A Classification Analysis of Customers Based on Ability to Repay Bank Loan

Annie Martina Viju
300210450
Data-Analysis
University of the Fraser Valley
annie.viju@student.ufv.ca

Anupam Sharma
300208103
Data-Analysis
University of the Fraser Valley
anupam.sharma@student.ufv.ca

Shanmukha Sree Veda Tippavajhala
300210776
University of the Fraser Valley
Shanmukha.Tippavajhala@student.ufv.ca

***Abstract* — In Today's world, the number of people applying for loans is increasing exponentially due to various reasons** [1]**. The bank organizations and employees are taking much time to analyze and classify whether customers can repay the loan amount or not. By addressing this problem, the project aims to classify the customers based on features like credit score, work experience, annual income, etc., and apply Supervised Machine Learning concepts. The analysis starts with data preprocessing then continues with normalization, sampling, and feature selection of customer details then build classification models using pandas, sklearn, and numpy and analyzes and compares with distance-based, tree-based, linear-based, and visualizes performance accuracies and makes effective classification of customers. The result of this binary classification analysis is to predict whether the customer will repay the loan based on his historical data.**

Keywords— Classification model, supervised machine learning, data preprocessing, normalization, sampling, feature selection, tree-based, linear classification, distance-based

I. INTRODUCTION

The business philosophy of the banking sector is to circulate money between loans and savings accounts at a certain interest rate [2]. If this doesn't happen or, if there has arisen an imbalance between the savings and loan customers then it results in huge losses for the bank. Therefore, the ability to analyze loan applications and customer repayments is important. The main goal of banks is to make their wealth available to more secure people. Even banks are approving loans after verifying and confirming documents provided by customers. However, there is no guarantee that the applicant deserves it. Therefore, sharp and accurate analysis of loan customers is becoming a big problem for bankers to evaluate them based on past historical data. Today, machine learning plays an important role in analyzing data and providing good predictions. Machine learning algorithms are known for their deep understanding of data and the impact of different parameters on output results. Bank loan status data is categorized under binary supervision classification problems. The project tries to classify loan applicants or customers based on their ability to repay according to different customer characteristics using Machine Learning models like Linear models, Tree-based models, and Distance-Based models from sklearn packages and evaluate accuracies and find the best classification model to classify bank loan customers.

II. DATA DESCRIPTION

The data set is taken from the Kaggle website [3] consisting of 100000 records with 18 features including the classification label "Loan Status".

Data Set Features:

1. Loan ID – String – A Unique ID for the Loan Application
2. Customer ID – String – A Unique ID for Customer
3. Current Loan – Numeric – Loan amount requested
4. Term – Categorical – Loan Term duration
5. Credit Score – Numeric – Customer available credit score
6. Annual Income – Numeric – Annual Income of a customer
7. Years in current job – Categorical – Work Experience
8. Home Ownership – Categorical – Type of Customer Accommodation
9. Purpose – Categorical – Loan Purpose
10. Monthly Debt – Numeric – Numeric – Monthly Installments

11. Years of Credit History – Numeric
12. Months since last delinquent - Numeric
13. Number of Open Accounts – Numeric – Number of Accounts
in use
14. Number of Credit Problems – Numeric
15. Current Credit Balance – Numeric – Available credit balance
16. Maximum Open Credit - Numeric
17. Bankruptcies - Boolean
18. Tax Liens – Boolean
Class Label: Loan Status – Boolean – Whether Customer Paid Loan or not.
Loan Status is the class label that gives information on whether the customer repays the complete loan or not.

III. DATA PRE-PROCESSING

Data preprocessing is a data mining technique that involves converting raw data into an understandable format. Data in the real world are often incomplete, inconsistent, and/or lack certain behaviors or trends, and may contain many errors. Data preprocessing is an effective method to solve such problems.

In the real world, data is often incomplete: missing attribute values, missing certain attributes of interest, or containing only summary data. Noisy: Contains errors or outliers. Inconsistencies: Include differences in code or name.

A. Handling Missing Values

In the Bank Loan Status data set columns like “Credit Score”, “Annual Income” and “Months since last delinquent” have missing values. As these columns’ data are linearly distributed the missing values can be directly replaced with their respective column means.

