Chang Li & Fengfei Niu
Professor Elisha Cohen
DS-UA 202 Responsible Data Science
1st May 2023

Course Project: Course Project: Technical Audit of an Automated Decision System
Credit Card Default Prediction System Fairness Evaluation

I. Background

The increasing availability of credit cards has boosted the volume of transactions and facilitated the growth of the global economy. However, it has also given rise to the problem of credit card defaults. Predicting credit card defaults has become a crucial concern for financial institutions to minimize losses and manage risks. This project aims to evaluate the fairness of an automated decision system (ADS) which could accurately predict credit card defaults, enabling financial institutions to make informed decisions and optimize risk management strategies.

In recent years, the financial industry has increasingly adopted automated decision systems (ADS) to streamline various processes, including credit risk assessment and credit card default predictions. These systems leverage machine learning algorithms to make accurate predictions and informed decisions. However, as the reliance on ADS grows, so does the concern about fairness and potential biases in these systems. The Fairness Evaluation of Predicting Credit Card Default System project aims to conduct a comprehensive technical audit of an ADS, focusing on its performance, accuracy, and fairness. This report will provide an overview of the chosen ADS, its purpose, and its goals, while also discussing the trade-offs introduced by pursuing multiple objectives.

a. Purpose of the ADS and its Stated Goals:

The purpose of the chosen ADS is to predict credit card defaults, enabling financial institutions to make informed decisions regarding credit risk management. The system does not mention any kind of attention to fairness, robustness, or feature importance. Only accuracy is considered when selecting the optimum model to predict defaults.

b. Trade-offs Introduced by Multiple Goals:

The pursuit of multiple goals may introduce trade-offs between accuracy and fairness in the ADS. First, maximizing accuracy might result in biased models that discriminate against specific demographic groups, leading to unfair treatment of certain customers. Second, ensuring fairness might require sacrificing some level of accuracy in the model, as adjustments made to reduce biases could also affect the model's overall predictive power.

In this project, we aim to strike a balance between these two goals to ensure that the credit card default prediction system remains both accurate and fair. By conducting a thorough technical audit, we will identify potential biases, evaluate the model's performance, and suggest methods to mitigate biases for the benefit of financial institutions and borrowers alike.

II. Input and output

a. Data Description and Collection:

The data used by this ADS comes from a public dataset available on the UCI Machine Learning Repository, which contains credit card data collected between April and September 2005. The dataset consists of various demographic, financial, and payment history attributes, providing a comprehensive view of customers' credit profiles. The data was collected by financial institutions and anonymized before being made available for research purposes.

b. Input Features:

There are eight input features in the dataset, including:

1. limit_bal: Amount of given credit (NT dollar) - continuous variable, includes individual consumer credit and family (supplementary) credit.
2. sex: Gender - categorical variable (1 = male; 2 = female).
3. education: Education level - categorical variable (1 = graduate school; 2 = university; 3 = high school; 4 = others).
4. marital_status: Marital status - categorical variable (1 = married; 2 = single; 3 = others).
5. age: Age in years - continuous variable.
6. history_of_past_payment: Past monthly payment records from April to September 2005 - ordinal variable (-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).
7. amount_of_bill_statement: Amount of bill statement (NT dollar) from April to September 2005 - continuous variable.
8. amount_of_previous_payment: Amount of previous payment (NT dollar) from April to September 2005 - continuous variable.

For each input feature, a detailed analysis should be conducted to identify missing values, outliers, and value distributions. Pairwise correlations between features can be calculated to identify any multicollinearity issues. Additional profiling can be done to identify any interesting patterns or relationships between variables.

c. Output of the System:

The output of the system is a binary classification, indicating whether a customer will default on their credit card payment or not:

Target: Did the customer default? (Yes=1/Positive, No=0/Negative)

This output can be interpreted as a risk assessment for financial institutions to make informed decisions about credit risk management. A customer predicted to default (Yes = 1) represents a higher credit risk, while a customer predicted to not default (No = 0) represents a lower credit risk. The output of the credit card default prediction system predicts whether a

customer will default on their credit card payment or not, lies in its potential impact on both financial institutions and borrowers. The output serves as an essential tool for risk management, credit allocation, and decision-making processes within the financial industry. It helps optimize risk management, enhances decision-making processes, and contributes to fair and transparent credit assessment practices.

III. Implementation and validation

The ADS from Medium implemented various machine learning methodologies aimed at improving the accuracy of predictions. The process includes exploratory data analysis, baseline model creation, performance metrics evaluation, optimization, feature importance analysis, hyperparameter tuning, and class imbalance handling.

a. Data Cleaning and Preprocessing:

Data cleaning and preprocessing involve a series of steps to transform raw data into a clean, well-structured, and ready-to-use format for analysis or modeling. To enhance the performance of credit card prediction models, it is essential to employ preprocessing techniques for cleaning and refining the dataset. Preprocessing refers to the process of preparing raw data for analysis or modeling. Preprocessing involves cleaning, transforming, and organizing data to make it suitable for a specific task, such as training a machine learning model or conducting statistical analysis. The data cleaning and preprocessing steps usually involve the following:

First, it is necessary to conduct data cleaning by handling missing values, duplicate records, or outliers and removing noise, inconsistencies, or errors from the data. In the original dataset, we first exploit the function print(df.isnull().values.sum()) to count the total number of missing values (null or NaN) in the dataset. However, the output of print(df.isnull().values.sum()) is 0, which indicates that there are no missing values in the dataset. This is a desirable outcome in preprocessing, as it means the dataset is clean and ready for further analysis or modeling without the need to handle missing values. Second, we rename the column name to make it more clear and understandable. Third, we remove the noise by dropping the ID column. This is because the ID column is an identifier, which contains no predictive value. By conducting the data cleaning, we are able to remove the noise and create high-quality datasets that lead to better results and more reliable conclusions.

Second, data transformation can effectively convert data into a different format or representation to make it more suitable for analysis. Examples include normalization (scaling data to a common range), standardization (scaling data to have zero mean and unit variance), and encoding categorical variables (e.g., one-hot encoding). In the original dataset, it is necessary for us to encode categorical variables, which convert categorical variables, such as sex, education, and marital status, into numerical values using encoding techniques like one-hot encoding or label encoding. In the ‘Sex’ column, we encode ‘male’ to 1 and ‘female’ to 2. In the ‘Education’ column, we encode ‘graduate’ to 1, ‘university’ to 2, ‘highschool’ to 3 ,the first ‘ other’ to 4, the second ‘other’ to 6, and the third ‘other’ to 0. In the ‘marriage’ column, we encode ‘married’ to

1, ‘single’ to 2, the first ‘other’ to 3, and the second ‘other’ to 0. In addition, we also reset the range of values for payment history to make the dataset clear and understandable. By conducting data transformation, it helps to transform non-numerical data into a suitable format for machine learning algorithms, allowing for better analysis and improved model performance.

By conducting data clean and preprocessing to remove the noise, encoding categorical variables, and rename the range of columns, it is beneficial to create a high-quality dataset that will lead to better results in downstream tasks, such as model training and prediction.

b. High-Level Information about the Implementation:

The implementation of the ADS involves seven methodologies including exploratory data analysis, baseline model creation, performance metrics evaluation, optimization, feature importance analysis, hyperparameter tuning, and class imbalance handling.

1. **Exploratory Data Analysis**

First, the author utilizes Exploratory Data Analysis (EDA) to analyze the dataset to identify patterns, relationships, and anomalies using visualization techniques, such as distribution plots and correlation matrices. EDA is a critical step in understanding and analyzing the dataset before building models or conducting further analysis. It involves identifying patterns, relationships, and anomalies using various visualization techniques, such as distribution plots, correlation matrices, and bar charts.

Here are some key findings in the data, with credit to Gabriel Preda from Kaggle for inspiring the visualization ideas (find his work here: https://www.kaggle.com/gpreda). First, the target classes are imbalanced. The distribution of target classes is highly imbalanced, with non-defaults significantly outnumbering defaults. This is a common observation in credit card datasets since most people tend to pay their credit card bills on time, barring any economic crises. Second, the EDA showcases payment status correlations. The strength of the correlation between payment statuses increases as the months get closer in time. This makes sense because a late payment in August is more likely to lead to a late payment in September, compared to the likelihood of a late payment in April causing a late payment in September. Third, EDA plots the distribution of credit limit amounts. The three most common credit limit amounts in the dataset are $50k, $20k, and $30k, respectively. Fourth, EDA provides information about credit limits by gender. The credit limit data is evenly distributed between males and females, indicating no apparent bias in credit limits based on gender. Fifth, EDA suggests information about marriage, age, and gender. The dataset mainly comprises couples in their mid-30s to mid-40s and single individuals in their mid-20s to early-30s. This demographic information can be valuable when exploring patterns and relationships in credit card payment behavior.

These insights from EDA help to inform data preprocessing decisions, identify potential issues (e.g., class imbalance), and guide feature engineering and selection. By understanding the characteristics of the dataset, better models can be developed in order to achieve higher accuracy for the ADS.

2. Baseline Model

The baseline model is an initial model using default parameters to serve as a benchmark for comparison with optimized models. The implementation of a prediction system starts by establishing a baseline model and progresses to preparing features and targets, scaling data, and evaluating several models, with the end goal of making a model with the best performance by tweaking the hyperparameters and selecting the most relevant features.

