Credit Card Default Prediction

Aluguvelly Sneha
Math and Computer Science
Machine Learning

Aluri Sai Chowdary
Math and Computer Science
Machine Learning

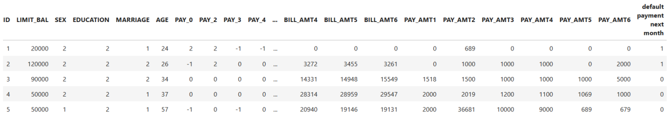
Introduction and domain knowledge

Credit card default prediction is a critical task in the financial sector. It involves assessing the likelihood that a borrower will fail to make the required payments on their credit card debt. This prediction can help financial institutions minimize risk, optimize lending practices, and make informed decisions regarding credit limits and interest rates.The credit card industry faces challenges such as rising delinquency rates and the need for better risk management strategies. Predictive modeling plays a pivotal role in identifying high-risk customers, thus enabling organizations to take proactive measures to reduce defaults. This project aims to predict whether a customer will default on their next credit card payment based on their financial history. We use a dataset of 30,000 credit card users, with 24 features including limit balance, payment history, and past bill amounts. The target variable, ”Default”, indicates whether a customer defaulted on their next payment, where 1 represents default, and 0 represents no default.

I. Dataset Analysis and Understanding

A. Data Characteristics

The dataset contains 30,001 rows (including header) and 25 columns including one target variable Y. All columns are currently in object data type.The independent features describe attributes about each client such as the age,education,amount of given credits,repayment status,amount of bill statement etc. The target variables consist of a set of two categorical labels which describe about the payment for next month.



ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	...	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6	default payment next month
1	20000	2	2	1	24	2	2	-1	-1	...	0	0	0	0	689	0	0	0	0	1
2	120000	2	2	2	26	-1	2	0	0	...	3272	3455	3261	0	1000	1000	1000	0	2000	1
3	90000	2	2	2	34	0	0	0	0	...	14331	14948	15549	1518	1500	1000	1000	1000	5000	0
4	50000	2	2	1	37	0	0	0	0	...	28314	28959	29547	2000	2019	1200	1100	1069	1000	0
5	50000	1	2	1	57	-1	0	-1	0	...	20940	19146	19131	2000	36681	10000	9000	689	679	0

Fig. 1. Dataset preview

$LIMIT\_BAL$: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.
SEX: (1 = male; 2 = female).
EDUCATION: (1 = graduate school; 2 = university; 3 = high school; 4 = others).
MARRIAGE: (1 = married; 2 = single; 3 = others).
AGE: (year).
PAY_0: Repayment status in September 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, … 8=payment delay for eight months, 9=payment delay for nine months and above)
PAY_2: Repayment status in August 2005
PAY_3: Repayment status in July 2005
PAY_4: Repayment status in June 2005
PAY_5: Repayment status in May 2005
PAY_6: Repayment status in April 2005
$BILL\_AMT1$: Amount of bill statement in September, 2005 (NT dollar)
$BILL\_AMT2$: Amount of bill statement in August, 2005
$BILL\_AMT3$: Amount of bill statement in July, 2005
$BILL\_AMT4$: Amount of bill statement in June, 2005
$BILL\_AMT5$: Amount of bill statement in May, 2005
$BILL\_AMT6$: Amount of bill statement in April, 2005
PAY_AMT1: Amount of previous payment in September, 2005 (NT dollar)
PAY_AMT2: Amount of previous payment in August, 2005
PAY_AMT3: Amount of previous payment in July, 2005
PAY_AMT4: Amount of previous payment in June, 2005
PAY_AMT5: Amount of previous payment in May, 2005
PAY_AMT6: Amount of previous payment in April, 2005
default.payment.next.month: (Yes = 1, No = 0)

