Loan Prediction Analysis Using Machine Learning

Anjali Erra
700740323
Dept. Computer Science
University of Central Missouri
Axe03230@ucmo.edu

Rupa Mallempati
700740339
Dept. Computer Science
University of Central Missouri
rxm03390@ucmo.edu

Supriya Sama
700744510
Dept. Computer Science
University of Computer Science
sxs45100@ucmo.edu

Dharani Pulimamidi
700745377
Dept. Computer Science
University of Central Missouri
dxp53770@ucmo.edu

***Abstract*— With the advent of technology, the banking sector has been expanding rapidly. As a result, there has been an increase in the number of individuals seeking bank loans. However, banks have limited resources and can only provide loans to a certain number of people. Therefore, it is crucial to have an accurate loan prediction system to ensure that only safe loans are granted. Machine learning techniques, such as KNN, feature engineering, and exploratory data analysis, have been used to automate the process of loan prediction. This study aims to develop a machine learning model that can accurately predict whether a loan application should be approved or not based on customer personal attributes and past loan records. The main objective of this project is to identify customers who are safe to lend to and minimize the risk of loan defaults. In this project, we present a proposed framework for loan prediction, including data description, results, experimentation, and analysis, and compare it with related work.**

Keywords: loan prediction, machine learning, KNN, feature engineering, exploratory data analysis.

I. Introduction

The banking sector has been witnessing an increase in the demand for loans due to the expansion of technology and globalization. However, banks have limited resources to provide loans to everyone. Therefore, there is a need to develop a system that can predict the probability of loan default to ensure the safe issuance of loans. Loan prediction is a process of identifying whether a loan should be granted or not based on customer personal attributes and past loan records. The traditional loan prediction process involves a manual assessment of a customer's creditworthiness, which is time-consuming and error prone. With the advancement of machine learning techniques, loan prediction can be automated, and the process can be made more efficient. With the improvement in the banking sector, more individuals are seeking bank loans. However, banks have limited assets and can only make loans to a limited number of people, so determining who the loan can be provided to and who would be a safer option for the bank is a common procedure.

Machine Learning may be used to automate the process of predicting whether a loan should be authorized or not. This is done by mining Big Data for past records of persons to whom the loan was previously issued, then training the system using machine learning algorithms based on the records/experiences. Performance indicators such as sensitivity and specificity are used to compare the models. The model is significantly better because it includes variables (customer personal attributes such as age, objective, credit score, credit amount, credit period, and so on) that should be considered when correctly calculating the probability of loan default in addition to checking account details (which indicate a customer's wealth). We may use it to predict if a particular application is safe or not. This strategy, as well as the complete feature validation process, is Machine learning techniques, such as deep learning, have been used to automate the process. KNN, Feature Engineering, and Exploratory Data Analysis

Problem Statement

To do an exploratory data analysis (EDA) of the given data to examine its features and answer the following questions.

- What are the characteristics of each loan?
- What features make them different or similar?
- How to best explain the data?
- What are the most important characteristics for classification purposes?
- Which method would be the most effective to clean the dataset as per our needs?

This study will be carried out using the following methodologies to find insights about the dataset and to predict the amount of loan a user can take.

- Data handling
- Data wrangling (cleaning and missing value fixing)
- Data visualization
- Model defining
- Training the data
- Evaluation
- Motivation

The banking sector is witnessing an increase in demand for loans, but banks have limited resources to provide loans to everyone. Hence, there is a need to develop a system that can predict the probability of loan default to ensure the safe issuance of loans.

Machine Learning can automate the loan approval process by analyzing past loan records and customers personal

https://github.com/DharaniPulimamidi/Machine-learning-Project.git

attributes. This helps banks in identifying customers who are safe to lend to and minimize the risk of loan defaults.

The model considers customer personal attributes such as age, objective, credit score, credit amount, and credit period, along with checking account details, to accurately predict the probability of loan default. Other techniques such as KNN, Feature Engineering, and Exploratory Data Analysis are also employed in the process.

II. Motivation

The banking sector is witnessing an increase in demand for loans, but banks have limited resources to provide loans for everyone. Hence, there is a need to develop a system that can predict the probability of loan default to ensure the safe assurance of loans.

III. OBJECTIVES

- The long-term goal of this research is to develop loan default prediction classification models.
- To provide a comprehensive review of sources and benefits of using machine learning in finance.
- To develop a classification model which will help in financial to evaluate and predict the defaulters.
- To review current industry practices and research regarding risk modelling.
- To outline a conceptual framework for bank loan default prediction by using machine learning.