B. Categorical to Numeric

Generally, we come across data with categorical features, but many machine learning algorithms do not support categorical data. So, it’s important to convert categorical data to numeric data to use machine learning algorithms for classification.

1. One Hot Encoding

One type of hot coding is the process of transforming categorical variables into a form that can be provided to an ML algorithm for better prediction. The input to this converter should be an array, such as an integer or a string, representing the values used by the classification (discrete) feature. These functions are encoded using a single-pass (also known as "one-of-K" or "dummy") encoding scheme. This will create a binary column for each category and return a sparse matrix or dense array. By default, the encoder derives categories based on the uniqueness Value of each function. Alternatively, you can specify Sort manually. OneHotEncoder previously assumed that the value range of the input features was [0, max (values)). This behavior is deprecated. We use One hot encoding for the feature data which are
independent of each other and adding weights may increase in bias and weight of the model. In our data set the columns “Home Ownership” and “Purpose” are the One Hot Encoded as their values are independent of each other.

Home Ownership_HaveMortgage	Home Ownership_Home Mortgage	Home Ownership_Own Home	Home Ownership_Rent	Purpose_Business Loan	Purpose_Buy House	I
0	1	0	0	0	0	
0	1	0	0	0	0	
0	0	1	0	0	0	
0	0	1	0	0	0	
0	0	0	1	0	0	
0	1	0	0	0	0	
0	1	0	0	0	0	
0	1	0	0	0	1	
0	0	0	1	0	0	
0	0	0	1	0	0	
0	0	0	1	0	0	
0	1	0	0	0	0	
0	0	1	0	0	0	
0	0	0	1	0	0	
0	0	0	1	0	0	
0	0	1	0	0	0	
0	1	0	0	0	0	
0	1	0	0	0	0	
0	0	0	1	0	0	
0	1	0	0	0	0	
0	0	0	1	0	0	
0	0	0	1	0	0	
0	0	0	1	0	0	
0	1	0	0	0	0	
0	1	0	0	0	0	
0	0	0	1	0	1	

Figure 1: Data Set after One Hot Encoding

2. Label Encoding

Label Encoding or Ordinal Encoding is the process of assigning continuous values starting with 1. This encoding is used when the feature data values are ordered, and the highest value is given to the feature data with high priority. In our data set, “Loan Status”, “Term”, and “Years in current job” are label encoded.

3. Normalization

Normalization is a technique often used when preparing machine learning data. The goal of normalization is to change the value of a numeric column in a data set to a common scale without distorting the difference in the value range. For machine learning, there is no need to standardize every data set. Only required if the functions have different ranges [4]

Min Max Normalization

Min Max Normalization is the process of converting values with different ranges to between 0 and 1. Min Max Normalization is used to transform data based on the Max and Min Values of feature column data.

In our dataset, columns such as "Current Years Worked", "Credit Years", and "Number of Open Accounts", “loan amount”, “credit score”, “annual income”, “monthly debt”, “months since last delinquent”, “current credit balance” and “maximum open credit”. are converted using "Minimum Maximum Normalization".

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Figure 2: Min Max Normalization for range (0,1)

Min Max normalization cannot be used when min and max are outliers that affect the normalization scale.

4. Dimensionality reduction

In statistics, machine learning, and information theory, dimensionality reduction or dimensionality reduction is the process of reducing the number of random variables considered by obtaining a set of main variables. Methods can be divided into feature selection
and feature extraction [5].

As part of feature engineering, the columns “Loan ID”, “Customer ID” are dropped as they are not affecting and influencing the dataset on machine learning algorithms.

5. Sampling the imbalance data

Training a machine learning model on an imbalanced dataset can introduce unique challenges to the learning problem. Imbalanced data typically refers to a classification problem where the number of observations per class is not equally distributed; often you'll have many data/observations for one class (referred to as the *majority class*), and much fewer observations for one or more other classes (referred to as
the *minority classes*) [6].

