Applying Machine Learning Techniques To Maximize The Performance of Loan Default Prediction

Vinay Padimi, Venkata Sravan Telu and Devarani Devi Ningombam

Department of Computer Science and Engineering,
GITAM Institute of Technology, GITAM University, Visakhapatnam, Andhra Pradesh, 530045, India
Emails: vinaypadimi@gmail.com, telusravan272@gmail.com, devaraninin@gmail.com

***** Correspondence: devaraninin@gmail.com

Abstract

In peer-to-peer (P2P) lending, borrowers would access loans with lower interest rates than what they usually got from traditional lenders. People can directly borrow from the P2P platform with the rules that make them easy to borrow loans and invest free funds into P2P, which can benefit both borrowers and lenders. However, the easy way to borrow loans comes with risks. One of the major issues is that borrowers may default on the loan taken. In such cases, they can get loans quickly from P2P online platforms without any bank interferences. Thus, the lender can calculate his risk for loan default. In this project, we consider the P2P lending data to predict the loan default reassuring the lender to continue providing loans in the future. In our analysis, we consider the Logistic Regression, Naive Bayes, Random Forest, K Nearest Neighbour, and Decision tree to classify loan data based on their likelihood of default. The simulation result in our algorithm provides a significant accuracy of 94.6%.

Keywords: KNIME, Machine Learning, Peer-to-peer Lending (P2P), Loan default, Hyperparameter Tuning, Dimensional Reduction.

1. Introduction

Loans are one of the sources to increase the overall money supply to overcome financial crises. Taking the loan and getting approval for the loan from banks is a long process. In current situations where almost everything is digitalized, people tend to choose an easy solution for every problem. One such case is applying for a loan on an online platform known as Peer-to-Peer Lending [1].

In the Peer-to-Peer Lending platform, people are willing to invest and borrow without any bank interfering. In the absence of interference, unsound information can be extracted transparently. Thus, these platforms provide users with access to loans with fewer interest rates so that the users can decide themselves whether to lend or borrow loans from safe users or riskier users, depending on their circumstances. Thus, the users can review the information provided by the platform to make their final verdict.

Many applications of artificial intelligence (AI) help solve issues in Peer-to-Peer Lending such as easy recognition of loan default, quickening loan procedures, reducing errors, etc. This project focuses on loan default by the user which is one major risk in Peer-to-Peer Lending. With the help of machine learning techniques, we can quickly identify borrowers who are at high risk of defaulting in the future [2]. The dataset we considered in this

Doi : https://doi.org/10.54216/JNFS.020204

Received: January 05, 2022 Accepted: March 22, 2022

project is taken from the European Peer-to-Peer Lending platform, Bondora [3] which provides consumer loans both secured and unsecured loans with 1 EUR minimum investment. According to OP bank which is one of the largest financial companies in Finland, we can get a secured loan of 10,000 euros and an unsecured loan of 2,000 euros [4]. So, we decided to eliminate records from the dataset where the loan amount is less than 2,000.

Two-dimension reduction techniques are used in this project to reduce the complexity (or dimension) and remove the features that are not contributing to learning. The high correlation filter technique is used to remove highly correlated features, and Backward feature elimination is used to remove features that do not contribute to learning. Machine learning supervised models like Logistic Regression, Random Forest, Naïve Bayes, etc., are used in this project to predict the loan status and work out the best approach for a lender to accurately classify whether the loan will default or not.

The main contributions in this paper are listed as:

1. This research is conducted using secondary open data from bondora that is available to the public. This limits the generalization of the model as it is specific towards the dataset used to train and test the model.
2. We focused on the probability of default in a default state to reduce the credit risk.
3. From the analysis, we have observed that our method provides an accuracy of 94.6%.

The overview of the paper is as follows: in Section 2, we present the related work, Section 3 presents proposed technique, Section 4 presents the performance analyses of models used and the paper is concluded in Section 5.

2. Related Work

In this section, we will discuss some of the existing works on loan default.

2.1. Credit Risk Analyses in Peer-to-Peer Lending

The paper [5] analyses credit risk caused by loan default in Peer-to-peer (P2P) online platforms. The author used Ensemble learning to determine the features that play a vital part in predicting credit risk. This paper mainly focused on Pre-processing and Cleaning, Feature Selection, and Classification as a part of the methodology. Tree-based ensemble machine learning c1assifiers like Decision Tree, Random Forest, Extra Trees, and Bagging is used to accomplish better performance. We have seen from the paper that the Decision Tree has the highest precision in comparative analyses of these algorithms, and Random Forest has the best accuracy. Thus, the author concluded that the Decision Tree is more potent in finding the best credits, while the Random Forest predictor model is more effective in identifying the defaults.

2.2. A Study on Predicting Loan Default based on the Random Forest Algorithm [6]

This paper [6] studies Lending Club, a P2P lending platform, and the risks faced in loan default. Similarly, from the previous paper [5], Decision trees, Logistic regression, Support Vector Machine (SVM), and Random Forest are used to predict loan default on the dataset. Moreover, operations like Oversampling, data cleaning, and dimension reduction are performed on the dataset for better learning performance. With an accuracy of 98%, the random forest algorithm outperforms the other three algorithms thus used for building a model.