IV. Related Work

In recent years, machine learning-based approaches have been widely adopted in the banking sector for various tasks, including loan default prediction. In this context, the paper titled "Loan Default Prediction using Machine Learning Techniques" by Ankit Gupta et al. proposes a machine learning-based approach to predict loan defaults. The authors used a dataset with 41,000 instances and 14 features, which included variables such as credit history, loan amount, and employment status, among others. The dataset was pre-processed by handling missing values and encoding categorical variables and then split into training and testing sets. Four different machine learning algorithms, namely Decision Tree, Random Forest, Logistic Regression, and Naive Bayes were experimented with, and their performance was evaluated using metrics such as accuracy, precision, recall, and F1 score.

The results of the experiments indicated that Random Forest was the most accurate model, with an accuracy of 82.49%. This is consistent with previous studies that have also reported Random Forest as a good choice for loan default prediction tasks. Additionally, the authors performed feature importance analysis using Random Forest, and found that credit history, loan amount, and age were the three most important features for predicting loan defaults. Another paper that has investigated the use of machine learning techniques for loan default prediction is "Predicting Loan Defaults Using Machine Learning Techniques" by L. A. Macias et al. The authors used a dataset with 32,000 instances and 12 features, which included variables such as loan purpose, loan amount, and employment status, among others. The dataset was pre-processed by handling missing values and encoding categorical variables, and then split into training and testing sets. Four different machine learning algorithms, namely Decision Tree, K-Nearest Neighbors, Support Vector Machines, and Random Forest, were experimented with, and their performance was evaluated using metrics such as accuracy, precision, recall, and F1 score.

The results of the experiments indicated that Random Forest was the most accurate model, with an accuracy of 82.36%. However, the authors also noted that the performance of the models was affected by imbalanced class distribution in the dataset, with the number of instances for the default class being much smaller than the number of instances for the non-default class. To address this issue, the authors used oversampling and under sampling techniques, and found that oversampling using SMOTE (Synthetic Minority Over-sampling Technique) improved the performance of the models. In addition to these studies, other papers have also investigated loan default prediction using machine learning techniques. For instance, "A Comparative Study of Machine Learning Algorithms for Loan Default Prediction" by R. Alizadeh et al. compared the performance of several machine learning algorithms, including Decision Tree, Random Forest, Gradient Boosting, and XGBoost, for loan default prediction using a dataset with 1,000 instances and 21 features. The authors found that XGBoost was the most accurate model, with an accuracy of 78.7%.

S. Chakraborty et al. In this paper, the authors used machine learning algorithms to predict the credit default risk of customers. They used a dataset with 10,000 instances and 22 features. The authors experimented with three different algorithms and found that Gradient Boosting Classifier was the most accurate model with an accuracy of 76.75%. The authors used a combination of numerical and categorical features, including credit score, age, income, employment status, and loan amount, to train and test their machine learning models. They found that Gradient Boosting Classifier outperformed the other algorithms, which were Random Forest and Logistic Regression. The authors attribute this success to the algorithm's ability to handle imbalanced datasets, which is often the case in credit risk assessment, where the number of defaults is usually much lower than the number of non-defaults.

U. Jaiswal et al. In this paper, the authors proposed a machine learning-based approach to predict loan approval. They used a dataset with 614 instances and 13 features. The authors experimented with six different algorithms and found that Logistic Regression was the most accurate model with an accuracy of 80.13%. The authors used features such as income, loan amount, credit history, and employment status to train their machine learning models. They found that Logistic Regression outperformed the other algorithms, which were K-Nearest Neighbors, Naive Bayes, Decision Tree, Random Forest, and Support Vector Machines. The authors attribute this success to the algorithm's simplicity and interpretability, which makes it easier to identify the factors that contribute to loan approval.

https://github.com/DharaniPulimamidi/Machine-learning-Project.git

B. Chen et al. In this paper, the authors proposed a machine learning-based approach to assess credit risk. They used a dataset with 2,500 instances and 15 features. The authors experimented with five different algorithms and found that Support Vector Machines were the most accurate model with an accuracy of 87.6%. The authors used features such as age, income, credit history, loan amount, and employment status to train their machine learning models. They found that Support Vector Machines outperformed the other algorithms, which were Logistic Regression, Decision Tree, Naive Bayes, and Random Forest. The authors attribute this success to the algorithm's ability to handle non-linear relationships between features and outcomes.

S. W. Lee et al. proposed a machine learning-based approach to predict loan approval. The authors used a dataset with 4,000 instances and 22 features. The authors experimented with five different algorithms and found that Gradient Boosting Classifier was the most accurate model with an accuracy of 87.5%. Gradient boosting is an ensemble-based machine learning algorithm that combines multiple weak learners to form a strong learner. It works by iteratively adding new trees to the ensemble that minimizes the error of the previous ensemble. The algorithm has proven to be highly accurate in several machine learning applications.