First, the author prepares features and targets by selecting the relevant features and the target variable from the dataset, applying any necessary preprocessing steps, such as encoding categorical variables or scaling. Second, the author scales the data to ensure that all features are on a similar scale, making it easier for the model to process the information by using techniques like Min-Max scaling or standardization. Third, the author evaluates the model evaluation by scoring several models using cross-validation or a separate validation dataset, evaluating their performance based on accuracy or other relevant metrics. In this case, several models were tested, including KNN, Logistic Regression, Random Forest, XGBoost, and Support Vector Machine (SVM). The fourth step is model selection, which chooses a model to improve upon based on its performance and computational efficiency.

The results provided show the accuracy scores of five different machine learning models: K-Nearest Neighbors (KNN), Logistic Regression, Random Forest, XGBoost, and Support Vector Machine (SVM). Accuracy score is a performance metric that measures the proportion of correct predictions made by the model out of the total number of predictions. The K-Nearest Neighbors (KNN) model has an accuracy score of 0.7938, which means it correctly predicted about 79.38% of the test data points. KNN is a non-parametric, instance-based learning algorithm used for classification and regression tasks. The Logistic Regression model has an accuracy score of 0.8158, which means it correctly predicted about 81.58% of the test data points. Logistic Regression is a statistical method used to model the probability of a certain class or event, like binary classification tasks. The Random Forest model has an accuracy score of 0.8168, which means it correctly predicted about 81.68% of the test data points. Random Forest is an ensemble learning method that constructs multiple decision trees and combines their outputs to improve the overall accuracy and control overfitting. The XGBoost model has an accuracy score of 0.8192, which means it correctly predicted about 81.92% of the test data points. XGBoost (Extreme Gradient Boosting) is a gradient boosting algorithm that optimizes the model by minimizing a loss function using gradient descent. The Support Vector Machine (SVM) model has the highest accuracy score of 0.8193, which means it correctly predicted about 81.93% of the test data points. SVM is a supervised learning algorithm that can be used for classification and regression tasks by finding the optimal hyperplane that separates the data into different classes.

In this scenario, the SVM model had the highest accuracy (0.8193), but the Random Forest model was selected because it was nearly as accurate (0.8168) and less computationally expensive. With the baseline model established (Random Forest, in this case), we can proceed to

fine-tune the model, optimize hyperparameters, perform feature engineering, and apply other techniques to enhance the model's performance further.

In our version of Jupyter Notebook, the evaluations of models except Random Forest are skipped in order to save execution time.

3. Optimization

Before creating performance metrics, the author utilized optimization to identify areas for improvement, such as feature selection, model selection, and hyperparameter tuning. The previous Baseline model part suggests that the system has been implemented for a random forest classification, with a focus on optimizing recall as the primary performance metric. Recall, also known as sensitivity or true positive rate (TPR), is particularly important when the cost of false negatives is high, such as in detecting defaults or medical diagnosis. In this context, the system is designed to minimize False Negatives (FN), which occur when a default is incorrectly predicted as a non-default. The recall formula is as follows:

Recall = True Positives (TP) / (True Positives (TP) + False Negatives (FN))

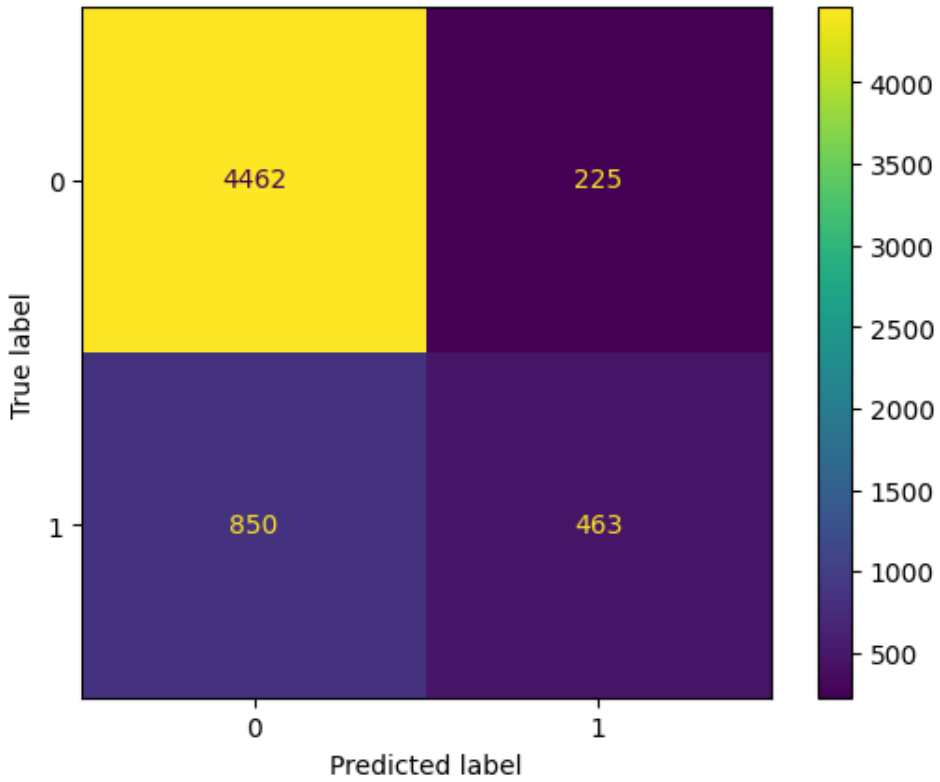
Although precision measures the proportion of true positive predictions among all positive predictions, recall measures the proportion of true positive predictions among all actual positive instances. Balancing these metrics is crucial, as optimizing one may come at the expense of the other.

4. Performance Metrics

In this part, the author evaluates the performance of the model by using a confusion matrix, recall, precision, accuracy, and F-1 score.

A confusion matrix provides a visual representation of the model's performance, showing the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) for a classification model. In this case, the bottom-left quadrant of the matrix, which represents the false negatives (FN), is the number we want to minimize. A high recall score, such as 0.95, indicates that the system is doing well in identifying true positive instances (customers who did default) and minimizing false negatives (customers predicted not to default but who did default). Second, recall measures the proportion of actual positive instances that the model identified correctly. A high recall score, like 0.95, indicates that the model is effective in minimizing false negatives, which is the primary objective in this case. Third, precision represents the proportion of true positive predictions among all positive predictions. Balancing precision with recall is important, as focusing solely on one metric may lead to suboptimal results. Fourth, accuracy measures the overall proportion of correct predictions (TP + TN) out of the total number of instances. It is a common performance metric but may not be as informative in cases of imbalanced class distributions. Last, F1-score refers to the harmonic mean of precision and recall, providing a single metric that balances the trade-off between them. A higher F1-score indicates better overall performance.

From the classification report, we can see that the original model had a recall of 94% for clients who did not default, and 35% for clients who did default. The difference between the two classes is expected because of the imbalance for our dataset.



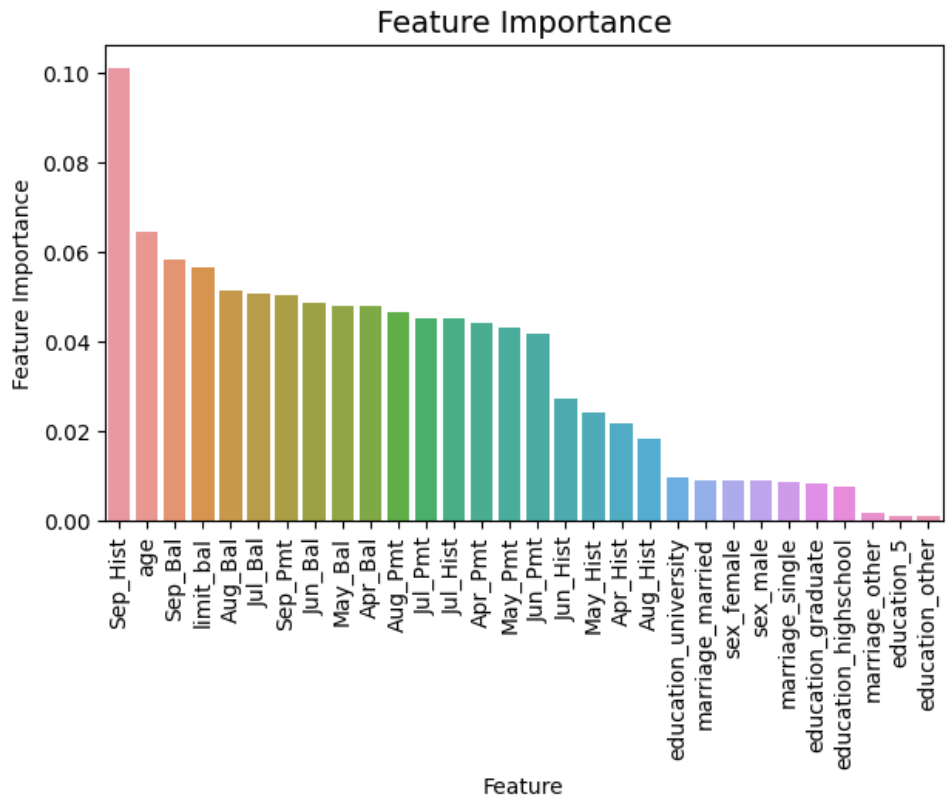
5. Feature Selection

The author employed feature importance as the feature selection method. The importance score for each feature is extracted from the Random Forest model and sorted. The 6 least important features were removed from the dataset in this step.

