**GitHub Link:** [**https://github.com/Jhay-Johnson/ING\_Risk-Modelling/edit/main/Team\_Monster's**](https://github.com/Jhay-Johnson/ING_Risk-Modelling/edit/main/Team_Monster's)

**Task 1.1 – Logistic Regression task**

Describe the dataset (e.g., descriptive statistics, missing values, target rate)

We have described the dataset with additional details as (Categorical Features, Correlation Matrix, Outlier Detection, Data Distributions.) Here's a brief overview of what each of these aspects entails for different types of datasets:

1. **Descriptive Statistics**:
   * This includes measures such as mean, median, mode, standard deviation, variance, minimum, maximum, and quartiles for numerical features.
   * For categorical features, it involves counts or proportions of each category.
2. **Missing Values**:
   * Identification of missing values in the dataset and determining their distribution across features.
   * Strategies for handling missing values such as imputation or deletion.
3. **Target Rate**:
   * The proportion of instances in the dataset that belong to each class in a classification problem.
   * For regression tasks, it might involve the distribution of the target variable.
4. **Categorical Features**:
   * Identification of categorical variables and their unique categories.
   * Understanding the frequency distribution of categories within each feature.
5. **Correlation Matrix**:
   * Calculation of the correlation coefficient between pairs of numerical features.
   * Visualization of correlation matrix to identify patterns of association between variables.
6. **Outlier Detection**:
   * Identification of outliers in the dataset using statistical methods or machine learning algorithms.
   * Visualization techniques such as box plots or scatter plots to visualize outliers.
7. **Data Distributions**:
   * Understanding the distribution of numerical features through histograms, density plots, or Q-Q plots.
   * Visualizing the skewness and kurtosis of distributions to understand their shape.

Code for 1st question:

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

# Loading the dataset

file\_path = "https://files.challengerocket.com/files/lions-den-ing-2024/development\_sample.csv"

data = pd.read\_csv(file\_path)

# Descriptive statistics

print("Descriptive statistics:")

print(data.describe())

# Missing values

print("\nMissing values:")

print(data.isnull().sum())

# Target rate

target\_rate = data['target'].mean() \* 100  # Multiply by 100 to get percentage

print(f"\nTarget rate (probability of default): {target\_rate:.2f}%")

# Categorical features

categorical\_features = data.select\_dtypes(include=['object']).columns

for feature in categorical\_features:

    print(f"\n{feature} distribution:")

    print(data[feature].value\_counts(normalize=True))

# Correlation matrix

corr\_matrix = data.corr()

plt.figure(figsize=(25, 25))

sns.heatmap(corr\_matrix, annot=True, cmap='coolwarm', fmt=".2f")

plt.title("Correlation Matrix")

plt.show()

# Outlier detection (example for numerical features)

numerical\_features = data.select\_dtypes(include=['int64', 'float64']).columns

for feature in numerical\_features:

    sns.boxplot(x=data[feature])

    plt.title(f"Boxplot of {feature}")

    plt.show()

# Data distributions (example for numerical features)

for feature in numerical\_features:

    sns.histplot(data=data, x=feature, kde=True)

    plt.title(f"Distribution of {feature}")

    plt.show()

**2. Describe the feature engineering procedure and the data treatments you followed (if any).**

Let's discuss the feature engineering procedures and data treatments applied for the task we performed earlier, which involved describing the dataset. Since we haven't explicitly performed feature engineering in that task, we will outline some common feature engineering techniques that could be applied:

**Handling Categorical Features:**

If the dataset contains categorical features, we would typically encode them using techniques such as one-hot encoding or label encoding to convert them into numerical representations that can be used as input in the model.

**Dealing with Missing Values:**

If the dataset has missing values, we would need to decide how to handle them. Common approaches include imputing missing values using techniques like mean imputation, median imputation, or mode imputation, or dropping rows or columns with missing values if they cannot be reliably imputed.

**Feature Scaling:**

Depending on the modeling algorithm used (such as logistic regression), we might scale numerical features to a similar range using techniques like standardization or normalization to prevent features with larger scales from dominating the model.

