Universidad Pontificia de Comillas
ICAI

FINAL PROYECT

Machine Learning

Lydia Ruiz Martínez

HELOC INSIGHTS: A DATA STUDY

Academic year 2023-2024

Abstract

This document presents the final project for the Machine Learning course, completed by Lydia Ruiz Martínez. This report contains a justified analysis of the results obtained and demonstrates the work throughout the statement.

The problem resolution has been implemented in Python. This project consists of the following development elements:

- 6 Modules: `config.py`, `classification_tools.py`, `cross_validation.py`, `logistic_regressor.py`, `model_tools.py`, and `utils.py`.
- 1 file: `Lydia_Ruiz_Martinez_Final_Project.ipynb`.
- 2 text files: `requirements.txt`, which specifies the dependencies (non-standard Python libraries) of the project and `README.md` with the steps to reproduce the results of the report.

Contents

1 Data Preparation and Exploratory Analysis

This section encapsulates the exploratory data analysis and preprocessing steps undertaken for the FICO dataset, laying the groundwork for subsequent machine learning modeling.

1.1 Overview

Initial exploratory data analysis revealed a balanced distribution of the target variable, 'RiskPerformance'. Through histograms and KDE plots, we visualized variable distributions. Pairwise scatter plots highlighted potential relationships, while a heatmap of the correlation matrix indicated various degrees of linear associations among features.

1.2 Preprocessing Techniques

- Missing Values: Special characters representing missing data were converted to NaNs (-8 and -7). Following this, some rows with missing target values or non-investigated records were excluded (-9). The KNN imputation strategy estimated remaining missing values using five nearest neighbors.
- Data Split: The dataset was bifurcated into training (80%) and testing (20%) sets, with the 'RiskPerformance' variable encoded for modeling. The training set consisted of 4156 samples, while the test set comprised 1039 samples.
- Feature Scaling: Standardization normalized feature scales to zero mean and unit variance, a necessary step for many learning algorithms.

We did not eliminate any variable nor did we account for outliers, as the box plots suggest that the outliers could be highly significant and explain the natural variability of the population.

1.3 Visualizations

Included below are histograms (Figures 1 and 2), pairwise plots with KDEs (Figure 3), heatmap of the correlation matrix (Figure 4), box plots for outlier identification (Figure 5), and Q-Q plots for distribution analysis (Figure 6). Each figure substantiates the discussed statistical observations and is positioned to complement the narrative of the data analysis.

1.4 Implications

Preprocessed data, devoid of missing values and anomalies, and augmented by EDA insights, optimizes the training process. The feature scaling ensures fairness in the algorithm's interpretive ability across all input variables, setting the stage for accurate model predictions.

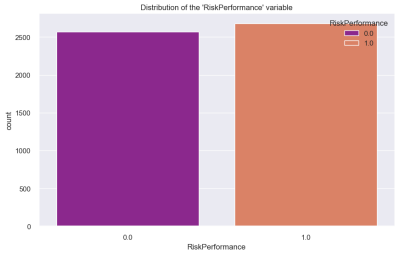


Figure 1: Distribution of the 'RiskPerformance' variable.

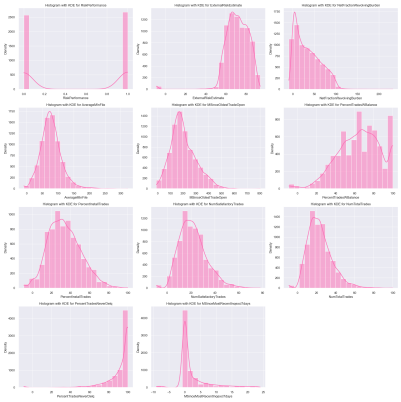


Figure 2: Histograms and KDEs for the input variables.

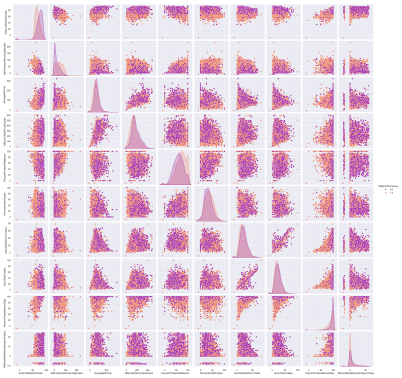


Figure 3: Pairwise scatter plots with KDEs.