2.3. Loan Default Prediction with Machine Learning Techniques [7]

Motivated by the increasing interest in Big Data and the development of machine learning techniques, this paper [7] mainly focused on the study of loan default which is a significant issue for loan firms, and how they performed predictions on the sophisticated machine learning models (i.e., XGBoost, Random Forest, AdaBoost, K Nearest Neighbour, and Multi-Layer Perceptron). This project collected the data from Xiamen International bank. The concluding statement by the authors in this paper is that the performance of Random Forest, K Nearest Neighbour,

Doi : https://doi.org/10.54216/JNFS.020204

Received: January 05, 2022 Accepted: March 22, 2022

and Multi-Layer Perceptron algorithms are weaker than other models, whereas the Boosting algorithms like XGBoost, AdaBoost perform better.

2.4. Loan Default Prediction Using Machine Learning: A Case of Mobile Based Lending [8]

In this paper [8], the authors focused on applying machine learning algorithms to predict the loan default on online mobile-based lending techniques. The proposed methodology involves Data Collection, Data Pre-processing, Data analyses, Model selection, and Performance Evaluation. The algorithms applied are logistic regressions, naïve Bayes, and decision trees. Among three classifiers, Decision trees performed better with 64% accuracy. The main challenge in this paper is the low accuracy rate. In this paper, to maximize the accuracy, we propose a hyperparameter tuning and dimension reduction technique.

3. Proposed Technique

3.1. Architecture/Framework

The Konstanz Information Miner, shortly known as KNIME, is a data analytics platform that assimilates diverse elements of data science using visual programming. The key features are:

i. It can work on different types of data and connect data from other sources like Salesforce, Aws, Azure, etc.
ii. We can visualize and customize data to bring maximum learning performance.
iii. We can build, optimize, and validate our ML model.
iv. Workflows are created with no coding, which helps better understanding.

3.2. Training Phase

3.2.1. Dataset Preparation

The dataset used in this project is a real time dataset publicly available at Bondora, a leading European P2P lending platform. The dataset is acquired from the public-reports page of the Bondora website [9]. It includes loan data from 2009 to present which contains 2,20,906 rows and 112 columns. The description for entire dataset is available at [9].

Data fields	Description
Rating	Rating issued by the Bondora
CreditScoreEsMicroL	Credit score given to borrower by calculating probability of default one month ahead.
BidsApi	The amount of investment offers made via Api
BidsManual	The amount of investment offers made manually
VerificationType	Method used for loan application data verification
Age	The age of the borrower when signing the loan application
Gender	Gender of the borrower
LoanDuration	Duration of loan in months
MonthlyPayment	Estimated amount the borrower must pay every month
Education	Education of the borrower when signing the loan application
HomeOwnershipType	The borrower type of property ownership
ExistingLiabilities	Existing liabilities of borrower
ActiveScheduleFirstPaymentReached	It shows whether the first payment date has been reached according to the active schedule
Restructured	The original maturity date of the loan has been increased by more than 60 days
PrincipalPaymentsMade	The principal amount owner received
Status	The current status of the loan application

Figure 1: Dataset after Pre-processing

For importing the dataset into our workflow CSV Reader node is used, which allows us to read dataset from our local computer.

Doi : https://doi.org/10.54216/JNFS.020204

Received: January 05, 2022 Accepted: March 22, 2022

3.2.2. Data Pre-processing

Data Pre-processing is the first step to start building a machine learning model. The model performance is mainly dependent on how well data is pre-processed. Data Cleaning, Feature Engineering, Data Transformation, Data Reduction, Data Scaling, and Data Partitioning are a few techniques performed on datasets to improve their quality[10].

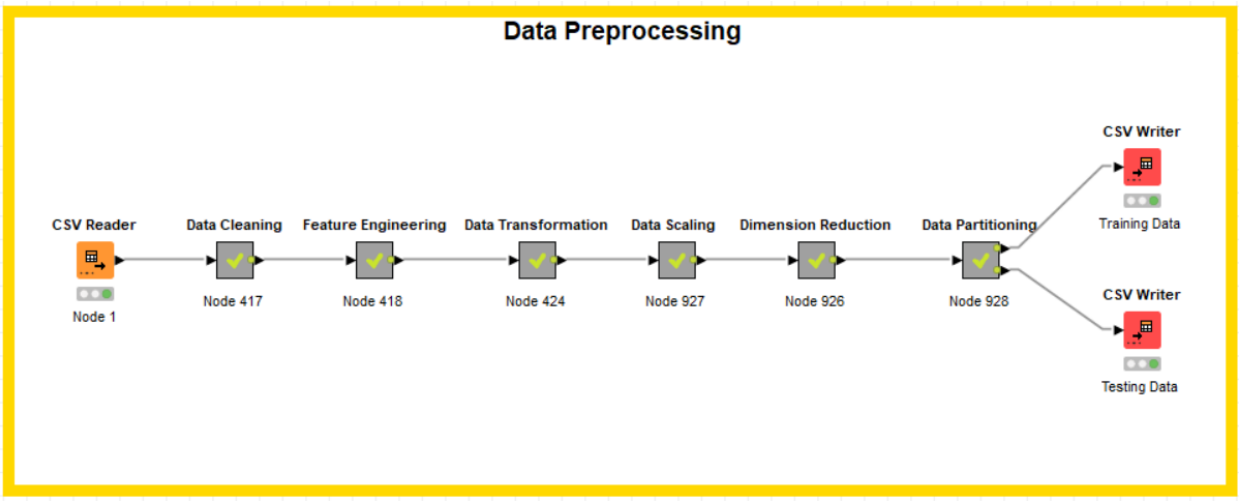

Figure 2: Workflow of Data Pre-processing

a) Data Cleaning: Data cleaning is a technique to improve the quality of data by handling incorrect data, anomaly data, duplicate data, and imputing missing data.