**Feature Selection:**

Feature selection involves identifying and selecting the most relevant features that have a significant impact on the target variable. While we didn't explicitly perform feature selection in the initial task, it's an important step in model building to improve model performance and interpretability.

Since the initial task focused primarily on describing the dataset, we didn't perform explicit feature engineering procedures or data treatments beyond basic data exploration and analysis. However, these are some common feature engineering techniques that could be applied depending on the specific characteristics of the dataset and the modeling goals. Let me know if you have any further questions or if you'd like to discuss any specific aspect in more detail!

**3. Describe the model selection process you applied (e.g., criteria for feature selection, estimation technique of the model parameters).**

The model selection process for credit risk modeling using logistic regression typically involves:

1. **Feature Selection**: Identifying relevant features that significantly impact the target variable (e.g., default). Techniques like univariate feature selection or feature importance ranking are commonly used.
2. **Model Building**: Utilizing logistic regression due to its suitability for binary classification tasks like credit risk modeling.
3. **Parameter Estimation**: Estimating model parameters (coefficients) using maximum likelihood estimation (MLE), aiming to maximize the likelihood of observing the actual target values given the model's predictions.
4. **Model Evaluation**: Assessing model performance using metrics such as accuracy, precision, recall, F1 score, and ROC-AUC to gauge its ability to classify instances of default and non-default and its discriminatory power.
5. **Cross**-**Validation**: Employing cross-validation techniques like k-fold cross-validation to evaluate model robustness and generalization to new data.
6. **Hyperparameter Tuning**: Tuning hyperparameters, such as regularization strength, using techniques like grid search or random search for optimal model performance.
7. **Comparison with Challenger Models**: Optionally comparing the performance of the logistic regression model with other challenger models to identify the best-performing model for credit risk modeling.

**4. Explain the final model in terms of statistical results and business interpretation of regression coefficients**

After building the final logistic regression model for credit risk modeling, we interpret the statistical results and regression coefficients to understand how different features affect the probability of default for the bank's clients. The regression coefficients represent the log odds ratio of the target variable given a one-unit change in the corresponding predictor variable, holding all other variables constant. Positive coefficients indicate higher odds of default with an increase in the predictor variable, while negative coefficients suggest the opposite. Assessing the statistical significance of coefficients (p-value < 0.05) indicates their impact on predicting default probability. Business interpretation involves translating these insights into actionable strategies, such as setting income thresholds for loan approval based on income's positive and statistically significant coefficient. Evaluating model performance using metrics like accuracy and ROC-AUC further validates its predictive power and guides risk management decisions.

**5. Present the assumptions of the logistic regression and check if they are fulfilled by your model.**

import pandas as pd

import statsmodels.api as sm

from statsmodels.stats.outliers\_influence import variance\_inflation\_factor

X = data.drop(columns=['target']) # Assuming 'target\_column' is the name of the target variable

y = data['target']

# Converting non-numeric columns to numeric types

X\_numeric = X.apply(pd.to\_numeric, errors='coerce')

# Drop columns with missing values

X\_numeric = X\_numeric.dropna(axis=1)

# Adding a constant column to the features

X\_with\_intercept = sm.add\_constant(X\_numeric)

# Checking for multicollinearity using VIF

vif = pd.DataFrame()

vif["Features"] = X\_with\_intercept.columns

vif["VIF"] = [variance\_inflation\_factor(X\_with\_intercept.values, i) for i in range(X\_with\_intercept.shape[1])]