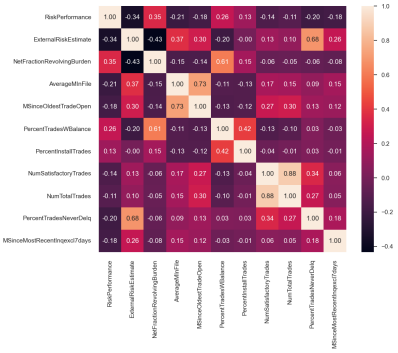


Figure 4: Heatmap of the correlation matrix.

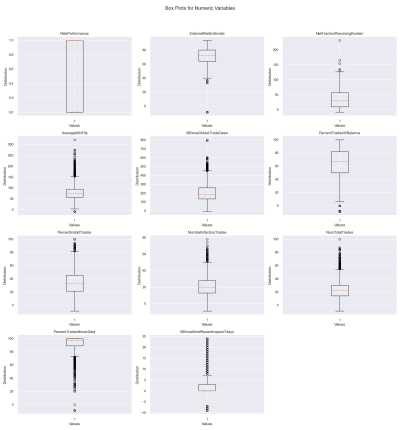


Figure 5: Box plots for numeric variables.

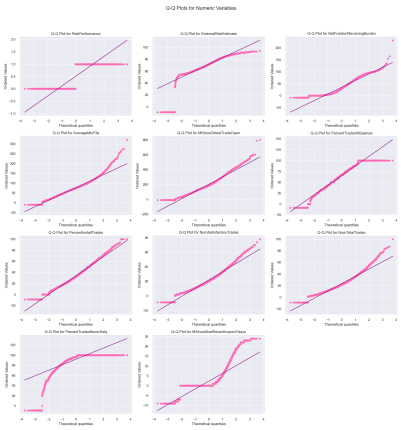


Figure 6: Q-Q plots for numeric variables.

2 Classification

2.1 Fitting process of classification models and comparative analysis of the fitted models

2.1.1 Logistic regression

Incorporating L1 regularization into our Logistic Regression model, we tuned the hyperparameter C to achieve optimal balance between complexity and accuracy. The model demonstrated robust performance with key metrics: accuracy at 74.87%, precision at 75.84%, and an F1 score of 75.63%. The evolution of model weights and ROC curve analysis are visually summarized in the accompanying figures.

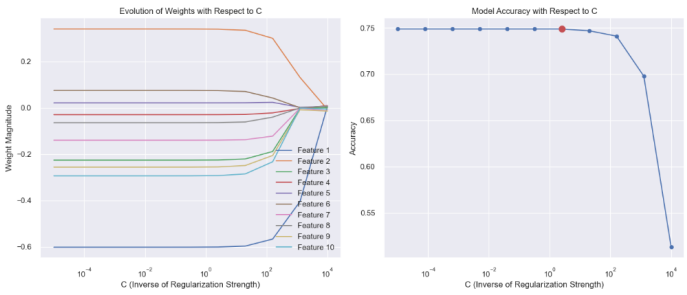


Figure 7: Evolution of weights with respect to C

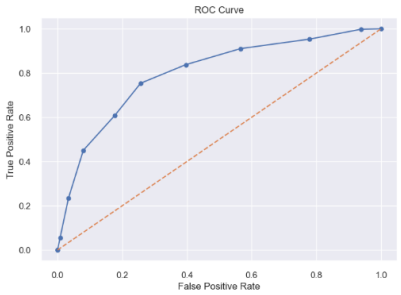


Figure 8: ROC Curve

2.1.2 KNN

Implementing K-Nearest Neighbors (KNN), we tuned the number of neighbors k to optimize the model. Cross-validation identified k=83 as the optimal number, with a mean score of 0.73, suggesting a balanced model that generalizes well to unseen data. Training and test accuracy rates were 73.64% and 74.49% respectively, indicating the model's reliability and a minimal risk of overfitting.