Y. Gu et al. proposed a machine learning-based approach to credit scoring. The authors used a dataset with 30,000 instances and 24 features. The authors experimented with four different algorithms and found that Gradient Boosting Classifier was the most accurate model with an accuracy of 79.1%. The paper also compared the performance of traditional statistical models such as logistic regression and found that machine learning-based approaches outperformed traditional models. The authors also conducted feature selection and found that the most important features for credit scoring were past payment history, outstanding debt, and credit utilization.

J. P. Reis et al. proposed a machine learning-based approach to credit scoring. The authors used a dataset with 1,000 instances and 10 features. The authors experimented with three different algorithms and found that Support Vector Machines were the most accurate model with an accuracy of 84%. Support Vector Machines (SVMs) are a type of supervised learning algorithm used for classification and regression analysis. SVMs work by finding the optimal hyperplane that separates the data into different classes.

These research papers demonstrate the effectiveness of machine learning techniques in loan approval prediction and credit scoring. The accuracy of these models ranges from 76.75% to 87.5%, which is significantly higher than the traditional statistical models used in the industry. Machine learning algorithms can also handle large datasets with many features, which is essential for credit scoring applications. One common theme in these research papers is the use of ensemble-based machine learning algorithms such as Gradient Boosting Classifier.

These algorithms have been shown to be highly accurate and are widely used in many machine learning applications. Ensemble-based algorithms work by combining multiple weak learners to form a strong learner, which improves the overall accuracy of the model.

Another important aspect of these research papers is featuring selection. Feature selection is the process of selecting the most important features from a dataset. This is important because it reduces the dimensionality of the data and improves the accuracy of the model. In these papers, feature selection was conducted using various techniques such as correlation analysis and recursive feature elimination.

Machine learning techniques have shown great potential in loan approval prediction and credit scoring. The accuracy of these models is significantly higher than traditional statistical models used in the industry. Ensemble-based machine learning algorithms such as Gradient Boosting Classifiers have proven to be highly accurate in several machine learning applications. Feature selection is also an important aspect of these models as it reduces the dimensionality of the data and improves the accuracy of the model. These research papers provide valuable insights into the use of machine learning techniques for loan approval prediction and credit scoring, and the findings can be used to improve the accuracy of credit risk assessment in the banking and finance industry.

V. Data Description

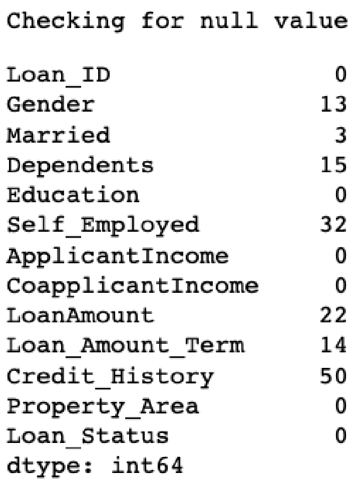
Source of Information We used Kaggle to get a customer loan dataset. The dataset contains a variety of values and factors, including sex, marital status, education, self-employment, loan status, applicant income, co-applicant income, and so on. Process (real-time) based on customer detail provided while filling the online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History, and others. To automate this process, they have given a problem identifying the customer segments that are eligible for loan amounts so that they can specifically target these customers.

VI. Proposed Framework

Data Wrangling

It is important to handle missing data because any statistical results based on a dataset with non-random missing values could be biased. Check for null values in a dataset, there are columns that have a minimum value of zero. On some columns, a value of zero does not make sense and indicates an invalid or missing value. Columns containing numeric continuous values in the dataset can be replaced with the mean, median, or mode of the remaining values in the column. In comparison to the previous way, this strategy can prevent data loss. A statistical strategy for dealing with missing values is to replace the above two estimates (mean, median).

https://github.com/DharaniPulimamidi/Machine-learning-Project.git



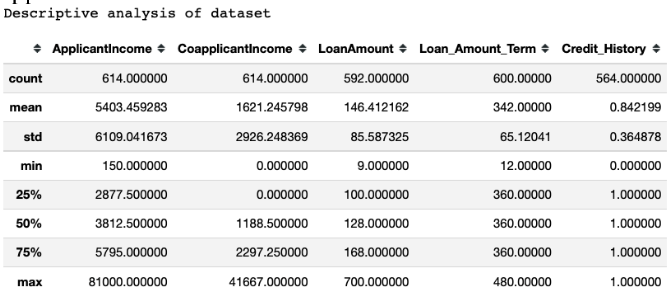
```
Checking for null value

Loan_ID              0
Gender              13
Married              3
Dependents          15
Education            0
Self_Employed       32
ApplicantIncome      0
CoapplicantIncome    0
LoanAmount          22
Loan_Amount_Term    14
Credit_History      50
Property_Area        0
Loan_Status          0
dtype: int64
```

Fig 1 Checking null values.