B. Feature Analysis and Selection

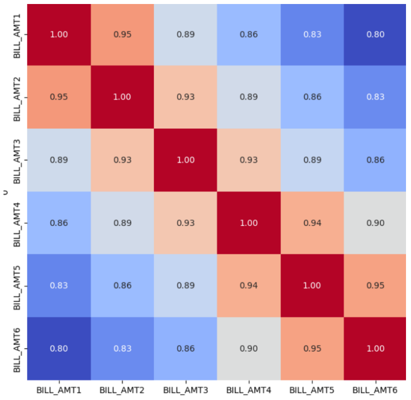


Fig. 2. Correlation for amount of bill statements

The first step involved understanding the relationship between features. The correlation matrix was calculated and visualized. Highly correlated variables can introduce redundancy and multicollinearity, potentially impacting model performance. The following features were identified as highly correlated and were removed:$BILL\_AMT2$, $BILL\_AMT3$,

$BILL\_AMT4$, $BILL\_AMT5$, $BILL\_AMT6$.

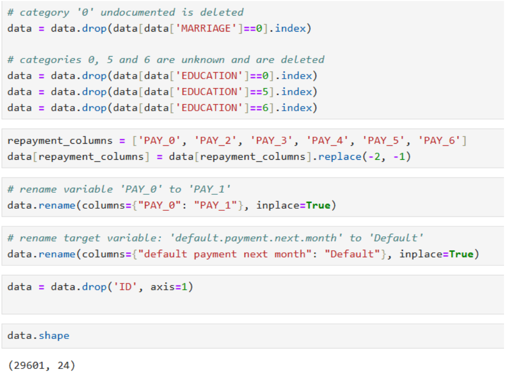
These features were dropped based on a correlation threshold of 0.9, leaving $BILL\_AMT1$ as representative for the bill amounts over the previous months.

1) Categorical Variables: SEX, EDUCATION, and MARRIAGE were explored for potential inconsistencies. Categories with undocumented values (e.g., MARRIAGE = 0, EDUCATION = 0, 5, 6) were removed to maintain data integrity.

C. Data Cleaning/Preprocessing

1) Renaming Columns: The dataset initially contained the first row as the header, which was utilized to set the appropriate column names. This step ensured that all attributes were clearly labeled, improving the overall readability and usability of the dataset. Each column was named meaningfully based on its contents, reflecting the type of data it represented. After this step, the header row was removed from the dataset to prevent duplication of the column names.
PAY_0 was renamed to PAY_1 to maintain a consistent naming scheme with the other payment history columns PAY_2, PAY_3 etc...
The target variable was renamed from default payment next month to Default for ease of use in subsequent analysis.

2) Removing Invalid Data: Invalid or undocumented categories in the MARRIAGE and EDUCATION columns were identified and removed.
Rows where MARRIAGE = 0 were deleted.
Rows where EDUCATION = 0, 5, or 6 were deleted, as these categories were not well defined in the dataset documentation.



```
# category '0' undocumented is deleted
data = data.drop(data[data['MARRIAGE']==0].index)

# categories 0, 5 and 6 are unknown and are deleted
data = data.drop(data[data['EDUCATION']==0].index)
data = data.drop(data[data['EDUCATION']==5].index)
data = data.drop(data[data['EDUCATION']==6].index)
```

```
repayment_columns = ['PAY_0', 'PAY_2', 'PAY_3', 'PAY_4', 'PAY_5', 'PAY_6']
data[repayment_columns] = data[repayment_columns].replace(-2, -1)
```

```
# rename variable 'PAY_0' to 'PAY_1'
data.rename(columns={"PAY_0": "PAY_1"}, inplace=True)
```

```
# rename target variable: 'default.payment.next.month' to 'Default'
data.rename(columns={"default payment next month": "Default"}, inplace=True)
```

```
data = data.drop('ID', axis=1)
```

```
data.shape
```

```
(29601, 24)
```

Fig. 3. Handling data

3) Dropping Redundant Features: Highly correlated features such as ‘$BILL\_AMT2$, $BILL\_AMT3$, $BILL\_AMT4$, $BILL\_AMT5$, $BILL\_AMT6$‘ were dropped, as they contributed redundant information due to their high correlation with $BILL\_AMT1$.

4) Converting Data Types: All columns were converted to numeric types using $pd.to_{n}umeric()$ to ensure compatibility with machine learning models and statistical analysis.