Handling Anomaly Data

We have anomaly data in the dataset, such as age less than 10, debt to income ratio greater than 100, etc. The "Row Filter (Labs)" node helps to filter rows based on multiple queries. Age >= 18, DebtToIncome < 100, Principal Balance >0 are Some queries included in this node.

Handling Error Data

This dataset has 617 error records getting loan amounts greater than the applied amount "Rule-based Row Filter" node is used to filter these records based on rules written. All row having $Amount$ > $AppliedAmount$ => TRUE are excluded.

Handling Empty Columns

It is the common practice in machine learning to drop the rows having more null/missing values in columns. 'DateOfBirth', 'County', 'City', and 'EmploymentPosition' are the features having only missing values throughout the dataset. The "Missing Value Column Filter" node is used in this scenario to remove columns having more than 50% missing values.

Handling the Missing Values

We have to eliminate missing values to increase the accuracy of the learning. We can either remove rows containing missing values/null-values or replace them with mean, median, mode, etc. The "Missing Value" node gets this job

Doi : https://doi.org/10.54216/JNFS.020204

Received: January 05, 2022 Accepted: March 22, 2022

done. Here, missing values for nominal data are replaced by the most frequent value and the mean value for numeric data.

Handling Duplicate & Constant Data

'LoanId' and 'LoanNumber' are the features that represent loan applications uniquely. Considering both are unnecessary, so we discard 'LoanId'. Similarly, multiple values of Income are erased since they are, as of now amassed in 'IncomeTotal'. "Column filter" is used to remove these duplicate features. "Constant Value Filtering" and "Unique Value Filtering" node helps handle constant and unnecessary data which is not required.

b) Feature Engineering: Our objective in this project is to determine if the loan will default. So, the target variable should be either default or not. We can create a target variable from this dataset using the field Status. Status is a nominal attribute whose values are 'Current', 'Repaid', and 'Late'. 'Repaid' can be treated as not default represented as 1, whereas 'Late' can be treated as default represented with 0. Since the 'Current' value is not required "Row Filter" node is used to exclude records with 'Current' value using pattern matching, and the "Rule Engine" node is used to represent 'Repaid' with 1 and 'Late' with 0.

c) Data Transformation: 'NrOfDependents' is stored as string which is converted into integer using the node "String To Number" and nominal data is converted into integer using the node "Category To Number".

d) Dimension Reduction: Feature selection is made in this section where Spearman's rank correlation is applied to data using "Rank Correlation," and in the range of 0.1 to 0.9, 0.3 is taken as the correlation threshold in the "Correlation filter" eliminate fields having high correlated features. The "Column Appender" node is used to append the target column to the dataset. 26 features are reduced in first dimension reduction.

We performed backward feature elimination to remove an unnecessary feature that does not contribute to model performance. "Feature Selection Loop Start" and "Feature Selection Loop End" are nodes used in backward feature elimination. "Feature Selection Loop Start" provides different strategies for feature selection, such as forward feature elimination, backward feature elimination, etc. After the second-dimension reduction, five features are eliminated.

e) Data Scaling: Depending on the data, we can either use standardization or normalization. The Normalization node in KNIME provides three types of normalization: Min-max normalization, Z-score normalization, and normalization by decimal scaling. Min-max normalization is rescaling values into the range [0,1]. Standardization or Z-score normalization is rescaling the data into Gaussian (0,1) distributed, i.e., a mean of 0 and a standard deviation of 1(unit variance).

Handling imbalanced data is done by either oversampling the minority class or undersampling the majority class column. Oversampling is performed if data is less, and Undersampling is done if data is more.

f) Data Partitioning: With the help of the "Partitioning" node, we split the dataset into 80% training and 20% testing data stratified over 'Status' and are respectively stored in the local computer using CSV Writer.

3.2.3. Model Building

In this paper, three steps are followed in this section, i.e., Selecting the model, optimizing, and validating.

First, we need to select ML algorithms to build a model. The next step is building the model using these algorithms. KNIME provides ML models into learner and predictor nodes. In the learner node, we develop our model, and it usually takes training data and outputs a learning model represented in a square shape at the end of the

Doi : https://doi.org/10.54216/JNFS.020204

Received: January 05, 2022 Accepted: March 22, 2022

learning node. This learning model and testing data are carried as input for the predictor node to predict the outcome.

After building a model, hyperparameter tuning and validation are done. Hyperparameter tuning refers to optimizing the model to bring maximum learning performance for a model. Parameter Optimization nodes help in performing this task. All algorithms have undergone hyperparameter tuning to bring maximum learning performance. All ML models cannot perform well on the same dataset, so we must identify which model is used for implementation, and for doing that, we need a validation dataset that helps us find an optimized model for a given task. The validation set is part of the training data, i.e., we further split training data into training and validation datasets where training data is used to train the model, whereas the validation dataset is used to validate our model. The model's performance should be more than 60% otherwise, considering this model does not yield good results.