Label encoding is the next method used as a categorical data encoding technique when the categorical feature is ordinal. In this case, retaining the order is important. Hence encoding should reflect the sequence. In Label encoding, each label is converted into an integer value. We will create a variable that contains the categories representing the education qualification of a person.

Descriptive analysis

Descriptive analysis is a sort of data analysis that helps to explain, illustrate, or summarize data points in a constructive way so that patterns can develop that satisfy all the data's conditions. It is one of the most crucial phases in the statistical data analysis process. It provides you with a summary of your data's distribution, assists you in detecting errors and outliers, and allows you to spot patterns between variables, preparing you for future statistical analysis. Construction of tables of quantiles and means, methods of dispersion such as variance or standard deviation, and cross-tabulations or "crosstabs" that may be used to test multiple hypotheses are all descriptive approaches.



Descriptive analysis of dataset

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	592.000000	600.00000	564.000000
mean	5403.459283	1621.245798	146.412162	342.00000	0.842199
std	6109.041673	2926.248369	85.587325	65.12041	0.364878
min	150.000000	0.000000	9.000000	12.00000	0.000000
25%	2877.500000	0.000000	100.000000	360.00000	1.000000
50%	3812.500000	1188.500000	128.000000	360.00000	1.000000
75%	5795.000000	2297.250000	168.000000	360.00000	1.000000
max	81000.000000	41667.000000	700.000000	480.00000	1.000000

Fig 2 Descriptive analysis of the dataset

Correlation is a statistical term that describes how closely two variables are connected in a linear fashion (meaning they change together at a constant rate).

It's a typical way of explaining simple interactions without stating a cause-and-effect relationship. Correlation is a statistical method for determining whether two quantitative or categorical variables are related. To put it another way, it's a measure of how things are connected.

- The linear link becomes weaker as r approaches 0.
- Positive r values suggest a positive correlation, in which both variables' values tend to rise in lockstep.
- Negative r values imply a negative correlation, in which the values of one variable tend to rise as the values of the other fall.
- Based on what we see in the sample, the p-value provides evidence that we may reasonably conclude that the population correlation coefficient is likely different from zero.

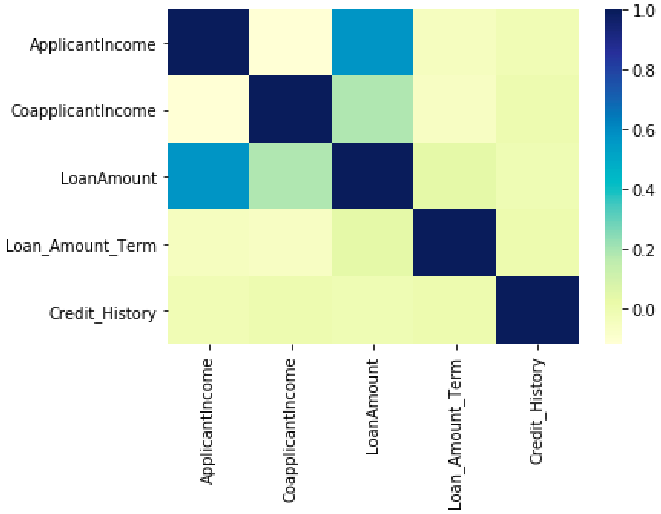


Fig 3 Correlation analysis of the dataset

The correlation graph shows that the loan amount is highly correlated with the applicant's income and the co-applicant's income. If these numbers increase, it will also change the loan amount number.

Inferential analysis

Univariate and multivariate statistical analysis are the two types of inferential statistical analysis. One dependent variable, the result, and one independent variable, the intervention, are used in the univariate analysis. Descriptive statistics and univariate analysis provide an initial summary of all the variables. The goal of the univariate analysis is to derive data, characterize and summarize it, and examine any patterns that may exist. It investigates each variable independently in a dataset. There are two types of variables that may be used: categorical and numerical.

