An Efficient Machine Learning Based Big Data Analytics Approach for Business Decision Support System

Rajni, Dr. S. Senthil Kumar

Research Scholar,Dept of Computer Science and Engineering,
NIMS Institute of Engineering and Technology (NIET),NIMS University, Rajasthan, Jaipur
Email: jhajhariarajni18@gmail.com

Associate Professor,Dept of Computer Science and Engineering,
NIMS Institute of Engineering and Technology (NIET),NIMS University, Rajasthan, Jaipur
Email: *senthil.kumar@nimsuniversity.org*

Abstract: in today's data-rich business landscape, effective decision-making hinges on robust decision support systems (DSS). This paper presents an innovative fusion of machine learning and big data analytics aimed at fortifying DSS for enhanced business intelligence. The framework meticulously integrates sophisticated data preprocessing techniques to ensure data integrity and relevance. Harnessing the capabilities of diverse machine learning algorithms, including neural networks, decision trees, and ensemble methods, this approach extracts intricate patterns and valuable insights from voluminous and diverse datasets. Its adaptability to dynamic business contexts facilitates continuous learning, ensuring the system's relevance and accuracy over time. Case studies spanning diverse business domains affirm the system's superior performance, showcasing its capacity to empower organizations with actionable insights for informed decision-making in today's rapidly evolving markets. The proposed framework's scalability and agility position it as a valuable asset for businesses seeking to navigate the complexities of contemporary data ecosystems. Through comprehensive experimentation and validation, its efficacy in driving accurate, timely, and informed decisions is underscored, highlighting its potential to not only optimize current business strategies but also pave the way for adaptive and competitive decision-making frameworks in the future.

Keywords: Cloud Computing, BSS, Big Data, Decision Tree

DOI: 10.53555/ecb/2022.11.12.255

1. INTRODUCTION

In the dynamic landscape of modern business operations, the proliferation of data has transformed the traditional paradigms of decision-making. The evolution of big data has heralded both unprecedented opportunities and challenges, demanding innovative strategies to harness its potential for informed and agile decision support. Amidst this backdrop, the fusion of machine learning techniques with big data analytics has emerged as a transformative approach, promising a paradigm shift in bolstering business decision support systems (DSS). This integration not only addresses the escalating complexities of large-scale data but also endeavors to empower organizations with actionable insights, fostering competitiveness and adaptability in an ever-evolving market landscape.

The amalgamation of machine learning and big data analytics represents a pivotal evolution in the realm of decision support systems, seeking to harness the vast reservoirs of data generated by contemporary businesses. This approach embodies a systematic framework designed to navigate the intricacies of data processing, employing sophisticated techniques for data pre-processing, integration, and analysis. By leveraging the prowess of diverse machine learning algorithms, this approach aims to distill meaningful patterns and predictive models from colossal datasets, thereby augmenting the decision-making capabilities of businesses across diverse domains.

Furthermore, the synergy between machine learning and big data analytics not only amplifies the scope of data analysis but also facilitates adaptive decision-making frameworks. The inherent adaptability of these systems to evolving data dynamics enables continual learning and refinement, ensuring the relevance and accuracy of decision support systems in the face of ever-changing business landscapes. This introduction sets the stage for exploring the intricacies and transformative potential of an efficient machine learning-based big data analytics approach as a cornerstone for robust business decision support systems.

Because of this, the P2P model simplifies the borrowing and loaning process while still providing economic benefits to all involved. While the P2P model streamlines the lending process and provides greater financial profits for both lenders and mortgagors, it also carries a higher risk of loan defaults than the traditional bank loan procedure. Because of the time savings from not having to manually verify and compare information, it is crucial to find ways to automate the creation of a reliable credit portfolio that lenders can use to determine whether or not to approve a loan application. Developing a comprehensive model for assessing credit risks is challenging due to the large number of quantitative and qualitative aspects that must be taken into account. Now that ML has advanced, numerous ML-based designs have been explored to produce accurate credit scores. A machine learning simulation for determining credit risk is presented here, making use of decision trees and the K-Nearest Neighbor technique.

A comprehensive evaluation of credit risk is necessary for identifying defaulters in the P2P lending model. This project involved the use of a machine learning model to Analyzing how the probability of default payment varies across different demographic variables. The ensemble methods proposed for evaluating credit

risks. This research made use of the credit dataset available through the University of California Irvine Machine Learning Repository (UCIMLR). This dataset includes thirty thousand samples, each of which includes twenty-four features used to approve or deny a credit application. The efficiency of the proposed system can be gauged by looking at how well it classifies financial data for purposes of calculating credit risk. The proposed models were shown to achieve an accuracy of 82.2% for the decision tree model and 81.2% for the K-Nearest Neighbor model on the selected credit data set.

2. BACKGROUND

In delving into related research, a foundational understanding of credit risk assessment terminology is essential. In the realm of financial services provided by banks and credit unions, the complexity arises from dealing with two primary risk categories:

1. Credit Risk Exposure: This pertains to the risk associated with the potential non-repayment, either in full or in part, of a loan by the borrower.

2. Market-related Possibilities: These threats emanate from the inherent volatility in the market values of commodities, securities, and services, often characterized by unpredictable actions.

Operational risks, stemming from the unpredictability of market forces and external factors, add another layer of challenge. Therefore, institutions engaged in credit finance, mortgages, venture capital, fundraising, etc., necessitate robust credit risk evaluation models. The growing integration of software robots aims to streamline processes, reducing human effort and time.

A high-level summary of research in credit risk assessment models using machine learning follows. Chen et al. assert that risk assessment has gained substantial momentum in the finance field [1]. The aftermath of the 2008 financial crisis has intensified the demand for reliable algorithms predicting business failures. Crook et al. delve into consumer risk assessment, considering factors like the client's ability to pay and timeliness in payments [2]. Galindo, Tamayo, and colleagues emphasize the importance of selecting appropriate predictors in financial risk models, proposing a model based on error curve research [3].