A. Proposed algorithms

In this section, we proposed different ML algorithms like Logistic Regression, Decision Tree, Naïve Bayes, Random Forest, and KNN for building the model [11].

Logistic Regression: It is used to understand the relationship between the independent and dependent variables by calculating probabilities using a logistic regression function. It helps us predict the likelihood of an event occurring.

*Decision tree:*The decision tree uses a tree-like model to make decisions on their possible consequences, event outcomes, etc. Using this result, it performs a complete analysis of each branch and identification of decision nodes that require further analysis.

Random forest: Random forest builds decision trees on different samples and takes their majority vote for classification. It is a supervised machine learning algorithm used in both Classification and Regression problems.

Naive Bayes: A naive Bayes classifier is simple to implement and uses the Bayes theorem to classify data. It is helpful for making predictions and forecasting output based on historical outcomes.

K-Nearest Neighbour: K-Nearest Neighbour uses the distance function to identify nearby data points to estimate the final result. Different distance functions are used to calculate the distance, such as Euclidean distance, Manhattan distance, etc.

B. Hyperparameter tuning

Doi : https://doi.org/10.54216/JNFS.020204

Received: January 05, 2022 Accepted: March 22, 2022

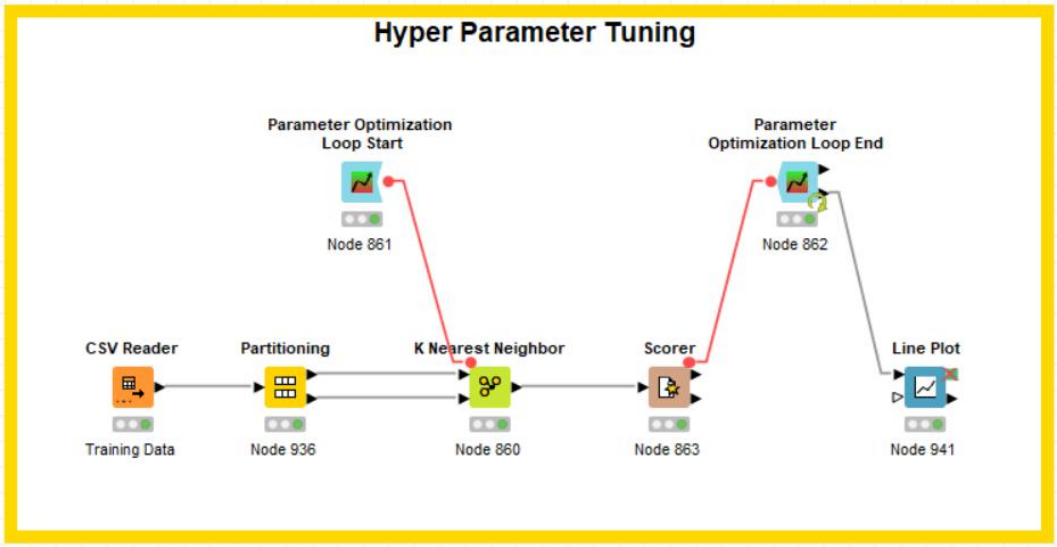

Figure 3: Workflow of Hyperparameter tuning on KNN

Figure 3, presents how hyperparameter tuning is performed in KNN. “Parameter Optimization Loop Start” and “Parameter Optimization Loop End” are two nodes in KNIME for hyperparameter tuning. “Parameter Optimization Loop Start” node helps us initialize parameters for the model and provides different strategies for better results. “Parameter Optimization Loop End” allows us to define whether the objective function should be minimized or maximized and outputs results of performance. Here we can either take the objective function as accuracy to be maximum or error to be minimized.

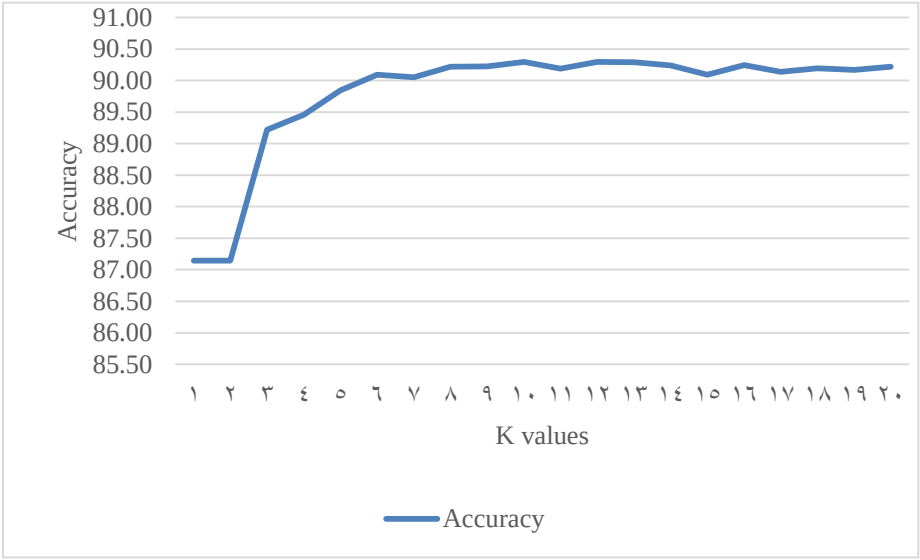

Figure 4: Workflow of Hyper parameter tuning for KNN

Figure 4 illustrates the results of hyperparameter tuning in K Nearest Neighbour (KNN). In figure 4, we can observe that from k value 6, accuracy is 90%. So, we consider k value 6 as an optimal parameter for the KNN model. Similarly, we perform hyperparameter tuning for all models.