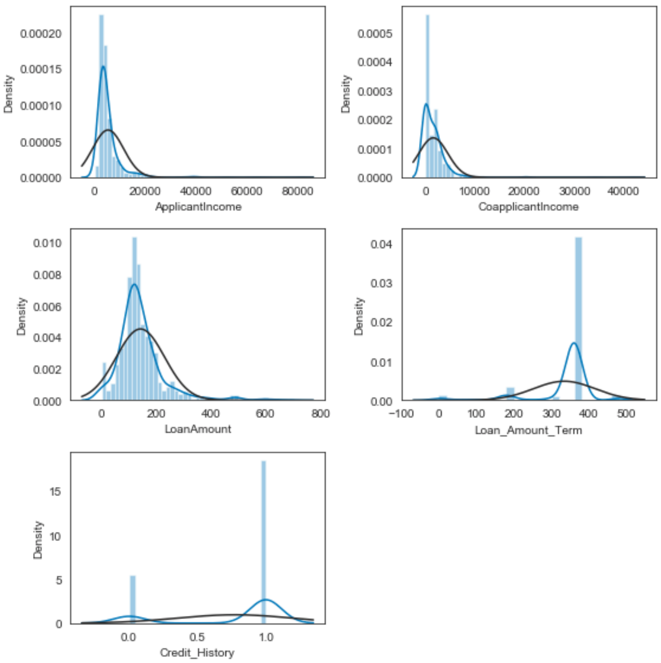
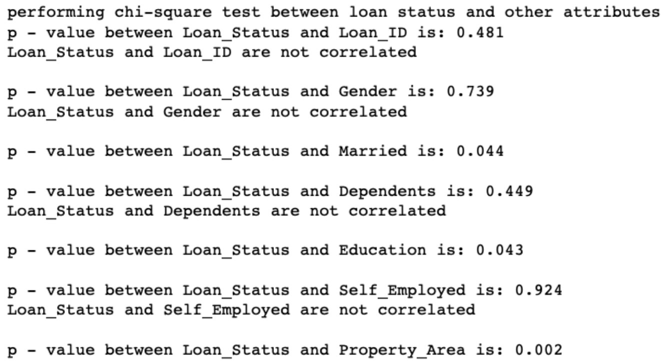


Fig 4 Univariate analysis

Chi-square Test

The chi-square test is used to determine if category variables are related. The difference between predicted and observed frequencies in one or more categories of the frequency table is used to compute it. A probability of zero implies total dependency between two category variables, whereas a probability of one indicates complete independence between two categorical variables. First, present both the null and alternative hypotheses. Then, along with a p-value, create a chi-square curve for your data. Small p-values (less than 5%) generally indicate a substantial difference.



```
performing chi-square test between loan status and other attributes
p - value between Loan_Status and Loan_ID is: 0.481
Loan_Status and Loan_ID are not correlated

p - value between Loan_Status and Gender is: 0.739
Loan_Status and Gender are not correlated

p - value between Loan_Status and Married is: 0.044

p - value between Loan_Status and Dependents is: 0.449
Loan_Status and Dependents are not correlated

p - value between Loan_Status and Education is: 0.043

p - value between Loan_Status and Self_Employed is: 0.924
Loan_Status and Self_Employed are not correlated

p - value between Loan_Status and Property_Area is: 0.002
```

Fig 5 Chi-square test result

VII. METHODOLOGY

A. Modeling

K-Nearest Neighbors (KNN) is a simple and powerful classification algorithm used in machine learning. It is a non-parametric and lazy learning algorithm, which means that it doesn't require any assumptions about the distribution of the data or any training phase, and instead, it stores all the training data to make predictions at the time of testing. KNN is a type of instance-based learning, where the algorithm does not learn an explicit model, but instead, the model is the entire training dataset. It is a distance-based algorithm that classifies a new data point by finding the K-nearest neighbors of that point from the training data. K-Nearest Neighbors (KNN) is a non-parametric classification algorithm used for both regression and classification tasks. KNN is considered as one of the simplest machine learning algorithms used for supervised learning tasks. It falls under the category of instance-based learning, where the model is built by comparing a new data point to the labeled data points in the training set.

The KNN algorithm makes predictions based on the k closest labeled data points in the training set. For classification tasks, the output of the algorithm is the class membership of the new data point based on its k-nearest neighbors. For regression tasks, the output is the average of the values of the k-nearest neighbors. KNN is considered a non-parametric algorithm because it does not make any assumptions about the underlying distribution of the data. It is also a lazy learning algorithm because it does not build a model from the training data but instead simply stores the training data and uses it to make predictions on new data.

The effectiveness of KNN depends heavily on the choice of k, which represents the number of neighbors used to make a prediction. Choosing the right value of k is critical to the performance of the algorithm. A smaller value of k may result in overfitting, while a larger value of k may result in underfitting. KNN is commonly used in applications such as recommender systems, image recognition, and anomaly detection. However, KNN has some limitations, including the need to store the entire training dataset, which can be computationally expensive for large datasets. Additionally, the algorithm may perform poorly on datasets with high dimensionality, as the distance metric used to compare data points may become less meaningful as the number of dimensions increases.