Twala advocates for machine learning in credit risk assessment, utilizing an ensemble classifier to categorize customers despite attribute noise levels [5]. Doumpos et al. highlight the issues of estimating profit/loss and calculating the probability of default in risk assessment [7]. Saha et al. propose a data-driven strategy for loan approval in California, incorporating both data mining and expert opinion [8].

Cai et al. (2020) leverage a blockchain platform for credit risk assessment, while Zhou et al. (2019) emphasize the efficiency gained through large clusters for distributed implementation [11][12]. Wang et al. (2018) present price forecasting models, and Deng et al. (2016) propose k-means clustering for dataset segmentation [14][15].

The study concludes by acknowledging the potential for further exploration, especially in the realm of peer-to-peer lending models utilizing machine learning. Table 5-1 succinctly encapsulates seminal works, highlighting their contributions and limitations. The authors underscore the need for continued research to unearth novel insights, enhancing the accuracy of borrower classification through machine learning methodologies.

3. THE PROPOSED METHODOLOGY

Credit risk assessment using statistical models is challenging because of the large number of attributes, both numerical and otherwise, that must be analysed. Machine Learning (ML) based models have become widely used for evaluating credit risks because of their ability to analyse large and complex data sets in time-critical applications. While the mentioned groups represent the broadest classifications of machine learning models, a wide variety of other groups and variants exist. The complete process for analysing credit data with the Machine Learning algorithm is depicted in Figure 1.

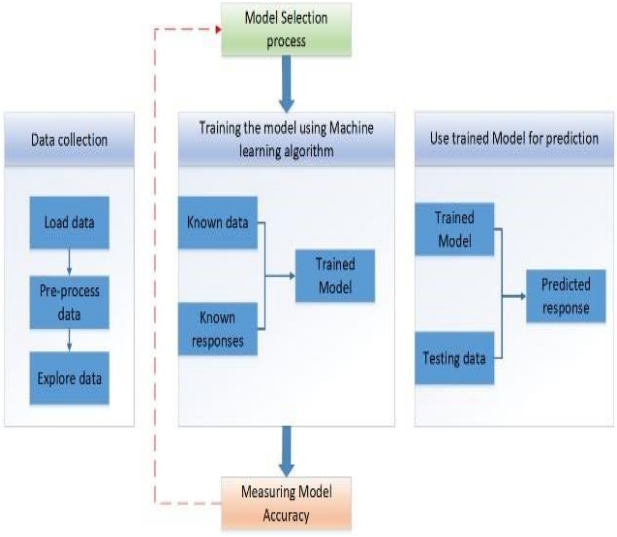


Fig. 1. Proposed Workflow

Method for implementing the three facets of the proposed setup Obtaining Data Is Step One. Step two: put the model through its paces with some machine learning training. Third, repeatedly apply the trained model to forecasting.

A. Classification of Borrowers using Binary Partitioning and Decision Trees

Decision Trees (DTs) are an example of a non-parametric learning method that takes its learning cues from the underlying decision rules of the training dataset. If the expected value of the outcome (or target) variable depends on several other (governing) variables, the tree draws probabilistic inferences about the outcome of events. The decision tree can be obtained by breaking down the initial data set (the root node) into smaller subsets (the child nodes). The practise of repeatedly subdividing a tree into smaller and smaller pieces is commonly known as "recursive partitioning." When some subset of nodes shares the target variable's value, recursive partitioning stops. Decision trees can be divided

into two major categories, classification trees and regression trees, depending on the type of data being analysed. Due to the necessity of making a difficult decision at each step of the partition, the implementation of decision trees is challenging because of the inherent error-proneness of the two-category recursive partitioning (often a greedy algorithm approach). Furthermore, the time complexity of the algorithm increases exponentially with the size of the training set and the number of dimensions. The training time complexity formula is O(n2+log2+d2) (1)

Here, we employ the notation for "number of dimensions" (d) and "number of data points" (n).

An example of a decision tree is shown in Figure 2.

Steps followed for credit risk assessment using Decision Tree:

Step 1: Input: Extract data set credit risk dataset.

Step 2: Apply and calculate probability of inaccurate classifications based on the Gini's Index

Step 3: The continued or recursive splitting generates binary trees at every decision node and the final Gini Index is computed as the weighted sum of all the individual splits.

An alternative partitioning approach is the partitioning based on the average information content of a random variable often termed as entropy

Step 4: The information gain can also be used as a metric for splitting the tree.

Step 5: Calculate Decision Tree's accuracy for credit data, as two identified class labels are Correct and Incorrect.

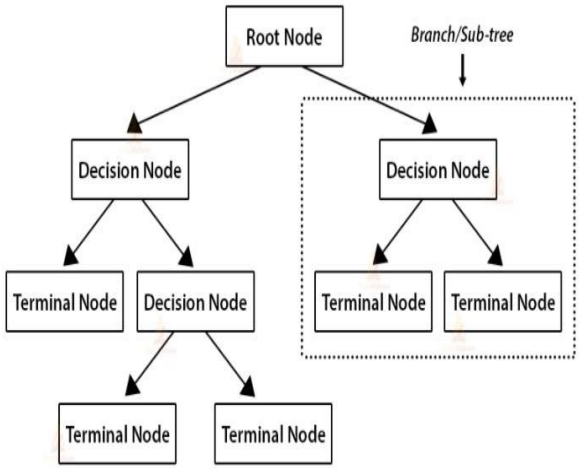


Fig. 2. A Typical Decision Tree Model

Gini's impurity function or Shannon's entropy can be used to determine the splitting condition, but the impurity function is preferred due to its reduced computational complexity. Deeper trees are more likely to overfit the model, while shallower trees are more likely to under-fit. Therefore, Hyper-parameter tuning is required for a perfect tree to emerge.

B. Classification of Borrowers using K-Nearest Neighbour (KNN)

The K-Nearest Neighbour (KNN) is a method to perform non-parametric classification, whose output is typically a class membership. The KNN approach tries to classify a given sample $'\boldsymbol{XX}'$ into a class $'\boldsymbol{CC}'$ based on the Euclidean Distance $'\boldsymbol{dd}'$ given by:

$$dd = \sqrt{f_1^2 + \cdots f_n^2} \quad \ldots\ldots(1)$$

Here, $\boldsymbol{dd}$ denotes the Euclidean Distance

$\boldsymbol{ff_{11}} \ldots \boldsymbol{ff_{nn}}$ denotes the data attributes or features.