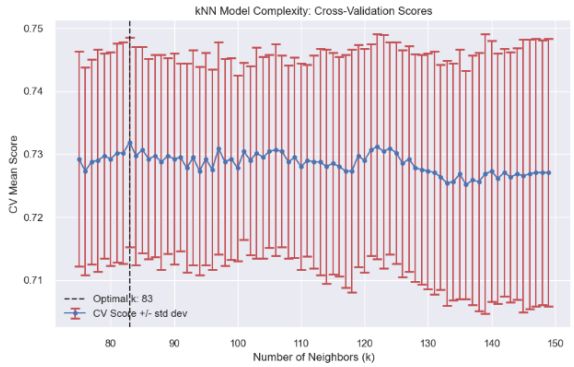


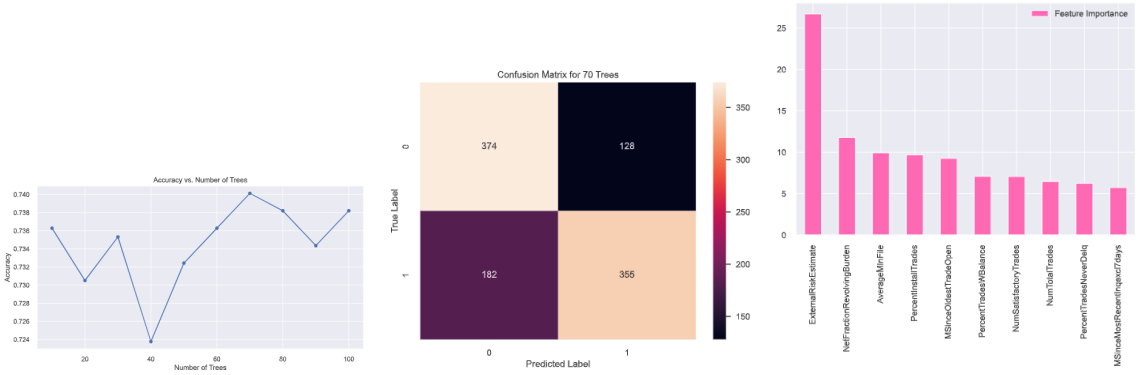
Figure 9: KNN model complexity, showing cross-validation scores with optimal k and corresponding accuracy rates. Error bars represent score variability across different folds.

2.1.3 Linear and Radial SVM

Moving to Support Vector Machines (SVM), we utilized both linear and radial basis function (RBF) kernels to classify HELOC repayment data. The linear SVM model was fine-tuned with a range of C values for regularization strength, resulting in an optimal C of 100. This model yielded an accuracy of approximately 73.53% on the test set. Subsequently, we employed an RBF kernel with a grid search over both C and the gamma parameter to account for non-linear relationships. The search converged on C=100 and $\gamma = 0.01$ as the best parameters, achieving a slightly higher test accuracy of approximately 74.78%. These models were calibrated using cross-validation scores, ensuring reliable probability estimates.

2.1.4 Decision Trees

First we started with bagging. As the number of trees increased, we observed fluctuations in accuracy, which plateaued around 70 trees, indicating a balance between variance and bias. Accuracy metrics for bagging with optimal trees showed a test accuracy of 73.82%, precision at 74.48%, and an F1 score of 73.83%. These metrics provide insights into the model's performance and the significance of different predictors.The bagging ensemble effectively reduced overfitting, enhancing predictive performance compared to a single estimator, as evidenced by the feature importance which highlights the most influential factors in predicting positive repayment behaviors.