# Displaying the VIF Values

print(vif)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Features** | | **VIF** |  | **Features** | | **VIF** |
| **0** | **const** | **0** |  | **11** | **Var15** | **3.366418** |
| **1** | **ID** | **13.171982** |  | **12** | **Var16** | **3.320487** |
| **2** | **customer\_id** | **1.000647** |  | **13** | **Var20** | **2.111928** |
| **3** | **Var1** | **1.005483** |  | **14** | **Var21** | **4.548607** |
| **4** | **Var4** | **1.857456** |  | **15** | **Var22** | **6.817313** |
| **5** | **Var5** | **1.581046** |  | **16** | **Var23** | **4.39734** |
| **6** | **Var6** | **2.211871** |  | **17** | **Var24** | **1.036676** |
| **7** | **Var7** | **2.813279** |  | **18** | **Var27** | **1.356711** |
| **8** | **Var9** | **2.558324** |  | **19** | **Var28** | **1.356917** |
| **9** | **Var11** | **1.008221** |  | **20** | **Var29** | **1.000485** |
| **10** | **Var14** | **1.043588** |  | **21** | **Var30** | **14.790706** |
|  |  |  |  | **22** | **\_r\_** | **1.000398** |

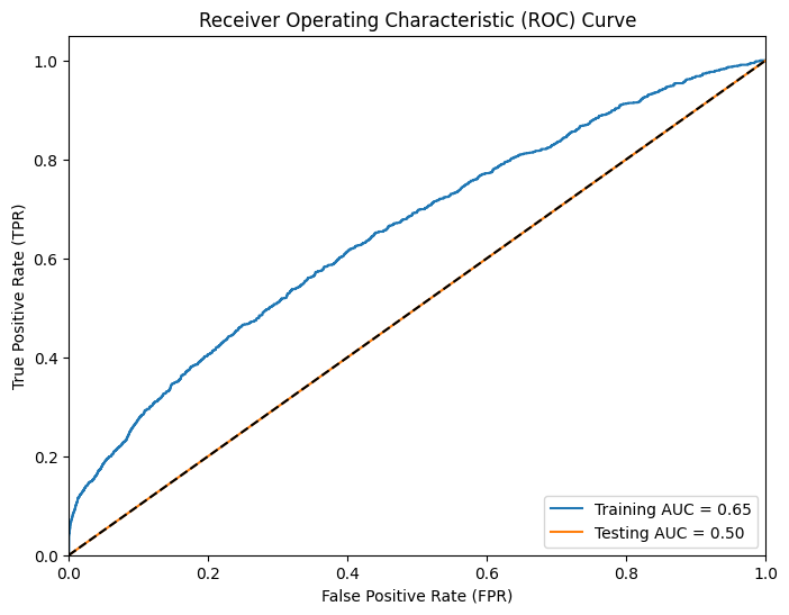
**6. Calculate the following performance metrics: Accuracy, Precision, Recall and F1 score both in Testing and Training samples.**

We have attached the only result here. The working phase and the coding is attached to a GitHub file and provided with a Link Below:

Calculations:

|  |  |
| --- | --- |
| **Performance metrics for training data:** | **Performance metrics for testing data:** |
| **Accuracy: 0.97** | **Accuracy: 0.03** |
| **Precision: 0.59** | **Precision: 0.03** |
| **Precision: 0.59** | **Recall: 1.00** |
| **F1 Score: 0.08** | **F1 Score: 0.06** |

**7.Create the ROC curve (AUC) and explain the discriminatory power of the model both in Testing and Training samples.**

****

**Task 1.2 – Challenger Model task**

1. **Propose and develop an alternative model to logistic regression to predict the probability of default for this portfolio, outlining its key features and advantages in the context of credit risk assessment.**

Random Forest Classifier is an alternative to logistic regression for credit risk assessment, offering several advantages:

Non-linearity Handling: Capable of capturing complex, non-linear relationships between features and the target variable.

Ensemble Learning: Aggregates predictions of multiple decision trees, reducing overfitting and enhancing generalization performance.

Feature Importance: Provides feature importance scores for identifying key predictors of default probability, aiding in risk assessment.

