A picture containing text, clock, sign

Description automatically generated

**ADTA 5230 – Data Analytics II**

**Dr. Mehmet Orhan**

**Final Project Report on Credit Risk Analyzer**

**By: Vamshi Krishna Vallam**

**Abstract**

The credit risk analyzer is a machine learning model designed to predict whether a borrower is likely to default on a loan or not. The model is built using a dataset of borrower data from a lending institution. The dataset is pre-processed to handle missing values, categorical data, and outliers. Feature selection is performed to identify the most important features for predicting credit risk. Several classification models are built and evaluated using metrics such as accuracy, sensitivity, and specificity. This model can be used by lending institutions to assess the credit risk of potential borrowers and make informed decisions about loan approvals. Future work includes incorporating additional features, exploring deep learning models, and performing more in-depth analyses of false positives and false negatives to improve the model's performance.

Future work can involve incorporating additional features such as borrower income and employment history and exploring the use of deep learning models for credit risk analysis. This project can be a useful tool for lending institutions to make data-driven decisions and reduce the risk of loan defaults.

**Contents**

Objective …………………………………………………………………………...4

Introduction………………………………………………………………………... 4

Machine Learning …………………………………………………………5

Supervised Learning………………………………………………..5

Unsupervised Learning……………………………………………..6

Semi-Supervised Learning………………………………………….7

Reinforcement Learning…………………………………………….7

Dimensionality Reduction…………………………………………..7

Data Collection.……………………………………………………………………..8

Methodology………………………………………………………………………..10

Data Cleaning………………………………………………………………14

Data Exploration……………………………………………………………14

Data Modeling………………………………………………………………17

Results………………………………………………………………………………25

Conclusion…………………………………………………………………………..27

Future Work…………………………………………………………………………27

Bibliography…………………………………………………………………………27

**Objective**

In this project, we will develop a credit risk analyzer using machine learning techniques that will help banks and different financial lending institutions to automate credit risk analysis.

**Introduction**

One of the most important tasks that banks, and financial organizations do is to evaluate the risk that is associated with lending loans or money to businesses and individuals. Analyzing the credit risk is the possibility of the borrower repaying the loan and the loss caused to the financer when the borrower faces difficulties to repay the loan due to any reasons. Credit fraud is one of the financial frauds and is an attractive topic for the scientific community. Credit card frauds and loan frauds have social consequences as they lead to funding terrorism, organized criminal activities, and narcotics trafficking. Detecting unauthorized transactions is a very complex problem and needs to be tracked and reported. One of the solutions to this problem is the use of Machine Learning techniques to detect patterns and trends in the data sets.

With the growth in machine learning and advanced analytics, the risk of credit analysis can be automated, and it can help lending institutions to make more informed decisions regarding loan approval or rejection. This project will help mitigate the complete loss of the money borrowed and its recovery.

**Machine Learning**

Machine Learning is a field of Artificial Intelligence that helps in understanding and building methods that use data to improve performance on some set of assigned tasks. Machine Learning algorithms are built on sample data, which is also known as training data to make decisions and predictions. Machine Learning algorithms are used in a wide range of applications such as medicine, speech recognition, spam email filtering, agriculture, computer vision, etc where it is difficult to develop traditional algorithms to perform the tasks. One of the subsets of machine learning is related to computational statistics, which aims on making predictions using computers. Data mining is a related field in machine learning that deals with exploring the data using unsupervised learning. Machine Learning uses data to build neural networks in a way that works like a biological brain. Machine Learning is also referred to as predictive analytics in different applications across business problems.

**Approaches**

Machine Learning is broadly divided into three categories which are Supervised Learning, Unsupervised Learning, and Reinforcement Learning.

**Supervised Learning**

In this, algorithms build a machine-learning model of a set of data that consists of inputs and outputs. This data is known as training data and consists of a set of training examples. Each training example is represented by an array or vector and the training data is represented using a matrix. Supervised machine-learning algorithms learn a function that can be used to predict the output associated with the new inputs. The supervised machine-learning algorithm that improves the accuracy of the outputs or predictions over a certain time is said to have learned to perform that task.

The different types of supervised machine-learning algorithms include classification, regression, and active learning. Classification algorithms are considered when the outputs have a limited value. Examples of classification algorithms include email filtering, loan defaulter or not, predicting diseases, etc. Regression algorithms are considered when the output has numerical values within a range. Examples of regression algorithms include price prediction, etc. Active learning is another type of supervised machine learning algorithm which is closely related to classification and regression, but the main goal of active learning is to learn from examples using similarity functions that measure the similarity between two objects. The application of active learning is recommendation systems, rankings, face verification, identity tracking, and speaker verification.

