Model Code and Documentation

1. **Introduction**
    - **Objective:** The objective of this project is to develop a machine learning model to predict the target variable based on provided GST analytics data.
    - **Model Used:** Random Forest Regressor was selected due to its robustness in handling large datasets and feature importance selection.
2. **Data Preparation**
    - **Dataset Overview:**
        - The training and testing datasets consist of multiple numerical columns along with a target column for prediction.
        - Non-numerical columns like `ID` were dropped since they don't contribute to the model's performance.
    - **Handling Missing Values:**
        - The datasets had missing values, which were handled by filling them with the column mean.
    - **Data Splitting:**
        - The training data was split into training (80%) and validation (20%) sets for cross-validation.
    - **Feature Scaling:**
        - All features were standardized using `StandardScaler` to improve the performance of the model.
3. **Model Development**
    - **Algorithm:**
        - Random Forest Regressor was used with 100 estimators and a fixed random state for reproducibility.
    - **Why Random Forest:**
        - Random Forest is chosen for its ability to handle high-dimensional datasets, robustness to overfitting, and importance scoring of features.
    - **Training:**
        - The model was trained on the scaled training set.
    - **Evaluation Metrics:**

    - The model was evaluated using **Mean Squared Error (MSE)** and **R^2 Score** for both validation and test datasets.

4. **Code Structure**
    - **Data Preprocessing:**
        - Dropping the `ID` column.
        - Handling missing values.
        - Splitting the data.
        - Standardizing the features.
    - **Model Training:**
        - Building and fitting the Random Forest Regressor.
    - **Evaluation:**
        - Calculating MSE and R^2 scores for validation and test datasets.
    - **Prediction Export:**
        - Exporting predictions to a CSV file for submission.
5. **Conclusion**
    - This model offers robust predictive performance with easy scalability for future improvements.

Model Performance Report

1. Overview of the Model

- **Algorithm**: Random Forest Regressor
- **Data**: GST Analytics Dataset (with missing values handled by mean imputation)
- **Target Variable**: A binary classification target (`Y_train`, `Y_test`)
- **Problem Type**: Regression-based predictive modeling to estimate values of the test dataset.

---

2. Dataset Overview

- The training and testing datasets (`X_Train_Data_Input.csv` and `X_Test_Data_Input.csv`) had the following missing values:

  Training Data Missing Values:

  - Column0: 9 missing values
  - Column3: 126,303 missing values
  - Column4: 127,710 missing values
  - Column5: 167,180 missing values
  - Column6: 3,850 missing values
  - Column8: 3,850 missing values
  - Column9: 732,137 missing values
  - Column14: 365,703 missing values
  - Column15: 16,456 missing values

  Testing Data Missing Values:

  - Column0: 2 missing values
  - Column3: 42,234 missing values
  - Column4: 42,710 missing values
  - Column5: 55,659 missing values
  - Column6: 1,234 missing values
  - Column8: 1,234 missing values
  - Column9: 243,853 missing values
  - Column14: 121,679 missing values
  - Column15: 5,485 missing values

Handling Missing Data:

- Missing values were filled using **mean imputation** across the dataset, allowing the model to proceed without any dropped rows.

---

3. Model Performance Metrics

The model was evaluated on both the validation set (20% split from the training data) and the test set provided.

- **Validation Metrics**:
  - **Mean Squared Error (MSE)**: 0.0177
  - **R^2 Score**: 0.792
- **Test Metrics**:
  - **Mean Squared Error (MSE)**: 0.0175
  - **R^2 Score**: 0.795

These scores indicate that the model performs well, with an MSE of approximately 0.0175 on the test data and a high R^2 score of ~0.795, showing strong predictive accuracy.

---

4. Key Observations

- The **Random Forest Regressor** effectively handled the binary classification problem and provided strong predictive capabilities on both the validation and test sets.
- **Mean Imputation** successfully dealt with the significant number of missing values in the dataset without impacting model performance drastically.
- Despite the presence of a large number of missing values (particularly in Columns 3, 4, 5, and 9), the Random Forest model showed good resilience and accuracy.