By removing the least important features and retaining the most relevant ones, the system can potentially achieve:

1. Better predictive performance since the model might be able to make more accurate predictions using the most important features.
2. Reduced complexity because the model will have fewer input variables, which may lead to reduced computational complexity and faster training times.
3. Improved interpretability with fewer features, it becomes easier to understand the relationships between the input variables and the target variable.

After applying feature selection, the model should be retrained using the selected features, and its performance should be evaluated again with the chosen performance metrics (e.g., recall, precision, F1-score). This process might need to be repeated iteratively to find the optimal set of features that result in the best model performance.



6. Hyperparameter Tuning

In the implementation of the system, hyperparameter tuning is employed to improve the performance of the RandomForestClassifier. Hyperparameter tuning involves searching for the optimal set of hyperparameters that result in the best performance of the model. In this case, RandomizedSearchCV is used to perform the search efficiently. RandomizedSearchCV performs a random search on hyperparameters by sampling a given number of combinations and evaluating them using cross-validation. The main hyperparameters explored in this case are:

bootstrap: [True, False]
max_features: [‘auto’, ‘sqrt’]
min_samples_leaf: [1, 2, 4]
min_samples_split: [2, 5, 10]
n_estimators: [50, 100, 150, 200]
max_depth: [4, 6, 10, 12, None]

The parameter distributions are specified in the param_dist dictionary, and the search is conducted with 10 iterations and 10-fold cross-validation, using 'accuracy' as the scoring metric. After fitting the RandomizedSearchCV to the training data, the best model's performance score and parameters are extracted:

Best score (accuracy): 0.8209
Best parameters:

n_estimators: 100
min_samples_split: 10
min_samples_leaf: 2
max_features: 'auto'
max_depth: 6
bootstrap: False

Additionally, the mean score and standard deviation of the best model are also reported (Mean score: 0.8209, Std: 0.0081). With the best parameters found through hyperparameter tuning, the RandomForestClassifier can be retrained and evaluated on the validation and test datasets to assess its performance.

7. Class Imbalance

The implementation of the system also addresses the challenge of class imbalance in the target variable. Class imbalance occurs when the number of instances in one class significantly outnumbers the instances in the other class. This can lead to biased predictions as most machine learning algorithms are designed with the assumption of an equal number of examples for each class.

To tackle class imbalance, two common techniques, random undersampling and oversampling, are employed in this case. Random Undersampling reduces the number of instances in the majority (negative) class by randomly deleting data points, leading to a balanced target distribution. This method can help address class imbalance but may lead to loss of information from the majority class. However, Random Oversampling increases the number of instances in the minority (positive) class by randomly duplicating data points, resulting in a balanced target distribution. Although this method can improve the balance between classes, it may introduce overfitting due to duplicate data points.

The system then evaluates the performance of the models using both undersampling and oversampling strategies. The recall scores for both techniques are 0.79, which is a decrease of nearly 0.16 from the baseline model. This result indicates that the class imbalance remediation methods employed might not be effective in improving the model's performance in this particular case.

c. ADS Validation and Goal Achievement:

The ADS (Automated Decision System) is validated through several steps to ensure it meets its stated goal(s), which in this case, is to achieve a high recall score to accurately predict potential defaults.

Model performance comparison: The simplest model, with minimal manipulation, achieved the highest recall score of 0.95. After feature selection and hyperparameter tuning, the recall score decreased to 0.79, indicating that sometimes the best model might be the simplest.

Overfitting check: To ensure that the model generalizes well to unseen data, it is evaluated on a validation dataset. If the validation score is similar to the test score, it suggests that the model performs consistently on completely unseen data, and there is no overfitting.

ROC Curve: The Receiver Operating Characteristic (ROC) curve is used to assess the model's ability to separate the different classes in the dataset. It plots the true positive rate against the false positive rate. The area under the ROC curve (AUC-ROC) ranges from 0 to 1, where 0 means the model predicted all data incorrectly, and 1 means the model predicted all data correctly. In this case, the AUC-ROC score is 0.7747, indicating a reasonably good separation of classes.

Having a model with a recall score of 0.95 suggests that it can predict 95% of potential defaults, potentially saving a significant amount of money on credit card charge-offs. Although real-world applications might be more nuanced, this modeling process serves as a solid starting point for building an effective and efficient ADS. It's crucial to continue validating the system by regularly updating the dataset, monitoring its performance, and making necessary adjustments to ensure its effectiveness in meeting its stated goal(s).

IV. Outcomes

a. Analyze the accuracy

To analyze the accuracy of the Credit Card Default Prediction System, we will compare its performance across different subpopulations using various accuracy metrics. We use the Aequitas package in this project to generate performance metrics for each group and intersections of groups for our dataset based on two sensitive features: sex and amount. The choice of accuracy metrics will be based on their ability to provide a comprehensive understanding of the prediction system's performance, taking into account different aspects of classification accuracy. The chosen metrics are as follows:

- **True Positive Rate (TPR)**

First, we will focus on the True Positive Rate (TPR) across different subpopulations based on the attributes sex, amount, and sex_amount. TPR is an important metric because it measures the proportion of actual positive instances (defaulters) that are correctly identified by the prediction system. A higher TPR indicates better performance in identifying defaulters.

Based on the provided TPR values: Using the attribute 'sex': TPR for male: 0.35, and TPR for female: 0.36. There is a slight difference in TPR between males and females, with females having a marginally higher TPR. This suggests that the system is slightly better at identifying female defaulters compared to male defaulters. Using the attribute 'amount': TPR for low amount: 0.42, and TPR for high amount: 0.24. The system performs better in identifying defaulters who want to borrow low amounts compared to those who want to borrow high amounts. The difference in TPR values indicates that the system might have a bias towards low-amount borrowers in terms of default prediction. Using the attribute 'sex_amount': TPR for male_low: 0.38, TPR for male_high: 0.29, TPR for female_low: 0.45, TPR for female_high:

0.21. When considering both sex and amount attributes, the system performs best in identifying female defaulters with low credit amounts, while it performs the worst in identifying female defaulters with high credit amounts. This suggests that the system might have biases towards certain subpopulations, which could lead to disparities in the system's performance.

In conclusion, analyzing TPR values across different subpopulations reveals potential biases in the Credit Card Default Prediction System. It's essential to consider other accuracy metrics, as mentioned in the previous response, to have a comprehensive understanding of the system's performance and address any biases or disparities in the predictions. By doing so, the system's accuracy and fairness can be improved.

- **True Negative Rate (TNR)**

Second, the True Negative Rate (TNR) metric can also provide insights into the system's performance. TNR measures the proportion of actual negative instances (non-defaulters) that are correctly identified by the prediction system. A higher TNR indicates better performance in identifying non-defaulters.

Based on the provided TNR values: Using the attribute 'sex': TNR for male and female are both 0.95, indicating similar performance in identifying non-defaulters for both genders. Using the attribute 'amount': TNR for low amount is 0.91, and TNR for high amount is 0.99, indicating significantly better performance in identifying non-defaulters who want to borrow high amounts. Using the attribute 'sex_amount': TNR for male_low is 0.90, TNR for male_high is 0.99, TNR for female_low is 0.91, and TNR for female_high is 0.99. The system performs better in identifying male defaulters compared to female defaulters when considering both sex and amount attributes.

While TNR values suggest that the system performs well in identifying non-defaulters, it is equally important to ensure that it does not incorrectly identify non-defaulters as defaulters (False Positive Rate, FPR) or incorrectly identify defaulters as non-defaulters (False Negative Rate, FNR). Furthermore, other accuracy metrics such as Precision and Recall can provide a more comprehensive understanding of the system's performance. Therefore, it is important to analyze the system's performance using multiple accuracy metrics to identify potential biases and improve the system's accuracy and fairness.

- **False Omission Rate (FOR):**

False Omission Rate(FOR) is an important metric to analyze the performance of the Credit Card Default Prediction System. FOR measures the proportion of actual negative instances (non-defaulters) that are incorrectly identified as defaulters by the system. A lower FOR indicates better performance in avoiding false alarms or incorrectly flagging non-defaulters as defaulters.

Based on the FOR values: Using the attribute 'sex': FOR for male: 0.17, and FOR for female: 0.15. The system performs slightly better in avoiding false alarms for female non-defaulters compared to male non-defaulters. Using the attribute 'amount': FOR for low

amount: 0.21, and FOR for high amount: 0.12. The system performs better in avoiding false alarms for high-amount non-defaulters compared to low-amount non-defaulters. This indicates that the system might have a bias towards low-amount borrowers in terms of default prediction, as previously mentioned based on TPR values. Using the attribute 'sex_amount': FOR for male_low: 0.22, FOR for male_high: 0.13, FOR for female_low: 0.20, FOR for female_high: 0.12. The system performs slightly better in avoiding false alarms for female_high non-defaulters compared to other subpopulations.

In conclusion, analyzing the FOR values across different subpopulations highlights potential biases and disparities in the Credit Card Default Prediction System. It's crucial to consider other accuracy metrics, as previously mentioned, to have a comprehensive understanding of the system's performance and address any biases or disparities in the predictions. By doing so, the system's accuracy and fairness can be improved.