C. Cross Validation

Doi : https://doi.org/10.54216/JNFS.020204

Received: January 05, 2022 Accepted: March 22, 2022

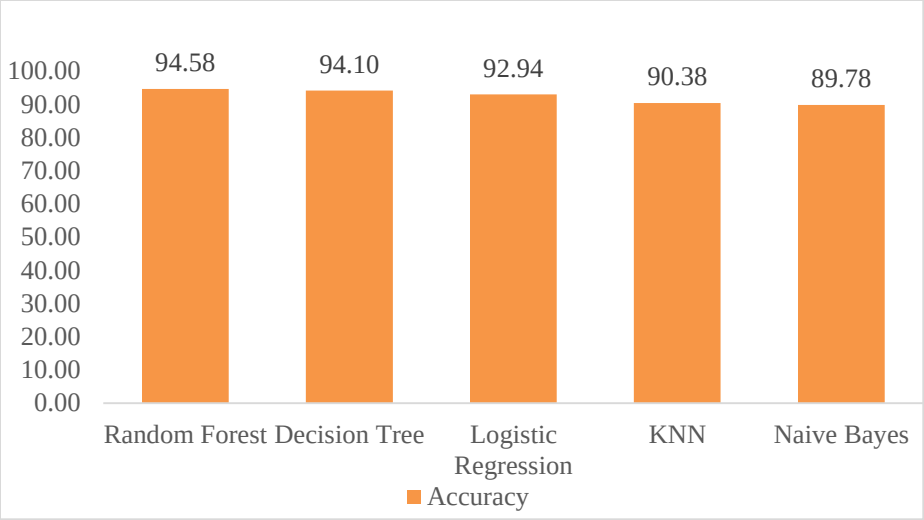

Figure 5: Cross Validation results

Cross-validation is validating our model more than once by creating a fixed number of folds of the data and performing an analysis of each fold to estimate the overall mean error. Cross-validation helps us understand data and gives us much more information about the performance of the algorithm. We considered ten folds in cross-validation as a standard procedure and analyzed errors in each fold. KNIME provides "X-Partitioner", which splits the dataset to data folds, and "X-Aggregator" runs the validation k times where k is no of folds in k fold cross-validation. "X-Aggregator" outputs the prediction table with prediction values appended and the error table, which contains errors in each fold.

The accuracy metric is used for analysing the performance of a model in the training phase. From figure 5, we can observe that all algorithms overall performance is greater than 60%, so we proceed with model testing.

3.3. Testing Phase

After validating our model, we proceed to model testing to ensure that logic learned will remain consistent, no matter how many times we execute. We perform cross-validation to make sure the model learns all the data, and in testing, we check how the model performs on unknown data. Based on the results from training and testing, we can check the performance of the model. Two situations that arise in this scenario are overfitting and underfitting. Underfitting arises because the model is incapable of accurately capturing the relationship between the input and target variable, generating a high error rate on both the training and unseen data. Overfitting happens when a model learns noise more than the input data in training to the degree that it negatively affects the models performance on unseen data. We use 80% training data and 20% testing data to test the model before deployment

3.4. Overall Phase

The dataset we considered in this project has 112 attributes and missing values in many fields. In section 3.2.2, different techniques are used to increase the quality of the dataset and increase the performance of learning we performed data pre-processing.

In section 3.2.2(a), we perform data cleaning where we improved the quality of the dataset. Section 3.2.2(b) presents feature engineering where the target variable has three unique values such as 'Current', 'Repaid', and 'Late'. We consider 'Repaid' as not-default, 'Late' as default, and remove 'Current' as it plays no role in loan

Doi : https://doi.org/10.54216/JNFS.020204

Received: January 05, 2022 Accepted: March 22, 2022

default. Section 3.2.2(c) presents data transformation that converts categorical data into numerical data. Data reduction is performed in section 3.2.2(d). The high correlation filter technique is the first-dimension reduction used, reducing the dataset to 21 features. After removing high correlated features, we used backward feature elimination to remove features that do not contribute to learning, i.e., after removing these features, we either get the same accuracy or improved accuracy. Finally, we performed data scaling in section 3.2.2€ and partitioning in section 3.2.2(f). After pre-processing, we got the dataset with dimensions of 64,202 rows and 16 columns.

In section 3.2.3, we have proposed model building which includes proposed algorithms, hyperparameter tuning, and cross-validation. Section 3.2.3(A) presents the proposed algorithms, and section 3.2.3(B) presents hyperparameter tuning for all proposed algorithms. After optimizing the model to its maximum performance, we perform cross-validating for validating the model in section 3.2.3(C). Finally, in section 3.3 model test is performed.