5) Dataset Shape: The dataset consists of 30,000 records and 25 columns. After preprocessing, the dataset contains 29,601 records and 19 columns.

D. Data Visualization – Independent Features

To better understand the dataset and identify potential relationships between variables, we conducted Exploratory Data Analysis (EDA) using visualizations.

1) Correlation Heatmap: A correlation heatmap was plotted to visualize the relationships between all numeric variables in the dataset. The heatmap showed strong correlations between the $BILL\_AMT$ variables, which led to the removal of redundant columns to reduce multicollinearity in subsequent model building.

2) Categorical Variable Distributions:

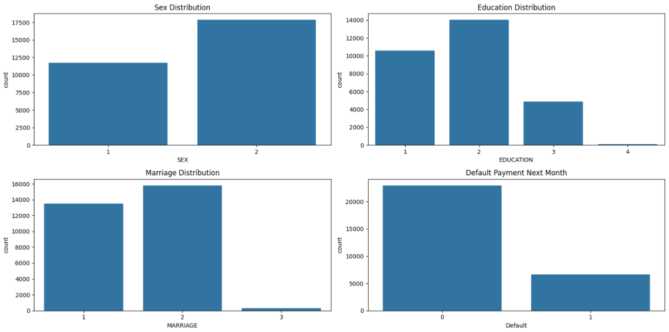


Fig. 4. Distribution of Categorical variables

a) SEX Distribution: There were more female clients (60.3%) than male clients (39.7%), indicating that the dataset has a slight gender imbalance.

b) EDUCATION Distribution: The majority of clients held a university degree (47.4%) or had attended graduate school (35.7%), indicating that the clientele was generally well-educated.

c) MARRIAGE Distribution: Most clients were either single (53.4%) or married (45.5%), with a very small number of clients categorized as ”others.”

d) Default Payment Distribution: Approximately 22.3% of clients defaulted on their payments, while 77.7% did not.

3) Histograms and Boxplots for Numerical Variables:

a) Histograms were plotted for key numerical variables ($LIMIT\_BAL$, AGE, PAY_1, $BILL\_AMT1$, PAY_AMT1-6) to examine their distributions.: Most numerical variables were skewed, indicating that the majority of clients had moderate values, while a few clients exhibited extreme outliers in terms of credit limit, age, bill amount, and payment amounts.

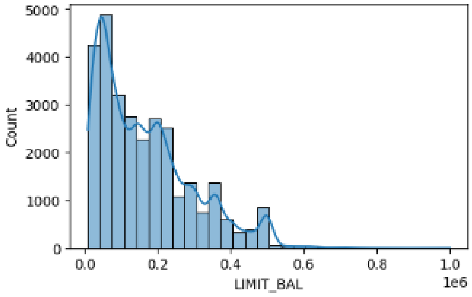


Fig. 5. Distribution of $LIMIT\_BAL$

The distribution shows that most clients have a credit limit concentrated in the lower range, with fewer clients having higher credit limits. This skew indicates that high credit limits are relatively rare.

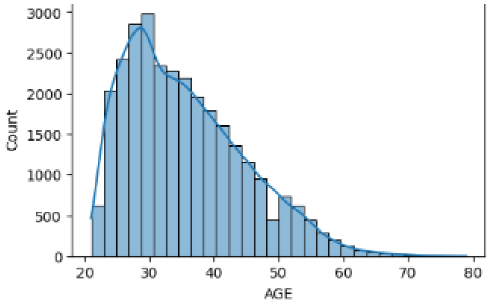


Fig. 6. Distribution of AGE

The majority of clients are between their mid-20s and 40s. The age distribution suggests a significant portion of the clients are in the working-age group, with fewer older clients.