Bank Loan Data set deals with class imbalance, with most of the class with loan repayable customers and very few not. So, for better results, accuracy, and classification analysis it’s important to sample the data set and make it balanced so that both the class label has an equal priority and balance.

5.1 Over Sampling

Oversampling is the method of adding more of the minority class, so it has more effect on the machine learning algorithm

SMOTE

Smote is a nearest neighbor technique based on Euclidean distance between data points in feature space.
There is an oversampling percentage that indicates the number of synthetic samples to be created, and the oversampling percentage parameter is always a multiple of 100. If the oversampling percentage is 100, for each instance, a new sample will be created. As a result, the number of instances of a few classes will double. Similarly, if the oversampling percentage is 200, the total number of samples in the minority
class will triple [7].

After SMOTE Oversampling the features are in the ratio 8:8

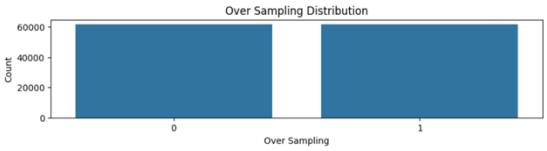

Figure 3: Oversampled Class Label distribution after SMOTE

5.2 Under Sampling

Under-sampling is the method of removing some of the majority class so it has less effect on the machine learning algorithm.

Neighbourhood Cleaning Rule

Editing the close distance removes samples from most categories that differ from one of the nearest neighbours. Sieves can be repeated, which is the principle of repeatedly editing adjacent distances. By changing the parameters of the internal nearest neighbour algorithm and increasing it at each iteration, all KNNs are slightly different from repeated edited nearest neighbors.

The compressed nearest neighbour uses 1-NN iteration to determine whether the sample should be kept in the data set.

The problem is that compressed nearest neighbours are sensitive to noise by retaining noise samples. One-sided selection also uses 1-NN and removes noise samples using Tomek links. The neighbour cleaning rule deletes some samples using the edited nearest neighbour distance. In addition, they use the 3 nearest neighbours to remove samples that are inconsistent with the rules.
After Neighbourhood Cleaning Rule Under sampling the features are in the ratio 2:6.

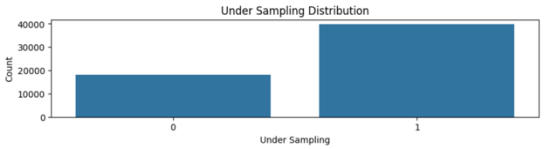

Figure 4: Under Sampling using Neighborhood Cleaning Rule

5.3 Balanced Sampling

Balanced Sampling is defined as Hybrid sampling, which is the mixture of both oversampling and under-sampling techniques.

SMOTE Tomek

Tomek's link and edited nearest neighbors are the two cleaning methods that have been added to the pipeline after applying SMOTE over-sampling to obtain a cleaner space. The ready-to-use class imbalanced-learn implements for combining over- and under-sampling methods is SMOTE Tomek [8].
After SMOTE Tomek Balance Sampling the features are in the ratio 6:6.

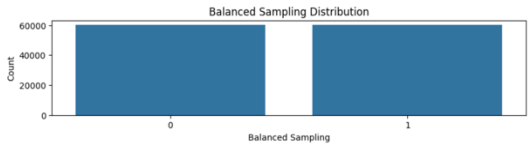

Figure 5: Balanced sampling using SMOTE Tomek

6. Feature Selection

Feature selection is one of the core concepts in machine learning, which greatly affects the performance of the model. The data capabilities we use to train your machine learning models have a huge impact on the performance you can achieve. Unrelated or partially related features may have a negative impact on model performance. Feature selection is the process by which you automatically or manually select those features that contribute the most to the predictor or output you are interested in.
The best feature of the feature selection algorithm is to reduce overfitting, improve accuracy, and reduce training time.

IV. MODEL EVALUATION

A. Linear Model

Linear Model is a supervised machine learning strategy that makes classification decisions based on the value of a linear combination of the characteristics.
If the input feature vector to the classifier is a real vector x, then the output score is

$$y = f(\vec{w} \cdot \vec{x}) = f\left(\sum_j w_j x_j\right),$$

Figure 6: Linear Model

Where w is vector weight and function f is a function that converts the dot product of two vectors into the desired output. And the weights are learned from labeled training samples [9].