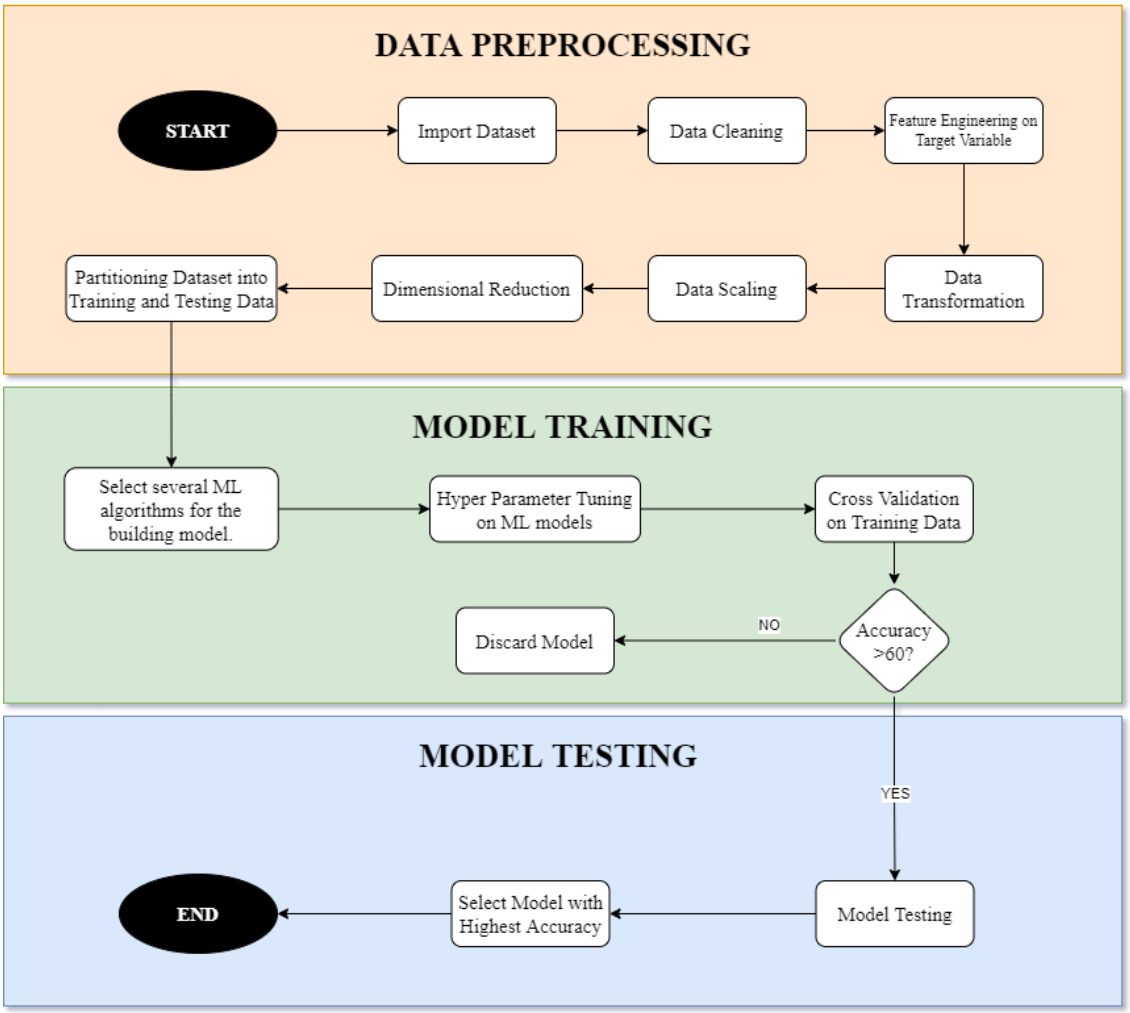

Figure 6: Proposed methodology

Algorithm 1: Pseudo Code of the Proposed Technique
Step 1: Importing the dataset into KNIME Work flow
Step 2: Preprocessing of the dataset → Cleaning the dataset to improve the quality → Perform feature engineering on target variable

Doi : https://doi.org/10.54216/JNFS.020204

Received: January 05, 2022 Accepted: March 22, 2022

→ Perform data transformation on required attributes → Scale the data on every attribute by performing Standardization → Eliminating the unwanted features using dimension reduction → Split the dataset into training and testing data
Step 3: Building a ML model → Select different ML algorithms → Perform Hyperparameter tuning to select optimal parameter for a model → Validating the training data with validation data
Step 4: If performance in validation is less than 60% then discard the model else proceed to testing
Step 5: Keeping the threshold value as 0.6. If overall accuracy is less than 60%, discard the model
Step 6: Testing the model with test data
Step 7: Select the model which has highest overall accuracy

4. Performance Analysis

"Scorer (JavaScript)" node in KNIME compares the target column and predicted column and outputs confusion matrix, class statistics table and overall class statistics. Confusion matrix contain True Positives, False Positives, True Negatives, and False Negatives. A true positive represents that the model accurately predicts the positive class, whereas a true negative shows that the model correctly predicts the negative class. A false positive outcome shows that the model incorrectly predicts the positive class, and false negative results in incorrectly predicting the negative class. which are used to describe the performance of a model by calculating performance metrics like Recall, Precision, Sensitivity, Specificity, F-measure.