A typical illustration of a two-class KNN is depicted in figure 3.

The KNN computes the Euclidean Distance ($\boldsymbol{dd}$) of the data sample $'\boldsymbol{XX}'$ from all the classes. The minim Euclidean distance governs the decision regarding a new data sample being classified into any class $'\boldsymbol{CC}'$.

$$yy = \min(dd_1, dd_2 \ldots \ldots dd_{nn}) \quad \ldots(2)$$

Thus for $'\boldsymbol{nn}'$ distinct classes, the nearest neighbour based on weighted Euclidean distance can be computed as:

$$dd_{ww} = \sqrt{\sum_{i=1}^{n} w_i(C_i - \sigma_i)} \quad \ldots.(4)$$

$\boldsymbol{dd_{ww}}$ denotes the weighted Euclidean Distance

$\boldsymbol{ww_{ii}}$ denotes the weight for element $'\boldsymbol{ii}'$

$\boldsymbol{CC_{ii}}$ denotes the component of $\boldsymbol{ii^{tttt}}$ feature vector.

$\boldsymbol{\sigma\sigma_{ii}}$ denotes the standard deviation of the $\boldsymbol{ii^{tttt}}$ feature vector.

Here, $\boldsymbol{yy}$ denotes the minimum Euclidean distance for the given data class. $\boldsymbol{dd_{11}}$, $\boldsymbol{dd_{22}}$ ... ... $\boldsymbol{dd_{nn}}$ denote the individual Euclidean Distances.

The weighted version of the KNN is obtained by assigning a weight to the k-nearest neighbour members. Mathematically, the weight assigned to an $\boldsymbol{ii^{tttt}}$ the nearest neighbour is given $\boldsymbol{ww_{nn,ii}}$ with the property of:

$$\sum^{nn}{}_{ii=1} ww_{nn,1} = 1 \quad \ldots(3)$$

The objective of the multi-class classification is to attain convergence of error rate for the classifier for $'\boldsymbol{nn}'$ distinct classes.

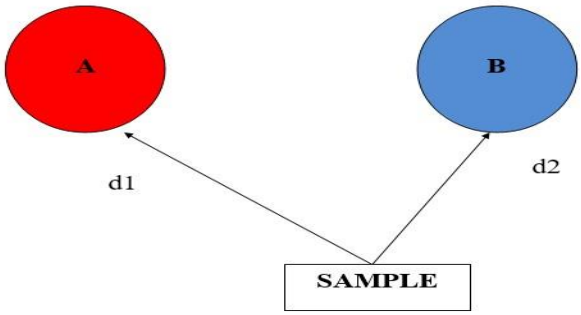


Fig. 3. Working of K-Nearest Neighbour

Steps followed to use KNN for credit assessment:

Step 1: Extract data set for training iterations.

Step 2: Split data set into training and testing vectors in the ratio of 75:25.

Step 3: Initialize weights (***ww***) randomly.

Step 4: Update weights using gradient descent to minimize the objective function J given by:

$$mm \quad = -\sum_{i=1}^{m}(v_i - v'_i)^2 \quad JJ \qquad \ldots(5)$$

Step.5: Compute the error matrix (cost function).

Step.6: Iterate steps (1-4) till the cost function ***JJ*** stabilizes.

C. Environment Setup and Data Set Details

To conduct their research, the authors consulted the credit risk dataset [11] available at UCI's Machine Learning Repository (UCIMLR). A dataset containing 30,000 approved and declined credit applications based on 24 attributes or features is used in the proposed work. This dataset encompasses details related to default payments, incorporating demographic factors, credit data, payment history, and credit card bill statements of clients in Taiwan during the period from April 2005 to September 2005. The dataset comprises 25 variables, including client IDs, credit limits in NT dollars, gender, education level, marital status, age, and repayment statuses across six months. The repayment statuses are denoted on a scale, ranging from timely payments to delayed payments.

Data Attributes:

1. ID: ID of each client
2. LIMIT_BAL: Amount of given credit in NT dollars (individual and family/supplementary credit)
3. SEX: Gender (1=male, 2=female)
4. EDUCATION: Education level (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
5. MARRIAGE: Marital status (1=married, 2=single, 3=others)
6. AGE: Age in years

7-13. PAY_0 to PAY_6: Repayment status from September 2005 to April 2005 (-1=pay duly, 1=payment delay for one month, ..., 9=payment delay for nine months and above)

14-19. BILL_AMT1 to BILL_AMT6: Amount of bill statement from September 2005 to April 2005 (NT dollar)

20-25. PAY_AMT1 to PAY_AMT6: Amount of previous payment from September 2005 to April 2005 (NT dollar)

26. default.payment.next.month: Default payment (1=yes, 0=no)

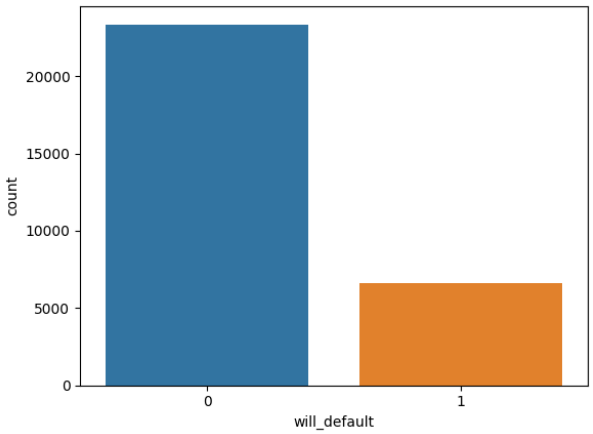


Fig. 4. Label Graph

Data is non equally distributed (figure 4). Non defaulters are much more in number (figure 5) than defaulters Will make it equally distributed using SMOTE.