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} > T, \\ 0 & \text{otherwise} \end{cases}$$

Figure 7: Linear Model Output Evaluation

Naïve Bayes

Naïve Bayes classifier is a probabilistic linear machine learning model that's used for classification task. The crux of the classifier is based on the Bayes theorem.

$$p(C_k \mid \mathbf{x}) = \frac{p(C_k)\, p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$$

Figure 8: Bayes Formula

Using Bayes theorem, we can find the probability of A happening, given that B has occurred. Here, B is the evidence and A is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one feature does not affect the other. Hence it is called naïve [10].

Linear Discriminant Analysis (LDA)

LDA operates under the assumption that the features are normally distributed within each class. It models the distribution of the features for each class separately. Using these distributions, it can calculate the likelihood of a new data point belonging to each class. Use Bayes theorem to flip things around and obtain
When we use normal (Gaussian) distributions for each class, this leads to linear or quadratic discriminant analysis. [11]

B. Tree Model

Decision Tree

Decision tree learning is to construct a decision tree from labeled training tuples. A decision tree is a structure like a flowchart, where each internal (non-leaf) node represents a test of an attribute,
each branch represents the test result, and each leaf (or terminal) node has a class label. [12]

C. Distance Model

k-NN:

In k-NN classification, the output is a class member. An object is classified by multiple votes of its neighbors, and the object is assigned to the most common category among its nearest k neighbors (k is a positive integer, usually small). If k =1, simply assign the object to the class of that single nearest neighbor. [13]

Raw Data:

Decision Tree

[[1944 2599]
[2437 12979]]

Naive Bayes

[[895 3648]
[14 15402]]

LDA Algo

[[898 3645]
[1 15415]]

K-NN Algo

[[1876 2667]
[2117 13299]]

	decisiontree	naive	LDA	knn
Accuracy	0.747	0.8165	0.8173	0.7603
Precision	0.833	0.8085	0.808	0.833
Recall	0.842	0.999	0.999	0.862
F1 Score	0.8375	0.893	0.894	0.8475

Over Sampling:

Decision Tree

[[2027 2516]
[2563 12853]]

Naive Bayes

[[2984 1559]
[5971 9445]]

LDA Algo

[[2578 1965]
[4247 11169]]

K-NN Algo

[[2206 2337]
[2857 12559]]

	decisiontree	naive	LDA	knn
Accuracy	0.745528	0.623	0.688	0.7397
Precision	0.836294	0.858	0.8503	0.8431
Recall	0.833744	0.613	0.7245	0.8146
F1 Score	0.835017	0.7149	0.78241	0.828649

Under-Sampling:

Decision Tree

[[2517 2026]

[4283 11133]]

Naive Bayes

[[979 3564]
[182 15234]]

LDA Algo

[[1014 3529]
[158 15258]]

K-NN Algo
[[2549 1994]
[4934 10482]]

	decisiontree	naive	LDA	knn
Accuracy	0.683902	0.812315	0.815271	0.652888
Precision	0.846037	0.810405	0.812157	0.840173
Recall	0.722172	0.988194	0.989751	0.679943
F1 Score	0.779213	0.890513	0.892202	0.751613

Balanced- Sampling

Decision Tree

[[1973 2570]
[2562 12854]]

Naive Bayes

[[3011 1532]
[6087 9329]]

LDA Algo

[[2532 2011]
[4152 11264]]

K-NN Algo

[[2171 2372]
[2963 12453]]

	decisiontree	naive	LDA	knn
Accuracy	0.742873	0.618267	0.691217	0.733
Precision	0.833377	0.858945	0.848512	0.84
Recall	0.833809	0.60515	0.730669	0.808
F1 Score	0.833593	0.710051	0.785194	0.824

V. EVALUATION TECHNIQUES

A. Train test split

The concept of test sequence segmentation is a technique used in machine learning to divide a data set into training and test data. This can be done easily by passing a panda data frame to the train_test_split () function or the scikit learning library. We will need to provide the proportion of the dataset that needs to be split. Based on experience or the most used convention, we use 80: 20 as the ratio of dividing the data set into training and test data sets. Similarly, we can try to build models for other ratios and observe performance. The test sequence split also provides an option to specify something called "random state", which is just an integer.
This helps ensure that we don't get a different training and test dataset each time we split the dataset. The following is a count of split test and training data sets.

