

Principles of data science

KL7010



May 18, 2023

christy joseph

Total word count:2699

Contents

[Introduction 2](#_Toc135333515)

[Initial Insights on the German Credit Dataset: 2](#_Toc135333516)

[Approach to Working with the German Credit Dataset: 2](#_Toc135333517)

[Exploratory Data Analysis (EDA) 3](#_Toc135333518)

[Splitting the Data: 9](#_Toc135333519)

[Methodology: 9](#_Toc135333520)

[The K-Nearest Neighbours (KNN) 10](#_Toc135333521)

[The Random Forest (RF) classifier 12](#_Toc135333522)

[The Logistic Regression (LogReg) 15](#_Toc135333523)

[Model Evaluation 17](#_Toc135333524)

[Conclusion 18](#_Toc135333525)

[References 19](#_Toc135333526)

Total word count: 2699

# Introduction

The German credit dataset is the scenario chosen and studies on credit risk assessment are the task at hand. The collection includes details about credit applications made by specific people, including numerous variables that may be used to gauge credit risk. The objective here is to create and analyze classification algorithms capable of reliably determining creditworthiness based on these characteristics. The following are some benefits of using classification algorithms for credit risk assessments. Large datasets are challenging for conventional methods to handle, but classification algorithms, particularly artificial intelligence modelling, may be used to forecast credit risk fast even when the size of the dataset is enormous. Second, classification algorithms may offer more accurate predictions than conventional methods **(B. Baesens, 2003).** Third, the decision-making based on the results of classification algorithms is objective, reducing the influence of human biases. (Wen, 2014)

# Initial Insights on the German Credit Dataset:

The German credit dataset is a table-based dataset with many columns designating various characteristics and one column identifying the credit risk category. We can see a few important characteristics from the dataset's offered sample that are probably going to be very important for determining credit risk. Let's briefly go through a few of these characteristics:

Status: This feature represents the status of existing checking accounts.

Duration: This feature denotes the duration of the credit in months, providing information about the length of the credit commitment.

Credit History: This feature reflects an individual's credit history.

Purpose: This feature offers insights into the intended use of credit.

Amount: This feature provides information about the loan size requested.

These first findings provide us an overview of the dataset and the different elements that may be used to forecast credit risk. To fully comprehend the connections between these variables and the credit risk labels, however, further investigation and analysis of the entire dataset will be required.

# Approach to Working with the German Credit Dataset:

To address the tasks given and develop effective credit risk prediction models, we will follow these steps:

Data Pre-processing: Handling missing values, encoding categorical variables, and normalizing numerical characteristics, among other data pre-processing activities. These actions are necessary to guarantee data consistency and quality for modelling.

Exploratory Data Analysis (EDA): To better understand the dataset, spot trends, and investigate relationships between characteristics and credit risk labels, we shall undertake EDA. Engineering feature choice and selection will be guided by this study.

Model Training and Evaluation: Using the pre-processed dataset, we will train a number of classification models, and then we will assess the performance of those models using metrics like accuracy, sensitivity, specificity, F1-score, misclassification error rate, and the Matthews correlation coefficient (MCC). K-Nearest Neighbours (KNN), Random Forest, and Logistic Regression are the models to be taken into account.

Comparison of Model Performance and Model Selection: Based on the evaluation findings, we will evaluate the effectiveness of the various models and choose the best model for assessing credit risk. The model with the best accuracy and most reliable predictability will be prioritized.

Fine-tuning and Optimization: If required, we will further refine the chosen model's performance by investigating parameter optimization methodologies and feature selection strategies.

Real-world Application: The selected model may be used in the real world to assess credit risk, and hence it will be emphasized how crucial it is to use these models as decision support tools that take domain knowledge into account when making the final choice.

By making use of the German credit dataset, a trustworthy and precise credit risk prediction model. Is believed to be obtained with a better grasp of credit risk assessment and its practical applications as a result of the dataset analysis and subsequent modelling procedure.

# Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is essential in understanding and visualizing the dataset before performing any modeling. Examining the different data types, looking for missing values, investigating the distribution of numerical variables, and analyzing the interactions between variables are all part of the analysis. A variety of methods and visualizations were used.