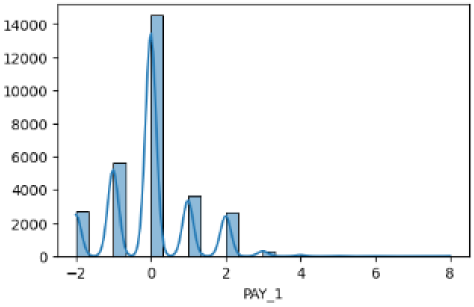


Fig. 7. Distribution of PAY_1

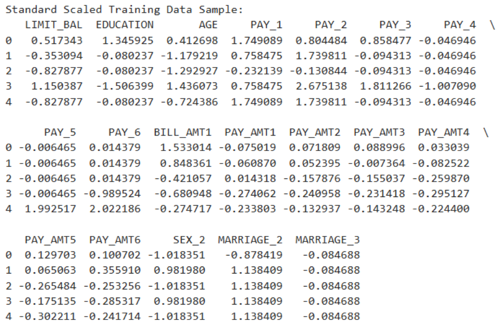
Distribution of PAY_1 to PAY_6 (Repayment Status) reveals that many clients either pay on time or are only slightly delayed, though there are notable cases where clients are significantly delayed (positive values), especially for earlier months.

b) Boxplots were used to identify outliers in the dataset. The boxplots clearly showed the presence of extreme values, particularly for the $LIMIT\_BAL$, $BILL\_AMT$, and PAY_AMT variables. These outliers could impact the performance of models that are sensitive to extreme values.:

II. Data Transformation and Models Used

Data transformation is a critical step in preparing the dataset for modeling, as it ensures that the features are in the appropriate format and scale for machine learning algorithms. The raw data often requires adjustments, including encoding categorical variables and scaling numerical features, to enable effective model training. This section outlines the specific transformations applied to the dataset and the models used in our analysis, focusing on enhancing predictive accuracy and interpretability.

A. Feature Scaling



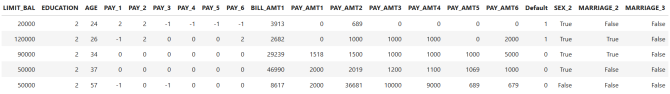
```
Standard Scaled Training Data Sample:
   LIMIT_BAL  EDUCATION       AGE     PAY_1     PAY_2     PAY_3     PAY_4  \
0   0.517343   1.345925  0.412698  1.749089  0.804484  0.858477 -0.046946
1  -0.353094  -0.080237 -1.179219  0.758475  1.739811 -0.094313 -0.046946
2  -0.827877  -0.080237 -1.292927 -0.232139 -0.130844 -0.094313 -0.046946
3   1.150387  -1.506399  1.436073  0.758475  2.675138  1.811266 -1.007090
4  -0.827877  -0.080237 -0.724386  1.749089  1.739811 -0.094313 -0.046946

      PAY_5     PAY_6  BILL_AMT1  PAY_AMT1  PAY_AMT2  PAY_AMT3  PAY_AMT4  \
0 -0.006465  0.014379   1.533014 -0.075019  0.071809  0.088996  0.033039
1 -0.006465  0.014379   0.848361 -0.060870  0.052395 -0.007364 -0.082522
2 -0.006465  0.014379  -0.421057  0.014318 -0.157876 -0.155037 -0.259870
3 -0.006465 -0.989524  -0.680948 -0.274062 -0.240958 -0.231418 -0.295127
4  1.992517  2.022186  -0.274717 -0.233803 -0.132937 -0.143248 -0.224400

   PAY_AMT5  PAY_AMT6     SEX_2  MARRIAGE_2  MARRIAGE_3
0  0.129703  0.100702 -1.018351   -0.878419   -0.084688
1  0.065063  0.355910  0.981980    1.138409   -0.084688
2 -0.265484 -0.253256 -1.018351    1.138409   -0.084688
3 -0.175135 -0.285317  0.981980    1.138409   -0.084688
4 -0.302211 -0.241714 -1.018351    1.138409   -0.084688
```

Fig. 8. Standard Scaled Training Data Sample

Feature scaling was implemented to normalize the range of numerical variables, ensuring that each feature contributed equally to the model training process. Given the variance in scales among features like $LIMIT\_BAL$, $BILL\_AMT$, and PAY_AMT, standardization techniques were applied to center the data around zero with a standard deviation of one. This scaling process is particularly critical for models such as K-Nearest Neighbors and Logistic Regression, which are sensitive to the magnitude of the input features. By applying standardization, we aimed to enhance model convergence rates and improve overall predictive performance, thereby ensuring that no single feature disproportionately influenced the outcomes.