B. ROC Curve

The receiver operating characteristic curve or ROC curve is a graphical diagram that illustrates the diagnostic capabilities of the binary classifier system when the discrimination threshold changes.
ROC curves were drawn by plotting the true positive rate (TPR) and False positive rate (FPR) at various threshold settings. The true positive rate is also called sensitivity, recall rate, or detection probability in machine learning. The false positive rate is also called false alarm probability and can be calculated as (1- specificity).

Raw Data

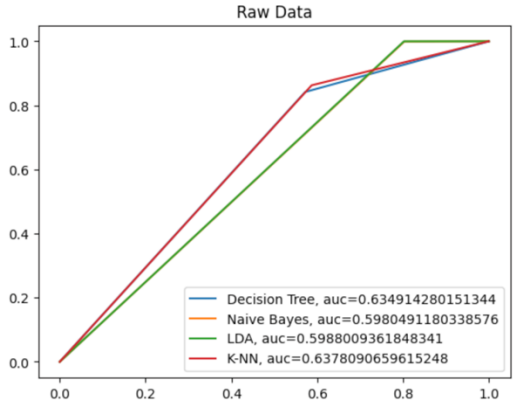

Figure 9: ROC for Raw Data

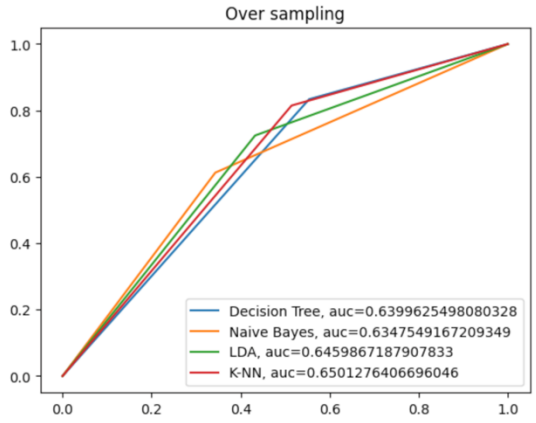

Figure 10: ROC for Over Sampled Data

Under sampled Balanced sampled

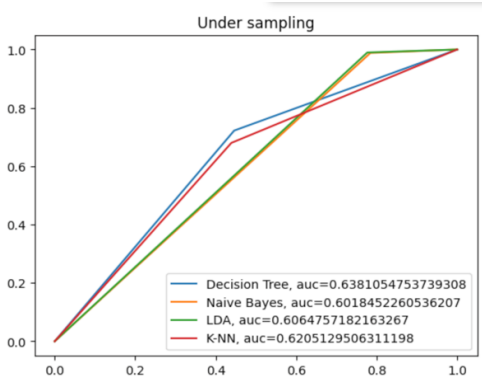

Figure 11: ROC for Under Sampled Data

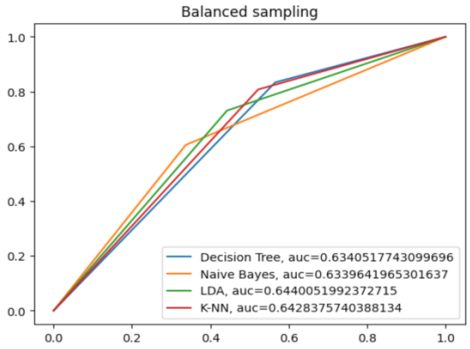

Figure 12: ROC for Balanced Sampled Data

C. Confusion Matrix

A table of confusion (sometimes also called a confusion matrix), is a table with two rows and two columns that reports the number of *false positives*, *false negatives*, *true positives*, and *true negatives*. [14]

		predicted	
actual		negative	positive
	negative	TN True positive	FP False Positive
	positive	FN False negative	TP True positive