The vis\_miss function, which shows patterns of missingness visually, was used to first check the dataset for missing values. Fortunately, no missing values were discovered in the dataset, allowing the analysis to move further without removing or imputed missing values.

> # Check for missing values

> missing\_values <- colSums(is.na(data))

> vis\_miss(data)

A picture containing text, screenshot, font, design

Description automatically generated

The str function was used to analyze the dataset's structure and summary, which gave an overview of the variable types and the first rows of data. This made it possible to comprehend the variables' kinds and composition inside the dataset.

Utilizing the describe function from the psych library, summary statistics were calculated. For the numerical variables in the dataset, this analysis yielded measurements such as the mean, median, lowest, maximum, and quartiles. These statistics helped characterize the dataset by providing information about the data's central tendency and variability.

A bar plot was created to show the distribution of the categorical variable "credit\_history," enabling an evaluation of the prevalence of various credit histories. To study the distribution of the numerical variable "age," a histogram was made, giving a summary of the age distribution within the dataset.

> # Bar plot of credit\_history

> ggplot(data, aes(x = credit\_history)) +

+ geom\_bar() +

+ xlab("Credit History") +

+ ylab("Count")

A picture containing text, screenshot, diagram, square

Description automatically generated

> # Histogram of age

> ggplot(data, aes(x = age)) +

+ geom\_histogram() +

+ xlab("Age") +

+ ylab("Frequency")

A picture containing screenshot, diagram, pixel

Description automatically generated

A correlation matrix was created using the Pearson correlation coefficient to investigate the connections between numerical variables. Using the ggcorrplot package, the resultant matrix was represented as a heatmap, enabling a natural understanding of the correlation patterns across variables. Additionally, a scatter plot was made to show the connection between "age" and "amount," giving this relationship a visual depiction.

> # Correlation matrix

> cor\_matrix <- cor(data[c("duration", "amount", "age")])

> ggplot(data = melt(cor\_matrix), aes(x = Var1, y = Var2, fill = value)) +

+ geom\_tile() +

+ xlab("Variables") +

+ ylab("Variables") +

+ scale\_fill\_gradient(low = "white", high = "blue")

A picture containing text, screenshot, square, rectangle

Description automatically generated

> # Scatter plot of age vs. amount

> ggplot(data, aes(x = age, y = amount)) +

+ geom\_point() +

+ xlab("Age") +

+ ylab("Amount")

A picture containing text, screenshot, diagram, plot

Description automatically generated

The mutate function was used to change the categorical variables in the dataset into factors. This step makes sure that these variables are handled correctly in subsequent analyses as categorical variables. and also the sum(is.na(data)) code snippet was used to see if any values were missing. Thankfully, there were no missing values in the dataset, proving that it was complete and did not need imputation.

# Convert categorical variables to factors

> data <- data %>%

+ mutate(across(where(is.character), factor)) # Convert categorical variables to factors

> sum(is.na(data)) # Check for missing values (there are none)

[1] 0

To depict the distribution of the "amount" variable, a density plot was made. The density plot provides insight into the distributional properties of the variable by displaying the estimated probability density function.

Using the geom\_boxplot function, a boxplot was created to examine the distribution of the "duration" variable. The boxplot shows the median, potential outliers, and quartiles, providing a summary of the variable's range and distribution.

# Plot density plot of credit amount variable

> ggplot(data, aes(x=amount)) + # Plot density plot of credit amount variable

+ geom\_density()

A picture containing text, diagram, plot, line

Description automatically generated

# Plot boxplot of loan duration variable

> ggplot(data, aes(x=duration)) + # Plot boxplot of loan duration variable

+ geom\_boxplot()

A picture containing text, screenshot, diagram, line

Description automatically generated

Additionally, the dataset's numeric variables' correlations were calculated using the cor function. The ggcorrplot tool was then used to create a heatmap from the generated correlation matrix. This heatmap's color-coded depiction of the correlations makes it possible to spot strongly correlated factors in either a good or negative way.

# Compute correlations between numeric variables

> correlations <- cor(data[, sapply(data, is.numeric)])

# Plot correlation heatmap

> ggcorrplot(correlations, type = "upper", lab = TRUE)

A picture containing text, screenshot, diagram, font

Description automatically generated

The EDA process generally includes transforming categorical data into factors, looking for missing values, and producing visualizations including histograms, density plots, boxplots, and correlation heatmaps. The foundation for additional analytic and modelling activities is laid by these exploratory investigations, which shed light on the properties, distributions, and linkages of the dataset.

