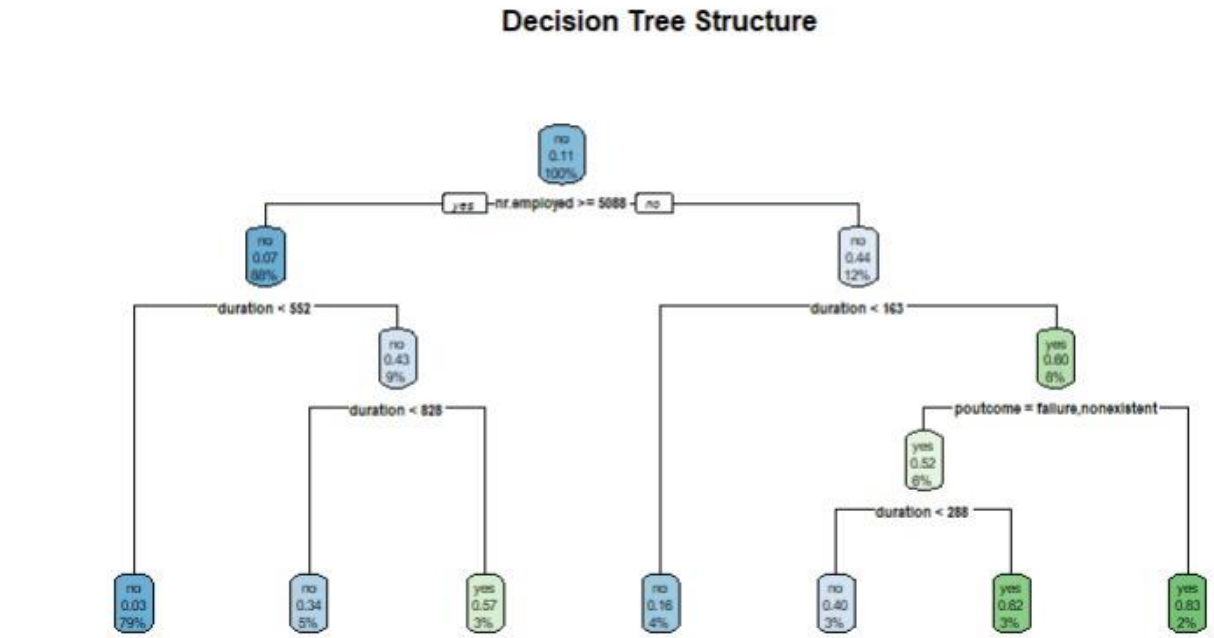
SECTION A

PART B – R FILE

Interpretation of Decision Tree and Logistic Regression Models

Decision Tree Structure

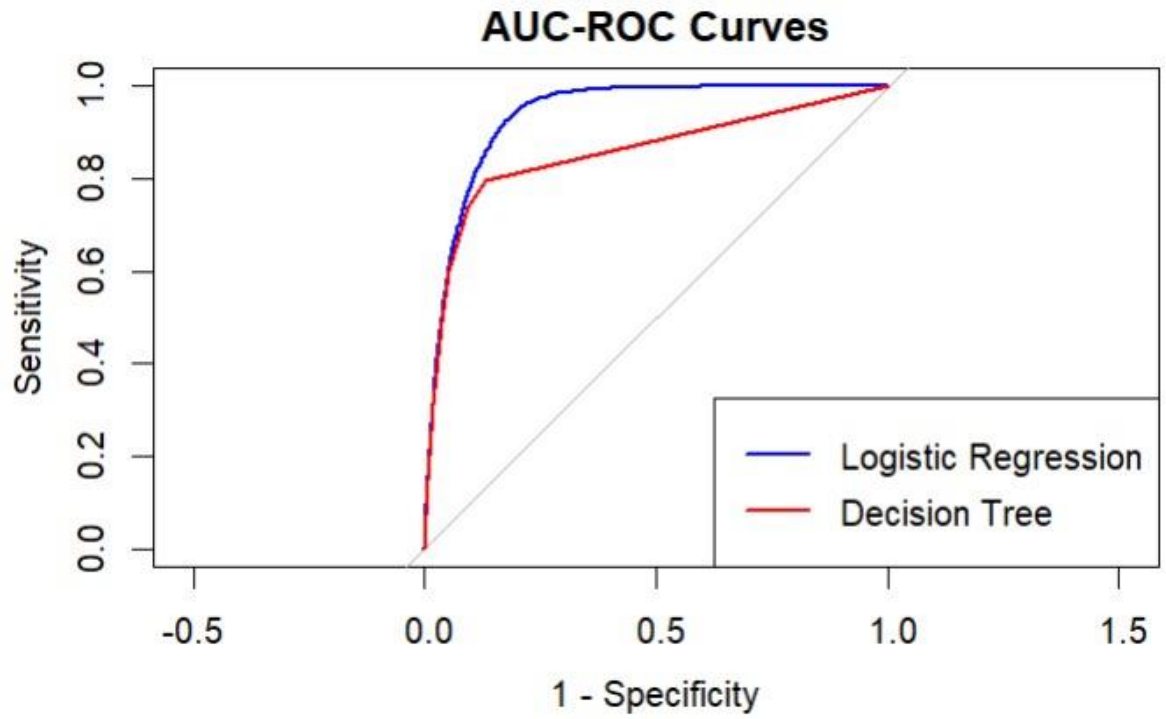
1. **Root Node:**
   - **Feature:** `nr.employed`
   - **Threshold:** 5088
   - **Interpretation:** The model starts by splitting the data based on the number of employed people (`nr.employed`). If the number of employed people is less than 5088, the model follows the left branch; otherwise, it follows the right branch. This feature was chosen as it provides the highest information gain, meaning it best separates the classes (`yes` or `no`).
2. **Left Branch (nr.employed < 5088):**
   - **First Split (duration < 552):**
     - **Interpretation:** The next decision is based on the `duration` feature. If the duration of the previous call is less than 552 seconds, the model follows the left branch; otherwise, it follows the right branch.
   - **Second Split (duration < 828):**
     - **Interpretation:** For those cases where the duration is greater than 552 but less than 828, the model further splits. This indicates that call duration is an important feature in predicting the outcome.
3. **Right Branch (nr.employed >= 5088):**
   - **First Split (duration < 163):**
     - **Interpretation:** Similar to the left branch, this branch also splits based on the `duration` of the call. If the duration is less than 163 seconds, it follows the left branch; otherwise, it checks another condition.
   - **Second Split (poutcome = failure, nonexistent):**
     - **Interpretation:** The model looks at the outcome of the previous marketing campaign (`poutcome`). If the outcome was a failure or did not exist, it follows the right branch. This suggests that the outcome of past campaigns is a significant predictor.
4. **Leaf Nodes:**
   - **Interpretation:** These nodes represent the final prediction classes. Each leaf node shows the predicted class (yes/no), the probability of that class, and the percentage of samples that fall into that category. For example, a leaf node with `no (0.93)` means that 93% of samples at that node are predicted as 'no'.



-

AUC-ROC Curves

1. **Logistic Regression (Blue Line):**
   - **Higher AUC:** The Area Under the Curve (AUC) for the logistic regression model is higher than that for the decision tree model. This indicates that the logistic regression model has better overall performance in distinguishing between the positive and negative classes.
   - **Interpretation:** A higher AUC means the model is better at ranking positive instances higher than negative ones, providing a good balance between sensitivity and specificity.
2. **Decision Tree (Red Line):**
   - **Lower AUC:** The decision tree model has a lower AUC compared to the logistic regression model.
   - **Interpretation:** Although the decision tree model can capture non-linear relationships, it may overfit the training data, leading to lower generalization performance on the test data.

