College of Engineering

Dept of Computer Science and Engineering

Professor: Salam Dhou

CMP 466

Title: Credit Score Classification

Khaled Mohamed - B00087968

Koushal Parupudi - B00087520

Abstract

Abstract— Accurate classification of credit scores is essential for financial institutions to assess the creditworthiness of individuals and make informed lending decisions. This research aims to develop a comprehensive solution for classifying credit scores using machine learning techniques. Our approach analyzed various attributes, such as income, debt-to-income ratio, credit history length, and number of credit inquiries, to determine an individual's creditworthiness. We utilized decision trees, KNN, Naïve Bayes, and SVM (linear and RBF) classifiers, fine-tuning hyperparameters through k-fold cross-validation and GridSearchCV, to ensure model robustness and prevent overfitting. The results indicate that Decision Trees performed the best among the classifiers with a testing accuracy of 91%, demonstrating the efficacy of our approach in accurately classifying credit scores and aiding financial institutions in making informed lending decisions.

Keywords— Machine learning, Credit score classification, Statistical approaches , Class imbalance, Hyperparameter tuning, Feature selection

Introduction

Access to credit is an important factor that allows individuals and enterprises to reach their goals. However, traditional statistical methods used to assess creditworthiness frequently have shortcomings and inadvertently overlook certain populations, preventing them from accessing crucial financial resources due to overlooked data or features in the decision-making procedure[8]. Consequently, our machine learning project is aimed at enhancing the credit scoring process by developing more precise models based on machine learning algorithms. The main goal of the project is to tackle the problems that undermine the ability of the financially marginalized population to use it for their benefit. As highlighted by [10], machine learning techniques have shown promising results in credit scoring compared to traditional logistic regression models. Our approach aims to address this by feeding our model with data such as transaction histories, spending patterns, and economic indicators so that it can learn to cover the previously overlooked financial behaviors[5]. This would result in a more accurate assessment of the creditworthiness and consequently would enable financial institutions to take more responsible and informed decisions. Furthermore, the COVID-19 pandemic has severely impacted the global economy, causing a significant corporate credit crisis[8]. Efficient credit scoring systems are crucial in this context to effectively control credit risk and stabilize the sustainable development of the global economy. With every evaluation and refinement of our model, our ultimate goal is to scale our data-driven approach in a responsible manner, ensuring that the positive impacts of better credit scoring extend to all. For example, this involves the credit seekers, borrowers and lenders at all levels, policymakers and regulators who play an important role in creating inclusive and fair credit systems. In conclusion, we plan to employ machine learning to establish a financial ecosystem in which a broader array of people can have access to credit that is more fair and based on the evolving financial behaviors of modern times. Our approach aligns with recent research efforts in developing automated machine learning pipelines for efficient and accurate credit scoring [11].

Literature review

Paper ID	Title	Objective	Methods	Dataset	Key Findings	Limitations
[1]	Experimental analysis of machine learning methods for	Evaluate the effect of feature selection methods on credit score classification.	Nine feature selection and sixteen classification techniques were used on seven benchmarked datasets.	7 benchmarked credit scoring datasets.	The combination of specialized feature selection and classification approaches improves	Generalizability and interpretability are restricted; findings may not be applicable generally, and some models are difficult to

	credit score classification				performance significantly, with the NRS-based selection and layered ensemble approach being the best performers.	interpret.
[2]	Machine Learning approach for Credit Scoring	To produce an advanced credit score and default prediction system employing a variety of machine learning techniques.	Machine learning models such as gradient boosting machines (GBM), natural language processing (NLP), and genetic algorithms were used for economic sector descriptions and rating attribution, with an emphasis on model interpretability using SHAP and LIME.	Credit Research Database (CRD) provided by Moody's, containing annual financial statements of 157,986 Italian companies from various sectors.	Strong out-of-sample performance demonstrated the utility of the complex machine learning framework as well as the significance of model interpretability.	Thepaper lacks detailed limitations discussion.However, a broader issue might include the model's particular to Italian firms as well as its complexity, which affects use and interpretability.
[3]	Use of Machine Learning Techniques to Create a Credit Score Model for Airtime Loans	To create a machine learning-based credit score model for airtime loans with the goal of assessing creditworthiness with limited data.	Applied logistic regression, decision trees, and random forests using several cross-validation techniques to classify defaulters and non-defaulters among airtime loan users.	Data on over three million loans from more than 41 thousand customers with a three-month repayment period.	Random forest models were the most effective at categorizing defaulters, highlighting the potential of nonlinear classification models for controlling increased default rates and expanding client bases in airtime lending.	The reliance on a single company's data may restrict generalizability; more demographic data might improve future studies.
[4]	Application Analysis of Credit Scoring of Financial Institutions Based on Machine Learning Model	To improve personal credit evaluation utilizing big data approaches and optimize the credit scoring process using various machine learning models.	pdC-RF technique was used for feature correlation and reduction, WOE coding, and the performance of logistic regression, random forest, and SVM models were compared.	Lending Club dataset featuring 96,781 samples with 145 characteristics, focusing on financial data and credit status.	Logistic regression was found most suitable for personal credit evaluation on the Lending Club dataset. The research demonstrated the potential of machine learning in improving the accuracy and efficiency of credit	The study's use of a single dataset (Lending Club) could limit its generalizability. Potential data biases, as well as the specific models used, could have an impact on the results' reliability.

					scoring systems.	
[5]	An ensemble classifier model to predict credit scoring - comparative analysis	To assess and compare the performance of base and ensemble classifier models in predicting creditworthiness using various machine learning methods.	Applied multiple classification techniques, such as Decision Trees, Logistic Regression, Nearest Neighbor, Support Vector Machine, Random Forest, Extratree, and others	Australian credit dataset from the UCI Machine Learning Repository containing features that are used to assess creditworthiness.	Ensemble models like Random Forest and Extratree classifiers showed higher accuracy among ensemble classifiers, while the SVM model was the most accurate among base classifiers.	The research primarily focuses on a single dataset which could affect the generalizability of the results to other types of credit data or diverse financial contexts.
[6]	Mathematical Modeling and Analysis of Credit Scoring Using the LIME Explainer: A Comprehensive Approach	To enhance credit scoring models using various machine learning algorithms and the LIME explainer, focusing on mathematical modeling and interpretability.	Employs a range of machine learning algorithms such as logistic regression, decision trees, support vector machines, and neural networks; uses LIME to increase model transparency.	The study utilized multiple publicly available credit datasets, specifically Australian, German, and South German datasets, to evaluate the performance of various machine learning models.	Demonstrated high accuracies with 88.84% for the Australian dataset, 78.30% for the German dataset, and 77.80% for the South German dataset; emphasized the role of LIME in improving interpretability.	The paper potential limitations could include the generalizability of the results across different types of credit datasets or real-world applicability.
[7]	Credit Score Classification Using Spiking Extreme Learning Machine	To improve the accuracy and efficiency of credit score classification using a novel machine learning approach inspired by biological neuron activities.	Employs a spiking neuron model (Leaky Nonlinear Integrate and Fire Model) into the Extreme Learning Machine (ELM) architecture to utilize interspike intervals as inputs for enhancing credit score classification.	Utilized five real-world credit scoring datasets, including Australian, German-categorical, German-numerical, Japanese, and Bankruptcy.	The Spiking Extreme Learning Machine (SELM) outperformed traditional and neural network-based classifiers in accuracy, AUC, H-measure, and computing efficiency, indicating a significant increase in credit scoring approaches.	The paper lacks limitation discussions, but challenges might include the complexity of the SELM model which could impact its broader application and understanding in the financial sector.
[8]	Bacs: Blockchain and AutoML-based technology for efficient credit	The paper proposes BACS, a blockchain and automated machine learning (AutoML) based classification model for efficient credit	BACS consists of credit data storage on blockchain, feature extraction, feature selection, modeling algorithm, hyperparameter	Four credit datasets are used: three from UCI (German, Taiwan, Australia) and one from Kaggle	BACS achieves significant performance improvements in terms of accuracy, specificity, and AUC compared to	The paper does not provide a detailed comparison of the proposed method with other state-of-the-art credit scoring approaches beyond RF