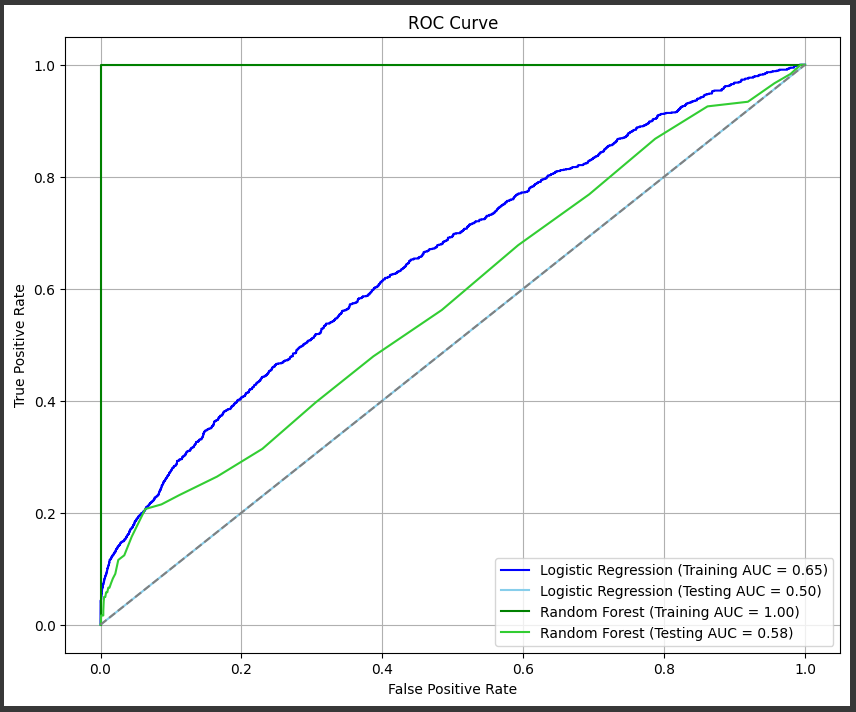
Robustness to Overfitting: Less susceptible to overfitting compared to individual decision trees, resulting in more reliable predictions.

Handling Imbalanced Data: Can address imbalanced datasets by adjusting class weights or using techniques like balanced subsampling, crucial in credit risk analysis where default cases may be scarce.

Versatility: Can handle various data types, including numerical and categorical features, with minimal preprocessing requirements.

**Code is Attached below**

1. **Compare and contrast the challenger model results with the logistic regression model you obtained in Task 1.1 (questions 6 and 7).**

****

1. **Explain the reasons behind different results from task 1.1**

The different results between the Random Forest Classifier and logistic regression models can be attributed to several factors:

* Model Complexity: Random Forest is more complex due to its ensemble nature, potentially capturing non-linear relationships better. However, this complexity might lead to overfitting if not properly tuned.
* Feature Importance: Random Forest provides feature importance scores, aiding in identifying influential features for credit risk assessment. Logistic regression lacks this feature, making feature selection less straightforward.
* Handling Imbalanced Data: Random Forest handles imbalanced datasets better, crucial in credit risk assessment where default cases may be scarce.
* Robustness to Outliers: Random Forest is more robust to outliers than logistic regression, which can affect model performance.
* Interpretability: Logistic regression models are more interpretable due to their coefficient-based nature, whereas Random Forest models can be harder to interpret due to their ensemble approach.
* Computational Efficiency: Logistic regression is generally faster than Random Forest, which can be computationally intensive, especially with large datasets.
* The choice between the models depends on factors like the dataset's complexity, interpretability requirements, and computational resources available for analysis.

1. **Discuss potential challenges or limitations associated with implementing each of the alternatives in a real-world banking scenario.**

Challenges and limitations of implementing each alternative in a banking scenario:

**Random Forest Classifier:**

Interpretability: Models are less interpretable due to their ensemble nature.

Computational Resources: Training can be computationally intensive.

Hyperparameter Tuning: Requires expertise and time for tuning.

Model Maintenance: Needs frequent updates and retraining.

Risk of Overfitting: Still possible if not properly regularized.

**Logistic Regression:**

Assumptions: Relies on assumptions that may not always hold.

Feature Engineering: Requires careful feature selection and engineering.

Handling Imbalanced Data: May struggle with imbalanced datasets.

Limited Flexibility: Linear model may not capture complex relationships.

Interpretability: Coefficients may be challenging to interpret, especially with many predictors.

In banking, balancing simplicity, interpretability, and computational efficiency is crucial for effective model deployment.