D. Descriptive Analysis (Exploratory Data Analysis)

Usage of Revolving Credit: The dataset reveals that, irrespective of being a defaulter or non-defaulter, a significant number of users are engaged with revolving credit services.

Defaulters with Revolving Credit: Notably, even among defaulters, a substantial number of individuals utilize revolving credit services.

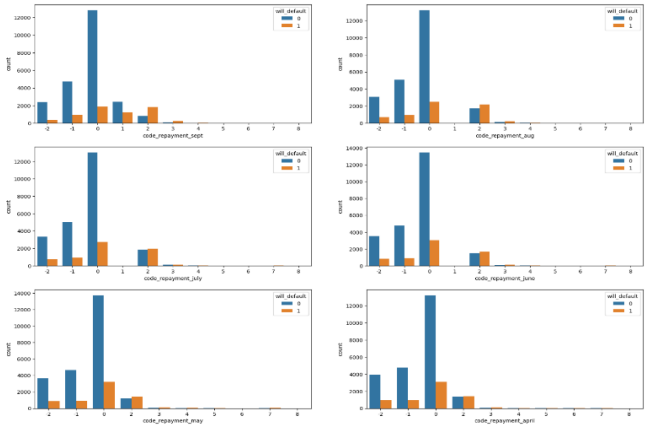


Fig. 5. Customer Payment Methods

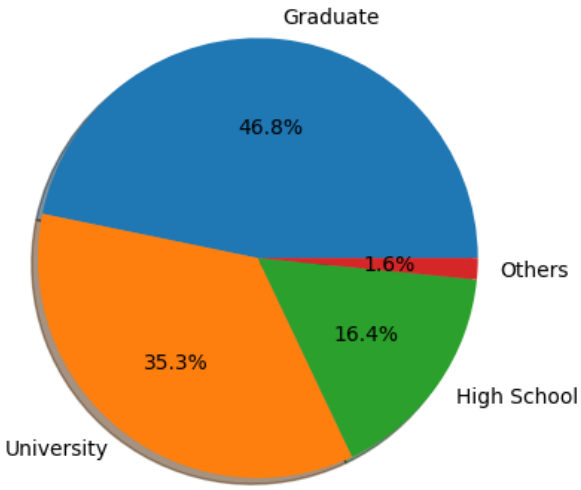


Fig. 6. Education Distribution of Customer

Rare Extended Delays: There are very few instances where users have delayed their payments for four months or more. This suggests that a vast majority of users tend to make timely payments

Marital Status Influence: It's observed that, in general, married users tend to utilize credit services more frequently than single users.

Marriage and Education Impact: Users belonging to the married category and having a graduate education background are notably more inclined to use credit services compared to other combinations of marital status and education levels. (figure 6)

Default Patterns: Users who do not have a graduate, university, or high school education tend to exhibit a higher default rate of approximately 30%-40%, regardless of their marital status.

Risk in Graduate Users with 'Other' Category: Graduate users categorized as 'Other' demonstrate a substantial 50% chance of defaulting on their credit card payments.

Interpretations Bill Amount Skewness: Across all months, the bill amounts exhibit a high degree of skewness. This indicates that the distributions are significantly asymmetric, which can impact the modeling and analysis process.

Presence of Negative Bill Values: Notably, some of the bill amounts contain negative values, signifying credit balances or overpayments, which should be carefully considered when analyzing financial behavior.

Almost 50% data covers Graduate section

Interpretations Bill Amount Skewness: Across all months, the bill amounts exhibit a high degree of skewness. This indicates that the distributions are significantly asymmetric, which can impact the modeling and analysis process.

Presence of Negative Bill Values: Notably, some of the bill amounts contain negative values, signifying credit balances or overpayments, which should be carefully considered when analyzing financial behavior

Highly Skewed Limited Distribution as shown in Figure

Fig 8 shows that limited balance feature shows positive relation with bill statement in all the months and marriage and age shows highly negative correlation.

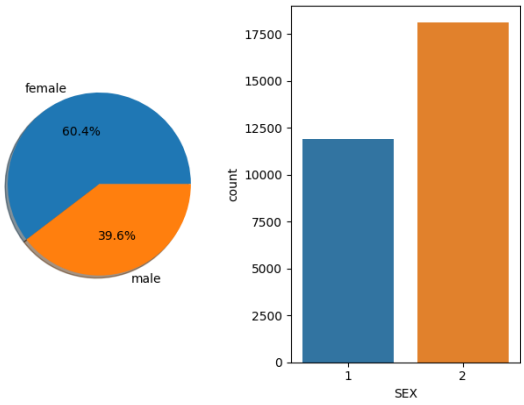


Fig. 7. Sex Distribution in the Data

E. Data Pre-processing and Feature Selection

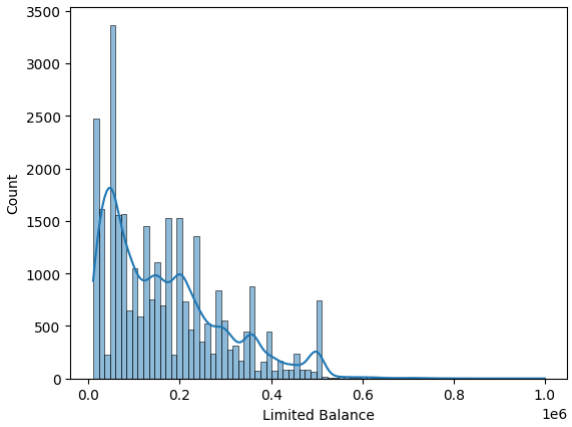
Streamlining and bettering the training process is possible with the help of data pre-processing. Averages are used to make up for gaps in data when necessary. Because they are irrelevant to recognising patterns, the ID numbers have been removed. Numeric values have been assigned to categorical characteristics like marital status.

The dataset is devoid of any null values. Bill amounts exhibit a wide range from 2,000 to 800,000 units.

To handle the missing or unknown values, we will begin by examining the value counts of these features. Subsequently, we will categorize and label them as 'Others' for effective data management and analysis.