(a) Accuracy vs. Trees (b) Confusion Matrix (c) Feature Importance

Figure 10: Comparative analysis of the bagging ensemble model

For the Random Forest classifier, we observed how the accuracy evolved with the number of trees. With 100 trees, the model reached an accuracy of about 74.39%, balancing variance and bias effectively. The confusion matrix for the 100-tree model shows a satisfactory distribution between true positives and true negatives, indicating a robust performance. Additionally, the feature importance graph reveals which predictors contribute most to the model's decisions, with 'ExternalRiskEstimate' being the most significant. With AdaBoost, an ensemble boosting method, we examined its influence on prediction accuracy for HELOC repayment. The model

enhanced decision boundaries by iteratively correcting mistakes of weak classifiers. This technique not only improved accuracy but also refined the precision, recall, and F1 score to approximately 75.45%. These results indicate AdaBoost's strength in combining multiple weak learners to form a more accurate and robust model. The accompanying visuals, including accuracy trends, confusion matrix, and feature importance, offer a detailed perspective on the model's performance and the relative influence of each predictor in the decision-making process.

2.1.5 MLP

Employing a Multi-layer Perceptron (MLP) Classifier, we achieved training and test accuracies of 73.79% and 74.30%, respectively. The model's loss decreased significantly within the first few epochs, indicating rapid learning, and stabilized quickly, evidencing a good fit. The MLP's performance affirms its capability to discern complex patterns in the HELOC repayment data.

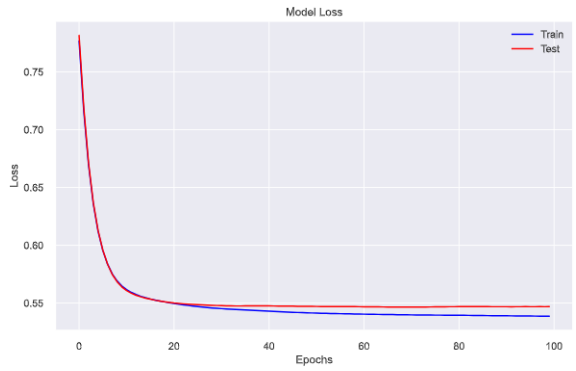


Figure 11: Loss reduction over epochs for the MLP Classifier.

2.2 Creativity and Innovation: Stacking and Voting Ensemble

In the final phase of our machine learning exploration, we introduced ensemble methods to enhance predictive accuracy. Ensemble techniques combine multiple models to capitalize on their individual strengths while mitigating their weaknesses. Specifically, we employed a StackingClassifier that integrates predictions from Bagging, Random Forest, and AdaBoost classifiers using Logistic Regression as the meta-model. This approach leverages the diverse perspectives of different models to reach a consensus prediction. Additionally, we utilized a VotingClassifier with a 'hard' voting scheme, which selects the most frequent prediction from the base classifiers to make a final decision. Our results indicated that the StackingClassifier achieved an accuracy of approximately 73.83% on the test set, slightly outperforming the VotingClassifier's accuracy of about 72.95%.

3 Unsupervised Learning

3.1 PCA fitting process and analysis

Principal Component Analysis (PCA) is a technique used to emphasize variation and bring out strong patterns in a dataset. We started by scaling the data and then applied PCA to reduce dimensionality while retaining the maximum variance.

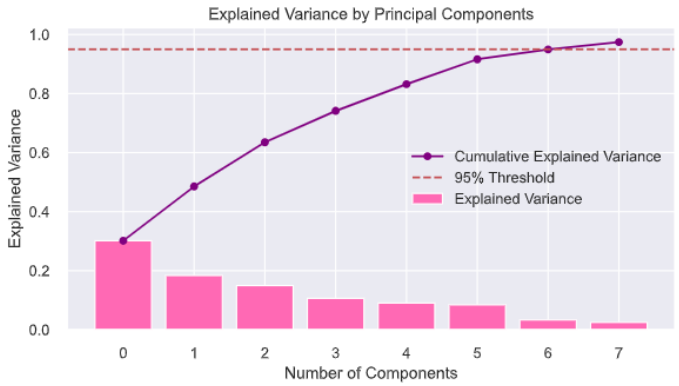


Figure 12: Explained Variance by Principal Components showing both cumulative explained variance and the percentage of variance explained by each principal component. The 95% threshold suggests the number of components to retain.

3.2 Identification and Fitting Process of Clustering Algorithms

To uncover the intrinsic groupings within the HELOC dataset, we explored and fitted multiple clustering algorithms. Each algorithm brings a unique approach to the task, offering different insights into the data's structure, each algorithm uses two clusters.