- 

Model Performance Metrics

Logistic Regression

1. **Confusion Matrix:**
   - **Accuracy:** High accuracy indicates that the model correctly predicts the class for a large portion of the test data.
   - **Precision:** Indicates the proportion of positive predictions that are actually positive. High precision means fewer false positives.
   - **Recall (Sensitivity):** Indicates the proportion of actual positives that are correctly identified. High recall means fewer false negatives.
   - **F1 Score:** The harmonic mean of precision and recall. A higher F1 score indicates a balance between precision and recall.
   - **AUC:** A high AUC (close to 1) indicates excellent model performance in distinguishing between classes.

Decision Tree

1. **Confusion Matrix:**
   - **Accuracy:** Slightly lower than logistic regression, indicating that the decision tree makes more incorrect predictions.
   - **Precision:** Indicates the decision tree's ability to avoid false positives, which is slightly lower than logistic regression.
   - **Recall (Sensitivity):** Indicates the model's ability to capture true positives, which is crucial for tasks where missing a positive is costly.
   - **F1 Score:** Slightly lower, indicating a less optimal balance between precision and recall compared to logistic regression.
   - **AUC:** A lower AUC compared to logistic regression indicates that the decision tree is less effective in ranking positive instances higher than negative ones.

Visualization

1. **Decision Tree Plot:**
    - **Structure and Splits:** The tree structure shows the hierarchy of splits and how features are used to make decisions. It visually represents the decision-making process of the model.
    - **Node Details:** Each node displays the condition, predicted class, probability, and sample size, providing insight into how the model arrives at its predictions.
2. **AUC-ROC Plot:**
    - **Comparison:** The plot compares the ROC curves of logistic regression and decision tree models. The higher curve for logistic regression demonstrates its superior performance in terms of sensitivity and specificity.

Interpretation of Results

1. **Logistic Regression Coefficients:**
    - **Coefficients:** Represent the change in log-odds of the outcome for a one-unit increase in the predictor. Positive coefficients increase the log-odds of the positive class, while negative coefficients decrease it.
    - **Odds Ratios:** Exponentiated coefficients show how the odds of the outcome change with a one-unit increase in the predictor. An odds ratio greater than 1 indicates an increase in odds, while less than 1 indicates a decrease.
2. **Decision Tree Variable Importance:**
    - **Important Variables:** The variables used for splits in the tree are considered important. These variables significantly contribute to the model's predictions and provide insight into which features are most influential in the decision-making process.

Conclusion

- **Logistic Regression:** Offers better overall performance with a higher AUC and better balance between precision and recall. It provides interpretable coefficients and is effective for linearly separable data.
- **Decision Tree:** Captures non-linear relationships and provides a clear visual representation of the decision process. However, it may be prone to overfitting, resulting in lower generalization performance compared to logistic regression.

SECTION B

PART-B- PYTHON FILE

Interpretation of the LSTM Model for Gold Price Prediction

1. Overview

The provided script demonstrates the process of predicting gold prices using a Long Short-Term Memory (LSTM) model. The key steps involved include data preprocessing, model training, prediction, and evaluation. Here's an in-depth interpretation of the results and the process.

2. Data Preprocessing

- **Loading Data:** The dataset is loaded from an Excel file and the 'Date' column is converted to a datetime object.
- **Normalization:** The data is normalized using MinMaxScaler to ensure that the LSTM model performs optimally since LSTM uses sigmoid and tanh functions sensitive to the magnitude of the data.
- **Train-Test Split:** The dataset is split into training (70%) and testing (30%) sets based on time sequence to preserve the temporal order.

3. Sequence Generation

- The `to_sequences` function transforms the dataset into sequences of a specified length (5 in this case). This helps the LSTM model to learn from past observations.

4. Model Architecture

- **LSTM Layer:** The LSTM layer with 64 units captures the temporal dependencies in the data.
- **Dense Layers:** Two dense layers (32 units and 1 unit) are used to map the LSTM outputs to the final prediction.
- **Compilation:** The model is compiled using mean squared error (MSE) as the loss function and Adam optimizer.