Outliers Consideration: Many outliers are present in the dataset, and these outliers may hold valuable information for our model.

Information Retention: Removing these outliers may result in the loss of valuable information, and careful



consideration is required when deciding how to handle them in the modeling process.

Fig. 8. Limited Balance Distribution

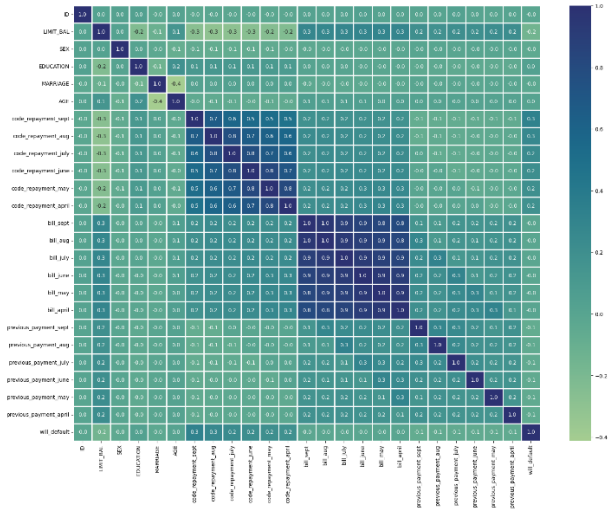


Fig. 9. Feature Correlation Matrix

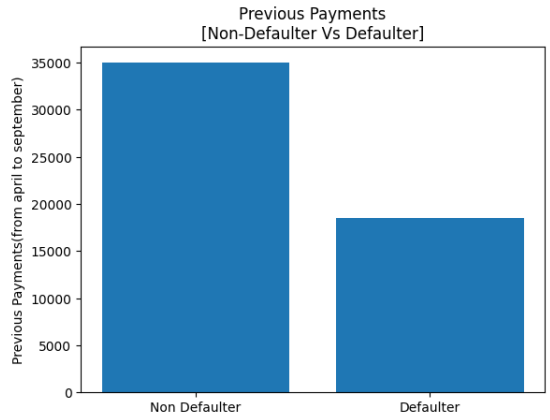


Fig. 10. Previous Payment Defaulter Distribution

F. Feature Engineering

Managing Model Complexity: The inclusion of a multitude of columns representing monthly bills poses a challenge in terms of model complexity. Simplifying the feature space becomes crucial to facilitate efficient model development.

Handling Continuous Features: The continuous nature of both bills and previous payments suggests an opportunity to enhance efficiency by consolidating them into a unified feature. This consolidation aims to reduce dimensionality while preserving valuable information, contributing to the creation of more streamlined and interpretable models.

Caution with Categorical Features: Notably, the 'payment_code' feature is categorical, necessitating a cautious approach in feature engineering. Applying the technique requires a specialized strategy to maintain the categorical meaning of this feature.

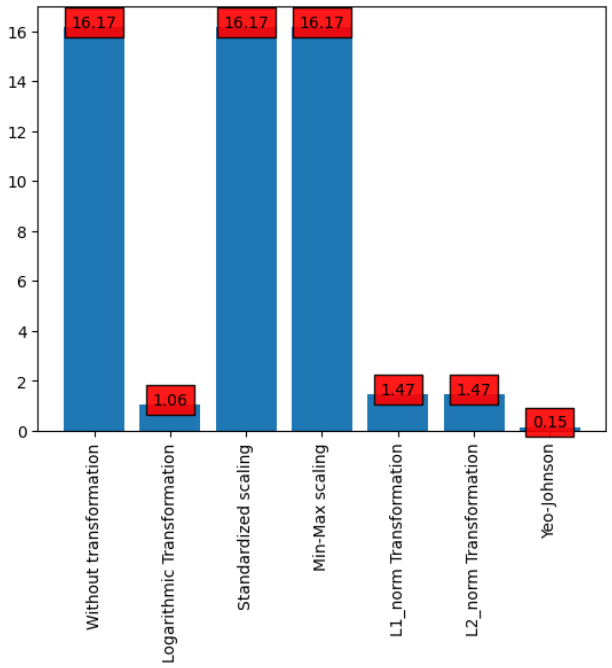


Fig. 11. Different types of Feature Scaling

G. Feature Scaling

As shown in figure 11 least skewness is in The Johnson feature scaling method effectively minimizes skewness, ensuring a distribution with very low skewness. Additionally, applying a logarithmic transformation contributes to reducing skewness further, resulting in a skewness value of 1.06. It's imperative to verify that there are no zero values in the feature when employing the logarithmic scaling technique to maintain its effectiveness and prevent mathematical inconsistencies. Yeo Johnson transformation promised the best normal distribution plot than others with skewness 0.15. In some case algorithm works fine without feature scaling, forcefully transforming into normal distribution can impact the accuracy of the model very much. We have trained model on both the dataset scaled or non-scaled

H. Data Partitioning

One of the most common data splitting ranges, according to the mentioned research, is 80:20. Seventy-five percent is used in the classroom, while the remaining twenty-five percent is stored for comparison with exam results. The 80:20 rule is often used as a rough guideline for the division of large datasets.

I. Performance Evaluation

Here, we implemented Decision Tree Classifier(DTC), Random Forest Classifier (RFC), Adaboost Classifier (ABC), Xegaboost Classifier (XBC), K-Neighbors Classifier (KNC), Logistic Regression (LRC) and Support Vector Machine (SVM) strategies for assessing credit-related dangers. All strategies have been compared to one another in terms of the accuracy with which they classify data in table I on unscaled data. It is also shown that the confusion matrix can be used to calculate the classification accuracy. What follows is a

discussion of the similarities and differences between the outcomes of these two approaches.