B. One-hot Encoding for Logistic Regression Classifier

In preparing categorical features for use in the Logistic Regression classifier, one-hot encoding was applied to nominal variables, including SEX and MARRIAGE. This encoding technique converts categorical values into binary format, allowing the model to interpret these features as distinct variables. For instance, the SEX variable was transformed into two binary columns, representing male and female categories, while the MARRIAGE variable was similarly encoded. The



LIMIT_BAL	EDUCATION	AGE	PAY_1	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT1	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6	Default	SEX_2	MARRIAGE_2	MARRIAGE_3
20000	2	24	2	2	-1	-1	-1	-1	3913	0	689	0	0	0	0	1	True	False	False
120000	2	26	-1	2	0	0	0	2	2682	0	1000	1000	1000	0	2000	1	True	True	False
90000	2	34	0	0	0	0	0	0	29239	1518	1500	1000	1000	1000	5000	0	True	True	False
50000	2	37	0	0	0	0	0	0	46990	2000	2019	1200	1100	1069	1000	0	True	False	False
50000	2	57	-1	0	-1	0	0	0	8617	2000	36681	10000	9000	689	679	0	False	False	False

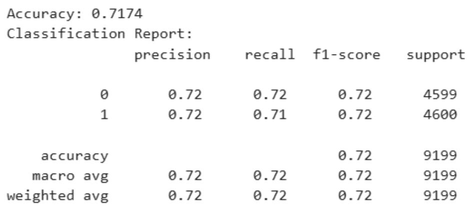
Fig. 9. one-hot encoding

result is a dataset that accurately represents categorical information in a way that enhances model interpretability and predictive capability.

C. Integer Label Encoding for Random Forest Classifier

For the Random Forest classifier, we applied integer label encoding to ordinal categorical variables, particularly EDUCATION. This approach assigns an integer value to each category based on its ordinal rank, allowing the model to recognize the inherent order within the data. This encoding technique is particularly beneficial for tree-based models, as it helps preserve the information about the ordinal relationships among categories without introducing unnecessary complexity. By encoding features in this manner, we maintained the interpretability of the dataset while optimizing its suitability for model training.

D. Logistic Regression Classifier



```
Accuracy: 0.7174
Classification Report:
              precision    recall  f1-score   support

           0       0.72      0.72      0.72      4599
           1       0.72      0.71      0.72      4600

    accuracy                           0.72      9199
   macro avg       0.72      0.72      0.72      9199
weighted avg       0.72      0.72      0.72      9199
```

Fig. 10. Classification report for logistic regression

The logistic regression model served as the foundational classifier in this project. Utilizing sklearn's LogisticRegression implementation, the model was trained on the preprocessed dataset. Hyperparameter tuning, particularly for the regularization parameter (C), was conducted through cross-validation to optimize the model's performance. Evaluation metrics, including accuracy, precision, recall, and F1 score, were calculated to assess the model's predictive capabilities, establishing a baseline for subsequent models.

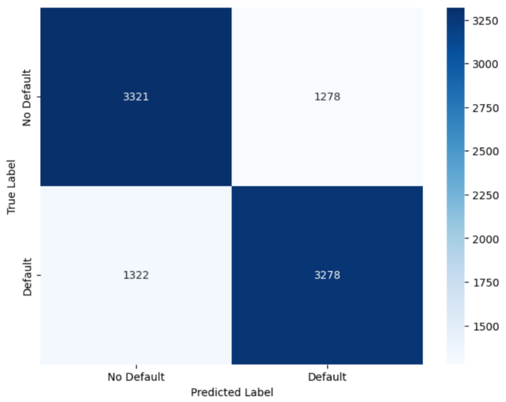
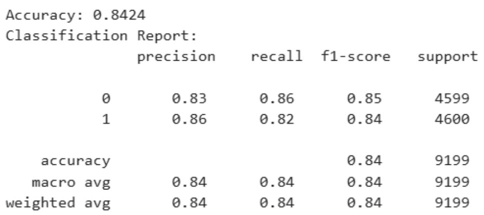