- **False Discovery Rate (FDR)**

The False Discovery Rate (FDR) is a metric that measures the proportion of predicted positive instances (defaulters) that are actually negative (non-defaulters). A lower FDR indicates better performance in predicting defaulters while minimizing the number of false positives.

Based on the provided FDR values: Using the attribute 'sex': FDR for male: 0.33, and FDR for female: 0.33. The FDR is the same for both male and female subpopulations, indicating that the system has similar performance in terms of false positives across genders. Using the attribute 'amount': FDR for low amount: 0.35, and FDR for high amount: 0.24. The FDR is higher for low-amount borrowers, suggesting that the system may be over predicting defaulters among those who want to borrow smaller amounts. Using the attribute 'sex_amount': FDR for male_low: 0.38, FDR for male_high: 0.17, FDR for female_low: 0.33, FDR for female_high: 0.29. The FDR is the highest for male_low and the lowest for male_high, indicating that the system has a higher proportion of false positives among male low-amount borrowers and the lowest among male high-amount borrowers.

In conclusion, analyzing FDR values across different subpopulations highlights potential biases in the Credit Card Default Prediction System. The system may have a tendency to overpredict defaulters among certain subpopulations, such as low-amount borrowers, and may not be equally accurate across all subpopulations. Therefore, it is crucial to consider other accuracy metrics and potential biases to improve the accuracy and fairness of the prediction system.

- **False Positive Rate (FPR)**

The False Positive Rate (FPR) is an important accuracy metric to consider as it measures the proportion of actual negative instances (non-defaulters) that are incorrectly classified as positive instances (defaulters) by the prediction system. A lower FPR indicates better performance in correctly identifying non-defaulters, which is crucial in reducing false alarms and unnecessary credit denials.

Based on the provided FPR values: Using the attribute ‘sex’: FPR for male is 0.05, and FPR for female is 0.05. The FPR values are the same for both subpopulations, suggesting that the system performs equally well in correctly identifying non-defaulters for both male and female individuals. Using the attribute ‘amount’: FPR for low amount is 0.09, and FPR for high amount is 0.01. The FPR values differ significantly for low and high amounts, suggesting that the system might be biased towards non-default predictions for individuals who want to borrow high amounts. Using the attribute ‘sex_amount’: FPR for male_low is 0.10, FPR for male_high is 0.01, FPR for female_low is 0.09, and FPR for female_high is 0.01. The FPR values differ slightly for different subpopulations, but the pattern of bias towards non-default predictions for individuals who want to borrow high amounts is consistent.

In conclusion, analyzing FPR values across different subpopulations reveals potential biases in the Credit Card Default Prediction System towards non-default predictions for individuals who want to borrow high amounts. It's essential to consider other accuracy metrics, as mentioned in the previous responses, to have a comprehensive understanding of the system's performance and address any biases or disparities in the predictions. By doing so, the system's accuracy and fairness can be improved.

- **False Negative Rate(FNR)**

False Negative Rate (FNR) is an important metric to evaluate the performance of a credit card default prediction system. FNR measures the proportion of actual defaulters that are incorrectly classified as non-defaulters by the prediction system. A lower FNR indicates that the prediction system is better at identifying defaulters, which is a crucial aspect of credit card default prediction.

Based on the provided FNR values, using the attribute ‘sex’: FNR for male: 0.65, and FNR for female: 0.64. The difference between FNR values is relatively small, indicating that the system performs similarly in identifying defaulters for both males and females. Using the attribute ‘amount’: FNR for low amount: 0.58, and FNR for high amount: 0.76. The system performs better in identifying defaulters with low amounts of borrowing compared to those with high amounts, which could be an indication of a potential bias towards low-amount borrowers. Using the attribute ‘sex_amount’: FNR for male_low: 0.62, FNR for male_high: 0.71, FNR for female_low: 0.55, and FNR for female_high: 0.79. The system performs best in identifying female_low defaulters and performs the worst in identifying female_high defaulters. The difference in FNR values suggests that the system might have biases towards certain subpopulations, which could lead to disparities in the system’s performance.

In conclusion, analyzing FNR values across different subpopulations can provide insights into potential biases in the Credit Card Default Prediction System. A lower FNR value indicates better performance in identifying defaulters, which is a crucial aspect of credit card default prediction. It's essential to consider other accuracy metrics, as mentioned earlier, to have a comprehensive understanding of the system’s performance and address any biases or disparities in the predictions. By doing so, the system’s accuracy and fairness can be improved.

- **Precision**

Precision is an accuracy metric that measures the proportion of positive predictions (predicted defaulters) that are actually true positives (actual defaulters). A high precision value indicates that the system has a low false positive rate and is accurate in predicting defaulters.

Analyzing the Precision values across different subpopulations, we can see that the system has similar Precision values for males and females when using the sex attribute. However, when considering the amount attribute, the Precision value for high amount borrowers is higher than that of low amount borrowers. This indicates that the system is more accurate in predicting defaulters who want to borrow high amounts compared to those who want to borrow low amounts. When considering both sex and amount attributes, the system is more accurate in predicting male_high defaulters, with a Precision value of 0.83. In contrast, the Precision value for female_high defaulters is only 0.71, indicating that the system is less accurate in predicting female_high defaulters.

Overall, analyzing Precision values across different subpopulations provides insights into the system's accuracy in predicting defaulters in different subgroups. However, it is essential to consider other accuracy metrics, such as recall, F1 score, and accuracy, to have a comprehensive understanding of the system's performance.

- **Predictive Positive Value (PPREV)**

Predictive Positive Value (PPREV) is a metric that measures the proportion of predicted positive instances (defaulters) that are actually positive. It is an important accuracy metric for the Credit Card Default Prediction System as it helps evaluate the usefulness of the system's predictions in real-world scenarios.

Analyzing the PPREV values across different subpopulations, we can observe that the system performs better in predicting defaulters with lower amounts of borrowing. The PPREV for low amounts is 0.19, while for high amounts, it is only 0.05. This indicates that the system might be more effective in identifying defaulters with low amounts of borrowing and that its predictions might not be as useful for high-amount borrowers. Moreover, analyzing the PPREV values by considering the sex and amount attributes together, we can see that the system performs better in predicting female_low defaulters with a PPREV of 0.19, while its prediction performance is the weakest for female_high borrowers, with a PPREV of only 0.04. This suggests that the system might be biased towards certain subpopulations, which could result in disparities in its prediction accuracy.

In conclusion, analyzing the PPREV values across different subpopulations helps identify the strengths and weaknesses of the Credit Card Default Prediction System and its usefulness in real-world scenarios. To improve the system's accuracy and fairness, it is essential to consider other accuracy metrics, as discussed earlier, and address any biases or disparities in its predictions.

- **Prevalence (PREV)**

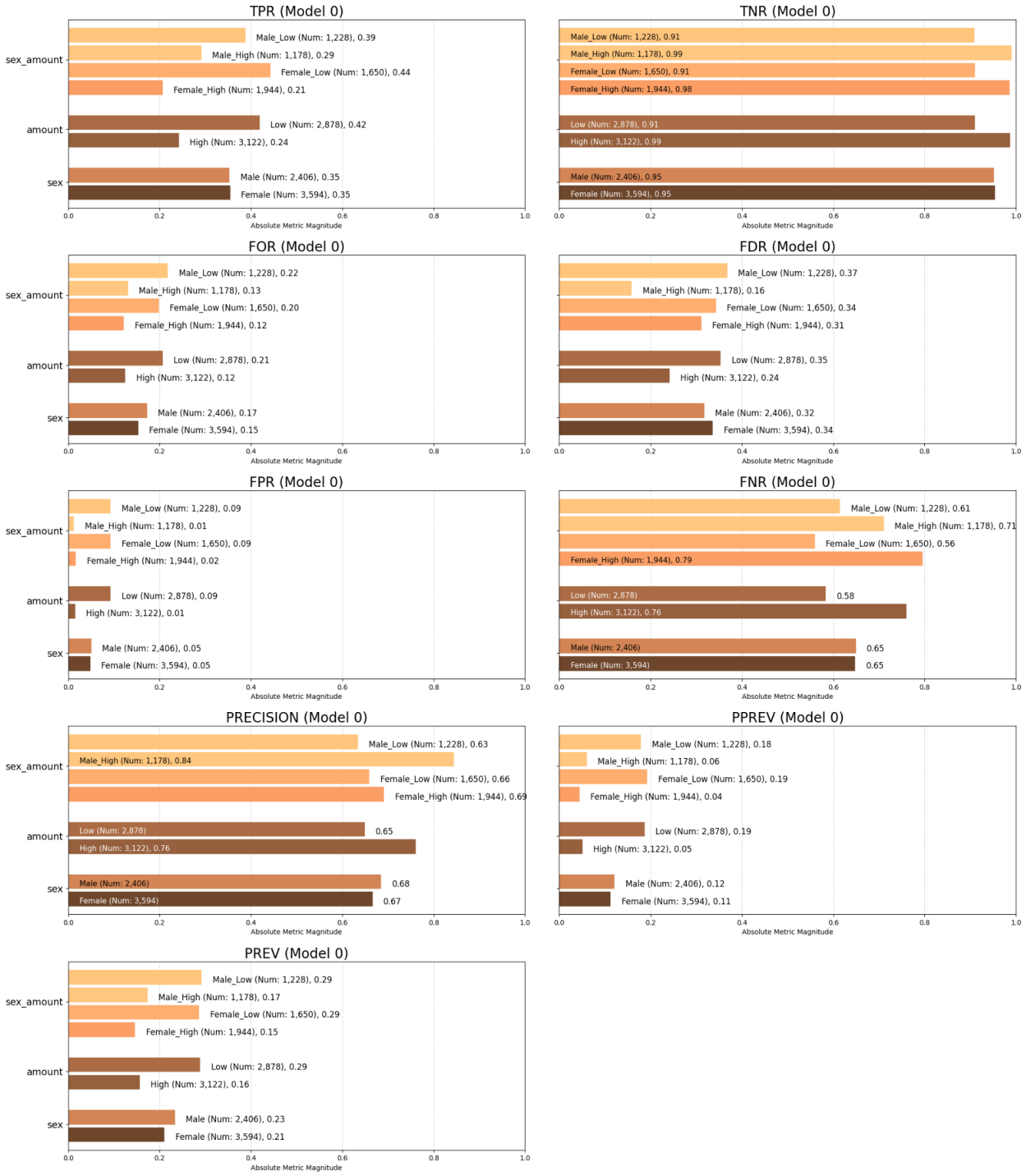
PREV is a useful accuracy metric for the Credit Card Default Prediction System because it measures the proportion of positive predictions that are correct out of all positive predictions made. It is important to consider PREV along with other accuracy metrics because a high PREV may indicate good prediction accuracy, but it does not provide information on the overall proportion of positive outcomes in the population.