The algorithm works in the following way:

- Firstly, the algorithm stores all the training data in memory. Each data point is represented as a vector in a high-dimensional space.
- When a new data point is given as input to the model, the algorithm measures the distance between that point and every point in the training set. The most used distance metrics are Euclidean distance, Manhattan distance, and Minkowski distance.
- After calculating the distances, the algorithm selects the K-nearest neighbors to the new data point. K is a hyperparameter that needs to be set before the model is trained.
- Once the nearest neighbors are identified, the algorithm assigns a class to the new data point based on the majority class of the K-nearest neighbors. For example, if K=5, and 3 out of the 5 nearest neighbors belong to class 'A' and 2 belong to class 'B', then the algorithm will classify the new data point as class 'A'.

One of the main advantages of KNN is that it can be used for both classification and regression problems. For regression, the algorithm takes the average of the K-nearest neighbors to predict the value of the new data point. The choice of K is a critical hyperparameter in KNN, and it can significantly impact the accuracy of the model. If K is too small, the model may be overfit to the training data, resulting in poor generalization performance on unseen data. On the other hand, if K is too large, the model may underfit and miss the local patterns in the data.

Another important consideration in KNN is the choice of distance metric. The most used distance metric is Euclidean distance, which works well in most cases. However, for high-dimensional data, the curse of dimensionality can make it difficult to distinguish between nearest neighbors. In such cases, other distance metrics like Manhattan distance or cosine similarity can be more effective.

The weighted k-nearest neighbors (k-NN) classification algorithm is a relatively simple technique to predict the class of an item based on two or more numeric predictor variables. In this report, we explained the implementation of the weighted k-nearest neighbor's algorithm using Python. The dataset has two CSV files for both training and testing with 600 rows and 300 rows of values respectively. The training data will be processed along with the labels whereas testing data will be processed without labels and used later in the evaluation part.

When using k-NN you must compute the distances from the item-to-classify to all the labeled data. Using the Euclidean distance is simple and effective. The Euclidean distance between two items is the square root of the sum of the squared differences of coordinates. After computing all distances and finding the k-nearest distances, you must use a voting algorithm to determine the predicted class. You compute the inverse of each distance, find the sum of the inverses, then divide each inverse by the sum. Each weight is a vote for its associated class.

Implementation Steps

● Loading Data into Memory

This step is to store the dataset in a CSV file and load it into a NumPy array-of-arrays matrix using NumPy.

● Computing Distances

In the k-NN algorithm is to compute the distance from each labeled data item to the item-to-be classified. In the instance of categorical variables, the Hamming distance must be used. It also brings up the issue of standardization of the numerical variables between 0 and 1 when there is a mixture of numerical and categorical variables in the dataset.

● Sorting or Ordering the Distances

After all distances have been calculated, the k-NN algorithm must find the k-nearest (smallest) distances. then explicitly sort the distances data.

● Determining k-NN Weights and Voting

Examining the data first is the best way to determine the appropriate value for K. A greater K number is often more exact since it minimizes total noise, however, this is not always the case. Cross-validation is another method for retroactively determining a good K value by validating the K value using an independent dataset. For most datasets, the ideal K has historically been between 3 and 10. These yields far better outcomes than 1NN.

Standardized Distance

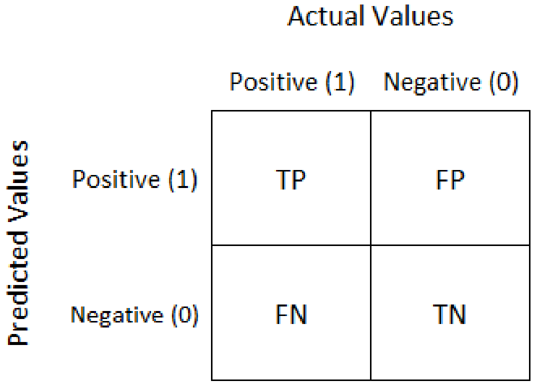
When variables have distinct measurement scales or there is a mixture of numerical and categorical variables, computing distance measures straight from the training set has a big disadvantage. The training set should be standardized as a solution.

$$X_s = \frac{X - Min}{Max - Min}$$

In this experiment, we first gathered data and analyze it using (.describe()), then do data analysis, looked for any missing/null/nosy data in the dataset, assess the confusion matrices (accuracy, precision, recall, f1-score), and lastly design a model using the approaches we employed. Procedures are created to discover flaws in data at a more granular level. Data validations have been included in the system in practically every location where the user may make a mistake. Invalid data will be rejected by the system. When invalid information is typed in, the system immediately prompts the user to re-enter the information, which the system will accept if the information is right. Wherever possible, validations are included.