	scoring classification	scoring	optimization, and model evaluation.	(Credit card).	standalone RF and SVM models.	and SVM.
[9]	Machine Learning for Credit Scoring: Improving Logistic Regression with Non-Linear Decision-Tree Effects	The paper proposes a new credit scoring method called penalised logistic tree regression (PLTR) that aims to improve the predictive performance of logistic regression while maintaining its interpretability.	PLTR uses short-depth decision trees to extract univariate and bivariate threshold effects, which are then included as predictors in a penalized logistic regression.	Kaggle "Give me some credit" dataset with 150,000 loans	PLTR outperforms logistic regression and is competitive with random forest in predictive performance while remaining interpretable.	Only univariate and bivariate threshold effects are considered for interpretability; including higher-order effects could further improve performance.
[10]		The paper proposes a credit scoring model (SIFS-PNN) that combines swarm intelligence-based feature selection with a PSO-trained artificial neural network (ANN) for classification to predict if a client is likely to default.	The Classification Module uses a Particle Swarm Optimization (PSO) tuned ANN to classify credit as good or bad.	Australian and Japanese credit scoring datasets	The proposed SIFS-PNN model achieved 93% and 91% accuracy on the Australian and Japanese datasets respectively, outperforming several previous models.	The study only uses two credit scoring datasets for evaluation. Testing on more datasets would strengthen the results.
[11]	Federated learning: Collaborative machine learning without centralized training data	The paper aims to introduce federated learning, a machine learning technique that enables collaborative model training across decentralized edge devices without sharing raw data, while addressing data privacy, security, and access rights issues.	Federated learning allows devices to download a shared prediction model, improve it using local data, and send a brief update summarizing the changes to the cloud.	The paper does not mention a specific dataset but uses Google Keyboard (Gboard) as an example application of federated learning.	Federated learning enhances model accuracy, predicts the next word based on user input, and powers the predictions strip in Gboard.	Communication in federated networks can be slower than local computing.
[12]	Credit score prediction using support vector machine and Gray Wolf Optimization	The paper proposes a credit score prediction method that combines Support Vector Machine (SVM) with Gray Wolf Optimization (GWO) to optimize the SVM hyperparameters for	SVM is used for credit score prediction. The SVM model is trained on historical credit data to predict credit scores for new applicants.	Data is collected from a financial institution, including attributes like gender, income, loan history etc.	SVM-GWO outperformed standard SVM in terms of accuracy (91% vs 86%), precision (92% vs 88%), recall (93% vs 89%), and F-measure (93% vs 89%).	The paper does not provide details on the size of the dataset used or how it was split into training and testing sets. The specific hyperparameter values found by GWO are not reported.

		improved prediction performance.				
[13]	A decision tree classifier based ensemble approach to Credit Score Classification	The paper proposes an ensemble-based approach for credit score classification that combines multiple machine learning algorithms to improve accuracy and resilience.	The ensemble approach uses a combination of base classifiers: Bagging Classifier, Extra Trees Classifier, Random Forest, Histogram Gradient Boosting Classifier, and XGBClassifier.	The "Credit Score Classification Dataset" from Kaggle is used	The proposed ensemble model achieved an accuracy of 92.25%, which is competitive with or better than several previous credit scoring models.	The paper does not provide details on the computational efficiency or scalability of the ensemble approach compared to other methods.
[14]	A study on credit scoring modeling with different feature selection and machine learning approaches	This study aims to build an optimum credit score prediction model by comparing different feature selection techniques and machine learning classifiers.	Five machine learning classifiers are evaluated: Bayesian, Naïve Bayes, Support Vector Machine (SVM), Decision Tree (C5.0), and Random Forest (RF).	publicly available German Credit dataset	The Chi-Square feature selection technique combined with the Random Forest classifier achieved the best performance, with 76.20% accuracy, 76.18% F-measure,	A limited number of feature selection methods and machine learning classifiers were tested.
[15]	Machine learning interpretability for a stress scenario generation in credit scoring based on Counterfactuals	This paper proposes a novel approach to evaluate the interpretability of machine learning (ML) credit scoring models using counterfactuals and a method to generate stress scenarios for cross-sectional data.	A data perturbation technique based on k-Nearest Neighbours (kNN) is proposed to generate synthetic data representing stress scenarios.	A dataset from Nationwide Building Society on 61,239 UK unsecured personal loans	Training SGB on augmented data with increased default rates negatively impacts model performance.	The paper does not provide details on the specific features in the Nationwide dataset due to confidentiality reasons.
[16]	A benchmark of machine learning approaches for Credit Score Prediction	The paper proposes a benchmarking study of machine learning approaches for credit risk scoring in peer-to-peer (P2P) lending platforms, considering accuracy and explainability of the models.	Three classifiers (Random Forest, Logistic Regression, Multi-layer Perceptron) are evaluated using different sampling strategies (under-sampling, over-sampling, hybrid methods) to handle class imbalance.	Data from Lending Club, a real P2P lending platform, consisting of 877,956 samples and 151 features.	Random Forest with Random Under Sampling (RF-RUS) emerged as the best method in terms of G-Mean (0.656), outperforming other sampling approaches and the classifiers without sampling.	The study is limited to one dataset from a single P2P lending platform. Evaluation on additional datasets would strengthen the findings.

[17]	Bridging accuracy and interpretability: A rescaled cluster-then-predict approach for enhanced credit scoring	The paper proposes a novel "Rescaled Cluster-then-Predict" approach to enhance both the interpretability and predictive performance of credit scoring models	The performance of the proposed approach is compared with XGBoost and Logistic Regression without clustering.	PAK dataset with 50,000 cases and the GMC dataset with 150,000 cases.	The proposed Rescaled Cluster-then-Predict approach achieves competitive performance compared to XGBoost while substantially improving interpretability.	As data volumes increase, the clustering aspect may become computationally demanding, potentially delaying the credit scoring process.

Dataset

The following database includes a tabular dataset with a description of banks' clients and focuses on their credit status. The dataset includes around 100,000 samples totally with 27 distinct features. A description of the 27 features are as the following:

1. **ID:** An identifier for each record in the dataset, typically a unique number or code.
2. **Customer ID:** An identifier for each customer, which can be used to link multiple records for the same customer.
3. **Month:** The month or time period to which the data corresponds. It may indicate when the data was collected or when the credit scoring assessment was made.
4. **Name:** The name of the customer or borrower.
5. **Age:** The age of the customer, which can be a factor in credit scoring.
6. **SSN:** The Social Security Number or a unique identification number for the customer, often used for verification purposes.
7. **Occupation:** The customer's occupation or employment status, which can provide insight into their financial stability.
8. **Annual Income:** The customer's total annual income, a key factor in determining creditworthiness.
9. **Monthly Inhand Salary:** The customer's monthly take-home salary after deductions.
10. **Num Bank Accounts:** The number of bank accounts held by the customer, which may indicate financial stability.
11. **Num Credit Card:** The number of credit cards owned by the customer.
12. **Interest Rate:** The interest rate associated with a loan or credit, if applicable.
13. **Num of Loan:** The number of loans currently held by the customer.
14. **Type of Loan:** The type or category of the loan(s), such as personal loan, mortgage, or car loan.
15. **Delay from due date:** The delay in making payments from the due date, which may indicate a history of late payments.
16. **Num of Delayed Payment:** The number of delayed or missed payments.
17. **Changed Credit Limit:** Whether there has been a change in the customer's credit limit.
18. **Num Credit Inquiries:** The number of times the customer's credit report has been accessed by creditors or lenders.
19. **Credit Mix:** The variety of credit types used by the customer, such as credit cards, loans, and mortgages.
20. **Outstanding Debt:** The total amount of outstanding debt owed by the customer.
21. **Credit Utilization Ratio:** The ratio of credit used to credit available, often associated with credit cards.
22. **Credit History Age:** The length of the customer's credit history, which can impact credit scores.
23. **Payment of Min Amount:** Whether the customer has consistently made at least the minimum required payments on loans or credit cards.
24. **Total EMI per_month:** The total Equated Monthly Installments (EMI) paid by the customer.
25. **Amount invested monthly:** The amount the customer invests or saves monthly.
26. **Payment Behavior:** An indicator of the customer's payment behavior, such as "good," "fair," or "poor."
27. **Monthly Balance:** The customer's monthly account balance or financial position.

The dataset contains 3 classes pertaining to the credit score which is given to the customer according to several calculations and factors to prove their creditworthiness. The following classes are poor (28998 occurrences), standard (53174 occurrences), and finally good (17828 occurrences) respectively.

Data preprocessing methods

Data preprocessing is an essential step in machine learning projects, particularly in sensitive applications like credit scoring. Our research uses various critical data preprocessing techniques to assure the data's integrity and quality before it is input into our machine learning models. This section describes the theoretical basis for the methods utilized, which include, encoding categorical variables, min-max scaling, among others.

- **Handling Missing Values**

Missing values in a dataset can have a major influence on the performance of machine learning models. To solve this, we utilize techniques such as deletion based on the nature of the data.

- **Encoding Categorical Variables**

Machine learning models require numerical input, thus encoding categorical variables is a crucial step. We utilize One-Hot Encoding for nominal data, where there is no ordinal connection, and Ordinal Encoding for ordinal data, where the categories are ordered. This conversion allows models to efficiently comprehend and use information from categorical data.