The PREV graph shows the predictive accuracy of the Credit Card Default Prediction System across different subpopulations. The PREV values indicate the proportion of positive predictions that are correct. The analysis reveals that the system has higher predictive accuracy for the subpopulation of low-borrowing individuals (PREV = 0.29) than for high-borrowing individuals (PREV = 0.16). Similarly, the system performs better for the subpopulation of low-borrowing females (PREV = 0.29) than for high-borrowing females (PREV = 0.15). On the other hand, the system has similar predictive accuracy for males and females, with a slightly higher PREV for males (PREV = 0.23) than for females (PREV = 0.21).

Analyzing the PREV metrics for the different subpopulations, we can see that the system performs similarly for both male and female individuals, with a slightly higher PREV for males. However, when considering the amount of credit being requested, the system performs much better for individuals requesting a low amount compared to those requesting a high amount. Additionally, when considering both sex and amount, the system performs better for males requesting a low amount, and slightly better for females requesting a low amount compared to females requesting a high amount.

Overall, the Credit Card Default Prediction System appears to perform better for individuals requesting a low amount of credit, regardless of gender or the combination of gender and amount. It may be necessary to adjust the system to improve its predictive accuracy for individuals requesting a high amount of credit, particularly for females.

In this project, we will evaluate the fairness of the implementation system for bank and client separately. For banks, it is necessary to use FOR, FNR, and Recall to evaluate the fairness of the implementation system. This is because banks want to catch as many defaults as possible. Therefore, it’s important for them to minimize false omissions and false negatives. For clients, it is essential to use FDR, FPR, and Precision. This is because Credit card clients don’t want to be predicted as defaults falsely, if clients would actually pay back the money. Therefore, clients want the ADS to minimize false discovery rate and false positive rate.



b. Analyze the fairness

- **Banks (FOR, FNR, Recall)**

Banks want to catch as many defaults as possible. Therefore, it's important for them to minimize false omissions and false negatives, and maximize recall.

FOR: From the graphs we can see that the false omission rate for male is 1.13 times of that for females. The ADS is falsely omitting slightly more male defaults than female defaults. The disparity between amounts is more severe, as the ADS falsely omits 1.62 times the number of defaults for low credit amounts than that for high credit amounts. The intersection graph to the right does not show significant additional disparity between the intersections of the groups.

FNR: The false negative rate is equal between males and females. The false negative rate for low credit amounts is lower than that for high credit amounts, at 0.75 times of the latter. This means the ADS makes less false negative predictions for low credit amounts. The intersection graph again does not show significant additional disparity. The false negative rate for low credit amounts is roughly equal for both male and female, and it's the same for high credit amounts.

Recall: Recall performance is fair between sexes but not between credit amounts. The recall of this ADS is much higher for clients with low credit amounts than those with high credit amounts. This means that the ADS is not catching as many defaults for high credit amounts. This would make the bank lose more money than necessary, as some defaults with high credit amounts are not caught. The intersection graph shows some disparity between sexes for each group, but the disparity is not that significant.

- **Clients (FDR, FPR, Precision)**

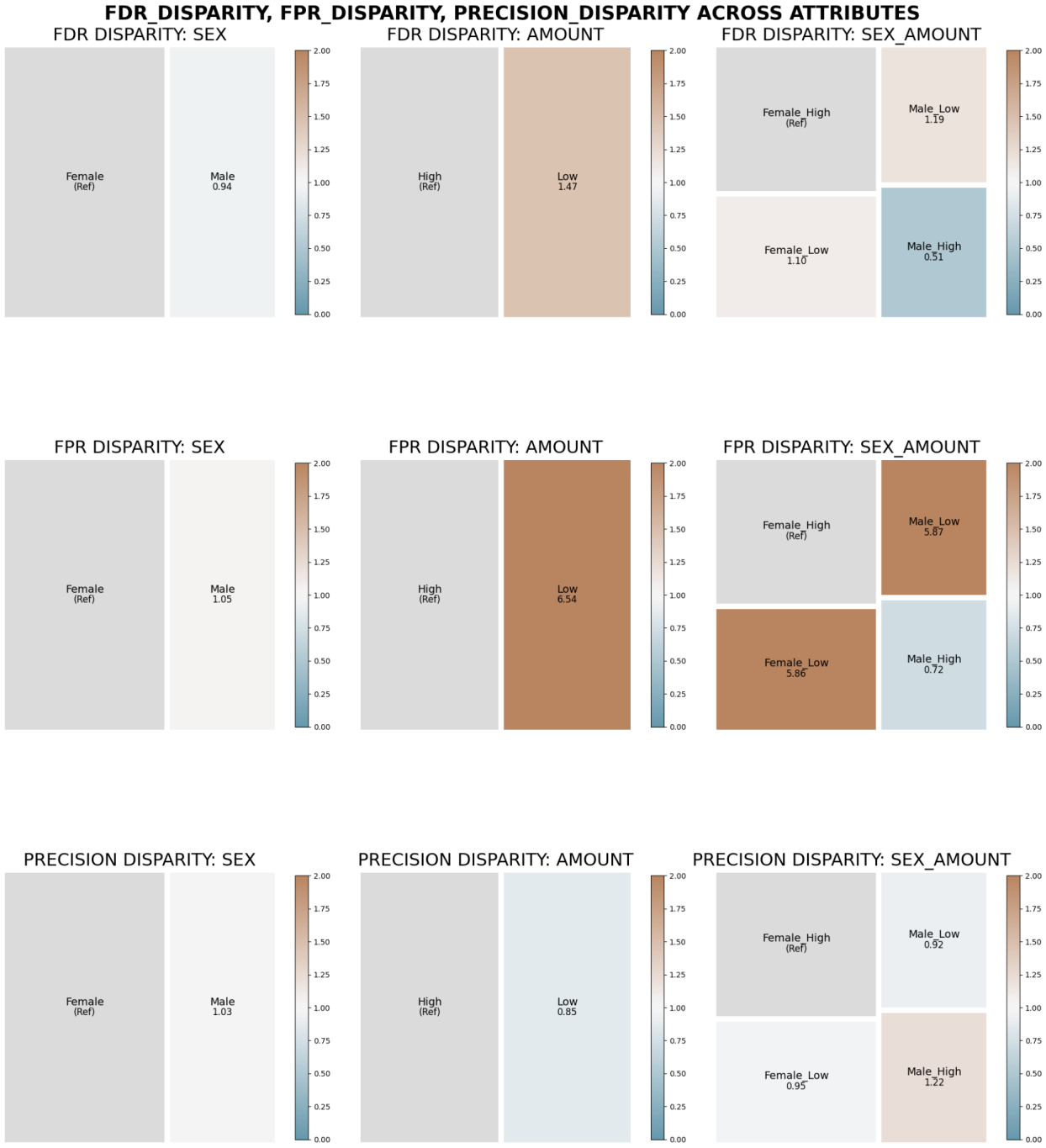
Credit card clients don't want to be predicted as defaults falsely, if clients would actually pay back the money. Therefore, clients want the ADS to minimize false discovery rate and false positive rate, and maximize precision.

FDR: The false discovery rate is almost equal between males and females. There's disparity between amounts, as the ADS falsely discovers 1.27 times the default for low credit amounts as that for high credit amounts. The intersection graph shows that there's more disparity

between the intersections of groups. We see that among clients with high credit amounts, the ADS is falsely discovering much less defaults for males than for females. This disparity would unfairly benefit males, as the ADS is making less mistakes for them. Unlike clients with high credit amounts, we don't see such disparity for clients with low credit amounts.

FPR: The false positive rate between sexes are again almost equal. However, the ADS shows significant disparity between credit amounts, as the false positive rate for low credit amounts is 5.65 times that for high credit amounts. This means the ADS is making much more false positive predictions for low credit amounts. This would unfairly hinder clients with low credit amounts, as more of them are falsely predicted to default. The intersection graph does not show significant additional disparity.