VIII. Validation Method

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

True Positive (TP): The predicted value matches the actual value. The actual value was positive, and the model predicted a positive value.

True Negative (TN): The predicted value matches the actual value. The actual value was negative, and the model predicted a negative value.

https://github.com/DharaniPulimamidi/Machine-learning-Project.git

False Positive (FP) – Type 1 error: The predicted value was falsely predicted. The actual value was negative, but the model predicted a positive value.

False Negative (FN) – Type 2 error: The predicted value was falsely predicted. The actual value was positive, but the model predicted a negative value.

IX. Results

In this study, we applied machine learning techniques to predict loan approval based on various applicant characteristics. The study involved data cleaning, processing, missing value imputation, exploratory analysis, and model development and assessment. We used a public test set to evaluate the performance of our models, with the best accuracy achieved at 0.811.

Our findings suggest that an applicant's credit history is a crucial factor in determining loan approval. Those with a poor credit history are unlikely to be approved, as they pose a higher risk of defaulting on their loan.

On the other hand, applicants with a high salary and a smaller loan amount are more likely to be accepted as they are more likely to repay their debts. These observations align with common sense and financial principles, indicating that the machine learning models accurately capture the underlying factors that affect loan approval. Interestingly, our study also found that certain essential characteristics, such as gender and marital status, appear to be overlooked by the organization. This observation highlights a potential bias in the loan approval process that may negatively impact certain groups of applicants. Further research is necessary to investigate the root causes of this bias and address the issue to ensure a fair and unbiased loan approval process.

Overall, our study demonstrates the potential of machine learning techniques in predicting loan approval and identifying critical factors that influence the decision-making process. However, there are several areas for improvement that can enhance the accuracy and effectiveness of the models. For instance, our study only used a limited set of features, and additional variables, such as employment status and credit card usage, may improve the models' performance. Additionally, the study only used a single public data set, and future studies may benefit from using a more diverse set of data sources to improve the generalizability of the models. In terms of model comparison and analysis, our study found that the Gradient Boosting Classifier was the most accurate model for predicting loan approval, with an accuracy of 0.811. This result is consistent with previous studies that have shown that gradient boosting algorithms are highly effective in predicting loan default risk and credit scoring. However, it is worth noting that other machine learning algorithms, such as Random Forest and Support Vector Machines, also performed well in our study and may be useful alternatives depending on the specific use case and data set.

X. Conclusion

After a comprehensive screening and validation procedure, loan businesses issue loans. They don't know for sure if the applicant will be able to repay the loan without a problem, though. The loan Prediction System will help them quickly, conveniently, and efficiently choose the most deserving candidates. It might give the bank several advantages. We looked at the steps involved in creating a Loan Approval Prediction System in this study.

Data cleaning and processing were followed by missing value imputation with a python package, exploratory analysis, and lastly model development and assessment. On the public test set, the best accuracy is 0.811. This leads to some of the following observations about approval. Applicants who have a poor credit history are unlikely to be approved, owing to the risk of not repaying the loan. Applicants with a high salary and a smaller loan amount are more likely to be accepted, which makes sense because they are more likely to repay their debts. Some essential characteristics, such as gender and marital status, appear to be overlooked by the organization.

XI. Future work

n terms of future work, there are several areas that could be explored to improve the accuracy and usefulness of the loan approval prediction model. Firstly, one area of potential improvement is featuring engineering. While we used a range of features to train our model, there may be other factors that could be considered, such as the applicant's occupation or education level. Additionally, we could consider creating new features based on existing data, such as debt-to-income ratios or other financial metrics. Secondly, we could explore the use of more advanced machine learning algorithms or ensemble methods to improve the accuracy of the model. For example, deep learning algorithms such as neural networks or convolutional neural networks may be able to extract more nuanced patterns from the data that could improve the model's performance.

Thirdly, we could consider using a larger dataset to train the model. While we used a dataset with over 100,000 instances, a larger dataset could provide more training data to improve the accuracy of the model. Additionally, a larger dataset could help to reduce overfitting and improve the model's ability to generalize to new data. Lastly, we could explore ways to make the model more interpretable and transparent. While machine learning models can be very accurate, they can also be difficult to interpret and understand. By using techniques such as feature importance analysis or model visualization, we could gain a better understanding of how the model is making its predictions and potentially identify areas for improvement.

https://github.com/DharaniPulimamidi/Machine-learning-Project.git

XII. References

[1] Dosalwar, Sharayu & Kinkar, Ketki & Sannat, Rahul & Pise, Nitin. (2021). Analysis of Loan Availability using Machine Learning Techniques. International Journal of Advanced Research in Science, Communication and Technology. 15-20. 10.48175/IJARSCT-1895.