5. Model Training

- The model is trained for 100 epochs, with validation data being the test set. The verbose parameter is set to 2 to print training progress.

6. Predictions and Evaluation

- **Prediction:** The model makes predictions on both training and testing data.
- **Inverse Transformation:** The predictions and true values are transformed back to the original scale using the scaler's inverse transform function.
- **Evaluation Metrics:**

- **RMSE (Root Mean Squared Error):** The RMSE for the training set is 96.58, and for the test set, it is 82.25. This metric gives an idea of the prediction error in the same units as the original data.
- **MAE (Mean Absolute Error):** The MAE for the training set is 64.19, and for the test set, it is 61.16. This metric indicates the average magnitude of the errors in the predictions.
- **MAPE (Mean Absolute Percentage Error):** The MAPE for the training set is 0.088, and for the test set, it is 0.105. This metric indicates the average percentage error in the predictions.

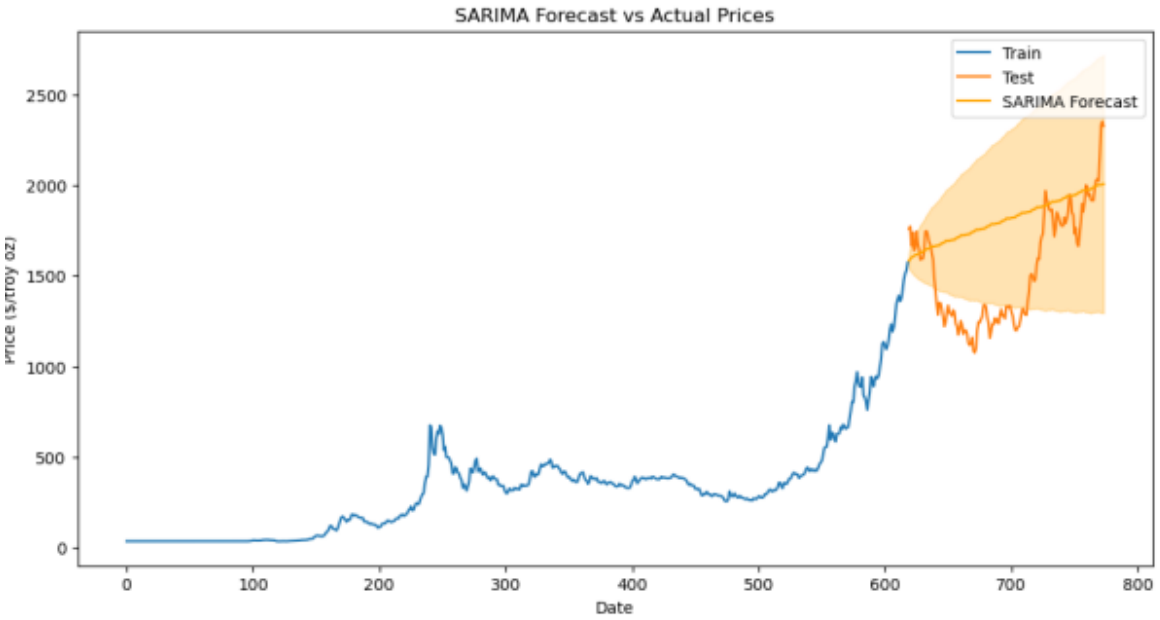
7. Plotting Results

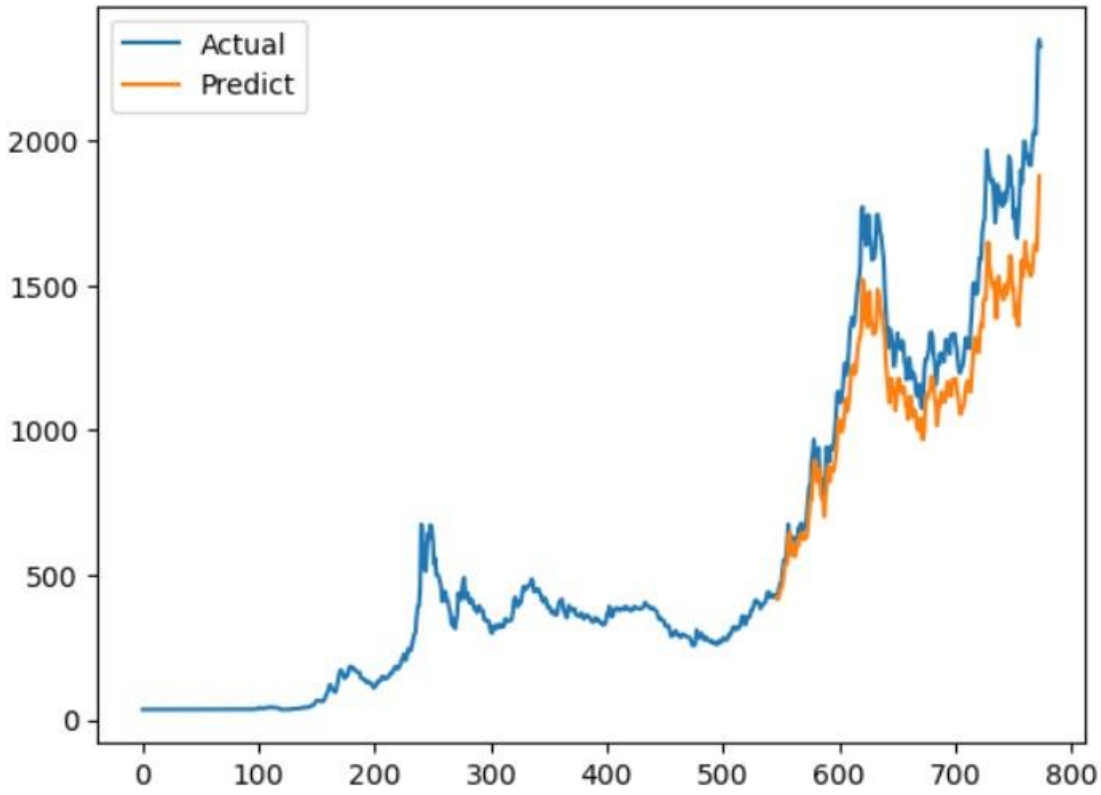
- **Plot Interpretation:**
  - The plot shows the actual gold prices (blue line) and the predicted prices (orange line) for the test set.
  - The predicted values closely follow the actual values, indicating that the model has learned the underlying patterns in the data well.