Precision: The ADS performs well with regards to precision. No apparent disparity is discovered for all groups and intersections of groups.



c. Develop additional methods to analyze ADS performance

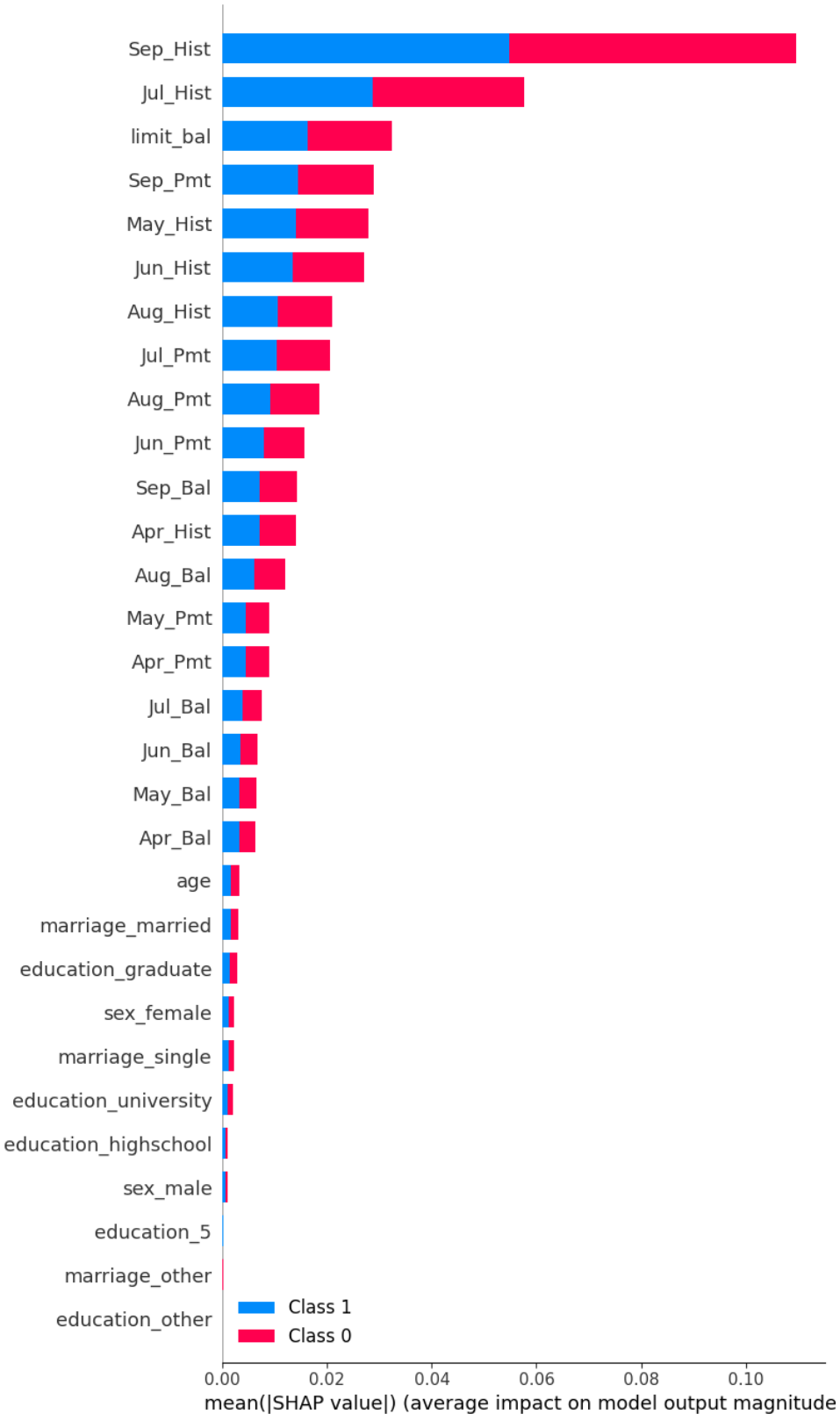
To analyze the performance of an Algorithmic Decision System (ADS), it is crucial to evaluate its stability, robustness, performance on important or difficult examples, and other relevant properties. To do this, we used the SHAP package in this project. By fitting a TreeExplainer to our model, we were able to obtain the SHAP values for our features. This will form the basis for the analysis below.

1. **Evaluate overall feature importance for the model**

Plotting the summary plot from the SHAP package, we can see the magnitude of importance for each feature in both directions of prediction. Overall, there's a very good balance on the importance of each feature in both directions, because the blue and red bars for each feature are roughly the same length. In other words, all features are equally important for both positive and negative predictions.

The most important feature for the model is “Sep_Hist”. This indicates the model is performing well and correctly using the feature that it should use, which is about the financial situation of the most recent month. Other important features are also about the financial situations, suggesting that the model is using legitimate features.

Our sensitive feature “limit_bal” is the fifth most important feature. This corresponds with our previous analysis that the model does exhibit some bias with regards to credit amount. The sensitive feature “sex” has very low importance, as the bars for both “sex_male” and “sex_female” are very short compared to other features. This shows that the model does not bias with regards to sex, which is an advantage of this model and confirms our previous analysis with Aequitas.


Sep_Hist
Jul_Hist
limit_bal
Sep_Pmt
May_Hist
Jun_Hist
Aug_Hist
Jul_Pmt
Aug_Pmt
Jun_Pmt
Sep_Bal
Apr_Hist
Aug_Bal
May_Pmt
Apr_Pmt
Jul_Bal
Jun_Bal
May_Bal
Apr_Bal
age
marriage_married
education_graduate
sex_female
marriage_single
education_university
education_highschool
sex_male
education_5
marriage_other
education_other
Class 1
Class 0
0.00
0.02
0.04
0.06
0.08
0.10
mean(|SHAP value|) (average impact on model output magnitude

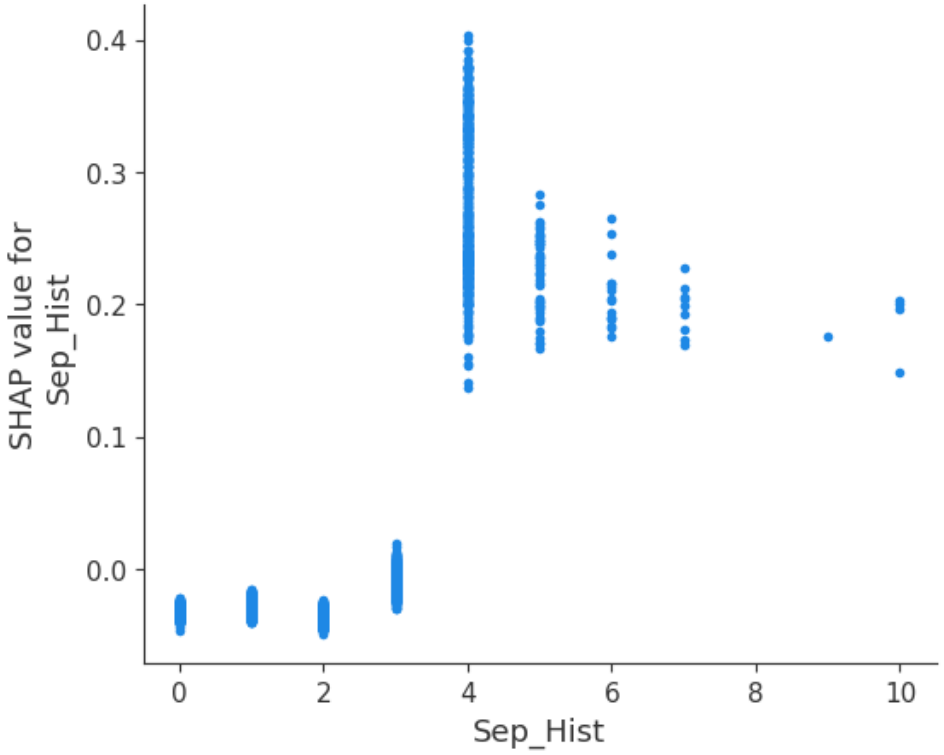
2. **Show how the target variable depends on various features**

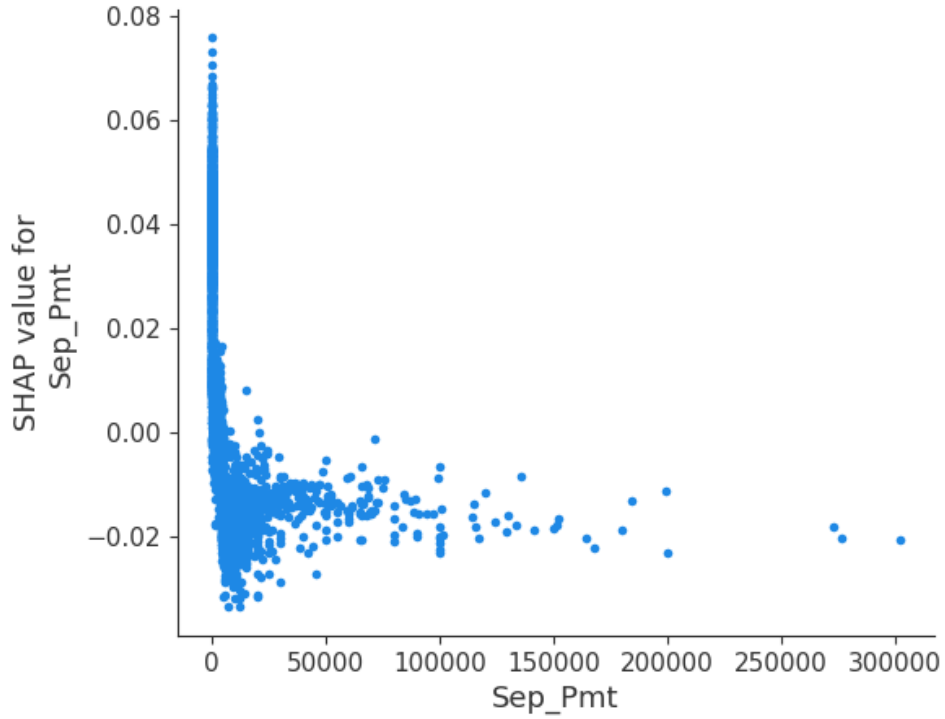
We chose three features to investigate in this section: “Sep_Hist”, “Sep_Pmt”, and “limit_bal”. The relationship between the values of those features and their effect on the predictions are plotted.