- **Scaling and Normalization**

Scaling changes the range of the data, whereas normalization changes the form of the distribution of data. These phases are critical when the dataset's characteristics cover multiple ranges. We implement Min-Max Scaling to rescale the data within a given range, often 0 to 1, ensuring that no one feature dominates owing to its size, as shown in the formula below.

$$X_{\text{Scaled}} = \frac{X - X(min)}{X(max) - X(min)}$$

Equation 1. Equation of Min-Max

Machine learning methods

Decision trees are a popular machine learning approach to classification and regression problems. They model decisions and the possible consequences using a tree-like structure, making them simple to understand and similar to human decision-making.

Essentials of Decision Trees:

Building Process: Decision trees divide data into branches based on feature thresholds, seeking to generate as homogeneous subsets as possible in terms of the goal variable. The procedure begins from the root and proceeds until a stopping requirement is reached**.**

Node Types: root nodes, internal nodes, and leaf nodes.

Pruning: To prevent overfitting, trees can be trimmed to remove less informative branches.

Criteria for splitting: Splits are calculated using metrics such as Gini impurity or entropy for classification and variance for regression, with the goal of maximizing information gain.

$$Gini\ index\colon\ 1\ -\ \sum_{i=0}^{c-1} pi(t)^2$$

Equation 2: Gini Index

$$Entropy\colon\ \ -\ \sum_{i=0}^{c-1} pi(t) log_2 pi(t)$$

Equation 3: Entropy

The K-nearest neighbor (KNN) approach is a machine learning algorithm for classification and regression that stores all existing instances before categorizing new data based on their similarity (distance). It detects the K-nearest neighbors of a new data point in the dataset, and the majority class of these neighbors determines the new point's categorization. The similarity of data points may be calculated using a variety of distance metrics, including Euclidean, Manhattan, and Maximum distance. The number of neighbors (k) must be carefully chosen since it has a substantial impact on the classification outcome, particularly in datasets with unbalanced classes. The algorithm's

efficiency is determined on the suitable selection of 'k', which may be adjusted using methods such as cross-validation to verify correctness and successfully manage data imbalances [18].

$$D(p,q) = \sqrt{\sum_i (p_i - q_i)^2}$$

Equation 4. Euclidean distance measure

The Naive Bayes classifier succeeds in supervised classification because it assumes conditional independence of features within a class. It computes the posterior probability for each class using Bayes' theorem, combining categorical and numerical data—direct computations for categorical characteristics and Gaussian distributions for numerical ones. To increase accuracy, it employs variable selection strategies to choose the best feature subset, which frequently involves the combination of many models with variable weighting. This successfully modifies each feature's impact depending on its contribution to accuracy, increasing the model's efficiency while compensating for its naive independence assumption [19].

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)}$$

Equation 5. Bayes Theorem

Support Vector Machines (SVM) are an effective set of supervised learning algorithms used for classification and regression. They distinguish themselves by generating a hyperplane or group of hyperplanes in a high-dimensional space to achieve optimal class separation. Linear SVMs do this by increasing the margin between the nearest data points in any class, hence reducing classification errors.However, for non-linear data, SVMs use kernel functions such as polynomial, radial basis function (RBF), and sigmoid to transfer inputs into higher-dimensional spaces where linear separation is possible, without doing direct computation in that space. This kernel method enables SVMs to effectively handle complicated datasets, making them applicable to a wide range of applications where data may not be linearly separable in the original input space[20].

Feature Selection and Dimensionality reduction

Feature selection is the process of identifying and selecting a subset of the most relevant features from the original feature set. The goal is to remove irrelevant, redundant, or noisy features that may negatively impact the model's performance or generalization ability. By focusing on the most informative features, feature selection can improve model accuracy, reduce overfitting, and speed up training and inference times. We have employed mutual information (MI) score as a feature selection technique that measures the amount of information shared between a feature and the target variable. It quantifies the reduction in uncertainty about the target variable when the value of a specific feature is known. Also, MI is a non-parametric method and does not make assumptions about the underlying data distribution and it can handle both continuous and categorical features. In the context of credit scoring, MI score can be used to evaluate the relevance of each feature in predicting creditworthiness. Nevertheless, the literature review proposed other filter methods, which assess the relevance of features independently of the learning algorithm. They use statistical measures such as correlation, mutual information, or chi-squared tests to rank or score features based on their relationship with the target variable[9]. Other examples include information gain, gain ratio, and correlation-based feature selection [15].

Dimensionality reduction techniques aim to transform the original high-dimensional feature space into a lower-dimensional representation while preserving the most important information. By reducing the number of dimensions, these techniques can alleviate the curse of dimensionality, improve computational efficiency, and facilitate data visualization. The main dimensionality technique used in this paper is Principal Component Analysis (PCA), which identifies the directions of maximum variance in the data and projects the data onto a lower-dimensional space formed by the principal components.

Evaluation metrics

Accuracy: Accuracy is a basic metric that measures the proportion of correctly classified instances among the total instances in a dataset. It is calculated as the number of correct predictions divided by the total number of predictions made as shown in the formula below.

$$\text{Accuracy} = \frac{TP+TN}{TP + TN + FN + FP}$$

Equation 6. Equation of Accuracy

where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively.

Precision: Precision measures the proportion of true positive predictions (correctly predicted positives) among all positive predictions made by the model. It is calculated as true positives divided by the sum of true positives and false positives as shown in the formula below.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Equation 7. Equation of Precision

Recall: The Recall metric measures the proportion of true positive predictions among all actual positive instances in the dataset. It is calculated as true positives divided by the sum of true positives and false negatives.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Equation 8. Equation of Recall

F1-Score: The F1-score is a metric that combines precision and recall into a single value. It is particularly useful when dealing with imbalanced classes. F1-score is the harmonic mean of precision and recall, and is calculated as shown below.

$$\text{F1 - SCORE} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Equation 9. Equation of F1-Score

Area Under the Curve (AUC): AUC is a metric used to evaluate the performance of a binary classification model. It represents the area under the receiver operating characteristic (ROC) curve, which plots the true positive rate against the false positive rate at various threshold settings.

Receiver Operating Characteristic Curve (ROC): The ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classification model across different threshold settings. It plots the true positive rate (sensitivity) against the false positive rate (1 - specificity)..

Confusion Matrix: A confusion matrix is a table that summarizes the performance of a classification model by counting the number of true positives, true negatives, false positives, and false negatives. It provides a detailed breakdown of the model's predictions, allowing for the calculation of various metrics like accuracy, precision, recall, and F1-score.

These evaluation metrics are essential tools for assessing the performance of machine learning models, providing insights into their effectiveness in correctly predicting classes, handling imbalanced data, and understanding the trade-offs between precision and recall.

Data Preprocessing

Upon examining the dataset, it was observed that it did not contain any missing values nor any duplicates, removing the need to perform any initial cleaning of data. However, the dataset contained numerical features that did not require to be preprocessed and categorical features that required to be encoded to ensure homogeneity numerical features. The first step was to rid the dataset of unimportant features that we determined would not contribute to the prediction of the output classes. These were - ID, Customer_ID, Name, SSN. Following this, we performed ordinal encoding on the features - Credit_score, Credit_mix and Payment_Behavior, and Label encoding on the features - Occupation and Payment_of_Min_Amount. For the feature – Type_of_Loan, encoding was not a simple task since the entries were a list of loan types for each customer (Example: auto loan, credit-builder loan, personal loan, home equity loan). Therefore, we extracted the unique loan types and performed one hot encoding. Thus, all the categorical features were converted to numerical features. Next, we investigated the class distribution and found that the dataset had some imbalance, where 2 classes were underrepresented and out of which, one was severely underrepresented as shown in Figure 1. As this would be detrimental for the models' training, we oversampled the underrepresented classes to the same sample size as the majority class. Thus, the distribution was balances, which can be observed in Figure 2.