4.1. Simulation Environment

This project is created using KNIME simulation environment. In KNIME, the Node Repository provides us nodes to create a workflow to work on the ML concepts. KNIME Hub has many workflows for different scenarios. We used those reference workflows to build our ML model.

4.2. Results and Discussions

We have analyzed various supervised algorithms like random forest, logistic regression, decision tree, naïve bayes and k-nearest neighbor. We calculated performance metrics and taken Receiver Operating Characteristic (ROC) and accuracy for comparative analyses.

Metrics [12] we considered in this project are

4.2.1 Precision: It is the percentage of positive identifications that were actually correct

$$Precision = \frac{True\ Positive}{(True\ Positive + False\ Postive)} \quad (1)$$

4.2.2 Recall: It tells us the percentage of true positives was correctly identified

$$Recall = \frac{True\ Positive}{(True\ Positive + False\ Negative)} \quad (2)$$

Doi : https://doi.org/10.54216/JNFS.020204

Received: January 05, 2022 Accepted: March 22, 2022

4.2.3 Accuracy: This is the measurement that indicates the complete correct predictions of the model.

$$Accuracy = \frac{True\ Positive+True\ Negative}{(True\ Positive+True\ Negativ+False\ Positive+False\ Negative)} \quad (3)$$

4.2.4 F-score: The F1-score metric combines precision and recall by taking their mean.

$$F-score = \frac{2(Recall \times Precision)}{(Recall+Precision)} \quad (4)$$

Algorithms	Precision		Recall		F-score	
	0	1	0	1	0	1
Decision Tree	0.954	0.933	0.931	0.955	0.942	0.944
Logistic Regression	0.938	0.917	0.915	0.94	0.926	0.928
Naïve Bayes	0.941	0.855	0.84	0.947	0.887	0.899
Random Forest	0.964	0.93	0.927	0.965	0.945	0.947
KNN	0.916	0.895	0.892	0.918	0.904	0.906

Table 1: Performance metric of ML Models

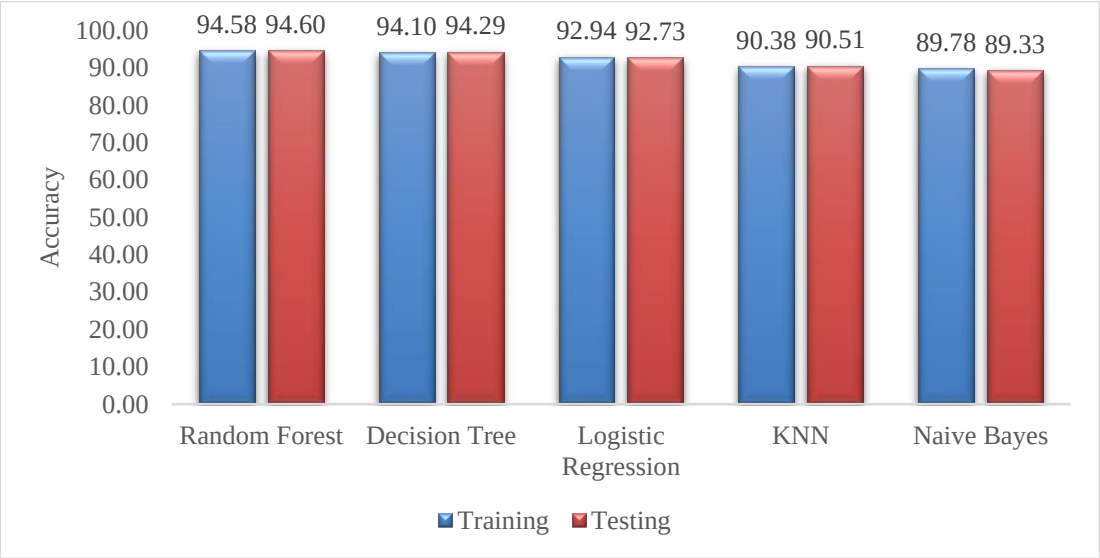

Figure 7: Comparative analyses on training and testing data using Accuracy

From figure 7 we can observe that there is no overfitting and under fitting problem. We performed cross validation on training to avoid overfitting and we have selected algorithms which perform at least moderate on data to avoid underfitting.