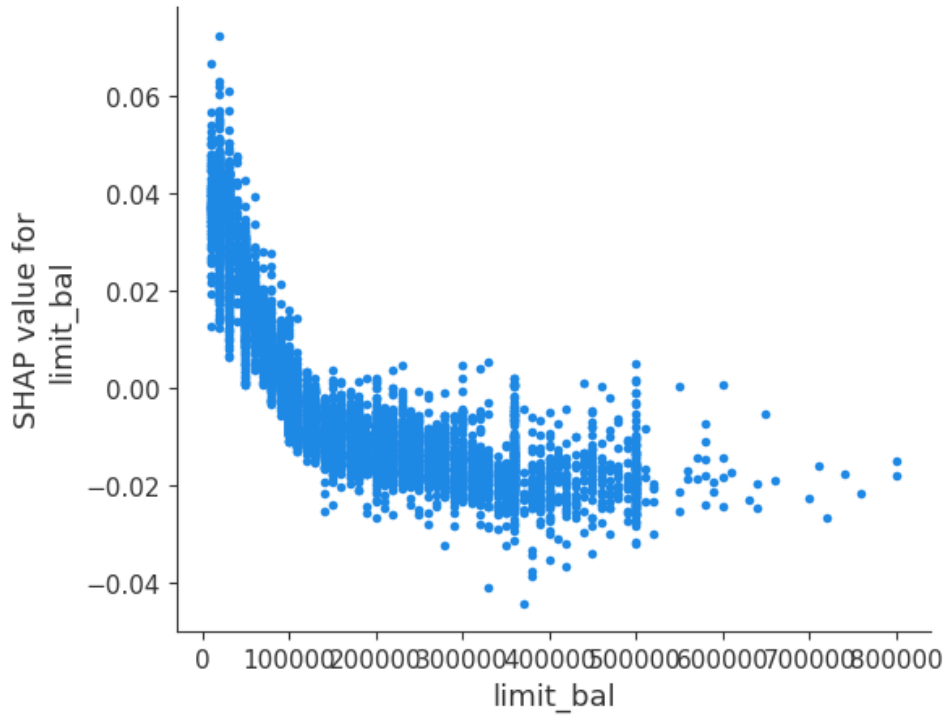
For “Sep_Hist”, we can see that values larger than 3 generally have a higher correlation with predicting that the client will default. This makes sense because a client that has payment delays more than 3 months is more likely to not payback at all and default.

For “Sep_Pmt”, We see that positive values mostly correlate with predicting that the client will not default. If the value for this feature is 0, which a lot of the data points are, the effect of this feature is mostly predicting that the client will default. This also makes sense because paying nothing for this month is not a good sign for the client’s financial situation.

For “limit_bal”, We see that there’s a negative correlation between the credit amount and default. This means the model predicts higher default probability if the credit amount is low, and lower default probability if the credit amount is high. This result verifies our analysis from Aequitas, as the model is using this sensitive feature to make predictions.







3. Explain individual predictions, both for correctly and incorrectly predicted clients

In this section, we first constructed two lists that contain the indices of clients in the test set that are misclassified. One of them is for false positives and the other is for false negatives.

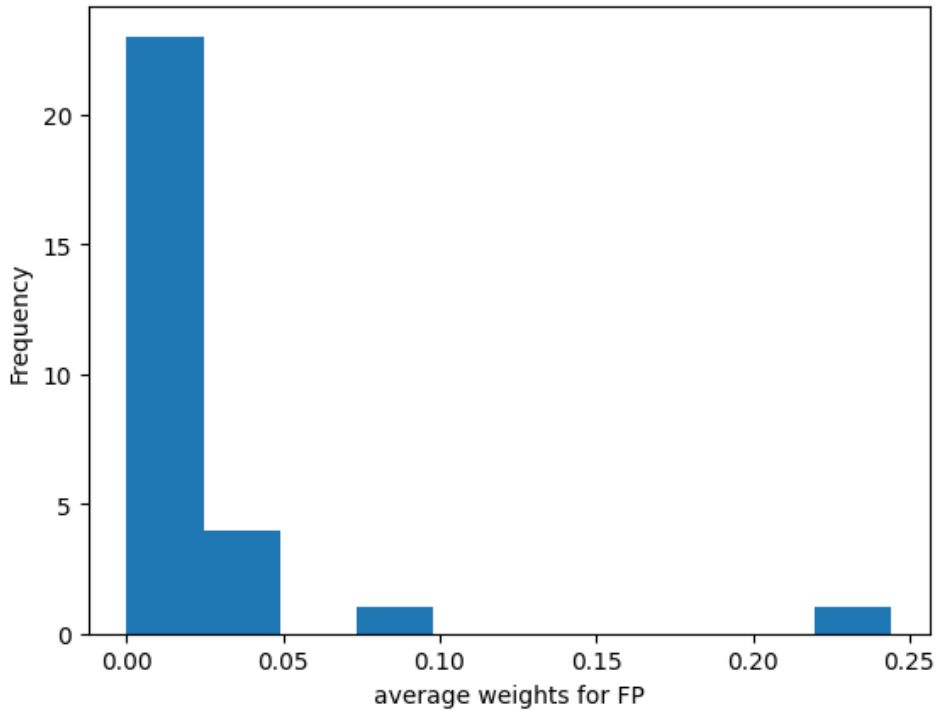
We choose two from each and see the individual feature explanations for those misclassifications.

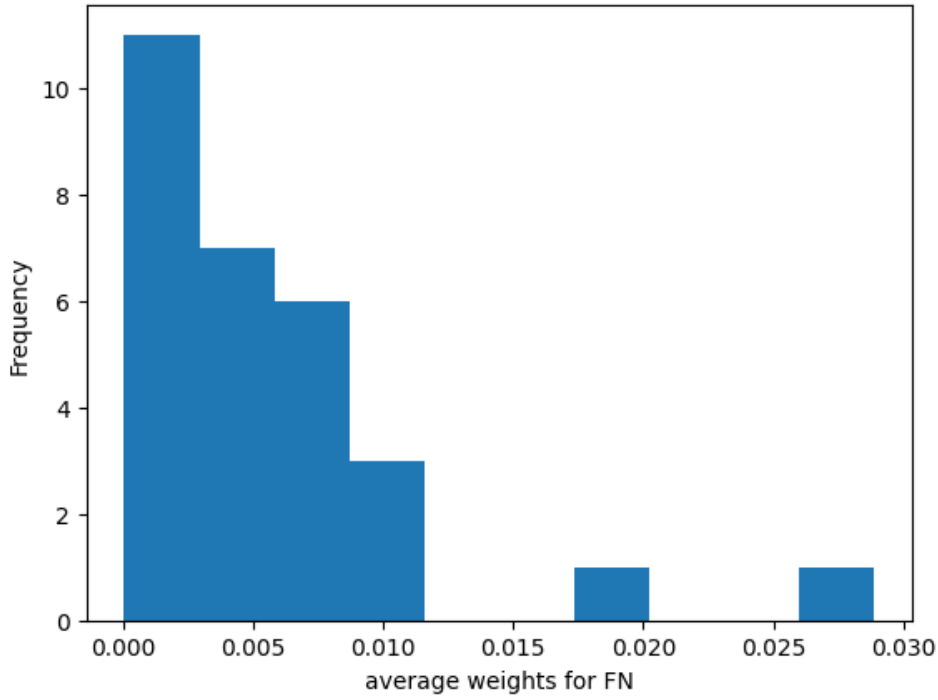
With the force plot from SHAP, we see that the features contributing to those two false positives are all payment delays from recent months. The values for those delays are quite high, so the ADS makes the prediction that the client will default. We see that for those two false positives, the ADS is doing the correct thing, and there's not much room for improvement, since all the relevant features are hinting that the client will default.

For false negatives, the most relevant features include payment delays and payment amounts. The feature values for both false negatives indicate low delay time and positive payment amounts. Similar to false positives, the ADS is working correctly as features indicate that the client wouldn't default, and there's not much room for improvements.

4. **Show which features contribute the most to misclassification**

For each misclassification, the features that contribute to the misclassification in the relevant direction are recorded. Dictionaries are used to accumulate their total counts of contribution to misclassification, as well as their total weights in those misclassifications. The average weight for each misclassification of the features are then calculated by dividing the two dictionaries. Plotting the average weight for both positive and negative directions of misclassification, we see that the distributions for average weights are both skewed to the right. This means a lot of features have relatively low average weights contributing to misclassification, and a few of the features have high average weights.