4. PERFORMANCE EVALUATION OF MODELS

The table I presents the performance metrics of various machine learning algorithms on a given dataset. It lists the names of the machine learning algorithms used in the analysis. Accuracy (Train) represents the accuracy of each algorithm on the training dataset. This indicates how well the algorithm predicts the target variable on the data it was trained on. Accuracy (Test) indicates the accuracy of each algorithm on a separate test dataset. This dataset is not used during the training phase, and the accuracy on this set gauges how well the algorithm generalizes to new, unseen data. Test Accuracy (MSE) provides the Mean Squared Error (MSE) for the test dataset. MSE is a measure of the average squared difference between the predicted and actual values. A lower MSE indicates better performance.

TABLE I. DIFFERENT CLASSIFIERS PERFORMANCE ON SMOTE BALANCED BUT UNSCALED DATA

Algorithm	Accuracy (Train)	Accuracy (Test)	Test Accuracy (MSE)
Support Vector Machine	61.94%	61.23%	0.3877
Decision Tree Classifier	99.94%	74.13%	0.2587
Adaboost Classifier	75.47%	75.35%	0.2465
Random Forest Classifier	99.94%	81.29%	0.1871
K-Neighbors Classifier	77.10%	65.25%	0.3475
Logistic Regression	56.08%	55.40%	0.4460
XGB Classifier	85.82%	80.42%	0.1958

Interpretation of the result:

- Decision Tree Classifier: Achieves very high accuracy on the training set (99.94%), but the accuracy drops on the test set (74.13%), indicating potential overfitting. The MSE is 25.87%.

- Random Forest Classifier: Similar to the Decision Tree, high accuracy on training (99.94%), but with a better generalization to the test set (81.29%) and a lower MSE (18.71%).

- Adaboost Classifier: Moderate accuracy on both training (75.47%) and test sets (75.35%) with an MSE of 24.65%.

- XGB Classifier: Shows good performance with high accuracy on both training (85.82%) and test sets (80.42%), and a relatively low MSE (19.58%).

- K-Neighbors Classifier: Achieves moderate accuracy on both training (77.10%) and test sets (65.25%) with an MSE of 34.75%.

- Logistic Regression: Performs less optimally with lower accuracy on both training (56.08%) and test sets (55.40%), and a higher MSE (44.60%).

- Support Vector Machine (SVM): Shows a moderate accuracy on the training set (61.94%) and test set (61.23%), with an MSE of 38.77%.

In summary, the XGB Classifier and Random Forest Classifier appear to be the top-performing algorithms based on the provided metrics, demonstrating a good balance between training accuracy, test accuracy, and MSE.

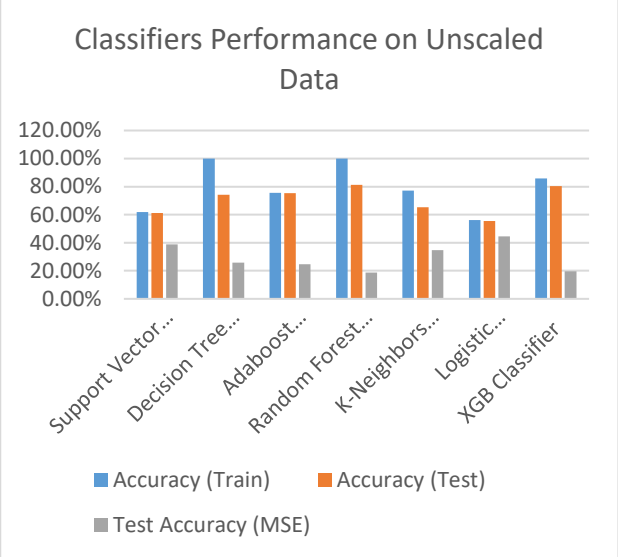


Fig. 12. Classifiers Performance on Unscaled Data

1) Performance of Models on Balanced but scaled Data

TABLE II. DIFFERENT CLASSIFIERS PERFORMANCE ON SCALED BALANCED DATA

Algorithm	Accuracy (Train)	Accuracy (Test)	Test Accuracy (MSE)
Support Vector Machine	77.71%	78.39%	0.2161
Decision Tree Classifier	99.98%	72.43%	0.2757
Adaboost Classifier	81.95%	82.32%	0.1768
Random Forest Classifier	99.97%	80.93%	0.1907
K-Neighbors Classifier	83.90%	78.83%	0.2117
Logistic Regression	77.71%	78.39%	0.2161
XGB Classifier	0.8675	0.8139	0.1861

Here are some conclusions based on the provided data:

Algorithm Performance:

The Decision Tree Classifier and Random Forest Classifier achieved high accuracy on the training set (close to 1.0), indicating potential overfitting. Adaboost Classifier and XGBClassifier performed well on both training and test sets, demonstrating good generalization. Support Vector Machine, Logistic Regression, and KNeighbors Classifier showed moderate performance. Decision Tree and Random Forest models exhibited high accuracy on the training set but comparatively lower accuracy on the test set, suggesting potential overfitting.

Consistency in Logarithmic Scaling: The models (except for Decision Tree and Random Forest) demonstrated similar accuracy on both the training and test sets, indicating that logarithmic scaling had a consistent impact across different algorithms.

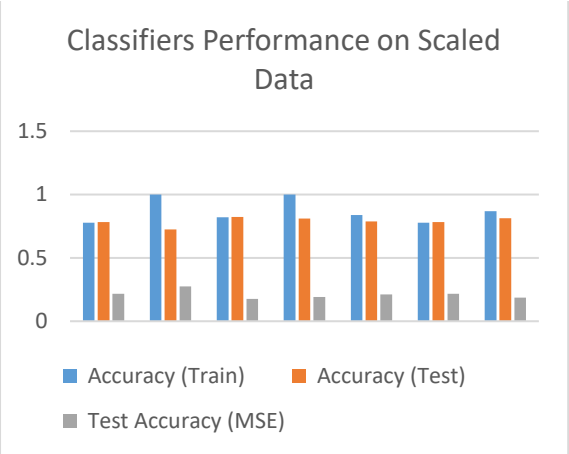


Fig. 13. Classifiers Performance on scaled Data

TABLE III. DIFFERENT CLASSIFIERS PERFORMANCE ON SCALED BUT UNBALANCED DATA

Algorithm	Accuracy (Train)	Accuracy (Test)	Test Accuracy (MSE)
Support Vector Machine	77.92%	77.75%	0.222533
Decision Tree Classifier	99.95%	73.23%	0.267733
Adaboost Classifier	82.02%	81.67%	0.183333
Random Forest Classifier	99.95%	81.08%	0.1892
K-Neighbors Classifier	83.74%	79%	0.21
Logistic Regression	77.92%	77.75%	0.222533
XGB Classifier	86.81%	81.33%	0.186667

Importance of Feature Scaling: The performance of algorithms, especially Support Vector Machine, can be sensitive to feature scaling. Logarithmic scaling is applied, which can help in normalizing the distribution of features.