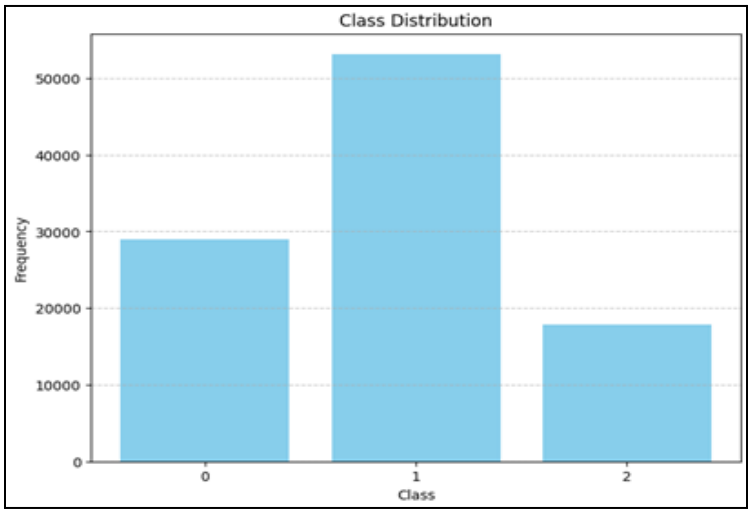


Figure 1: Class distribution before oversampling

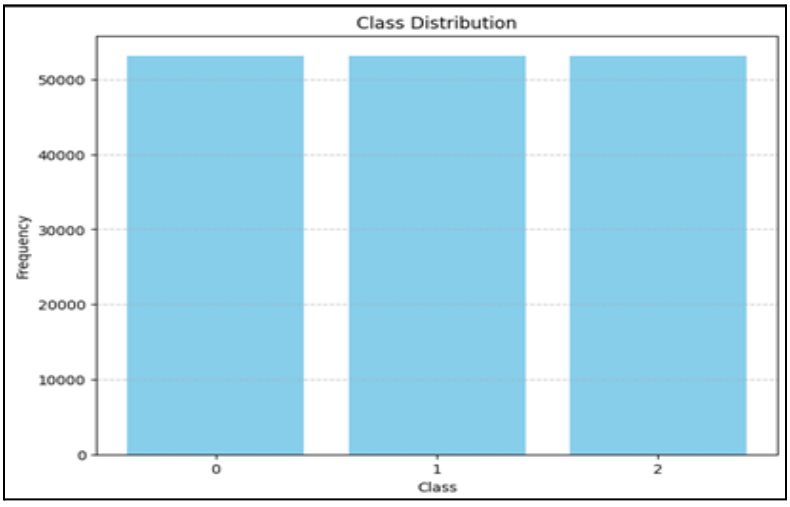


Figure 2: Class distribution after oversampling

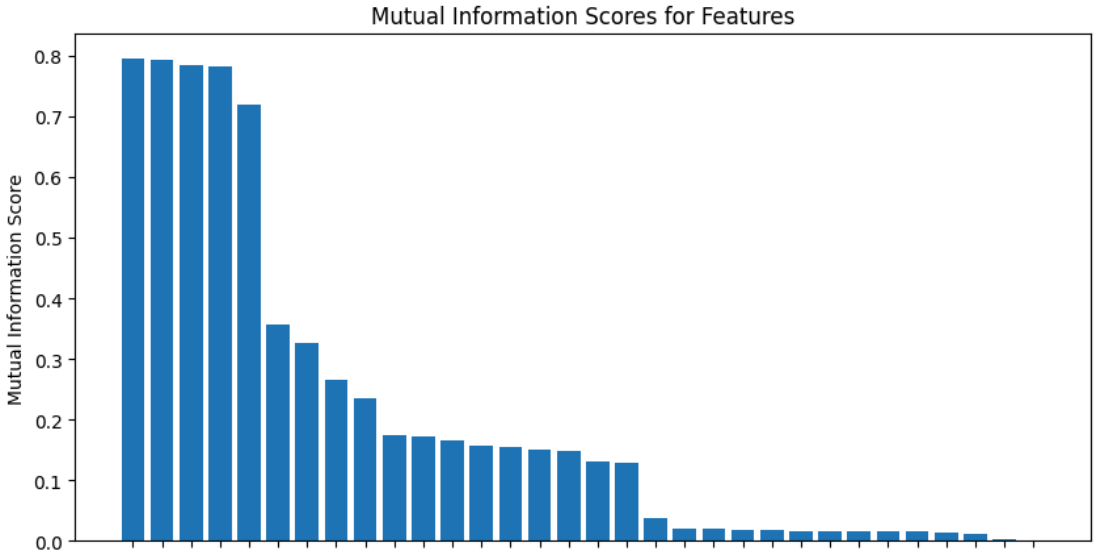
Finally, the last step included scaling the features using minmax scaling to bring the features to the same scale and balance the influence on the prediction.

Dimensionality Reduction

Given our dataset's dimensionality, it's crucial to assess our models both before and after dimensionality reduction. This process helps us understand if the reduction in features impacts model performance. If our models perform similarly before and after reduction, it suggests that many features were redundant and didn't contribute significantly to predictive power. Essentially, it means that the dataset's dimensionality is well-captured by fewer features, preserving most of the data's variance and essential characteristics. Moreover, comparable performance levels indicate that dimensionality reduction effectively addresses potential overfitting while improving model interpretability and computational efficiency. By streamlining the model without sacrificing accuracy, dimensionality reduction enables us to create a more generalizable model. Therefore, we'll assess our models before and after dimensionality reduction using Principal Components Analysis (PCA), ensuring that we retain 95% of the data's variance. By reducing the feature space we can reduce the complexity of our models drastically, which has various implications during real-world deployment.

Feature selection:

In our analysis of the credit score dataset, we employed mutual information (MI) score as a metric to evaluate the relevance of each feature in predicting creditworthiness. After computing the MI scores for all features, we selected the top 10 features with the highest MI scores. These selected features represent the variables that exhibit the strongest statistical dependence with the target variable, i.e., the credit score. By focusing on these top-ranking features, we aim to streamline our predictive model and enhance its accuracy by leveraging the most informative attributes of the dataset. This feature selection process enables us to prioritize the most relevant aspects of the data while discarding potentially redundant or less impactful variables, ultimately improving the efficiency and interpretability of our credit score prediction model.



These are the best features with MI scores from left to right on the **X-axis**.

Feature 1: Monthly Inhand Salary
Feature 2 : Annual Income
Feature 3: Amount Invested Monthly
Feature 4 : Outstanding Debt
Feature 5 : Total EMI per Month
Feature 6 : Monthly Balance
Feature 7 : Credit Mix
Feature 8 : Interest Rate

Feature 9 : Delay from Due Date

Machine learning results:

Decision trees:
For decision trees, we started by running the algorithm with hyperparameter tuning, to find the best hyperparameter pertaining to maximum depth while keeping all the features of the dataset. This resulted in the graph seen below in Figure x.

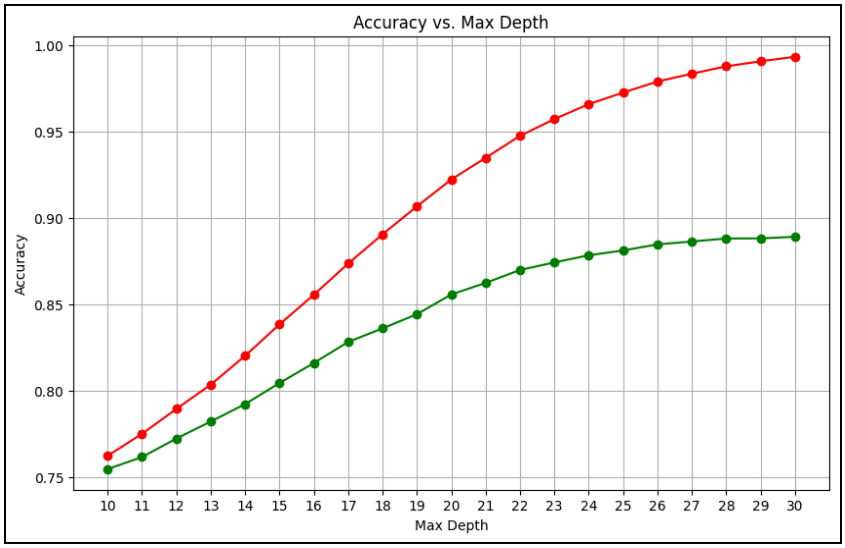


Figure 3: Decision trees with hyperparameter tuning

The decision tree with the optimal hyperparameter was determined to be of max_depth 27 since the accuracy was not significantly lower than the accuracies at max_depth 28 onwards. Next, we ran the decision tree with max_depth set to 27 on 10 features, incrementally training the model to see if the model can perform well with lesser features. Below is Figure 3 showing a graph of the model's performance with an incremental number of features and the respective classification report and confusion matrix of the best performing model in Figure y.