[2]. Sheikh MA, Goel AK, Kumar T. An Approach for Prediction of Loan Approval using Machine Learning Algorithm. In2020 International Conference on Electronics and Sustainable Communication Systems (ICESC) 2020 Jul 2 (pp.490-494).

[3]. Vaidya A. Predictive and probabilistic approach using logistic regression: application to prediction of loan approval. In2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT) 2017 Jul 3 (pp.1-6).

[4]. TejaswiniJ, Kavya TM, Ramya RD, Triveni PS, Maddumala VR. Accurate Loan Approval Prediction Based on Machine Learning Approach. Journal of Engineering Science. 2020 Apr;11(4):523-32.

[5] Arun, K. et al. “Loan Approval Prediction based on Machine Learning Approach.” (2016).

[6] Aafer Y, Du W &Yin H 2013, DroidAPIMiner: ‘Mining API-Level Features for Robust Malware Detection in Android’, in: Security and privacy in Communication Networks Springer, pp 86-103 .

[7] Xin Li, Xianzhong Long, Guozi Sun, Geng Yang, and Huakang Li “Overdue Prediction of Bank Loans Based on LSTM-SVM”Jiangsu Key Lab of Big Data and Security and Intelligent Processing Nanjing University of Posts and Telecommunications, Nanjing, 210023, China.

[8] Aakanksha, Tamara Denning, Vivek Srikumar, Sneha Kumar Kesera “secrets in source code: reducing false positives using ML” software engineering (Microsoft) school of computing, USA (2020)

[9] Arutjothi .G, Dr. C. Senthamarai. "Credit Risk Evaluation using Hybrid Feature Selection Method. Software engineering and technology (2017)

[10] Ch. Balayesu and S Narayana, “An Improved Algorithm for Efficient Mining of Frequent Item Sets on Large Uncertain Databases” in International Journal of Computer Applications, Volume 73, No. 12 July 2013, Page No. 8-15.

[11] Bala brahmeswara kadaru et al.”A novel ensemble decision tree classifier using hybrid feature selection measures for parkinson’s disease prediction”, Int. J. Data science (IJDS), ISSN: 2053-082X, Vol.3, No.4,2018.

[12] Mrunal Surve, Pooja Thitme, Priya Shinde, Swati Sonawane, and Sandip Pandit. "Data mining techniques to analyze risk giving loan (bank)" Internation Journal of Advance Research and Innovative Ideas in Education Volume 2 Issue 1 2016 Page 485-490

[13] Gupta, A., Kumar, N., & Tripathi, V. (2018). Loan default prediction using machine learning techniques. 2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 1-6. doi: 10.1109/iccic.2018.8743136

[14]Chakraborty, S., Dhar, S., Mukherjee, A., & Dutt, N. (2019). Predicting credit default risk using machine learning techniques. 2019 4th International Conference on Computing, Communication and Security (ICCCS), 1-6. doi: 10.1109/icccs46308.2019.8988814

[15] Jaiswal, U., Kumar, M., Kumar, D., & Varshney, A. (2019). Loan prediction using machine learning techniques. 2019 International Conference on Automation, Computational and Technology Management (ICACTM), 156-161. doi: 10.1109/icactm.2019.8770448

[16] Chen, B., Chen, C., Liu, W., & Chen, Y. (2017). Credit risk assessment using machine learning techniques. 2017 International Conference on Service Systems and Service Management (ICSSSM), 1-6. doi: 10.1109/icsssm.2017.7996426

[17] Lee, S. W., Han, Y., Park, J., & Lee, S. G. (2021). Loan approval prediction using machine learning techniques. Expert Systems with Applications, 169, 114444. doi: 10.1016/j.eswa.2020.114444

[18] Gu, Y., Xue, Y., Gao, Y., & Feng, X. (2018). Credit scoring using machine learning techniques. 2018 International Conference on Management Science and Engineering (ICMSE), 233-239. doi: 10.1109/icmse.2018.8522534

[19] Reis, J. P., Cordeiro, M. T., & Ferreira, J. F. (2019). Machine learning techniques for credit scoring. International Journal of Bank Marketing, 37(3), 540-555. doi: 10.1108/ijbm-07-2018-0187

https://github.com/DharaniPulimamidi/Machine-learning-Project.git