3.2.1 KMeans Clustering

KMeans clustering partitions the data into K distinct, non-overlapping subsets. We employed the Elbow Method to determine the optimal cluster count, as demonstrated in the distortion score plot below.

3.2.2 Hierarchical Clustering

In contrast to KMeans, Hierarchical clustering does not require the number of clusters to be specified a priori and provides a tree-based representation of the observations, called a dendrogram.

3.2.3 Gaussian Mixture Models (GMM)

GMMs offer a probabilistic approach to clustering, assuming that the data is generated from a mixture of several Gaussian distributions with unknown parameters. GMMs are flexible and can accommodate clusters of different sizes and shapes.

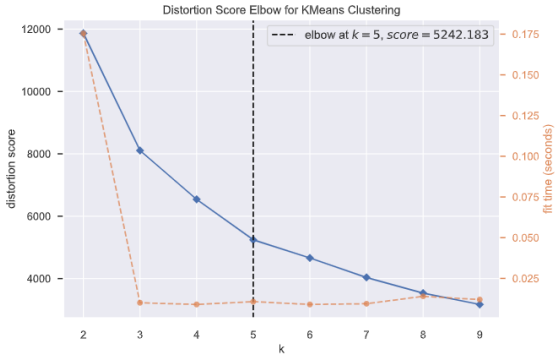


Figure 13: The Elbow Method for determining the optimal number of clusters in KMeans. A clear elbow is visible at $k = 5$, suggesting it as the optimal cluster count; however we used 2 clusters.

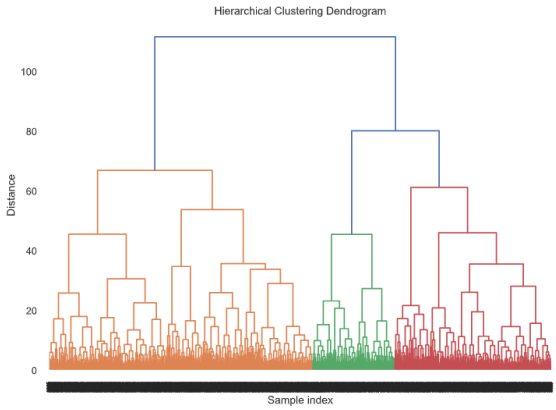


Figure 14: Dendrogram from Hierarchical Clustering.

3.3 Creativity and Innovation: t-SNE for Visualization

The use of t-SNE (t-distributed Stochastic Neighbor Embedding) stands as an innovative technique in our analysis. It provides a powerful method for visualizing the high-dimensional data in two dimensions while preserving the local structure of the data.

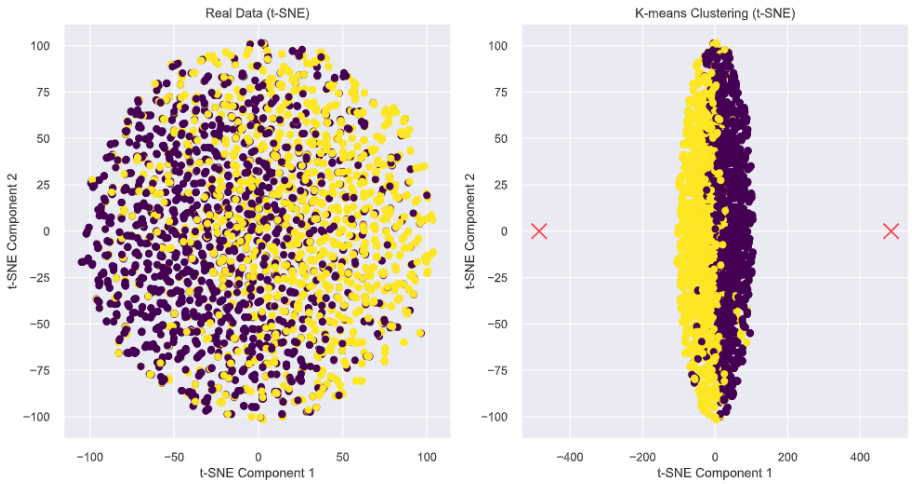


Figure 15: t-SNE visualization highlighting the clusters in the data. The use of t-SNE allows for a more nuanced understanding of the cluster distribution and separation.