Recommendation for Model Training: Both scaled and non-scaled datasets should be considered for training models, as the impact on accuracy can vary depending on the algorithm. These conclusions provide insights into the strengths and potential pitfalls of each algorithm and highlight the importance of appropriate preprocessing techniques for optimal model performance.

J. Performance of Models on unbalanced but Unscaled Data

When working with scaled but unbalanced data, several observations might arise from the results:

1. Accuracy Disparity: The accuracy scores for different algorithms might showcase varying levels of effectiveness. While some algorithms perform well on the training set, they might struggle with generalization, leading to lower accuracy on the test set. This disparity could indicate overfitting on the training data.

2. Class Imbalance Impact: Unbalanced data could affect model performance. Algorithms might have biases toward the majority class, resulting in high accuracy for that class but lower accuracy for the minority class. This imbalance might not accurately reflect the model's true predictive capability.

3. Model Selection Consideration: The choice of algorithms becomes crucial in handling unbalanced data. Some algorithms might handle imbalanced datasets better by incorporating techniques like weighted classes, ensemble methods, or specialized algorithms designed for imbalanced data.

4. Test Accuracy vs. MSE: It's interesting to observe the test accuracy along with the Mean Squared Error (MSE). A lower MSE implies better regression performance. Comparing accuracy and MSE provides insights into how well the models perform in both classification and regression tasks.

5. Model Comparison: Comparing various algorithms in this context helps identify which ones are more robust to the imbalanced nature of the dataset. The model's performance on both the training and test sets offers insights into its ability to generalize.

AdaboostClassifier and XGBClassifier outperforms in terms of accuracy among all the remaining models. Even scaling and balancing the dataset did not contribute much (as per what accuracy says)

1) Further Analysis on the best algorithm using Model Optimization

Beyond these scores, deeper analysis, like confusion matrices, precision-recall curves, or F1 scores, can provide a more comprehensive understanding of how the models handle class imbalance and their overall effectiveness. AdaboostClassifier and XGBClassifier outperforms in terms of accuracy among all the remaining models.

Performing hyperparameter tuning using GridSearchCV with an AdaBoostClassifier. This method can help find the best combination of hyperparameters for your AdaBoost model.

The code is using `GridSearchCV` to search through different hyperparameters for the AdaBoostClassifier:

- `n_estimators`: Number of estimators (weak learners) in the ensemble.

- `algorithm`: The algorithm to use for boosting ('SAMME' or 'SAMME.R').

- `learning_rate`: The contribution of each weak learner in the final combination.

The `cv=5` parameter indicates 5-fold cross-validation, splitting the data into 5 parts for training and validation.

After fitting the GridSearchCV on your training data (`X_train` and `y_train`), the best parameters for the

AdaBoost model will be stored in `best_parameters` using `clf.best_params_`.

This approach is beneficial for optimizing the performance of your AdaBoost model by testing various combinations of hyperparameters and selecting the set that yields the best accuracy based on the cross-validation scores.

This output provides a comprehensive evaluation of a classification model's performance, typically obtained after applying machine learning techniques to a dataset.

2) Optimization Results:

The overall accuracy of the model is 81.96%, indicating the proportion of correctly predicted instances (both true positives and true negatives) out of the total number of instances.

- Precision: Indicates the proportion of correctly predicted positive instances among all instances predicted as positive. For class 0, precision is 0.84, and for class 1, precision is 0.67.
- Recall (Sensitivity): Measures the proportion of actual positives that were correctly identified. For class 0, recall is 0.95, and for class 1, recall is 0.34.
- F1-Score: Harmonic mean of precision and recall, providing a balance between the two. For class 0, the F1-score is 0.89, and for class 1, the F1-score is 0.45.

- Support: Number of actual occurrences of the class in the specified dataset.

Confusion Matrix:

- True Negative (TN): 5596 instances were correctly predicted as class 0.
- False Positive (FP): 277 instances were wrongly predicted as class 1.
- False Negative (FN): 1076 instances were wrongly predicted as class 0.
- True Positive (TP): 551 instances were correctly predicted as class 1.

Interpretation:

The model exhibits higher accuracy and precision for predicting class 0 (no default), with a high number of true negatives and a relatively low number of false positives. However, the model's performance is weaker in predicting class 1 (default), with lower recall, precision, and F1-score. It correctly identifies fewer instances of class 1 (low recall) and also misclassifies some class 0 instances as class 1 (moderate false positives). The overall assessment highlights the model's strengths in predicting class 0 but indicates a need for improvement in identifying and predicting class 1 instances more accurately.