**Unsupervised Learning**

Unsupervised learning algorithms take a set of data that only contains input labels, and it finds structure in the data such as grouping, and clustering of data values. The algorithms learn from the test data that has not been labeled on its own and classify or categorize them. Unsupervised machine learning algorithms identify common patterns in the data and react based on the presence or absence of such patterns. Unsupervised learning is widely used in the area of density estimation in statistics, which includes determining the probability density function. However unsupervised learning spans additional disciplines including summarizing and explaining data aspects. According to one or more predetermined criteria, cluster analysis is the division of a set of data into subsets (referred to as clusters) so that observations drawn from the same cluster are similar while observations collected from different clusters are dissimilar.

**Semi-Supervised Learning**

Between supervised learning (with labeled training data) and unsupervised learning is semi-supervised learning (with completely labeled training data). Many machine learning researchers have discovered that unlabelled data, when utilized in conjunction with a little amount of labeled data, can generate a significant gain in learning accuracy even though some of the training examples lack training labels. Although the training labels in weakly supervised learning are generally noisy, constrained, or imprecise, they are frequently less expensive to collect, leading to larger useful training sets.

**Reinforcement Learning**

Reinforcement Learning is a type of machine learning algorithm that looks at how software agents should behave in a given environment to maximize a theoretical total reward. Owing to its generality, the area is explored in many different academic fields, including statistics, genetic algorithms, multi-agent systems, game theory, control theory, operations research, and information theory. Often, a Markov decision process is used to represent the environment in machine learning (MDP). Dynamic programming approaches are used in many reinforcement learning systems. Since precise models are impractical, reinforcement learning methods are applied. These techniques do not need knowledge of an accurate mathematical model of the MDP. In autonomous vehicles or while learning to play a game against a human opponent, reinforcement learning algorithms are used.

**Dimensionality Reduction**

By obtaining a collection of principal variables, dimensionality reduction reduces the number of random variables being considered. In other words, it is a method of lowering the "number of features" dimension of the feature set. Most dimensionality reduction strategies can be divided into two categories: feature extraction and feature deletion. The principal component analysis is one of the most used techniques for dimensionality reduction (PCA). PCA entails downscaling data with greater dimensions (like 3D) to a smaller area (e.g., 2D). As a result, the model retains all original variables without affecting the data, resulting in a smaller dimension of the data (2D rather than 3D). Several dimensionality reduction techniques operate under the assumption that high-dimensional data sets are located along low-dimensional manifolds, which gave rise to the fields of manifold learning and manifold regularization.

**Data Collection**

**Dataset Source**

The dataset is retrieved from Kaggle. Below is the link to the dataset.

[**https://www.kaggle.com/datasets/laotse/credit-risk-dataset?resource=download**](https://www.kaggle.com/datasets/laotse/credit-risk-dataset?resource=download)

**Dataset Sample**

**Graphical user interface, table, Excel

Description automatically generated with medium confidence**

**Dataset Distribution**

The dataset has 32581 rows and a total of 12 columns. This indicates that the data has information on 32581 applicants with a variety of applicant details.

The dataset's various columns include the following:

1. **person\_age**- It is the applicant’s age.
2. **person\_income**- It provides details regarding the applicant's earnings.
3. **Person\_home\_ownership**- It provides details on the applicant's residence. Rent, Mortgage, Own, and other figures can be found in the education column.
4. **Person\_emp\_length**- It provides details regarding the applicant's employment time.
5. **Loan\_intent-** It provides details regarding the loan's objective or the kind of loan the applicant needs.
6. **Loan\_grade-** It provides details on the applicant's loan grade.
7. **Loan\_int\_rate-** It provides details on the interest rate that will apply to the applicant's loan.
8. **Loan\_amnt-** It is the amount of loan that is required of the applicant.
9. **Loan\_status-** It provides the loan's status. If it is "0," it is a non-defaulter, and if it is "1," it is a defaulter.
10. **Loan\_percent\_income-** It provides data on the income percentage.
11. **cb\_person\_default\_on\_file-** It provides details regarding the applicant's prior defaults.
12. **cb\_person\_cred\_hist\_length**- It provides details about the applicant's credit history going back over a certain amount of time.

**Dependent Variable**

A variable that is predicted or explained by one or more independent variables is referred to as a dependent variable.

The dependent variable present in the dataset is the Loan\_Status. A value of ‘0’ indicates the applicant is a loan defaulter and a value of ‘1’ is a defaulter.

**Independent Variables**

The variance in the dependent variable is explained or predicted using independent variables.

