Iteration 4 Report

AI-Driven Financial Risk Management System

1. Dataset Description

Dataset: We are using the *Default of Credit Card Clients Dataset* from Kaggle.

Link: https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset

Description: This dataset contains financial and demographic data for 30,000 clients, including variables such as credit limit, payment history, bill amounts, and default status. It provides a comprehensive view of credit risk behavior, enabling predictive modeling for default likelihood. The data is structured in tabular format with 25 input features and one binary target variable (`default.payment.next.month`).

Relevance and Suitability: The dataset is well-suited to our project - *AI-Driven Financial Risk Management System* - because it directly supports the prediction of financial default risk, which is central to identifying potential credit issues. Its large sample size and diversity of variables make it ideal for building robust machine learning models for classification.

2. Tools and Methodologies

Tools and Libraries:

- Python 3.10 for implementation
- Pandas, NumPy for data cleaning and preprocessing
- Matplotlib, Seaborn for visualization and EDA
- Scikit-learn for model training, evaluation, and optimization
- GitHub for version control and collaboration
- Jupyter Notebook / VS Code as development environments

Methodologies: We follow a structured machine learning pipeline:

1. **Data Cleaning** : Handling missing values, encoding categorical variables, and feature scaling.
2. **Exploratory Data Analysis (EDA)** : Identifying correlations, outliers, and variable importance.
3. **Model Development** : Building Logistic Regression and Random Forest models.
4. **Model Optimization** : Applying `GridSearchCV` and `RandomizedSearchCV` for hyperparameter tuning.
5. **Evaluation** : Using metrics like accuracy, precision, recall, F1-score, and ROC-AUC.

These tools were chosen for their open-source reliability and strong community support, ensuring scalability and reproducibility across all stages.

3. Preliminary Timeline

Week	Milestone	Deliverables
Week 1	Data Cleaning and Preprocessing	Final cleaned dataset ready for modeling
Week 1	Exploratory Data Analysis (EDA)	Visualizations, summary statistics, correlation matrix
Week 2	Model Development	Train all models (Logistic Regression, Random Forest, Gradient Boosting, XGBoost, etc.)
Week 2	Model Evaluation and Optimization	Tuned models with validation metrics (GridSearchCV, RandomizedSearchCV)
Week 3	Integration and Testing	End-to-end pipeline completed and validated
Week 4	Final Documentation and Submission	Final report, GitHub sync, Overleaf PDF, and presentation ready

4. Team Member Contributions

Ishwarya Rajkumar: Data Preparation and Exploration (EDA) Cleaned and preprocessed data, handled missing values, encoded categorical variables, and conducted visual exploratory analysis (histograms, scatter plots, heatmaps).

Afrah Fathima: Model Building and Evaluation Developed and evaluated Logistic Regression and Random Forest baseline models, analyzed metrics such as accuracy, precision, recall, and ROC-AUC, and performed cross-validation.

Lakshmi Mahadevan: Hyperparameter Tuning and Optimization Executed model optimization using `GridSearchCV` and `RandomizedSearchCV`, fine-tuned hyperparameters, monitored for overfitting, and integrated early stopping and regularization.

All Members: Integration and Final Testing Collaboratively built the final pipeline, ensured code consistency across modules, finalized documentation, and prepared Iteration 3 deliverables.

5. Progress and Next Steps

Progress to Date:

- Completed data cleaning and preprocessing.
- Performed EDA and baseline model training.
- Achieved initial evaluation results with satisfactory accuracy and ROC-AUC.
- Began hyperparameter tuning and model optimization.

Next Steps:

- Finalize hyperparameter tuning for both models.
- Conduct full validation and testing on unseen data.
- Complete visualization dashboard for final report.
- Update Overleaf documentation and push final scripts to GitHub.

Project Links: GitHub Repository: https://github.com/iishwarya1415/DS5110-Project