
```

Second part of the decision tree: data associated with Facebook, Apple, and Netflix

Accuracy and Confusion Matrices for Training and Testing Data

To measure the performance of our model on both the training and testing datasets, we started the process by constructing confusion matrices, which we later employed to evaluate accuracy. The accuracy rate on the training data reached approximately 55% (0.5537044), while the accuracy rate of testing data was approximately 47% (0.4721907). Furthermore, the error rate was at 44.6% (0.4462956) for the training data and 52.27% (0.5278093) for the testing data.

```
Evaluation on training data (7923 cases):

          Decision Tree
        ----------------
        Size      Errors

         236 3536(44.6%)   <<
```

The error rate on training data

The accuracy percentage significantly improved when "Source" was introduced as a variable in the decision tree, in comparison to the tree model containing four variables. This enhancement is particularly remarkable because it showed the critical influence of the "Source" variable in refining the model's predictive capabilities. By incorporating "Source," the decision tree had the ability to capture nuanced patterns and relationships within the data, resulting in more accurate and reliable predictions.

c. Boosting

Since the decision tree model with "Source" as one of the variables performed better, we used it for boosting with different trial settings.

Boosting with 5 Trials

- Average tree size: 177.4
- Accuracy on Training Data: 52% (0.5241701)
- Accuracy on Testing Data: 45% (0.4517594)

When boosting is applied with 5 trees, the tree size decreases on average. However, the accuracy on both training and testing data decreases as well. This suggests that the model's performance on unseen data is not improving significantly.

Trial	Decision Tree		
	Size	Errors	
0	236	3536(44.6%)	
1	168	3823(48.3%)	
2	166	3827(48.3%)	
3	165	3822(48.2%)	
4	152	4011(50.6%)	
boost		3768(47.6%)	<<

The error rate over 5 trials

Boosting with 10 Trials

- Average tree size: 164.8
- Accuracy on Training Data: 51% (0.5196264)
- Accuracy on Testing Data: 45% (0.4517594)

With 10 boosting trials, the tree size continues to decrease slightly (from 174 to 164), but the accuracy remains similar to the 5-trial scenario. This implies that additional boosting trials may not be leading to substantial accuracy improvements.

Trial	Decision Tree		
	Size	Errors	
0	236	3536(44.6%)	
1	168	3823(48.3%)	
2	166	3827(48.3%)	
3	166	3827(48.3%)	
4	166	3827(48.3%)	
5	166	3827(48.3%)	
6	163	3839(48.5%)	
7	153	3890(49.1%)	
8	146	4023(50.8%)	
9	118	4185(52.8%)	
boost		3806(48.0%)	<<

The error rate over 10 trials

Boosting with 30 Trials

- Average tree size: 159.9
- Accuracy on Training Data: 51% (0.5183643)

- Accuracy on Testing Data: 44% (0.4494892)

As we increase the number of trials to 30, the average tree size decreases further. However, there is little change in accuracy. The testing accuracy remains relatively stable.

Trial	Decision Tree	
	Size	Errors
0	236	3536(44.6%)
1	168	3823(48.3%)
2	166	3827(48.3%)
3	166	3827(48.3%)
4	166	3827(48.3%)
5	166	3827(48.3%)
6	166	3827(48.3%)
7	166	3827(48.3%)
8	166	3827(48.3%)
9	166	3827(48.3%)
10	166	3827(48.3%)
11	166	3827(48.3%)
12	166	3827(48.3%)
13	166	3827(48.3%)
14	166	3827(48.3%)
15	166	3827(48.3%)
16	166	3827(48.3%)
17	165	3834(48.4%)
18	166	3830(48.3%)
19	163	3858(48.7%)
20	157	3890(49.1%)
21	157	3941(49.7%)
22	149	3954(49.9%)
23	155	3956(49.9%)
24	155	3999(50.5%)
25	146	4136(52.2%)
26	130	4157(52.5%)
27	126	4204(53.1%)
28	119	4174(52.7%)
29	115	4336(54.7%)
boost		3816(48.2%)

The error rate over 30 trials

Boosting with 50 Trials

- Average tree size: 158.9
- Accuracy on Training Data: 0.5183643
- Accuracy on Testing Data: 0.4494892

Similar to the 30-trial scenario, increasing the number of trials to 50 has minimal impact on accuracy. The accuracy remains nearly unchanged.

