**Credit Risk Analysis – Final Project Report**

**1. Objective**

The objective of this project is to design and implement a machine learning-based system that can evaluate the creditworthiness of individuals based on their financial history and behavioral patterns. This system aims to assist financial institutions in identifying **high-risk borrowers**—those who are more likely to default on their loans. By deploying such a system, banks and lending organizations can proactively manage risk, reduce non-performing assets (NPAs), and improve loan portfolio quality.

In a real-world setting, lenders often rely on traditional credit scores, which may not capture the complete financial behavior of an individual. Our machine learning approach goes beyond credit scores by analyzing multiple financial and demographic variables to flag high-risk profiles, thereby supporting **data-driven decision-making** in the financial sector.

**2. Dataset Overview**

We employed the **"Give Me Some Credit" dataset** available on Kaggle, which is a rich collection of synthetic financial data representing thousands of potential loan applicants. It consists of **32,561 records** with **12 input features** and **1 binary target variable**. Each row represents an individual along with various financial attributes.

**Key Features:**

* person\_age: Age of the applicant in years
* person\_income: Total annual income in USD
* person\_emp\_length: Employment length in years
* person\_home\_ownership: Homeownership status (e.g., Rent, Own, Mortgage)
* loan\_intent: Purpose of the loan (e.g., Personal, Education, Home Improvement)
* loan\_grade: Internal risk grade assigned by a financial institution
* loan\_amnt: Total loan amount requested
* loan\_int\_rate: Interest rate charged on the loan
* cb\_person\_cred\_hist\_length: Credit history length (years)
* cb\_person\_default\_on\_file: Indicates if the person has defaulted before
* loan\_status: Target column (0 = low-risk, 1 = high-risk)

This dataset is ideal for binary classification problems and mirrors real-world challenges in financial risk modeling, such as **missing values**, **class imbalance**, and **feature variety** (both numerical and categorical).

**3. Data Preprocessing**

Preprocessing is an essential step in any machine learning project, particularly when dealing with financial data that is often messy, incomplete, or inconsistent.

**3.1 Handling Missing Data**

We identified missing values primarily in two columns:

* person\_emp\_length: Employment length
* loan\_int\_rate: Interest rate

Rather than imputing values, which might introduce bias or noise, we **dropped rows** with missing data. This decision preserved the integrity of the data and ensured that the models were trained on complete, reliable inputs. However, in enterprise-level production systems, data imputation strategies could be explored (e.g., using regression or mean based on similar income groups).

**3.2 Categorical Feature Encoding**

The dataset includes multiple categorical variables like:

* person\_home\_ownership
* loan\_intent
* loan\_grade
* cb\_person\_default\_on\_file

Since machine learning models require numerical inputs, we used **One-Hot Encoding** to convert these variables into a binary matrix. This approach prevents the model from assuming any ordinal relationship among the categories.

For example:

* person\_home\_ownership = ['Rent', 'Own', 'Mortgage'] → becomes:
  + home\_Rent
  + home\_Own
  + home\_Mortgage

We used the parameter drop\_first=True to eliminate one dummy variable from each category group, avoiding multicollinearity.

**3.3 Feature Scaling**

In datasets with numerical features having different ranges (like income vs. credit history), it's important to scale them so that no feature dominates others in model training. We applied **Standard Scaling**, which transforms each feature to have:

* Mean = 0
* Standard Deviation = 1

This ensures optimal performance, especially for algorithms sensitive to feature magnitude (e.g., Gradient Boosting, SVM).

**3.4 Class Imbalance and SMOTE**

We found a significant imbalance in the target variable:

* Class 0 (low-risk): ~91%
* Class 1 (high-risk): ~9%

Training a model on such imbalanced data would lead it to predict mostly the majority class. To address this, we used **SMOTE (Synthetic Minority Over-sampling Technique)**. SMOTE creates synthetic samples of the minority class by interpolating between existing data points. This balanced the training data, enabling the model to learn from both classes effectively and improving metrics like **recall** and **F1-score**.

**4. Feature Engineering**

We enhanced the predictive power of the model by refining and engineering meaningful features. Examples include:

* **Debt-to-Income Ratio**: Calculated by dividing loan\_amnt by person\_income to estimate the applicant’s repayment burden.
* **Credit Utilization Proxy**: Combined loan\_int\_rate with cb\_person\_cred\_hist\_length to reflect a blend of risk and experience.
* **Risk Flags**: Converted cb\_person\_default\_on\_file into a binary flag that directly signals previous credit behavior.

Feature engineering is often domain-specific, and in financial modeling, it plays a vital role in boosting model performance by embedding business logic into data.

**5. Model Selection**

We trained and evaluated three powerful machine learning models:

**5.1 Random Forest Classifier**

* Ensemble of decision trees trained on bootstrapped data.
* High accuracy and robustness.
* Feature importance rankings available.
* Struggled with class imbalance before SMOTE.

**5.2 Gradient Boosting Classifier**

* Sequential tree-building algorithm.
* Each tree tries to correct the errors made by the previous one.
* Tends to perform better than random forest on structured data.
* Sensitive to hyperparameters, but great generalization after tuning.

**5.3 XGBoost Classifier**

* Enhanced version of Gradient Boosting with:
  + Regularization
  + Handling of missing data
  + Fast training using parallel computation
* Achieved the **highest accuracy and ROC-AUC score** among all models.
* Reduced overfitting through tree pruning and max\_depth control.

**6. Evaluation and Results**

We evaluated model performance on test data using the following metrics:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1 Score** | **ROC-AUC** |
| Random Forest | 89% | 0.62 | 0.49 | 0.55 | 0.91 |
| Gradient Boosting | 91% | 0.65 | 0.54 | 0.59 | 0.92 |
| **XGBoost** | **92%** | **0.68** | **0.58** | **0.63** | **0.93** |

* **Precision** indicates how many flagged high-risk customers were actually high-risk.
* **Recall** shows how many high-risk individuals the model successfully identified.
* **F1 Score** balances both precision and recall.
* **ROC-AUC** reflects the model's ability to distinguish between risk classes.

The **XGBoost Classifier** consistently outperformed the others and was selected as the final model for deployment.

**7. Challenges Faced**

1. **Imbalanced Data**: Initial model results were misleading due to dominance of class 0. We resolved this using SMOTE and by evaluating performance using recall and F1-score instead of accuracy alone.
2. **Missing Values**: Missing interest rates and employment length could have led to biased models. We handled this with conservative row deletion but explored the impact of imputation strategies.
3. **Model Interpretability**: Financial models must be explainable. We used feature importance from Random Forest and XGBoost to interpret decisions and justify risk classification.
4. **Overfitting Risks**: Some models performed very well on training data but poorly on validation. We introduced regularization, used cross-validation, and tuned hyperparameters to prevent overfitting.

**8. Conclusion and Future Work**

We successfully developed a credit risk prediction system that can classify customers as high-risk or low-risk based on a set of financial and behavioral attributes. This solution is scalable and can be adapted to real-time loan processing systems in banks and credit institutions.

The **XGBoost model** was found to be the most effective, with a balanced trade-off between recall, precision, and speed. By integrating this model with front-end systems, institutions can automate risk detection and improve overall lending strategies.

**Future Enhancements:**

* Incorporate time-series data such as monthly repayment logs.
* Use explainable AI (like SHAP values) for deeper transparency.
* Integrate with credit bureau APIs for real-time credit scoring.
* Add deep learning models for long-term behavior modeling.