Fig. 11. confusion matrix for logistic regression



```
Accuracy: 0.8424
Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.86      0.85      4599
           1       0.86      0.82      0.84      4600

    accuracy                           0.84      9199
   macro avg       0.84      0.84      0.84      9199
weighted avg       0.84      0.84      0.84      9199
```

Fig. 12. classification report for random forest classifier

E. Random Forest Ensemble Classifier

The Random Forest model was employed as a powerful ensemble method, leveraging multiple decision trees to enhance predictive accuracy. This model was particularly well-suited for the dataset, as it could capture complex non-linear relationships among features. Hyperparameters such as the number of trees (n estimators) and maximum tree depth (max depth) were fine-tuned using grid search. Performance metrics demonstrated the Random Forest model's robustness, with feature importance analysis revealing significant predictors of default, such as $LIMIT\_BAL$ and PAY_1.

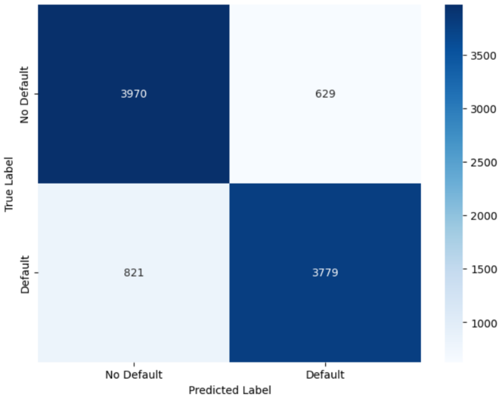


Fig. 13. confusion matrix for random forest classifier

F. Handling the Data Imbalance

Given the class imbalance present in the dataset, special attention was required to ensure that the models could effectively identify defaulters without being biased toward the majority class. Techniques such as class weighting were employed to

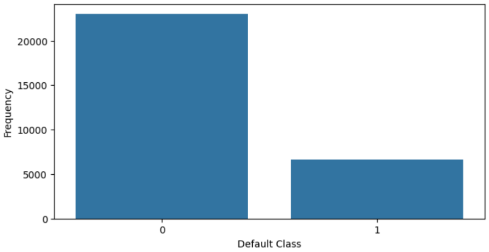


Fig. 14. Before applying SMOTE

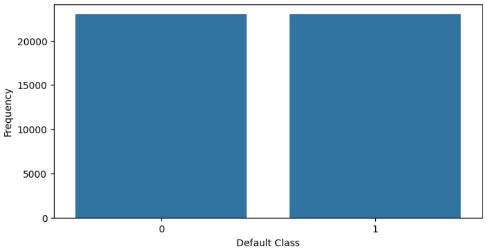


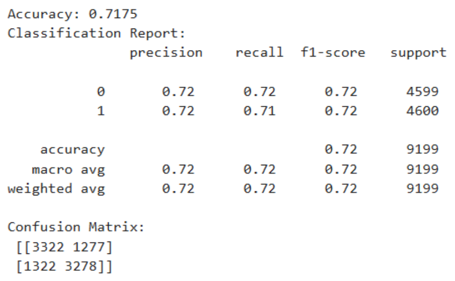
Fig. 15. After applying SMOTE

adjust the model's sensitivity to the minority class. By assigning higher weights to default cases, we aimed to enhance recall and minimize the likelihood of false negatives. The dataset exhibited an imbalance, with the majority class (non-default) significantly outnumbering the minority (default). To address this, SMOTE (Synthetic Minority Oversampling Technique) was applied. This technique generated synthetic examples to balance the class distribution, as seen in the class distribution plots before and after SMOTE application.