Trial	Decision Tree	
	Size	Errors
0	236	3536(44.6%)
1	168	3823(48.3%)
2	166	3827(48.3%)
3	166	3827(48.3%)
4	166	3827(48.3%)
5	166	3827(48.3%)
6	166	3827(48.3%)
7	166	3827(48.3%)
8	166	3827(48.3%)
9	166	3827(48.3%)
10	166	3827(48.3%)
11	166	3827(48.3%)
12	166	3827(48.3%)
13	166	3827(48.3%)
14	166	3827(48.3%)
15	166	3827(48.3%)
16	166	3827(48.3%)
17	166	3827(48.3%)
18	166	3827(48.3%)
19	166	3827(48.3%)
20	166	3827(48.3%)
21	166	3827(48.3%)
22	166	3827(48.3%)
23	166	3827(48.3%)
24	166	3827(48.3%)
25	166	3827(48.3%)
26	166	3827(48.3%)
27	165	3827(48.3%)
28	168	3820(48.2%)
29	164	3831(48.4%)
30	166	3837(48.4%)
31	164	3856(48.7%)
32	163	3862(48.7%)
33	160	3892(49.1%)
34	157	3886(49.0%)
35	156	3929(49.6%)
36	155	3947(49.8%)
37	158	3983(50.3%)
38	155	3985(50.3%)
39	152	4025(50.8%)
40	157	3996(50.4%)
41	153	4092(51.6%)
42	142	4137(52.2%)
43	134	4184(52.8%)
44	128	4169(52.6%)
45	120	4296(54.2%)
46	117	4267(53.9%)
47	114	4252(53.7%)
48	119	4271(53.9%)
49	122	4238(53.5%)
boost		3816(48.2%)

The error rate over 50 trials

Boosting with 100 Trials

- Average tree size: 158.3

- Accuracy on Training Data: 0.5177332
- Accuracy on Testing Data: 0.4483541

Even with 100 trials, there is no significant improvement in accuracy. The tree size continues to decrease slightly, but the testing accuracy remains relatively stable.

Conclusion

Below is the table summarizing the results of boosting trials:

Trials	Average tree size	Accuracy on Training Data	Accuracy on Testing Data
1(original)	236	53% (0.5537044)	47% (0.4721907)
5	177.4	52% (0.5241701)	45% (0.4517594)
10	164.8	51% (0.5196264)	45% (0.4517594)
30	159.9	51% (0.5183643)	44% (0.4494892)
50	158.9	51% (0.5183643)	44% (0.4494892)
100	158.3	51% (0.5177332)	44% (0.4483541)

The analysis reveals that increasing the number of boosting trials beyond 10 trials does not substantially improve the accuracy of the classifier. The accuracy on the testing data tends to stabilize, and additional boosting trials may not lead to meaningful enhancements.

d. Bagging and Random Forests

Bagging

When implementing bagging in the improvement process, we used different values of trials (2, 3, 4, 5), which determines the number of variables sampled at each split. The code trains multiple bagging models using the randomForest() function with different trial values and evaluates their performance on both training and test data using the predict() function.