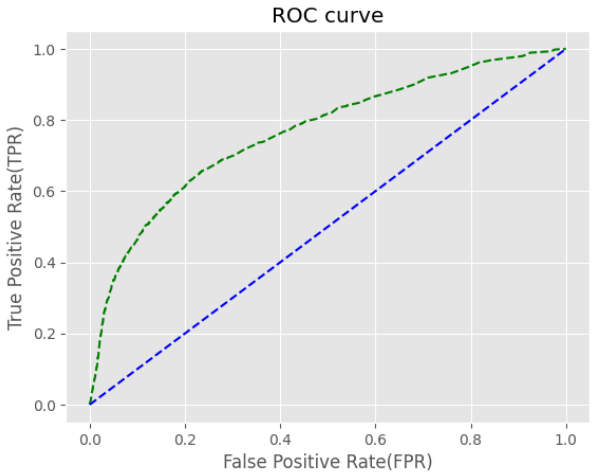


Fig. 14. ROC curve for the best model

TABLE IV. COMPARATIVE ANALYSIS WITH PRE-EXISTING WORK

Research Approach	Algorithm Used	Dataset Used	Classification Accuracy
Existing work that used K-NN for analysing Tunisian Bank customer's behaviour for checking risk of loan repay [14]	K-Nearest Neighbour Classifier	Tunisian Banks records used	Classification rate 88%
Corporate financial records with features liquidity, solvability, capital, profit margins, return on investment analysed [15]	Ensemble Learning Method by combining K-NN	Corporate financial records	Improved performance by combining methods and got result in order 83%
Presented our work used to decide whether a credit application is accepted or rejected	Decision Tree and the K-Nearest Neighbour models separately	Credit dataset which has 30,000 samples with 24 features each	Proposed models attain accuracy of 82.2% and 81.2%

5. CONCLUSION

The model's performance, as evidenced by an accuracy of 81.96%, shows a reasonable level of correctness in predictions across both classes. However, a deeper examination through the precision, recall, and F1-score reveals some nuances. For the 'no default' class (0), the model demonstrates higher precision (0.84) and recall (0.95), indicating it reliably identifies and correctly classifies instances where there is no default. This is further supported by a high F1-score (0.89), implying a balanced performance in this category. Conversely, for the 'default' class (1), the model's precision (0.67) and recall (0.34) are notably lower. It struggles to accurately identify instances where defaults occur, leading to a lower F1-score (0.45). The model tends to miss several instances of actual defaults (low recall) while also occasionally

misclassifying 'no default' instances as 'default' (moderate false positives).

In conclusion, while the model shows strength in predicting instances without defaults, its performance in identifying default cases requires improvement. Enhancing the model's ability to detect defaults accurately—reducing false negatives and improving recall for the 'default' class—would enhance its overall efficacy and reliability in practical applications.

REFERENCE

[1] Verma Shruti, Maan Vinod, "Comparative Analysis of Pig and Hive," International Journal of Research in Advent Technology, Vol.6, No.5, 2018. E-ISSN: 2321-9637. Available online at www.ijrat.org. 585 http://www.ijrat.org › paper ID-65201829

[2] N. Chen, B. Ribeiro, and A. Chen, "Financial credit risk assessment: a recent review," Artif Intell Rev, vol. 45, no. 1, pp. 1–23, Jan. 2016, DOI: 10.1007/s10462-015-9434-x.

[3] J. N. Crook, D. B. Edelman, and L. C. Thomas, "Recent developments in consumer credit risk assessment," European Journal of Operational Research, vol. 183, no. 3, pp. 1447–1465, Dec. 2007, DOI: 10.1016/j.ejor.2006.09.100.

[4] J. Galindo and P. Tamayo, "Credit Risk Assessment Using Statistical and Machine Learning: Basic Methodology and Risk Modeling Applications," Computational Economics, vol. 15, no. 1, pp. 107–143, Apr. 2000, DOI: 10.1023/A:1008699112516.

[5] S. Oreski and G. Oreski, "Genetic algorithm-based heuristic for feature selection in credit risk assessment," Expert Systems with Applications, vol. 41, no. 4, Part 2, pp. 2052–2064, Mar. 2014, DOI: 10.1016/j.eswa.2013.09.004.

[6] B. Twala, "Multiple classifier application to credit risk assessment," Expert Systems with Applications, vol. 37, no. 4, pp. 3326–3336, Apr. 2010, DOI: 10.1016/j.eswa.2010.10.018.

[7] L. Yu, S. Wang, and K. K. Lai, "Credit risk assessment with a multistage neural network ensemble learning approach," Expert Systems with Applications, vol. 34, no. 2, pp. 1434–1444, Feb. 2008, DOI: 10.1016/j.eswa.2007.01.009.

[8] M. Doumpos, K. Kosmidou, G. Baourakis, and C. Zopounidis, "Credit risk assessment using a multicriteria hierarchical discrimination approach: A comparative analysis," European Journal of Operational Research, vol. 138, no. 2, pp. 392–412, Apr. 2002, DOI: 10.1016/S03772217(01)00254-5.

[9] P. Saha, I. Bose, and A. Mahanti, "A knowledge-based scheme for risk assessment in loan processing by banks," Decision Support Systems, vol. 84, pp. 78–88, Apr. 2016, DOI: 10.1016/j.dss.2016.02.002.

[10] M. R. Sousa, J. Gama, and E. Brandão, "A new dynamic modeling framework for credit risk assessment," Expert Systems with Applications, vol. 45, pp. 341–351, Mar. 2016, DOI:10.1016/j.eswa.2015.09.055.

[11] Yeh, I. C., & Lien, C. H., "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," Expert Systems with Applications, 36(2), 2473-2480, 2009.

[12] Shousong Cai, Jing Zhang, "Exploration of the credit risk of P2P platform based on data mining technology", Journal of Computational and Applied Mathematics, Volume 372, 2020.

[13] H. Zhou, G. Sun, S. Fu, J. Liu, X. Zhou and J. Zhou, "A Big Data Mining Approach of PSO-Based BP Neural Network for Financial Risk Management With IoT," in IEEE Access, vol. 7, pp. 154035-154043, 2019, DOI: 10.1109/ACCESS.2019.2948949.

[14] T. Qiu, H. Wang, K. Li, H. Ning, A. K. Sangaiah and B. Chen, "SIGMM: A novel machine learning algorithm for spammer identification in industrial mobile cloud computing", IEEE Trans. Ind. Informat., vol. 15, no. 4, pp. 2349-2359, Apr. 2019. DOI: 10.1109/TII.2018.2799907

[15] Wang Bao, Ning Lianju, Kong Yue, "Integration of unsupervised and supervised machine learning algorithms for credit risk assessment", Expert Systems with Applications, Volume 128, 2019, Pages 301-315. https://doi.org/10.1016/j.eswa.2019.02.033.