4 Conclusions

In our project, variable importance was initially hypothesized through distribution and correlation analyses. The logistic regression coefficients highlighted variable significance, and the presence of these variables in the correlation matrix served as preliminary validation.

The logistic regression analysis revealed that certain features retained significant weights despite increasing regularization, suggesting their importance in the model. However, the challenge of multicollinearity, particularly between two highly correlated variables (NumSatisfactoryTrades and NumTotalTrades), could obscure the interpretation of these coefficients. ExternalRisk Estimate, AverageMinFile, and NetFractionRevolvingBurden emerged as highly significant features, whereas MSinceMostRecentInqexcl7days showed negligible importance.

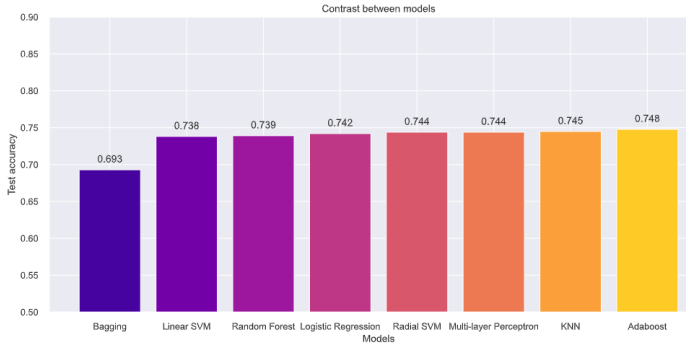


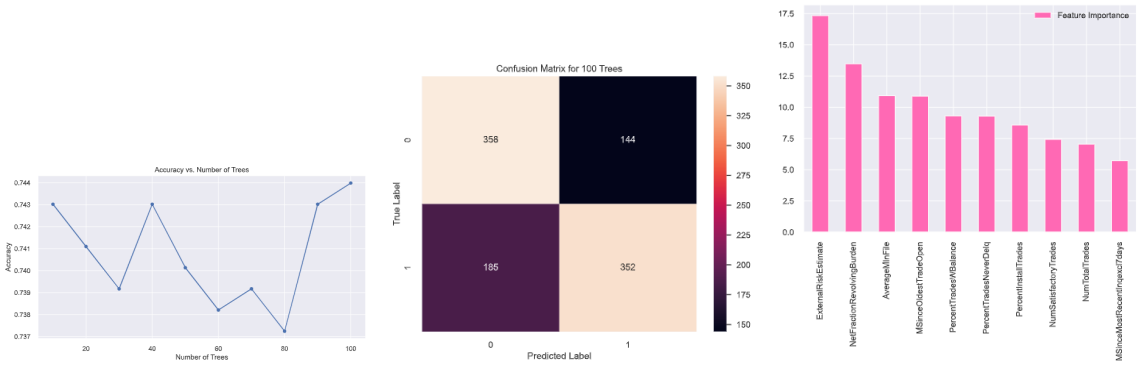
Figure 16: Comparison between the different trained models

The ensemble techniques, especially Random Forest, stood out in our analysis. Not only did it demonstrate robust accuracy, but it also maintained interpretability, which is a valuable asset in financial contexts. It balanced the clarity of decision trees with the precision of more complex models, outperforming other classifiers including SVC, which was initially a strong candidate but later overshadowed by Adaboost's superior performance. AdaBoost further enriched our modeling by combining multiple weak learners and amplifying their collective accuracy. This boosting approach fine-tuned our predictions, as reflected by the F1-score and other metrics. Despite the allure of clustering for classification, it's clear that supervised learning algorithms like Random Forest and AdaBoost, which utilize labeled data for training, ultimately provide more reliable and actionable insights for risk assessment in financial services. The best classification algorith was therefore Adaboost.