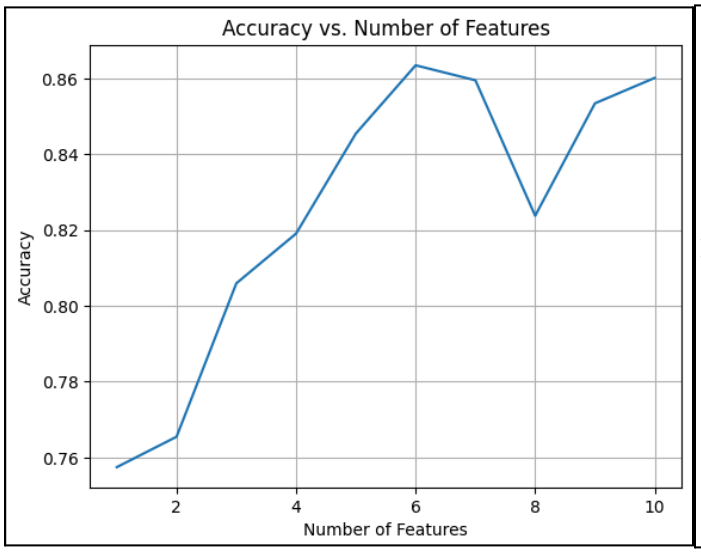


Figure 4: Accuracy for each feature

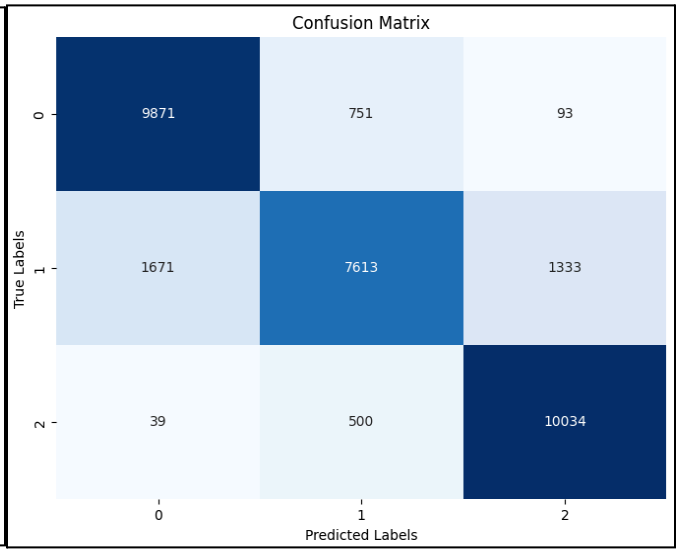


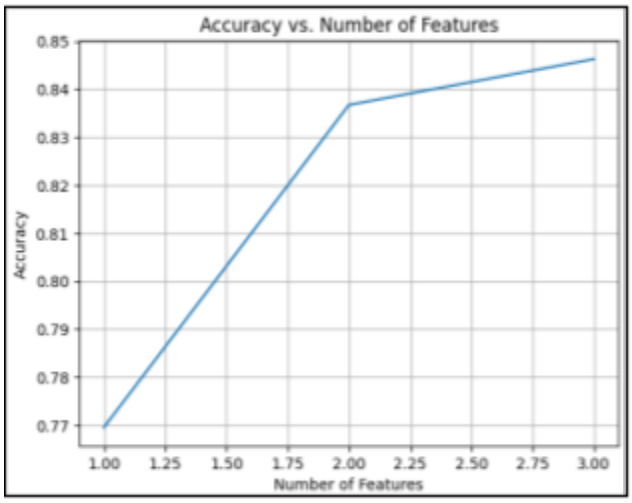
Figure 5 : Confusion matrix

Class	Precision	Recall	F1-score	Support
Good	0.85	0.92	0.89	10715
Poor	0.85	0.72	0.78	10617
Standard	0.88	0.95	0.91	10573
Accuracy			0.86	31905
Macro Average	0.86	0.86	0.86	31905
Weighted Average	0.86	0.86	0.86	31905

Table 1 : Classification report

The above table indicates that the best performing model is the model with the top 6 features having an accuracy of 86%.

Finally, running the model with reduced dimensions produced the following results.



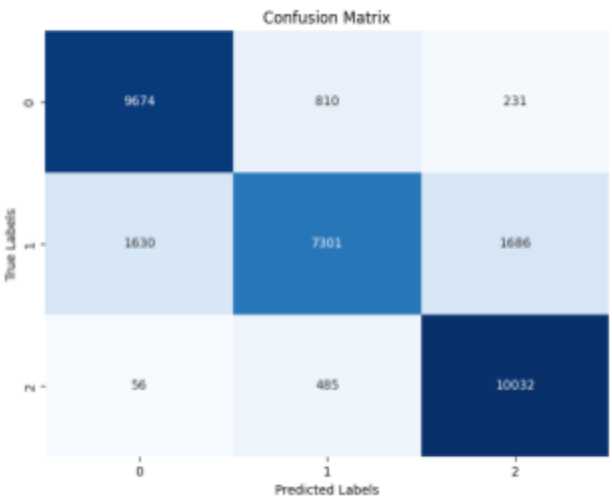


Figure 6 : Accuracy per feature Figure 7: Confusion Matrix after reducing dimensions

Class	Precision	Recall	F1-score	Support
Good	0.85	0.92	0.88	10715
Poor	0.85	0.69	0.76	10617
Standard	0.84	0.95	0.89	10573
Accuracy			0.85	31905
Macro Average	0.85	0.85	0.84	31905

Weighted Average	0.85	0.85	0.84	31905

Table 2 : Classification Report with reduced dimensions

Analyzing the table, we can say that the model is very good at identifying class good and class standard. Class poor seems to be the most difficult for the model to classify, as it has the smallest recall and F1-score. To conclude, this model performed slightly worse (only off by 0.01) than the model with 6 features as some information may have been lost in the reduced dimensions and the model appears to be performing well overall, but there is room for improvement in identifying class 1. Reducing the number of features from 6 to 5 may have negatively affected the model's performance.

Naive Bayes:

We began evaluating the Naive Bayes Classifier by running the model with all the features. Figure x and y are the confusion matrix and the classification report. The report shows that the model performed worse than the decision tree model with all the features.

Class	Precision	Recall	F1-score	Support
Good	0.72	0.74	0.73	10715
Poor	0.66	0.36	0.47	10617
Standard	0.60	0.87	0.71	10573
Accuracy			0.65	31905
Macro Average	0.66	0.65	0.63	31905
Weighted Average	0.66	0.65	0.63	31905

Table 3: Classification report for Naive Bayes.

Analyzing the classification report, we can see the model performs well at identifying classes 0 and . Class good has the highest precision (0.72) and F1-score (0.73), indicating the model accurately predicts this class. Similarly, the class standard has a high recall (0.87), meaning the model catches most of these data points, with a moderate F1-score (0.71). However, class poor presents a challenge. Here, the model has the lowest recall (0.36) and F1-score (0.47), suggesting it frequently misclassified data points belonging to this class.In conclusion, while the model achieves moderate overall accuracy, the inaccuracy achieved with class 1.0 highlights a potential limitation of Naive Bayes classifiers. The Naive Bayes assumption of feature independence might not hold true for all datasets, leading to issues in handling complex relationships between variables.

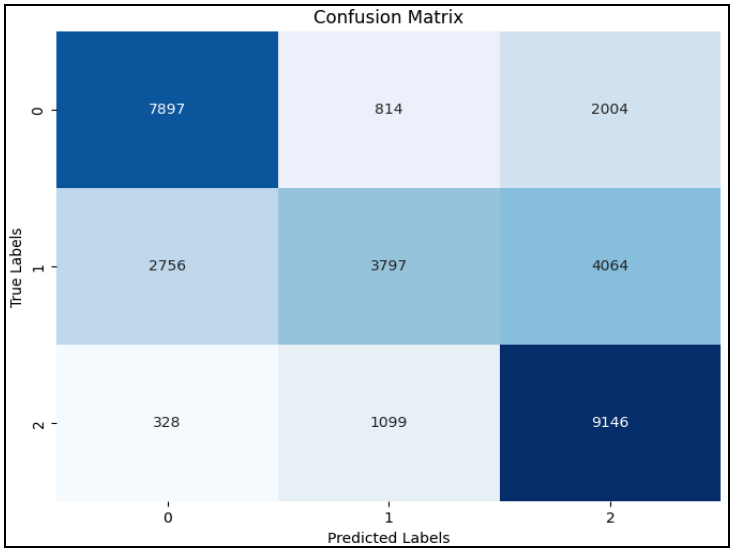
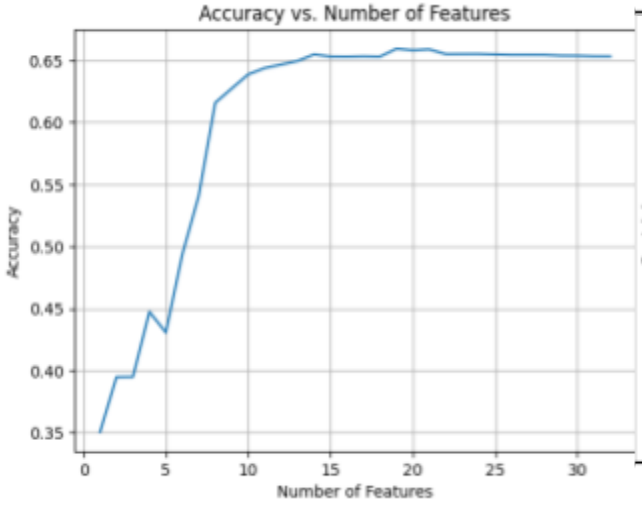


Figure 8: Confusion matrix for Naive Bayes.