Doi : https://doi.org/10.54216/JNFS.020204

Received: January 05, 2022 Accepted: March 22, 2022

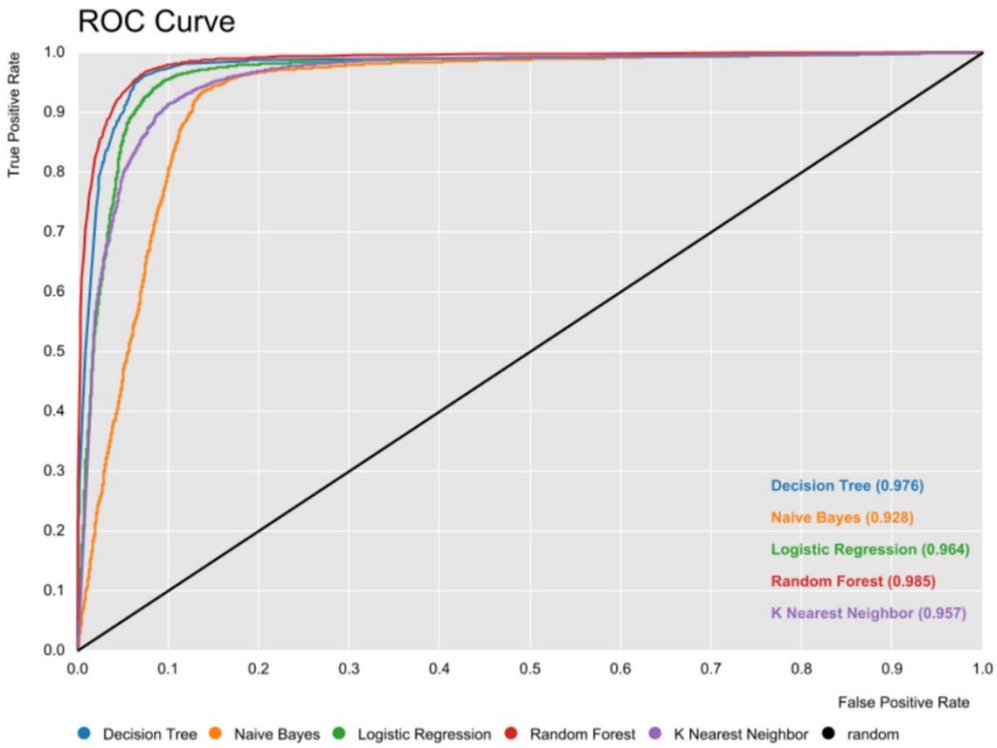

Figure 8: Receiver Operating Curve (ROC) Comparative analyses on ML models

ROC graph shows the performance of a classification model, and the Area Under Curve (AUC) represents the degree of separability. Figure 8 illustrates that random forest have the highest AUC.

5. Conclusion

Loan default prediction is a crucial and challenging problem in today's world. In this project, we explored and analyzed the machine learning algorithms to identify whether the person is defaulting on a loan or not.

We have used five algorithms random forest, logistic regression, decision tree, KNN algorithms, and naive-bias algorithm. In a comparative analysis, tree-based classifiers like random forest and decision trees are performing better when compared to other models. Among the tree-based classifier, random forest outperformed the decision tree by 0.3% accuracy, and with random forest having the highest AUC, we can conclude that random forest is used best-suited model for this scenario. we used model writer to save the model for deployment.

In the future study, we will experiment with the extensive dataset and other machine algorithms to further improve the model's performance and try to investigate with different P2P datasets to find the accuracy with additional attributes.

Conflicts of Interest: "The authors declare no conflict of interest."

References

[1] Y. Zuo, "CLUSTERING ANALYSIS TO SUPPORT LENDER'S DECISION-MAKING IN P2P LENDING - Bondora case study: borrower's creditworthiness classification," 2015. doi: 10.13140/RG.2.2.11598.48965.

Received: January 05, 2022 Accepted: March 22, 2022

[2] "How Machine Learning Will Transform P2P Lending," Lending Times, Jul. 23, 2019. https://lending-times.com/2019/07/23/how-machine-learning-will-transform-p2p-lending/ (accessed Apr. 05, 2022).
[3] "Bondora.com," Bondora.com. https://www.bondora.com/ (accessed Apr. 05, 2022).
[4] "Loans | loan for all sorts of situations." https://www.op.fi/private-customers/loans-and-homes/loans (accessed Apr. 05, 2022).
[5] V. L, N. Subramanyam, K. S, C. M, and L. N, "Credit Risk Analysis in Peer-to-Peer Lending System," Sep. 2016, pp. 193–196. doi: 10.1109/ICKEA.2016.7803017.
[6] L. Zhu, D. Qiu, D. Ergu, C. Ying, and K. Liu, "A study on predicting loan default based on the random forest algorithm," Procedia Computer Science, vol. 162, pp. 503–513, 2019, doi: 10.1016/j.procs.2019.12.017.
[7] L. Lai, "Loan Default Prediction with Machine Learning Techniques," in 2020 International Conference on Computer Communication and Network Security (CCNS), Xi'an, China, Aug. 2020, pp. 5–9. doi: 10.1109/CCNS50731.2020.00009.
[8] G. T. Kisutsa, "Loan Default Prediction Using Machine Learning : a Case of Mobile Based Lending," p. 51.
[9] "Bondora.com," Bondora.com. https://www.bondora.com/ (accessed Apr. 14, 2022).
[10] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, "A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data," Frontiers in Energy Research, vol. 9, 2021, Accessed: Apr. 14, 2022. [Online]. Available: https://www.frontiersin.org/article/10.3389/fenrg.2021.652801
[11] "Top 10 Machine Learning Algorithms: Supervised, Unsupervised Learning & More | Simplilearn," Simplilearn.com, Nov. 09, 2016. https://www.simplilearn.com/10-algorithms-machine-learning-engineers-need-to-know-article (accessed Apr. 14, 2022).
[12] S. Ghoneim, "Accuracy, Recall, Precision, F-Score & Specificity, which to optimize on?," Medium, Apr. 08, 2019. https://towardsdatascience.com/accuracy-recall-precision-f-score-specificity-which-to-optimize-on-867d3f11124 (accessed Apr. 14, 2022).

Doi : https://doi.org/10.54216/JNFS.020204

Received: January 05, 2022 Accepted: March 22, 2022