---

5. Recommendations

- **Further Tuning**: While the results are satisfactory, additional hyperparameter tuning (e.g., adjusting the number of estimators, max depth, etc.) could potentially further improve model performance.
- **Alternative Imputation**: Other imputation methods, such as K-nearest neighbors or regression imputation, could be explored to see if they improve the model's prediction accuracy.

- **Feature Engineering**: Further analysis of the dataset and engineering of features could help improve the model's ability to capture underlying patterns, especially in areas with significant missing data.

Conclusion: The model shows robust predictive accuracy with strong R^2 and MSE scores on both validation and test sets, demonstrating its effectiveness in solving this binary classification problem in the GST analytics dataset.

CODE:

```
# Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

# List the files in the current working directory
import os
print(os.listdir())

# Load the datasets (make sure the filenames match exactly)
X_train = pd.read_csv('X_Train_Data_Input.csv')
Y_train = pd.read_csv('Y_Train_Data_Target.csv')
X_test = pd.read_csv('X_Test_Data_Input.csv')
Y_test = pd.read_csv('Y_Test_Data_Target.csv')

# Drop the 'ID' column as it is non-numeric and not useful for training
X_train = X_train.drop(columns=['ID'])
X_test = X_test.drop(columns=['ID'])

# Check if there are any missing values in the dataset
print("Missing values in X_train:\n", X_train.isnull().sum())
print("Missing values in X_test:\n", X_test.isnull().sum())

# Handle missing values if present (you can impute or drop missing
rows)
X_train = X_train.fillna(X_train.mean())
X_test = X_test.fillna(X_test.mean())
```

```python
# Split the data into training and validation sets
X_train_split, X_val, Y_train_split, Y_val = train_test_split(X_train,
Y_train['target'], test_size=0.2, random_state=42)

# Feature scaling (Standardizing the data)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train_split)
X_val_scaled = scaler.transform(X_val)
X_test_scaled = scaler.transform(X_test)

# Model construction: Using Random Forest Regressor
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train_scaled, Y_train_split)

# Testing the model on validation data
Y_val_pred = model.predict(X_val_scaled)
print("Validation MSE:", mean_squared_error(Y_val, Y_val_pred))
print("Validation R2 Score:", r2_score(Y_val, Y_val_pred))

# Predicting on test data
Y_test_pred = model.predict(X_test_scaled)
print("Test Predictions:", Y_test_pred)

# Evaluate performance on test data
print("Test MSE:", mean_squared_error(Y_test['target'], Y_test_pred))
print("Test R2 Score:", r2_score(Y_test['target'], Y_test_pred))

# Export the predictions for submission
predictions = pd.DataFrame(Y_test_pred, columns=['Y_Predictions'])
predictions.to_csv('Predictions.csv', index=False)

# Download the predictions for local use
from google.colab import files
files.download('Predictions.csv')
```

OUTPUT:

```
['.config', 'Y_Train_Data_Target.csv', 'X_Test_Data_Input.csv',
'X_Train_Data_Input.csv', 'Y_Test_Data_Target.csv', 'drive',
'sample_data']
Missing values in X_train:
 Column0          9
Column1          0
Column2          0
Column3     126303
Column4     127710
```

```
Column5     167180
Column6       3850
Column7          0
Column8       3850
Column9     732137
Column10         0
Column11         0
Column12         0
Column13         0
Column14    365703
Column15     16456
Column16         0
Column17         0
Column18         0
Column19         0
Column20         0
Column21         0
dtype: int64
Missing values in X_test:
 Column0          2
Column1          0
Column2          0
Column3      42234
Column4      42710
Column5      55659
Column6       1234
Column7          0
Column8       1234
Column9     243853
Column10         0
Column11         0
Column12         0
Column13         0
Column14    121679
Column15      5485
Column16         0
Column17         0
Column18         0
Column19         0
Column20         0
Column21         0
dtype: int64
Validation MSE: 0.017708225851613246
Validation R2 Score: 0.7919612567735097
Test Predictions: [0. 0. 0. ... 0. 0. 0.]
Test MSE: 0.017511220520041983
Test R2 Score: 0.7949578795107081
```