Following this, we ran Naive Bayes classifier with incremental number of features



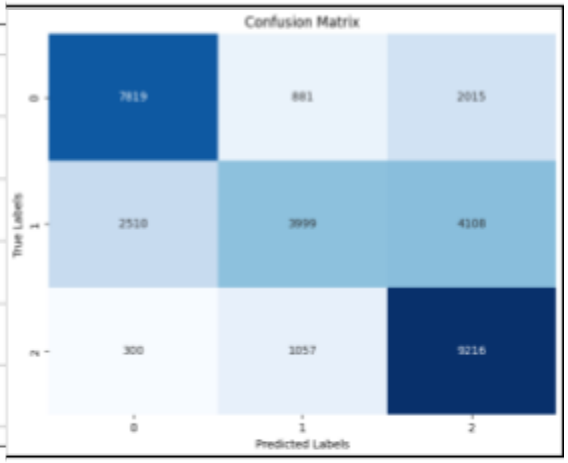


Figure 9 : Accuracy per feature for Naive Bayes. Figure 10 : Confusion matrix for Naive Bayes.

Class	Precision	Recall	F1-score	Support
Good	0.74	0.73	0.73	10715
Poor	0.67	0.38	0.48	10617
Standard	0.60	0.87	0.71	10573
Accuracy			0.66	31905

Macro Average	0.67	0.66	0.64	31905
Weighted Average	0.67	0.66	0.64	31905

Table 4 : Classification report with incremental features, Naive Bayes.

In conclusion, running Naive Bayes with incremental features has resulted in a slight accuracy increase. This slight increase could be attributed to adding features that are truly informative about the target variable and can provide the model with more data points to learn from. This allows the model to capture more complex relationships and make better predictions

Following are the results after running PCA.

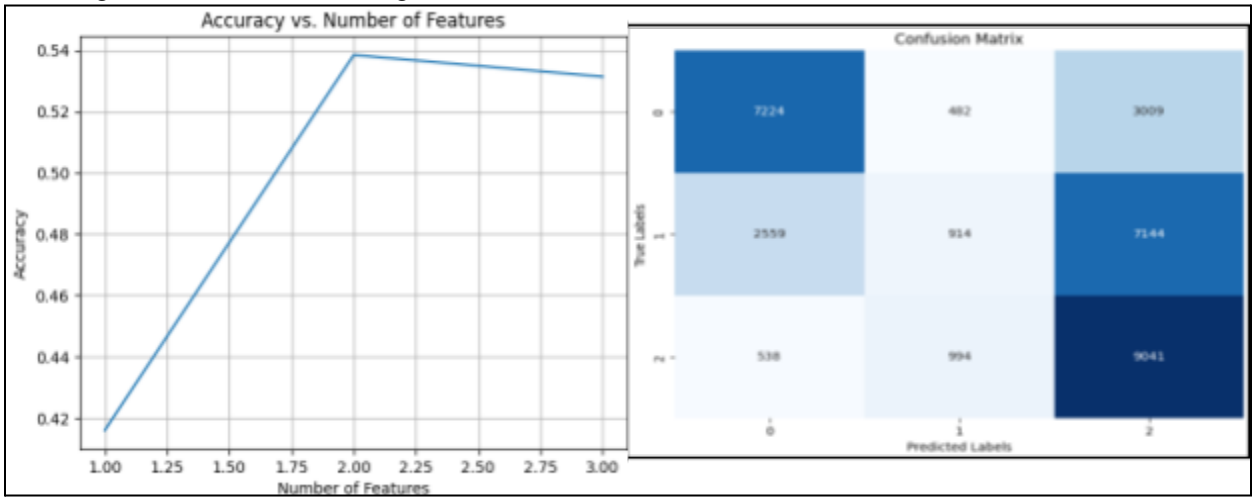


Figure 11: Accuracy per feature after PCA.

Figure 12: Confusion matrix after running PCA.

Class	Precision	Recall	F1-score	Support
Good	0.70	0.67	0.69	10715
Poor	0.38	0.09	0.14	10617
Standard	0.47	0.86	0.61	10573
Accuracy			0.54	31905
Macro Average	0.52	0.54	0.48	31905
Weighted Average	0.52	0.54	0.48	31905

Table 5 : Classification Report after running PCA (Naive Bayes)

Analyzing the classification report, we can say that applying PCA with Naive Bayes appears to be ineffectual in this case as the accuracy is 54%. It appears that PCA might have discarded informative features crucial for classification, especially for class poor. Additionally, Naive Bayes assumes feature independence, which might not be true for the

transformed features created by PCA. This could further prevent the model's ability to learn the relationships between variables and make accurate predictions.

Following are the results after running K-nearest neighbor (KNN):

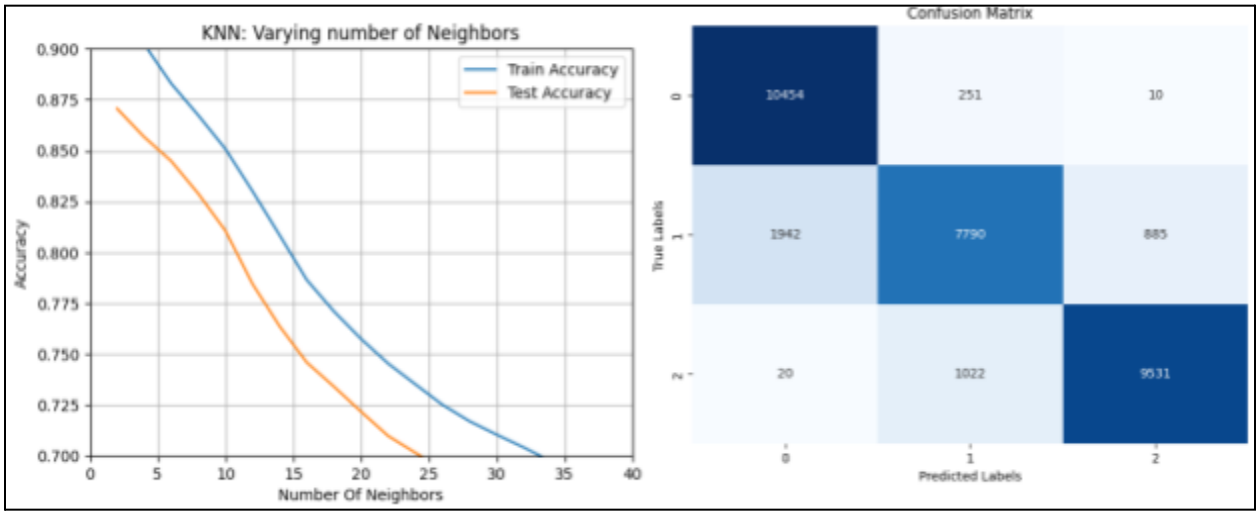


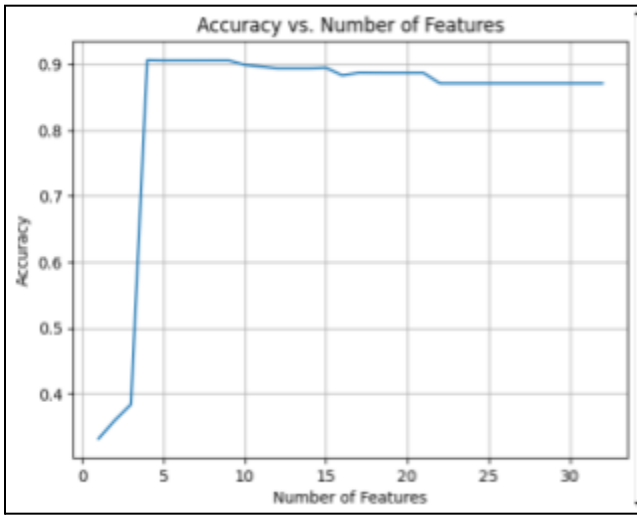
Figure 13: Accuracy with respect to number of neighbors Figure 14: Confusion Matrix of KNN

Class	Precision	Recall	F1-score	Support
Good	0.84	0.98	0.90	10715
Poor	0.86	0.73	0.79	10617
Standard	0.91	0.90	0.91	10573
Accuracy			0.87	31905
Macro Average	0.87	0.87	0.87	31905
Weighted Average	0.87	0.87	0.87	31905

Table 6 : Classification report after running KNN

Analyzing this table, we can conclude that the k-Nearest Neighbors (kNN) algorithm demonstrated strong performance on this classification task, achieving an overall accuracy of 0.87 along with high precision and recall across the 3 classes shown above. The model's success can be attributed to the balanced nature of the dataset across the three classes, high precision and recall values for each class, and consistent f1-scores. These metrics indicate that the model is able to effectively learn the patterns and make accurate predictions for each class, with minimal false positives and false negatives. The good performance of KNN suggests that the dataset has a structure where similar instances tend to belong to the same class, making it suitable for similarity-based classification.