As the number of variables sampled at each split (trials) increases from 2 to 5, the training and testing accuracy rates remain relatively consistent. The accuracy rate was 56% for the training

data and about 48% for the testing data. The error rate on the test data also remains fairly stable, ranging from 51.53% to 52.21%. It appears that the accuracy and error rates in bagging are not significantly affected by changes in trials. Below is the table summarizing the bagging results:

mtry	Accuracy Rate on Training Data	Error Rate on Training Data	Accuracy Rate on Testing Data	Error Rate on Testing Data
2	55.66% (0.5566073)	44.34% (0.4433927)	47.79% (0.4778661)	52.21% (0.5221339)
3	56.19% (0.5619084)	43.81% (0.4380916)	48.35% (0.4835414)	51.65% (0.5164586)
4	56.19% (0.5619084)	43.81% (0.4380916)	48.35% (0.4835414)	51.65% (0.5164586)
5	56.19% (0.5619084)	43.81% (0.4380916)	48.47% (0.4846765)	51.53% (0.5153235)

Accuracy and error rate with bagging on training and testing data

Random Forests

Random Forests are applied with different numbers of trees (10, 50, 100, 200) in the ensemble. The training accuracy rates for Random Forests increase slightly from 55.79% with 10 trees to 56.22% with 200 trees. The training error rates for Random Forests decrease slightly from 44.21% with 10 trees to 43.78% with 200 trees. In terms of testing data, the accuracy rates remain relatively consistent, hovering around 48% for all numbers of trees. The error rates on the testing data show a decreasing trend as the number of trees increases, ranging from 51.99% with 10 trees to 48.52% with 200 trees. Below is the table summarizing the random forest result:

Trees	Accuracy Rate on Training Data	Error Rate on Training Data	Accuracy Rate on Testing Data	Error Rate on Testing Data
10	0.557869	0.442130	0.480136	0.519863
50	0.561403	0.438596	0.480136	0.519863
100	0.561529	0.438470	0.482406	0.517593
200	0.562160	0.437839	0.485811	0.514188

Accuracy and error rate with random forests on training and testing data

Conclusion

Bagging results in stable accuracy and error rates over different values of mtry. There was no significant improvement with bagging. On the other hand, random forests show a slight improvement in the accuracy of the training data and a reduction in the training error with an increase in the number of trees. However, this trend is not as pronounced in the test data, where the accuracy rates remain stable.

Q4. Comparative Analysis

Based on the analysis we did in the previous section, this is a summary of the classifiers and their performance:

- Linear regression models showed high predictive quality, with R-squared values close to 1 and low mean squared error. The multivariate linear regression model performed the best.
- Logistic regression had moderate predictive performance based on the AUC metric, slightly better than random guessing. Unemployment rate and previous closing price were the most significant predictors.
- Naive Bayes classifier performed similar with or without Laplace smoothing, suggesting it was not very effective for this dataset.
- Decision tree models had low accuracy rates around 20% with just economic indicators. Including stock source improved accuracy to ~50%.
- Boosting the decision tree did not meaningfully improve accuracy. Bagging and random forests also showed limited improvements.

Overall, the linear regression models, especially multivariate regression, showed the strongest predictive performance for stock prices based on the economic indicators. The linear relationship between the predictors and stock prices was most effectively captured by regression.

Logistic regression and decision trees provided additional insights into predictive factors like unemployment rate and stock source. But their accuracy was lower compared to linear regression.

For predicting stock prices based on economic indicators, we would use the multivariate linear regression model, which had the best predictive quality with R-squared and MSE values indicating a very close fit to the data. The linear model is interpretable and flexible to include multiple relevant predictors.

Contribution Section

Question 1: developed by Srikanth Parvathala;

Question 2: developed by Vijay Arni;

Question 3: developed by Ya-Ting Yang;

Question 4: developed by Srikanth Parvathala, Vijay Arni, and Ya-Ting Yang

Srikanth Parvathala prepared 100% of the R code for question 1;

Vijay Arni prepared 100% of the R code for question 2;

Ya-Ting Yang prepared 100% of the R code for question 3;

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

Reference

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

Confusion Matrix https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/

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

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

Logistic regression detailed overview https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc

Sharing insights elevates their impact. (n.d.). S&P Global. https://www.spglobal.com/marketintelligence/en/mi/products/us-monthly-gdp-index.html

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