Submitted By:

Mahnoor Atique (SP23-BAI-023)

Fizza (SP23-BAI-016)

Submitted To:

Sir Aksam Iftikhar

Semester:

SP24 (Semester 4)

Section:

SP23-BAI-A

Introduction:

The objective of this project is to predict the likelihood of credit card default among clients using a dataset from the UCI Machine Learning Repository. The dataset, titled "Defaults of Credit Card Clients," contains various features related to demographic information, credit history, and payment behavior of clients. The target variable is binary, indicating whether a client defaulted on their credit card payment in the following month.The dataset comprises 30,000 records with 23 features.

Methodology:

Preprocessing:

1. **Data Loading & Inspection**: The dataset is loaded into a Pandas DataFrame. The data is inspected for missing values, feature types, and class distribution. The target variable is highly imbalanced. Features were selected for modeling, excluding the target variable and the ID column. The dataset was split into training (80%) and testing (20%) sets.
    - **Scaling**: StandardScaler is used to normalize the feature space, ensuring all features have a mean of 0 and standard deviation of 1.
    - **Dimensionality Reduction**: PCA (Principal Component Analysis) is applied to retain 95% of the variance in the dataset, reducing dimensionality and mitigating the risk of overfitting.
2. **Handling Class Imbalance**: SMOTE is used to generate synthetic samples for the minority class (non-defaulters), balancing the dataset. This helps improve the model's ability to detect the minority class.

Algorithms Applied:

1. **LightGBM**:
    - An efficient gradient boosting framework, LightGBM, is used for its speed and performance in classification tasks.
2. **Support Vector Machine (SVM)**:
    - SVM with the radial basis function (RBF) kernel is applied to the dataset. Similar to LightGBM, hyperparameters like C, kernel, and gamma are optimized using RandomizedSearchCV.
3. **Random Forest Classifier:**
    - A bagging-based ensemble learning method that builds multiple decision trees and combines their outputs.
    - Default parameters were initially used, followed by hyperparameter tuning.
4. **XGBoost Classifier:**
    - A gradient boosting framework known for efficiency and high performance.
    - Initial evaluation conducted using default parameters, followed by tuning.

Optimization Techniques:

- **Hyperparameter Tuning**: Both RandomizedSearchCV and GridSearchCV were utilized to optimize hyperparameters for both classifiers, enhancing model accuracy and generalization capabilities.

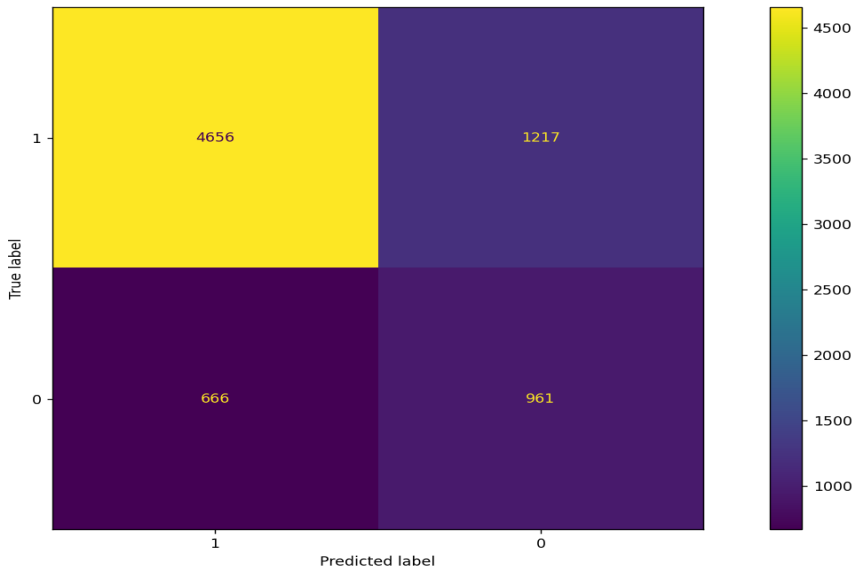
Results

The following evaluation metrics were used to compare the performance of the models:

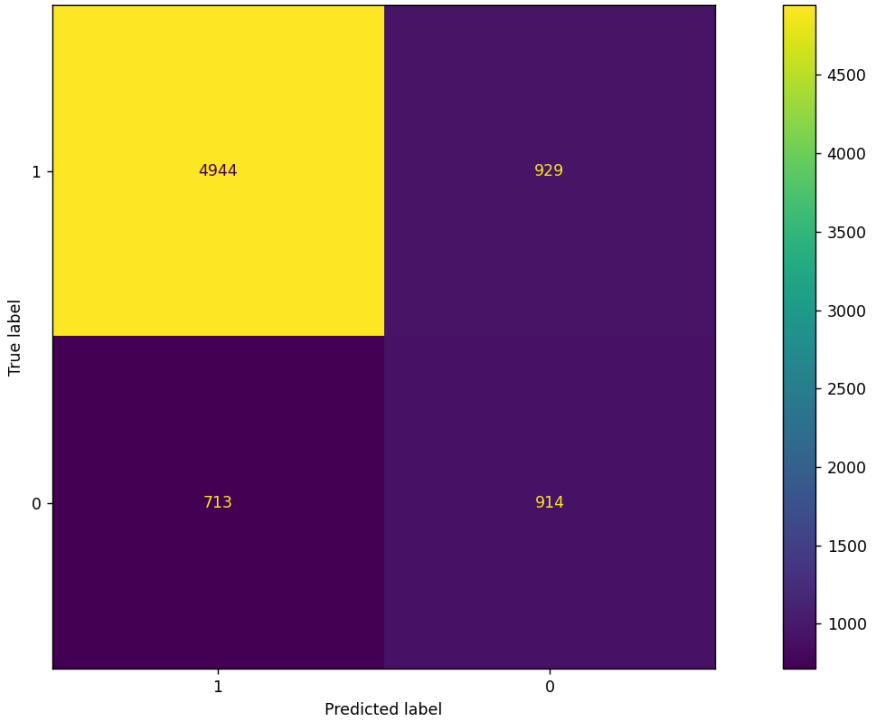
- **Accuracy**: The proportion of correct predictions.
- **Precision**: The proportion of positive predictions that were correct.
- **Recall**: The proportion of actual positives that were correctly identified.
- **F1-Score**: The harmonic mean of precision and recall.

Confusion Matrix:

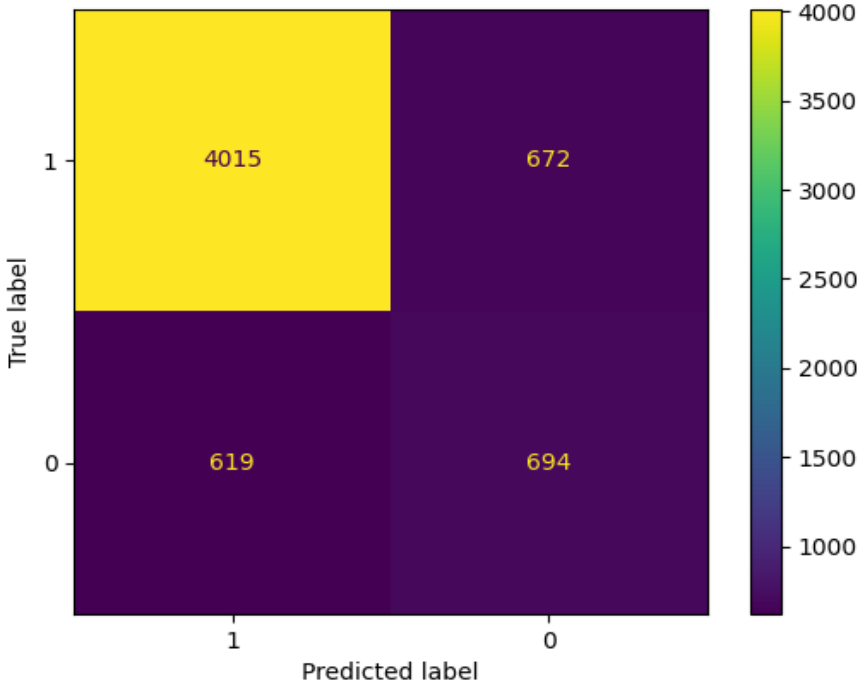
LightGbm:



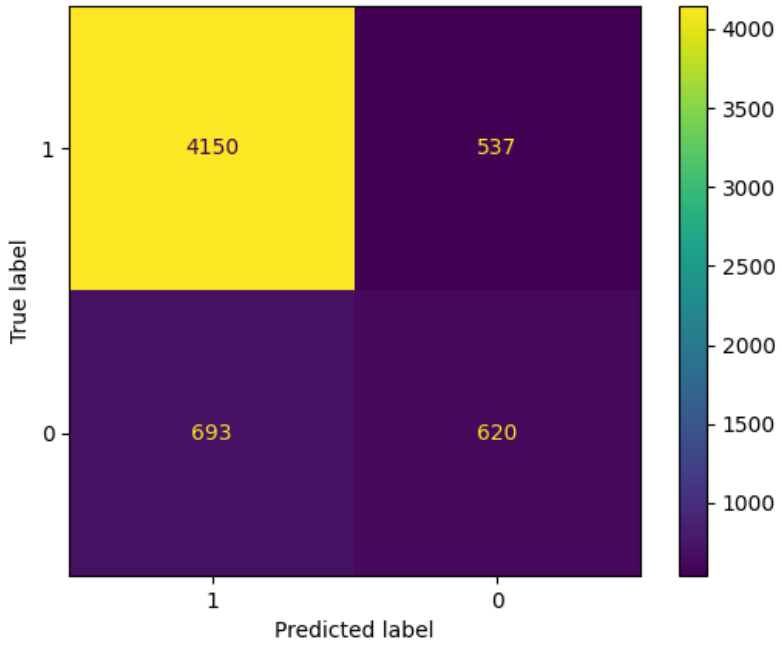
SVM:



Random Forest:

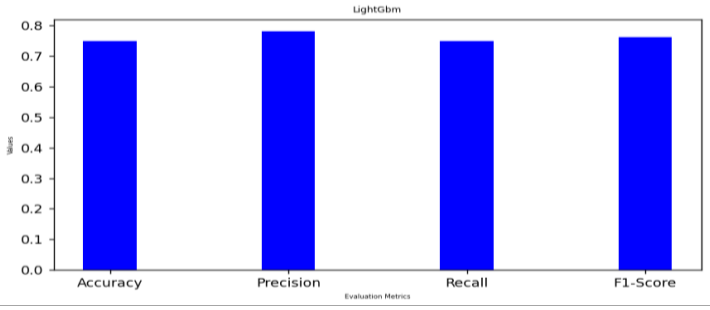


XG Boost:

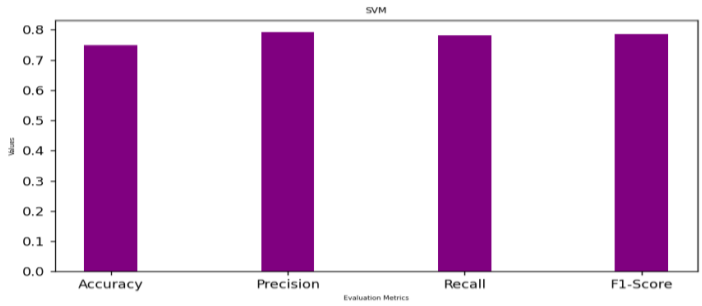


Bar Chart:

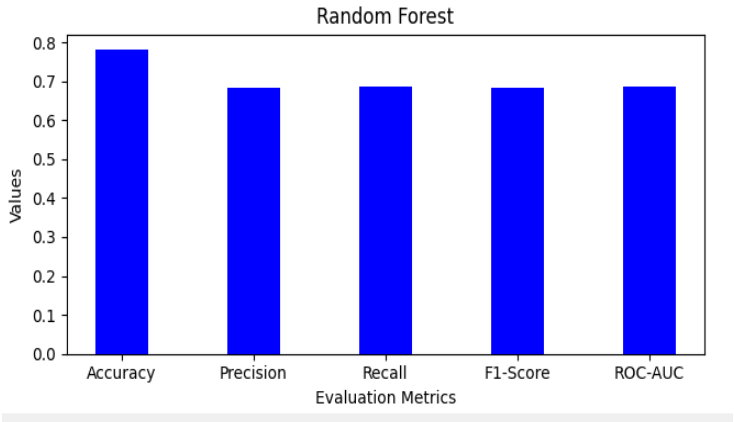
LightGbm:



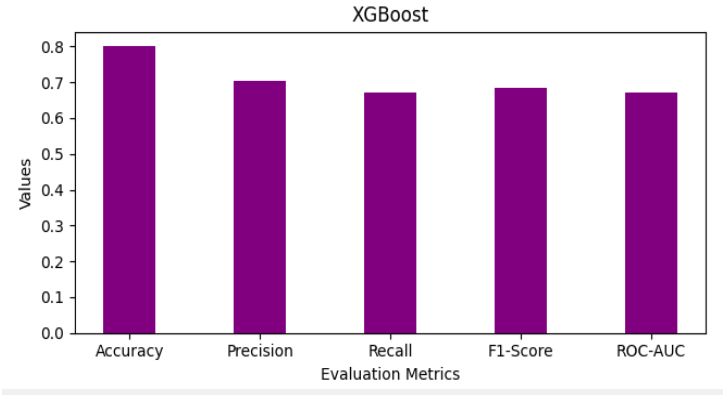
SVM:



Random Forest:



XG Boost:



Comparison of Machine Learning Algorithms on "Defaults of Credit Card Clients: "

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Best-Hyperparameters**	**Execution time(secs)**	**Remarks**
LightGBM	0.75	0.78	0.74	076	force_col_wise': True, 'learning_rate': 0.1, 'metric': 'binary_error', 'num_leaves': 40, 'objective': 'binary'	278.1	Efficient, high-performance, robust model.
SVM	0.78	0.79	0.78	0.78	'kernel': 'rbf', 'gamma': 0.01, 'degree': 3, 'C': 1.0	431.7	Reliable, solid, but slower.
Random Forest	0.77	0.79	0.78	0.78	'n_estimators': 100, 'min_samples_split': 5, 'max_features': 'sqrt', 'max_depth': 15, 'bootstrap': False	300	Robust performance, Generalized, slower
XG Boost	0.80	0.79	0.81	0.80	'subsample': 0.7, 'n_estimators': 150, 'max_depth': 7, 'learning_rate': 0.05, 'colsample_bytree': 0.8	168	Fast Execution

Visualizations

Confusion matrices for both models were plotted to visualize true positive, false positive, true negative, and false negative rates, providing insights into model performance.

Analysis

Insights:

- Both LightGBM and SVM performed similarly, with LightGBM achieving slightly better results in terms of precision and recall, while SVM had a marginally lower F1-score.
- The feature scaling and PCA helped to reduce dimensionality, while SMOTE effectively addressed the class imbalance issue.
- Both Random Forest and XGBoost demonstrated improved performance post-tuning, with XGBoost marginally outperforming Random Forest in most metrics.
- The use of SMOTE effectively mitigated the class imbalance issue, enhancing the minority class recall for both classifiers.

Algorithm Comparison:

- **LightGBM**: This model showed strong performance due to its boosting nature, and its hyperparameter tuning with RandomizedSearchCV and GridSearchCV helped to fine-tune the model for better accuracy.
- **SVM**: While SVM also performed well, it was slightly less efficient compared to LightGBM, particularly with the computational time involved in hyperparameter tuning.
- **Random Forest** showed robustness against overfitting due to its ensemble nature but had slightly lower recall compared to XGBoost.
- **XGBoost** provided superior performance in accuracy and F1-score, attributed to its advanced boosting mechanism and ability to handle complex data patterns.

Challenges :

- **Imbalanced Data**: The imbalanced nature of the dataset posed a challenge in model training. However, the application of SMOTE successfully mitigated this issue.
- **Feature Engineering**: The effectiveness of PCA in reducing dimensionality while preserving variance was crucial. However, selecting the right amount of variance to retain in PCA was a challenge that required careful tuning.
- **Hyperparameter Tuning**: Both models required extensive tuning using RandomizedSearchCV and GridSearchCV to find the optimal parameters. The tuning process was time-consuming but necessary for improving the model's performance.