Following are the results after running KNN for incremental features:



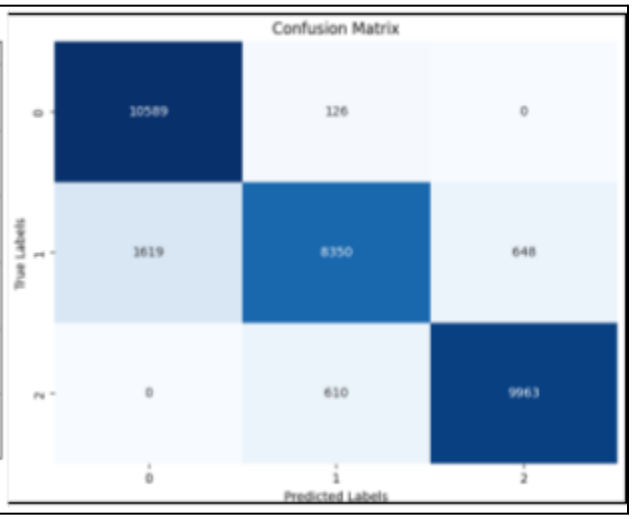


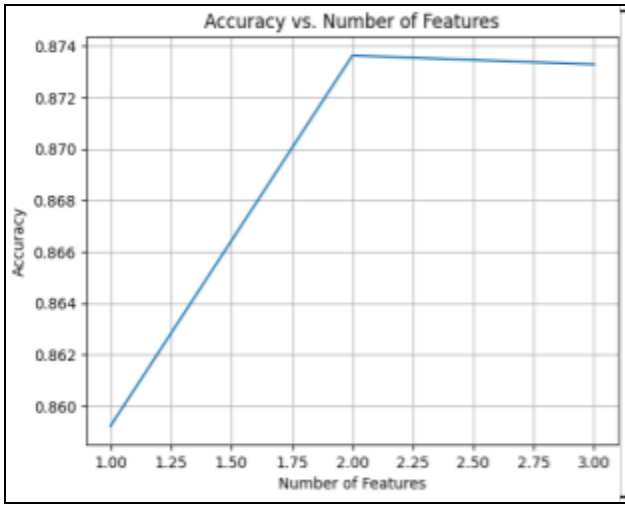
Figure 15: Accuracy with respect to number of features (KNN) Figure 16 : Confusion matrix for incremental features, (KNN)

Class	Precision	Recall	F1-score	Support
Good	0.87	0.99	0.92	10715
Poor	0.92	0.79	0.85	10617
Standard	0.94	0.94	0.94	10573
Accuracy			0.91	31905
Macro Average	0.91	0.91	0.90	31905
Weighted Average	0.91	0.91	0.90	31905

Table 6 : Classification report after running KNN with incremental features

Analyzing the table, we can say that the KNN model achieved a higher accuracy,precision and recall across the 3 classes, when we ran it with incremental features. The inclusion of incremental features has most likely provided the KNN with additional discriminative information, enabling it to better distinguish between the classes. The strong performance suggests that the selected features are highly relevant to the classification task at hand and that the model has successfully learned the complex relationships between these features and the target variable.

After applying PCA, here are the KNN results:



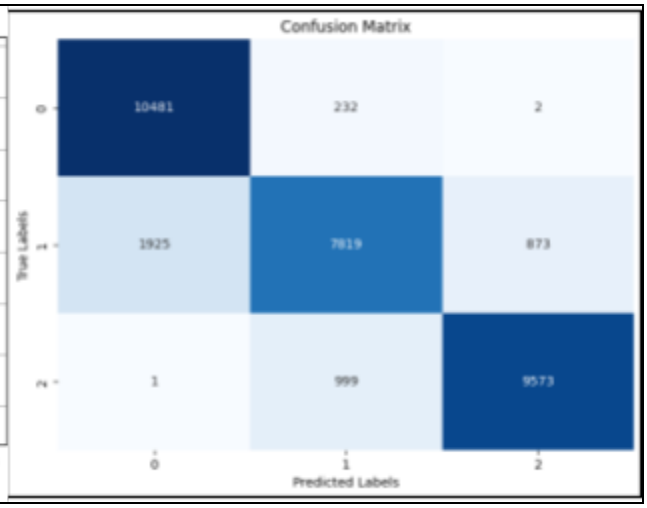


Figure 17: Accuracy per feature after applying PCA, (KNN) Figure 18: Accuracy per feature after applying PCA, (KNN)

Class	Precision	Recall	F1-score	Support
Good	0.84	0.98	0.91	10715
Poor	0.86	0.74	0.80	10617
Standard	0.92	0.91	0.91	10573
Accuracy			0.87	31905
Macro Average	0.87	0.87	0.87	31905
Weighted Average	0.87	0.87	0.87	31905

Table 7 : Classification report after applying PCA, (KNN)

When comparing the results to the report without PCA, we observe a slight decrease in performance across all metrics. The precision, recall, and f1-score for each class have marginally dropped. This dip in performance can be attributed to the information loss that occurs during the dimensionality reduction process with PCA. While PCA can be effective in removing noise and redundant features, it may also lead to some loss of discriminative information that is crucial for the classification task. As a result, the model's ability to distinguish between the classes may be slightly compromised.

Before sharing the SVM results, it's important to note that the model ran for over 15 hours without producing any output. This was due to the dataset's large size (100k samples). The reported SVM results are based on a subset of the data with only 10 selected features. Attempting to use all original features resulted in excessively long runtimes without any output.

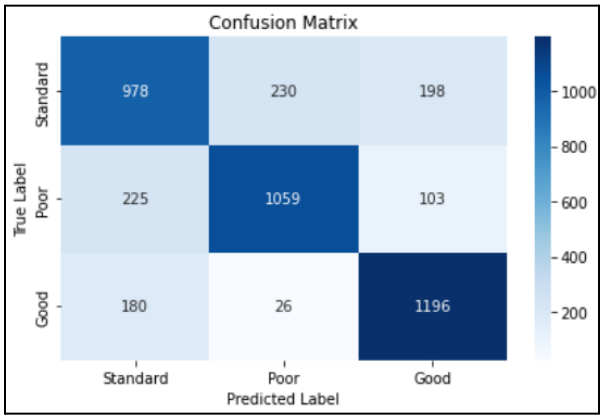
SVM Polynomial:

Class	Precision	Recall	F1-Score	Support
Good	0.81	0.78	0.79	10715

Poor	0.72	0.68	0.7	10617
Standard	0.78	0.85	0.82	10573
Accuracy			0.77	31905
Macro Avg	0.77	0.77	0.77	31905
Weighted Avg	0.77	0.77	0.77	31905

Table 8 : Classification report after running SVM with a POLY kernel

In conclusion, the SVM model with a polynomial kernel achieved an acceptable accuracy of 77%, demonstrating its ability to classify most data points correctly. While the F1-scores indicate a balanced performance across classes, there's room for improvement, particularly for Class 1 (precision: 0.72, recall: 0.68). This suggests the model is struggling to differentiate between Class 1 and other classes.

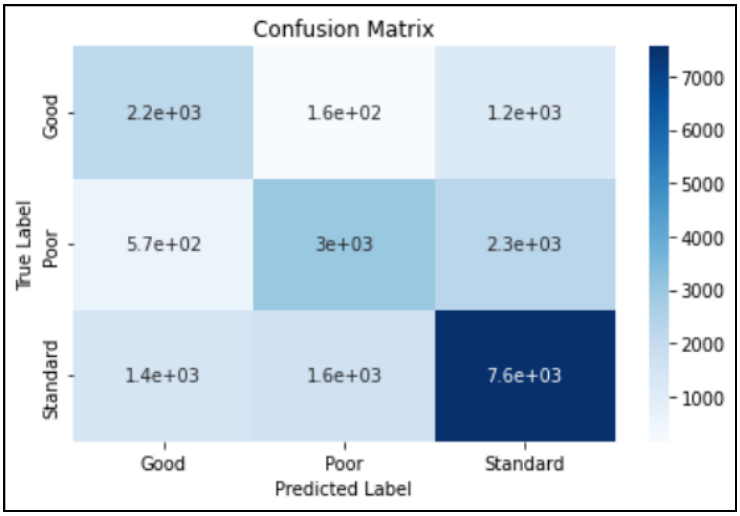


SVM Linear

Class	Precision	Recall	F1-Score	Support
Good	0.53	0.62	0.57	10715
Poor	0.63	0.51	0.56	10617
Standard	0.68	0.71	0.7	10573
Accuracy			0.64	31905
Macro Avg	0.61	0.62	0.61	31905

Weighted Avg	0.64	0.64	0.64	31905

In conclusion, the sentiment classification model achieved a moderate overall accuracy of 64%. While the model performs decently on the Standard class, it struggles with both Good and Poor sentiment, as evidenced by the lower F1-scores in these categories.