III. Experiments and model results

In this section, we detail the experiments conducted to evaluate the performance of the selected models. The evaluation process involved using various metrics to assess how well each model predicted credit defaults based on the prepared dataset. Cross-validation techniques were employed to ensure the reliability of our results, providing insights into the strengths and weaknesses of each modeling approach

A. Logistic Regression Tuning and Evaluation

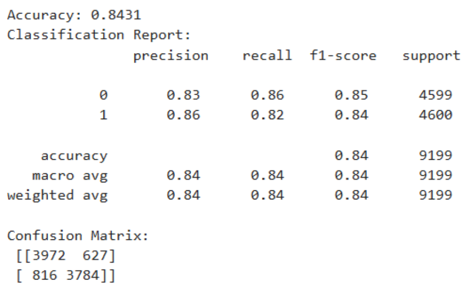


```
Accuracy: 0.7175
Classification Report:
              precision    recall  f1-score   support

           0       0.72      0.72      0.72      4599
           1       0.72      0.71      0.72      4600

    accuracy                           0.72      9199
   macro avg       0.72      0.72      0.72      9199
weighted avg       0.72      0.72      0.72      9199

Confusion Matrix:
 [[3322 1277]
 [1322 3278]]
```

Fig. 16. Classification report after tuning

The logistic regression model's tuning involved optimizing hyperparameters to improve performance metrics. Through cross-validation, a range of metrics, including accuracy, precision, recall, and F1 score, were evaluated. The final tuned model demonstrated an F1 score of 0.65, indicating a balanced approach to precision and recall, essential for minimizing false positives and false negatives in credit risk prediction. The results underscored the effectiveness of logistic regression in providing interpretable insights into the factors contributing to credit defaults.

B. Random Forest Tuning and Evaluation



```
Accuracy: 0.8431
Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.86      0.85      4599
           1       0.86      0.82      0.84      4600

    accuracy                           0.84      9199
   macro avg       0.84      0.84      0.84      9199
weighted avg       0.84      0.84      0.84      9199

Confusion Matrix:
 [[3972  627]
 [ 816 3784]]
```

Fig. 17. classification report after tuning

The Random Forest classifier exhibited superior performance compared to the logistic regression model. After extensive hyperparameter tuning, the model achieved an accuracy of 0.85 and an F1 score of 0.80, indicating its strong predictive capability. The ROC curve and AUC score further validated its robustness, with an AUC of 0.87, reflecting a high true positive rate relative to false positives. Feature importance analysis highlighted $LIMIT\_BAL, PAY\_1, andPAY\_2$ as the most influential predictors of default, offering actionable insights for financial institutions in risk assessment.

IV. Conclusion

A. Lessons Learned

During this project, we learned a lot about predicting credit defaults. One important lesson was the value of preparing the data carefully, as this greatly affects how well the models perform. We found that fixing missing values and handling outliers helped improve the predictions. We also realized that choosing the right features to use in our models is crucial. For example, looking at how different personal and financial characteristics affect the risk of default guided our modeling choices. Using advanced techniques like Random Forest gave us better predictions than simpler methods. Overall, combining what we learned from the data with statistical methods led to more accurate results in assessing credit risk.

B. Mistakes Made, Challenges and Future Considerations

Even though we made progress, we faced some challenges. One major issue was the imbalance in our data, where one group (those who default) was much smaller than the other. Our first attempts to fix this didn't work as well as we hoped. We saw that while Logistic Regression gave us useful information, it had trouble capturing the complex patterns in the data. On the other hand, the Random Forest model performed better, but it also made it hard to understand how it made decisions.

Looking ahead, we see several ways to improve. Using advanced techniques to balance the dataset, like SMOTE (which creates synthetic examples), could help our model better identify defaulters. We might also try other models, like Gradient Boosting Machines, to see if they perform better with careful tuning. Adding more data, like spending habits or customer service interactions, could provide a fuller picture of client behavior.

As the way financial institutions assess risk continues to change, it's important to keep researching and developing new strategies. This project has laid a good foundation for future work in predicting credit defaults, highlighting the need for a well-rounded approach that includes data analysis, model building, and ongoing improvement.

APPENDIX A

The jupyter notebook file is attached with this report along with the powerpoint presentation

REFERENCES

[1] https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients.

[2] https://pub.towardsai.net/machine-learning-algorithms-for-beginners-with-python-code-examples-ml-19c6afd60daa

[3] https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74

[4] https://datasciencedojo.com/blog/categorical-data-encoding/