Figure 13: Confusion Matrix

D.Accuracy

When we use the term accuracy, we usually mean classification accuracy. It is the ratio of the number of correct predictions to the total number of input samples. It works only if the number of samples belonging to each category is equal. [15]

Accuracy:

$$AC = \frac{TN + TP}{TN + FP + FN + TP}$$

Figure 14: Accuracy

E. Precision

Precision (P) is defined as the number of true positives (TP) over the number of true positives plus the number of false positives (FP). [16]

Precision:

$$precision : \frac{TP}{FP + TP}$$

Figure 15: Precision

F. Recall

Recall (R) is defined as the number of true positives (TP) over the number of true positives plus the number of false negatives (FN).

Recall aka. True Positive Rate:

$$recall = \frac{TP}{FN + TP}$$

Figure 16: Recall

G.F1-score

F1-score is defined as the harmonic mean of precision and recall.

$$F_1 = \left(\frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} \right) = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}.$$

Figure 17: F1-Score

VI. RESULTS

For the Bank Loan Data, over-sampling tends to improve performance metrics like AUC while under-sampling can be more variable. Balanced sampling methods generally maintain performance comparable to raw data scenarios.

VII. LESSONS LEARNED

Ability to understand a machine Learning problem and categorize them based on the problem statement. Capabilities to Load and preprocess the raw data and convert it into machine learning algorithm compatible. Understand the importance of feature engineering and feature selection on the model accuracies. Got to understand different model evaluation measures and statistical testing.

VIII. REFERENCES

[1] [Online]. Available: https://www.ijrte.org/wp-content/uploads/papers/v7i4s/E2026017519.pdf

[2] [Online]. Available: https://www.simple.com/blog/how-do-banks-work

[3] [Online]. Available: https://www.kaggle.com/omkar5/dataset-for-bank-loan- prediction

[4] [Online]. Available: https://medium.com/@urvashilluniya/why-data-normalization-is-necessary-for-machine-learning-models-681b65a05029#:~:targetText=Normalization%20is%20a%20technique%20often,dataset%20does%20not%20require %20normalization

[5] [Online]. Available: https://en.wikipedia.org/wiki/Dimensionality_reduction#:~:targetText=In%20statistics%2C%20machine%20learning%2C%20and,feature%20selection%20and%20feature%20 extraction

[6] [Online]. Available: https://www.jeremyjordan.me/imbalanced-data/

[7] [Online]. Available: https://medium.com/towards-artificial-intelligence/application-of-synthetic-minority-over- sampling-technique-smote-for-imbalanced-data-sets- 509ab55cfdaf

[8] [Online]. Available: https://imbalanced-learn.readthedocs.io/en/stable/auto_examples/under-sampling/plot_comparison_under_sampling.html

[9] [Online]. Available: https://en.wikipedia.org/wiki/Linear_classifier#:~:targetText=In%20the%20field%20of%20machine,linear%20combination%20of%20the%20characteristics

[10] [Online]. Available: https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c

[11] [Online]. Available: https://learn-ca-central-1-prod-fleet01xythos.content.blackboardcdn.com/5e99d3f427e82/24386702?X-Blackboard-S3-Bucket=learn-ca-central-1-prod-fleet01-xythos&X-Blackboard-Expiration=1718938800000&X-Blackboard-Signature=ICf3BB9pqYJ8TvJVqxrVqiHT99rxZ3T7ZdES

[12] [Online]. Available: https://en.wikipedia.org/wiki/Decision_tree_learning

[13] [Online]. Available: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

[14] [Online]. Available: https://en.wikipedia.org/wiki/Decision_tree_learning

[15] [Online]. Available: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

[16] [Online]. Available: https://learn-ca-central-1-prod-fleet01-xythos.content.blackboardcdn.com/5e99d3f427e82/24386702?X-Blackboard-S3-Bucket=learn-ca-central-1-prod-fleet01-xythos&X-Blackboard-Expiration=1718938800000&X-Blackboard-Signature=ICf3BB9pqYJ8TvJVqxrVqiHT99rxZ3T7ZdES