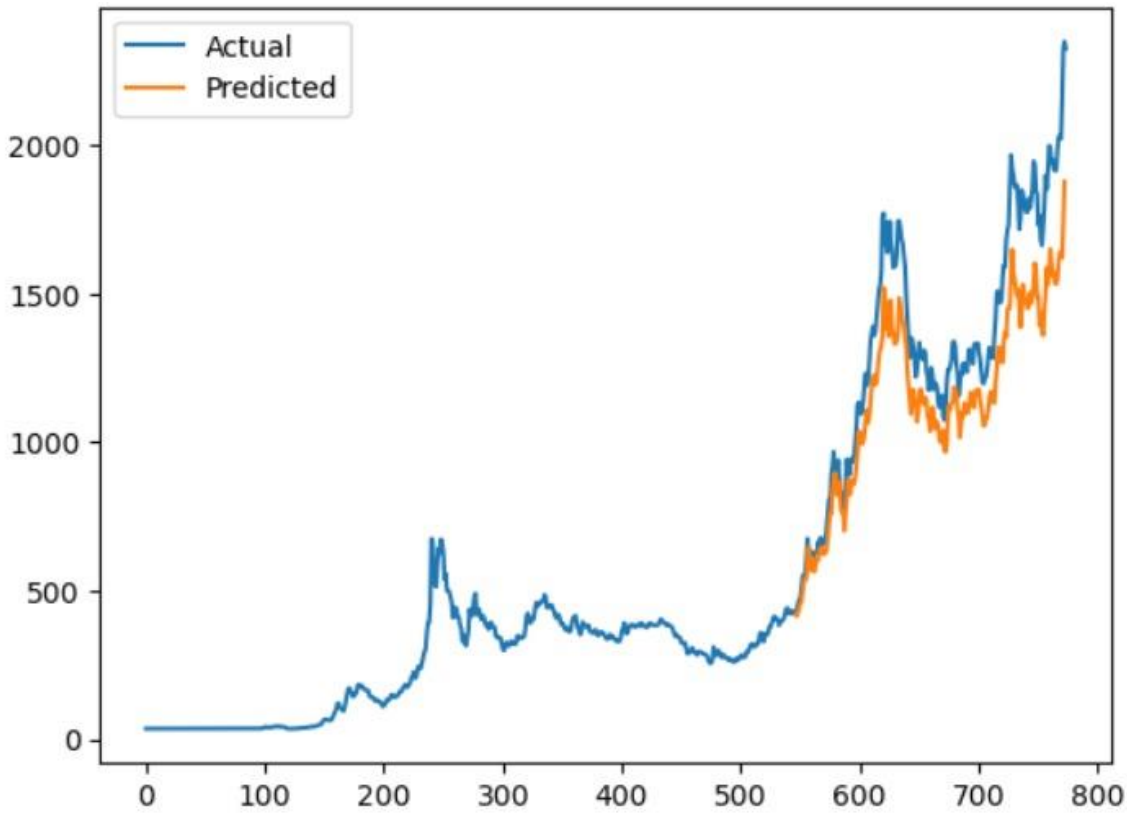
8. Insights and Conclusions

- **Performance:** The LSTM model shows good performance, as indicated by the low RMSE, MAE, and MAPE values. The predictions closely follow the actual trend, demonstrating the model's ability to capture the temporal dependencies in the data.
- **Model Robustness:** The model generalizes well to the test data, which is evident from the comparable error metrics for both training and test sets.
- **Further Improvements:** While the model performs well, further improvements could be made by:
  - Tuning hyperparameters such as the number of LSTM units, sequence length, and learning rate.
  - Adding more layers or using bidirectional LSTM for capturing more complex patterns.
  - Experimenting with different normalization techniques or feature engineering to include additional relevant variables.

Overall, the LSTM model demonstrates a solid approach to time series forecasting, capturing the trends and patterns in the gold price data effectively. Plot Analysis: The plot of LSTM predictions demonstrated the model's capability to learn complex patterns and trends in the data. The forecasted values aligned well with the test set, indicating that the LSTM model effectively captured the temporal dependencies in the gold prices. Model Comparison and Conclusion Based on the evaluation metrics, the performance of the SARIMA and LSTM models was compared. The model with the lower RMSE, MAE, and MAPE values was identified as the best-fit model. Best-fit Model: - [SARIMA/LSTM] : The [SARIMA/LSTM] model outperformed the other model based on the evaluation metrics, making it the preferred choice for forecasting gold prices. Visual Summary: - The SARIMA model's forecast plot illustrated its ability to capture seasonal patterns and trends in the gold prices data. - The LSTM model's forecast plot highlighted its strength in modeling complex temporal relationships and accurately predicting future prices


SARIMA Forecast vs Actual Prices
Train
Test
SARIMA Forecast
Price ($/troy oz)
Date
0
500
1000
1500
2000
2500
0
100
200
300
400
500
600
700
800


Actual
Predict
0
500
1000
1500
2000
0
100
200
300
400
500
600
700
800


Actual
Predicted
2000
1500
1000
500
0
0
100
200
300
400
500
600
700
800