The variables person\_age, person\_income, person\_home\_ownership, person\_emp\_length, Loan\_intent, Loan\_grade, Loan\_int\_rate, Loan\_amnt, Loan\_Percent\_income, cb\_person\_default\_on\_file, cb\_person\_cred\_hist\_length are the independent variables present in the dataset.

**Methodology**

Firstly, we will be loading the dataset in RStudio and loading the necessary libraries.

Graphical user interface, text, application

Description automatically generated

Ggplot2

ggplot2 is an open-source data visualization package for the statistical programming language R. ggplot2 can serve as a replacement for the base graphics in R and offers a number of defaults for online and print displays of common scales.

Caret

Caret is used for difficult regression and classification issues, the caret package, which stands for Classification and Regression Training, offers routines to speed up the model-training procedure. The package makes use of a variety of R packages but tries to avoid loading them all at once (by removing formal package dependencies, the package startup time can be greatly decreased).

Rpart.plot

Rpart is a powerful machine-learning library in R that is used for building classification and regression trees. This library implements recursive partitioning and is very easy to use.

We will then display the top five rows of the dataset. Below is the R code and the output to display the first five rows of the dataset.

The head () function is used to display the top rows of the dataset. The number indicated in the function will return the top number of rows of the dataset.

Similarly, the tail () function is used to display the bottom rows of the dataset.

Table

Description automatically generated

Now we’ll be checking the different summary statistics of the dataset. The code and the output are provided below.

The summary () function in R can be used to produce a statistical summary for a dataset. For each variable in the dataset, this function computes and produces a summary of statistics. This will result in a summary of statistics for each variable in the dataset, along with the frequency table for categorical variables and the median, quartile, and minimum and maximum values for numeric variables.

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

**Data Cleaning**

We’ll be finding the missing values in each column of the dataset. The code and the output of the same are given below.

Graphical user interface

Description automatically generated with medium confidence

The column ‘person\_emp\_length’ has 895 missing values in it and ‘loan\_int\_rate’ has 3116 missing values in it.

To remove the missing values from the dataset, we can use the below code. The output after the missing values is removed and is shown below.

Graphical user interface, text, application

Description automatically generated

**Data Exploration**

1. Below is the code and visualization for the distribution of the loan intent.

Graphical user interface, text

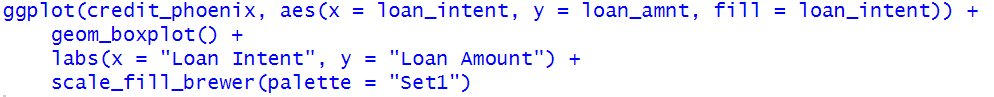
Description automatically generated with medium confidence

Chart, bar chart

Description automatically generated

From the above visualization, the loan intent for education is higher, which means most people are opting for education loans. The intent for taking Home Improvement loans is the least.

1. Below is the box plot of the loan intent where we can see the median loan amount and the outliers present in the dataset.



Chart, box and whisker chart

Description automatically generated

1. Below is the plot for the age distribution of applicants taking loans.

Text

Description automatically generated

Chart, histogram

Description automatically generated

From the above plot, applicants between the age group 20-25 are applying for a high number of loans. People of the age group around 40 are applying for the least loans.

1. Below is the plot for the relationship between income and loan intent.

Graphical user interface, text

Description automatically generated

Chart, line chart

Description automatically generated

**Data Modelling**

The practice of conceptually representing data and its relationships in a particular domain is known as data modeling. This format can be used to plan and implement a database or to get insights into patterns and relationships within the data.

It is crucial to keep in mind the objectives and specifications of the project throughout the data modeling process, as well as the requirements of the users who will be utilizing the data model. You may design a data model that is accurate, efficient, and useful for your needs by using an organized and iterative process.

We will be splitting the data into training and validation sets and dropping the columns that are not required.

Training Data

Data used to train a machine learning model is referred to as "training data" in machine learning. It is a portion of the dataset that is utilized to train the model to predict outcomes correctly.

Typically, the independent variables (sometimes referred to as features) and the dependent variable are both present in the training data (also known as the target variable or outcome variable). By modifying its parameters to reduce the discrepancy between its predicted values and the actual values in the training data, the model is trained on this data.

Validation Data

A section of the dataset used to assess the performance of a trained machine learning model is referred to as validation data (also known as validation set or development set) in machine learning. It is a distinct subset of training data that is not utilized during training.

The model's generalization ability—or, more specifically, how effectively it can make precise predictions on brand-new, untested data—is evaluated using the validation data. The model may overfit the training data, which means that it may perform well on the training data but badly on fresh, unseen data. For this reason, it is crucial to employ a validation set in addition to the training set.