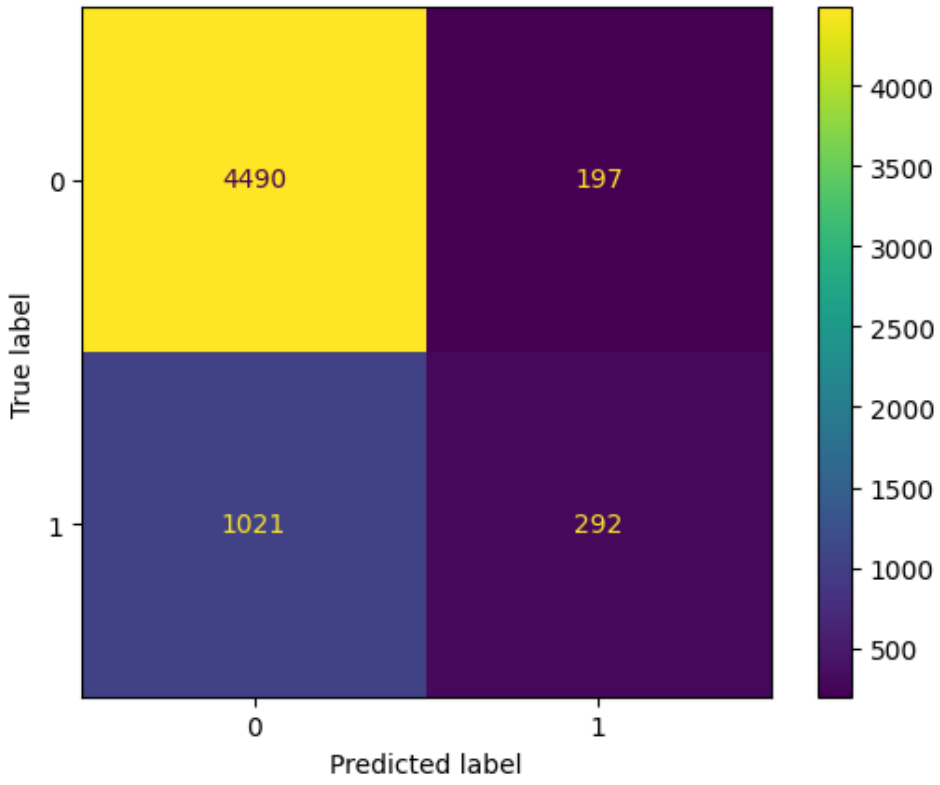
5. **Propose alternative models with modified feature sets that have potentially better performance**

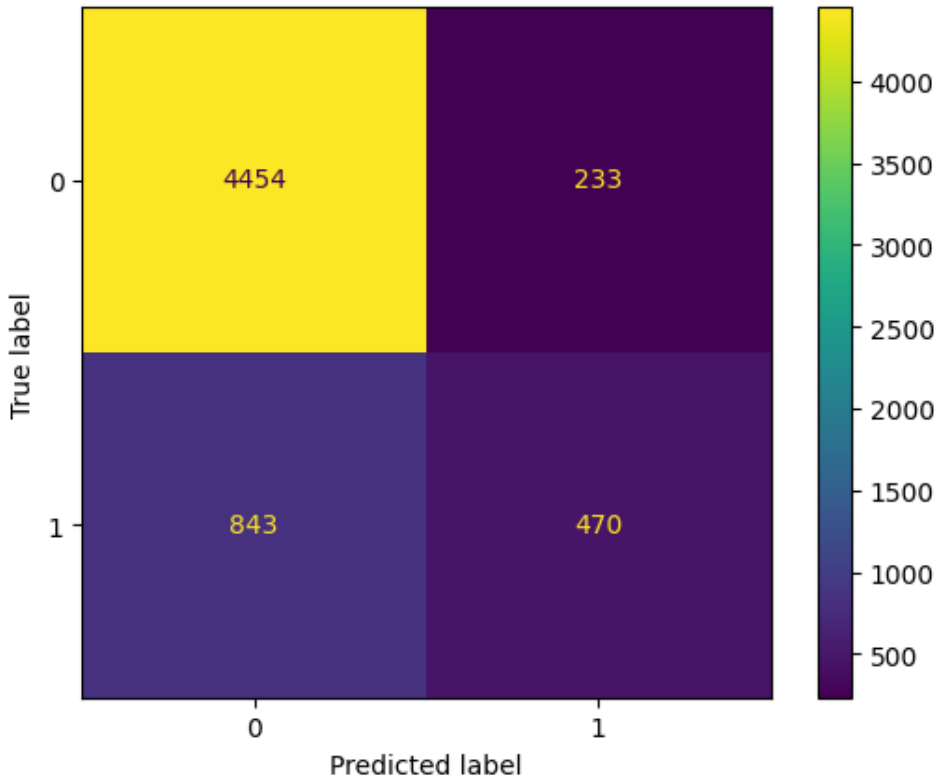
To potentially improve the original models, two feature selection strategies are considered in this project.

The first strategy is to remove the features that have the most average weight from the previous section. However, the two features that contribute most to misclassification are also the two most important features for the model. As expected, removing those two features made the accuracy of the model notably worse.

The second strategy is to remove the least predictive features for the model as indicated by the summary plot we got previously. The feature importance values were dynamically generated and the 5 least predictive features were selected for removal. The resulting model performs very slightly worse than the original model, indicating that this strategy has almost no effect on model performance.

Overall, those two alternative models weren't able to improve upon the original model, indicating that the original model is well-tuned and trained, using all the features to its benefit. Removing any feature would most likely not improve the model's performance.


4490
197
1021
292
0
1
True label
0
1
Predicted label
4000
3500
3000
2500
2000
1500
1000
500


4454
233
843
470
0
1
True label
0
1
Predicted label
4000
3500
3000
2500
2000
1500
1000
500

V. Summary

a. Do you believe that the data was appropriate for this ADS?

The dataset contains financial records for clients, including past payment amount, bill statement, and repayment status. These data are suitable for predicting credit card defaults. However, the dataset also contains 5 potentially sensitive features: credit amount, sex, education, marriage, and age. It's unclear whether using such features to predict defaults is fair or not, as the model might implicitly catch onto those features and discriminate against certain groups of clients.

b. Do you believe the implementation is robust, accurate, and fair? Discuss your choice of accuracy and fairness measures, and explain which stakeholders may find these measures appropriate.

The model is fairly robust. From the SHAP analysis, we can see that the model is correctly using features that describes the financial situation of the clients rather than some other irrelevant feature, and the most important feature for the model is "Sep_Hist", which describes the situation of credit card payment of the most recent month. The least important features for this model are the indicator variables that come from the sensitive features. The two models that we proposed alternatively both perform worse than the original model, which suggests that the original model is using all features to its benefit, and no features contribute significantly to misclassifications.

The model is not accurate in general. From general performance graphs we see that the model is overly optimistic across all groups, as the amount of predicted defaults is always lower than the actual amount. This would severely impact the profitability of banks, as they cannot use this ADS to effectively catch all clients that would default.

The fairness of the ADS with regards to sex is satisfactory. However, the ADS imposes significant disparity with regards to credit amount. This bias is apparent if we look at the ADS's false positive rate and recall performance. The false positive rate for clients with low credit amounts is much higher than that for clients with high credit amounts. This would unfairly put those with low credit amounts at a disadvantage, as they are falsely predicted to default when they actually won't. Recall for low credit amounts is much higher for clients with low credit amounts, and this implies the ADS is not catching as many defaults for clients with high credit amounts. This would make the banks lose more money than if the ADS can achieve the same recall for high credit amounts.

c. Would you be comfortable deploying this ADS in the public sector, or in the industry? Why so or why not?

This ADS should not be deployed in the public sector or the industry. The accuracy of the model is lacking, and it's unfair with regards to credit amounts. A good model should contain many more features about a client's financial situation such as income and savings, as well as the client's background information such as place of residency and overall economic status.

However, potentially sensitive features should not be included, as models can catch on to those features in the training process, and potentially make biased predictions using those sensitive features.

d. What improvements do you recommend to the data collection, processing, or analysis methodology?

The ADS has potential for improvement. The ADS we chose implemented a Random Forest to predict defaults. However, some columns in the feature set form a time series, as they are financial records across continuous months. It would almost be certain that data in the time series are not independent of each other, and a recurrent neural network has the potential to achieve better accuracy and fairness. It's important to note, however, that the goal of this ADS is to predict whether the client will default, not the future value of our features. Therefore, some translation needs to be done in order for the recurrent neural network to output a single value as our target variable.

Work Cited

Dominguez, Marcos. “Predicting Credit Card Defaults with Machine Learning.” Medium, The Startup, 1 Mar. 2021, medium.com/swlh/predicting-credit-card-defaults-with-machine-learning-fcc8da2fdafb.

Kuo, Chris. “Explain Any Models with the SHAP Values — Use the KernelExplainer.” Medium, The Startup, 6 Nov. 2019, https://towardsdatascience.com/explain-any-models-with-the-shap-values-use-the-kernelexplainer-79de9464897a

Yeh , I-Cheng. “Default of Credit Card Clients Data Set.” UCI Machine Learning Repository: Default of Credit Card Clients Data Set, UCI Machine Learning Repository, 2009, archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients#.