Conclusion:

In conclusion, our study explored various machine learning techniques and data preprocessing methods for credit score classification. We evaluated the performance of decision tree classifiers, k-Nearest Neighbors (kNN), Naive Bayes and SVM with linear and Polynomial kernels and dimensionality reduction using Principal Component Analysis (PCA). The decision tree classifier demonstrated varying performance across different classes, with Class 2 consistently showing the highest precision, recall, and F1-score. The overall accuracy ranged from 63% to 69%, indicating decent performance but with room for improvement. The model trained on all features (both numeric and categorical) achieved the highest accuracy of 69%, highlighting the importance of incorporating diverse feature types. The kNN algorithm exhibited strong performance, achieving an overall accuracy of 87%. The model's success can be attributed to the balanced dataset, high precision and recall values, and consistent F1-scores across all classes. The good performance of kNN suggests that the dataset has a structure where similar instances tend to belong to the same class, making it suitable for similarity-based classification. Furthermore, we explored the impact of incremental features on the classification performance. The inclusion of incremental features resulted in an impressive overall accuracy of 91%, indicating that these additional features provided valuable discriminative information for distinguishing between classes. Also, we applied PCA for dimensionality reduction in combination with the kNN classifier. The results showed a slight decrease in performance compared to the models without PCA, with an overall accuracy of 87%. This dip can be attributed to the information loss during the dimensionality reduction process. However, the model still maintained a satisfactory level of accuracy and performance across all classes. Finally, the SVM model with a polynomial kernel achieved an acceptable accuracy of 77%, demonstrating its ability to classify most data points correctly. While the F1-scores indicate a balanced performance across classes, there's room for improvement, particularly for Class poor (precision: 0.72, recall: 0.68). This suggests the model is struggling to differentiate between Class poor and other classes. For the SVM model with a Linear Kernel, the model achieved a moderate overall accuracy of 64% and the model performs decently on the Standard class but it struggles with both Good and Poor class, as evidenced by the lower F1-scores in these categories. Integrating these findings, our study offers a comprehensive analysis of various machine learning approaches for credit score classification, highlighting their strengths and areas for further refinement.

Contributions:

Koushal: Red text
Khalid: yellow highlighter
Omar: Blue text
Almarzooqi: green highlighter

References

[1]T. Diwakar et al, "Experimental analysis of machine learning methods for credit score classification," Progress in Artificial Intelligence, vol. 10, (3), pp. 217-243, 2021. Available: http://aus.idm.oclc.org/login?url=https://www.proquest.com/scholarly-journals/experimental-analysis-machine-learning-methods/docview/2560320782/se-2. DOI: https://doi.org/10.1007/s13748-021-00238-2.

[2]A. R. Provenzano *et al.*, "Machine Learning approach for Credit Scoring," *arXiv.org*, Jul. 20, 2020. https://arxiv.org/abs/2008.01687

[3] B. Dushimimana, Y. Wambui, T. Lubega, and P. E. McSharry, "Use of Machine Learning Techniques to Create a Credit Score Model for Airtime Loans," *Journal of Risk and Financial Management*, vol. 13, no. 8, p. 180, Aug. 2020, doi: https://doi.org/10.3390/jrfm13080180.

[4] Y. Wu and Y. Pan, "Application Analysis of Credit Scoring of Financial Institutions Based on Machine Learning Model," *Complexity*, vol. 2021, pp. 1–12, Oct. 2021, doi: https://doi.org/10.1155/2021/9222617.

[5] A. Safiya Parvin and B. Saleena, "An ensemble classifier model to predict credit scoring - comparative analysis," 2020 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS), Dec. 2020. doi:10.1109/ises50453.2020.00017.

[6] A. Aljadani et al, "Mathematical Modeling and Analysis of Credit Scoring Using the LIME Explainer: A Comprehensive Approach," Mathematics, vol. 11, (19), pp. 4055, 2023. Available: http://aus.idm.oclc.org/login?url=https://www.proquest.com/scholarly-journals/mathematical-modeling-analysis-credit-scoring/docview/2876580075/se-2. DOI: https://doi.org/10.3390/math11194055.

[7] V. Kuppili, D. Tripathi, and D. Reddy Edla, "Credit score classification using spiking extreme learning machine," *Computational Intelligence*, vol. 36, no. 2, pp. 402–426, Nov. 2019. doi:10.1111/coin.12242

[8] F. Yang, Y. Qiao, Y. Qi, J. Bo, and X. Wang, "Bacs: Blockchain and AutoML-based technology for efficient credit scoring classification," *Annals of Operations Research*, Jan. 2022. doi:10.1007/s10479-022-04531-8.

[9]E. Dumitrescu, S. Hué, C. Hurlin, and S. Tokpavi, "Machine Learning for Credit Scoring: Improving Logistic Regression with Non-Linear Decision-Tree Effects," *European Journal of Operational Research*, vol. 297, no. 3, Jun. 2021, doi: https://doi.org/10.1016/j.ejor.2021.06.053.

[10] I. Singh, N. Mishra, A. Joshi, and N. Agarwal, "An approach for credit-scoring using swarm intelligence for feature selection and PSO trained Ann Based Classification," *2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN)*, May 2023. doi:10.1109/vitecon58111.2023.10157909

[11] Abhishek V A, Binny S, Johan T R, Nithin Raj, and Vishal Thomas, "Federated learning: Collaborative machine learning without centralized training data," *international journal of engineering technology and management sciences*, pp. 355–359, Sep. 2022. doi:10.46647/ijetms.2022.v06i05.052

[12] D. Balakrishnan *et al.*, "Credit score prediction using support vector machine and Gray Wolf Optimization," *2023 3rd International Conference on Intelligent Technologies (CONIT)*, Jun. 2023. doi:10.1109/conit59222.2023.10205673

[13] A. Maurya and S. Gaur, “A decision tree classifier based ensemble approach to Credit Score Classification,” *2023 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, Nov. 2023. doi:10.1109/icccis60361.2023.10425039

[14] S. K. Trivedi, “A study on credit scoring modeling with different feature selection and machine learning approaches,” *Technology in Society*, vol. 63, p. 101413, Nov. 2020. doi:10.1016/j.techsoc.2020.101413

[15] A. C. Bueff *et al.*, “Machine learning interpretability for a stress scenario generation in credit scoring based on Counterfactuals,” *Expert Systems with Applications*, vol. 202, p. 117271, Sep. 2022. doi:10.1016/j.eswa.2022.117271

[16] V. Moscato, A. Picariello, and G. Sperlí, “A benchmark of machine learning approaches for Credit Score Prediction,” *Expert Systems with Applications*, vol. 165, p. 113986, Mar. 2021. doi:10.1016/j.eswa.2020.113986

[17] H.-W. Teng, M.-H. Kang, I.-H. Lee, and L.-C. Bai, “Bridging accuracy and interpretability: A rescaled cluster-then-predict approach for enhanced credit scoring,” *International Review of Financial Analysis*, vol. 91, p. 103005, Jan. 2024. doi:10.1016/j.irfa.2023.103005

[18] B. Chao and H. Guang Qiu, "Air pollution concentration fuzzy evaluation based on evidence theory and the K-nearest neighbor algorithm," Frontiers in Environmental Science, 2024.Available:http://aus.idm.oclc.org/login?url=https://www.proquest.com/scholarly-journals/air-pollution-concentration-fuzzy-evaluation/docview/3028034955/se-2. DOI: https://doi.org/10.3389/fenvs.2024.1243962.

[19] V. Lemaire, F. Clérot and M. Boullé, "An Efficient Shapley Value Computation for the Naive Bayes Classifier," ArXiv.Org, 2023. Available: http://aus.idm.oclc.org/login?url=https://www.proquest.com/working-papers/efficient-shapley-value-computation-naive-bayes/docview/2844446380/se-2.

[20] S. Li et al, "Hybrid Method with Parallel-Factor Theory, a Support Vector Machine, and Particle Filter Optimization for Intelligent Machinery Failure Identification," Machines, vol. 11,