To improve the model's performance on the validation set, the hyperparameters of the model, such as the learning rate, regularization, or number of hidden layers, are often tuned using the validation data. Up until the target level of performance is attained, the model is iteratively trained on the training data and validated on the validation data.

The model can be evaluated on a different test dataset after it has been tuned on the validation data to get a fair assessment of its performance. To evaluate the final performance of the model, a completely hidden subset of the original dataset is employed, referred to as the test dataset.

Below is the code for splitting the data into train and validation.

Text, letter

Description automatically generated

We are using the Decision tree classifier for the classification.

**Decision Trees**

Decision trees are a type of supervised machine learning model that can perform both classification and regression problems, these are often referred to as CART algorithm: Classification and Regression Tree.

The idea behind Decision Trees is to repeatedly partition the dataset until all the data points that belong to each class are isolated by using the dataset features to produce yes/no questions. Through this procedure, the data is arranged in a tree form. Every time you ask a question, you're adding a node to the tree. Moreover, the root node is the first node. A question's answer divides the dataset according to the importance of a feature and adds additional nodes. If you opt to terminate the process after a split, the final nodes formed are termed leaf nodes.

Diagram

Description automatically generated

While you respond to a question, branches are also being formed and the feature space is being divided into several regions. All the data points on one branch of the tree indicate that the answer to the query that the rule in the preceding node implies is Yes. Other remaining data points are located at a node on the other branch. With each split or branch in the tree, you reduce the feature space, and each data point will only be associated with one region. The objective is to partition the feature space and apply rules continuously until there are no more rules to apply or data points to collect. The next step is to give each leaf node's data points a class.

The technique attempts to fully divide the dataset such that each leaf node, or node that doesn't further divide the data, belongs to a single class. We refer to these as pure leaf nodes. Yet, most of the time you get mixed leaf nodes, where not all the data points belong to the same class. The data points in each leaf node can only be assigned to one class by the algorithm at the end. Since every data point in a pure leaf node belongs to the same class, this problem is already solved. In contrast, the algorithm assigns the most prevalent class among all the data points at mixed leaf nodes.

Graphical user interface, text, application, email

Description automatically generated

Evaluating the model’s performance.

Confusion Matrix

A confusion matrix is a table that is used to describe how well a classification system performs. The output of a classification algorithm is shown and summarized in a confusion matrix. The confusion matrix is utilized for the performance evaluations of the methods used after the classification. It compares the predicted and actual values of a target variable and computes several metrics that can be used to evaluate the performance of the model.

There are four possible outcomes from the confusion matrix:

True positive (TP): The target variable is in a positive class, as anticipated by the model.

False positive (FP): The target variable was supposed to belong to the positive class according to the model, but it does.

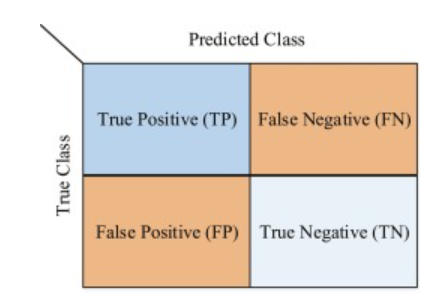
True Negative (TN): The target variable's membership in the negative class was accurately predicted by the model, which is known as a true negative (TN).

False negative (FN): The model predicted that the target variable belongs to the negative class, but it belongs to the positive class.

Accuracy: A popular performance parameter used to assess a predictive model's effectiveness is accuracy. It is described as the percentage of accurate predictions among all of the model's predictions. To put it another way, accuracy refers to how well the model can predict the values of the dependent variable.

The following formula can be used to determine a model's accuracy:

Accuracy = (True Positives + True Negatives) / (True Positives + False Positives + True Negatives + False Negatives)



Source: <https://www.sciencedirect.com/topics/engineering/confusion-matrix#:~:text=A%20confusion%20matrix%20is%20a,performance%20of%20a%20classification%20algorithm>.

**Sensitivity**

Sensitivity, commonly referred to as recall or true positive rate, is a performance parameter used in statistics and machine learning to assess how well a prediction model performs in accurately detecting positive cases.

Sensitivity quantifies the percentage of real positive cases that the model correctly classifies as positive. The formula is as follows:

Sensitivity = True Positives / (True Positives + False Negatives)

**Specificity**

Specificity is a performance indicator used in statistics and machine learning to assess how well a predictive model performs in accurately detecting negative cases.