Clustering methods were not as effective as the classification algorithms used in Part 2 of the project. This is because classification algorithms are specifically tuned to minimize prediction errors based on known labels. For instance, logistic regression calculates the probability of an instance belonging to each class, a task distinct from K-Means clustering, which focuses solely on grouping similar instances without reference to labels.

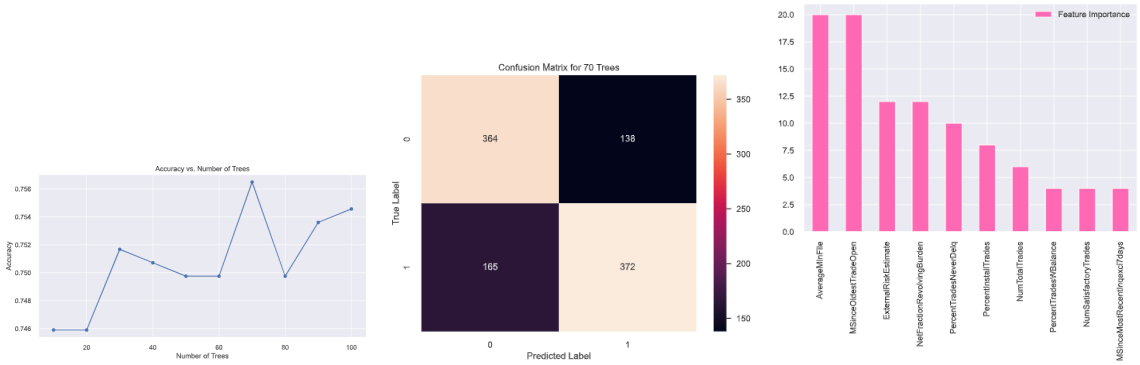
A Appendix

A.1 Decision Trees



(a) Accuracy vs. Trees (b) Confusion Matrix (c) Feature Importance

Figure 17: Comparative analysis of the random forest ensemble model



(a) Accuracy vs. Trees (b) Confusion Matrix (c) Feature Importance

Figure 18: Comparative analysis of the adaboost ensemble model

A.2 Clustering Visualizations

Additional visualizations of clustering algorithms applied to the PCA-transformed HELOC dataset are provided here for reference.

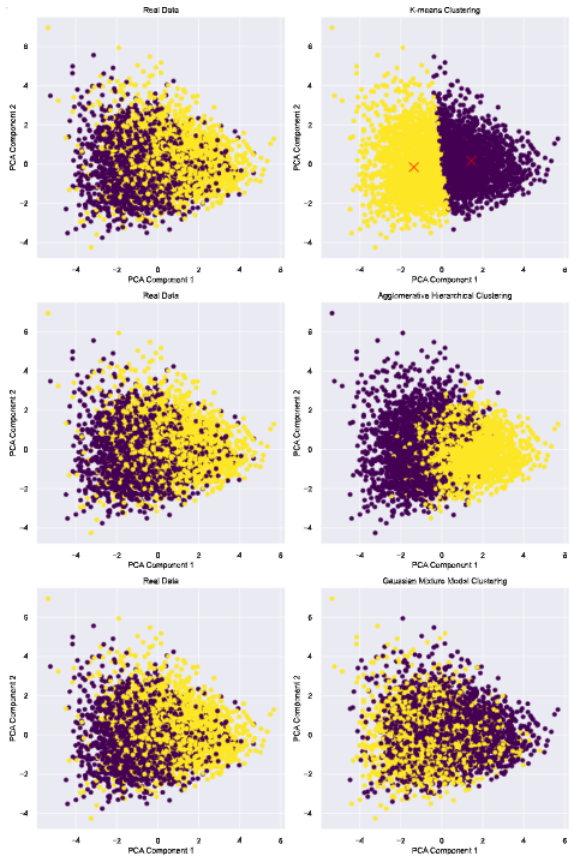


Figure 19: Comparative visualization of real data and clustering results using PCA components. From top to bottom: Real data, KMeans Clustering, Agglomerative Hierarchical Clustering, and Gaussian Mixture Model Based Clustering.