The percentage of actual negative cases that the model correctly classifies as negative is known as specificity. The formula is as follows:

Specificity = True Negatives / (True Negatives + False Positives)

**Positive Predictive Value**

In binary classification analysis, it is a statistical indicator used to assess the proportion of real positive outcomes in comparison to all the positive predictions a model makes.

The formula for PPV is:

PPV = True Positive / (True Positive + False Positive)

**Negative Predictive Value**

It is a statistical indicator used in binary classification analysis to assess the ratio of actual negative results to all the model's negative predictions.

NPV = True Negative / (True Negative + False Negative)

**Prevalence**

The percentage of positive cases (or instances) in a dataset is referred to as prevalence. Prevalence is the proportion of cases that fall into the "positive class" in binary classification issues where the goal is to predict one of two probable outcomes.

Prevalence = (True Positives) / (True Positives + False Positives)

**Detection Rate**

The true positive rate (TPR), often referred to as the detection rate (DR), or sensitivity, is a performance indicator used in binary classification to assess the percentage of positive cases that the model properly recognizes.

The detection rate can be calculated as:

Detection Rate = (Number of true positives) / (Number of true positives + Number of false negatives)

**Detection Prevalence**

Detection Prevalence is a binary classification performance indicator that accounts for both detection rate (true positive rate) and prevalence (proportion of positive instances in the dataset).

The detection prevalence can be calculated as:

Detection Prevalence = (Number of true positives) / (Number of true positives + Number of false positives) \* (Prevalence)

**Balanced Accuracy**

In binary classification, the balance of the classes in the dataset is taken into consideration by the performance statistic known as balanced accuracy. The true positive rate (TPR) and true negative rate (TNR), which quantify the proportion of correctly detected positive and negative cases, respectively, are averaged to form this term.

The formula for balanced accuracy is:

Balanced Accuracy = (True Positive Rate + True Negative Rate) / 2

Text

Description automatically generated

**Results**

We have built a decision tree classification model. We have obtained different performance metrics to evaluate the machine learning model.

Accuracy = 0.9066, this means that the model has correctly predicted the class label of 90.66% of instances in the dataset.

Sensitivity = 0.9835, this means that the model has correctly identified 98.35% of the positive instances in the dataset.

Specificity = 0.211, which means that the model has correctly identified 62.11% of the negative instances in the dataset.

Positive Predictive Value = 0.9061, which means the model predicts the correct positive class label 90.61% of the time.

Negative Predictive Value = 0.9099, which means the model predicts the correct negative class label 90.99% of the time.

Prevalence = 0.7880, which means the model correctly predicted 78.80% of instances belong to the positive class.

Detection Rate = 0.7550, which means the model correctly predicted 75.5% of the positive instances in the dataset.

Detection Prevalence = 0.8553, which means that the model correctly identified 85.53% of instances belonging to the positive class.

Balanced Accuracy = 0.8023, which means that the model correctly identifies both positive and negative instances with a balanced accuracy of 80.23%.

Overall, the model seems to have high sensitivity and positive predictive value, showing that it is effective at accurately recognizing positive events. It may not be as effective in correctly identifying negative cases, though, as the specificity and balanced accuracy are both rather low. The high prevalence of the positive class may point to a dataset with an unequal distribution of classes. To ascertain the model's applicability in practice, it would be crucial to further evaluate its performance, particularly with regard to the false positive and false negative rates.

**Conclusion**

We developed a credit risk analyzer for this project that can forecast whether a borrower will default or not. To improve the performance of the model, we used feature selection, feature preprocessing, and various classification models. The random forest model, which had an accuracy of 90.66%, performed the best. This model can be used by lending organizations to analyze the credit risk of potential borrowers and make educated judgments about loan approvals.

**Future Work**

By including further features like borrower income and employment history in the future, we can enhance the model's performance. We can investigate the application of deep learning models to credit risk assessment. Finally, we can conduct a more thorough study of the false positives and false negatives to comprehend the model's limitations and pinpoint areas that need development.

**Bibliography**

1. <https://en.wikipedia.org/wiki/Machine_learning>
2. <https://towardsdatascience.com/decision-tree-classifier-explained-in-real-life-picking-a-vacation-destination-6226b2b60575>
3. <https://en.wikipedia.org/wiki/Ggplot2>
4. <https://cran.r-project.org/web/packages/caret/vignettes/caret.html>
5. <https://www.pluralsight.com/guides/explore-r-libraries:-rpart>
6. <https://www.sciencedirect.com/topics/engineering/confusion-matrix#:~:text=A%20confusion%20matrix%20is%20a,performance%20of%20a%20classification%20algorithm>.