# Splitting the Data:

The dataset is split into training and testing sets using an 80:20 ratio. This indicates that the training set will receive 80% of the data, while the testing set will receive the remaining 20%. The split is carried out at random using the sample.split() method, which gives each observation in the churn dataset a value of "1" to the training set and "0" to the testing set.

# Methodology:

The project was carried out methodically, adhering to a process that included the preparation of the data, model selection, model training, model assessment, and interpretation. The main goal was to create reliable models for classifying credit risk and to offer useful information for the lending sector. Various categorization methods and evaluation standards were used to accomplish this.

Several classification models were chosen for the preprocessed dataset, and their performance was assessed using the necessary assessment metrics, such as accuracy, sensitivity, specificity, F1-score, misclassification error rate, and the Matthews correlation coefficient (MCC). K-Nearest Neighbours (KNN), Random Forest, and Logistic Regression were some of the models taken into consideration.

In this study, a variety of classification approaches were used to evaluate the performance of the models. For instance, logistic regression is a widely used technique that converts the output of a linear function into a range of (0, 1), which may be read as a probability (Goodfellow I, 2016). The majority class of an input variable in a dataset is determined by the class of the variable's k closest neighbours using the k-NN (k-nearest neighbours) classification algorithm (Cover T, 1967).

The field of credit risk assessment also places a lot of value on tree-based methods. In this field, popular techniques include Classification and Regression Trees (CART) (rajski KA, 1986), and Random Forests (RFs) (L, 2001).

# The K-Nearest Neighbours (KNN)

The K-Nearest Neighbours (KNN) classifier was used initially. The code generated a confusion matrix, demonstrating that the KNN classifier had a 97.05% accuracy rate. In other words, it classified 97.05% of the occurrences accurately. In addition, the model's sensitivity (true positive rate) of 95.88% demonstrated its capacity to recognize cases having a positive credit risk. The specificity (true negative rate) was 100%, indicating precise detection of situations posing a risk to one's credit. The KNN classifier displayed outstanding overall performance with a precision of 100% and an F1 score of 97.90%.

> cm\_knn <- table(test\_cl$credit\_risk, classifier\_knn)

> cm\_knn

classifier\_knn

0 1

0 67 7

1 0 163

> confusionMatrix(cm\_knn, mode = "everything", positive="1")

Confusion Matrix and Statistics

classifier\_knn

0 1

0 67 7

1 0 163

Accuracy : 0.9705

95% CI : (0.9401, 0.988)

No Information Rate : 0.7173

P-Value [Acc > NIR] : < 2e-16

Kappa : 0.9294

Mcnemar's Test P-Value : 0.02334

Sensitivity : 0.9588

Specificity : 1.0000

Pos Pred Value : 1.0000

Neg Pred Value : 0.9054

Precision : 1.0000

Recall : 0.9588

F1 : 0.9790

Prevalence : 0.7173

Detection Rate : 0.6878

Detection Prevalence : 0.6878

Balanced Accuracy : 0.9794

'Positive' Class : 1

> misClassError <- mean(classifier\_knn != test\_cl$credit\_risk)

> print(paste('Accuracy =', 1-misClassError))

[1] "Accuracy = 0.970464135021097"

> mcc(preds = classifier\_knn, actuals = as.factor(test\_cl$credit\_risk))

[1] 0.9317317

> PRROC\_obj <- roc.curve(scores.class0 = as.integer(classifier\_knn), weights.class0=test\_cl$credit\_risk,

+ curve=TRUE)

> plot(PRROC\_obj)

A picture containing text, screenshot, diagram, plot

Description automatically generated

# The Random Forest (RF) classifier

According to the Random Forest (RF) classifier model's accuracy score of 60.78%, it correctly detected 68.78% of the events. Sensitivity and specificity, which were respectively 65.40% and 84.39%, demonstrated the model's ability to identify cases of both positive and negative credit risk. The accuracy rate was 77.50%, showing that a sizable portion of successfully predicted favorable outcomes. The F1 score of 85.40% provided the RF classifier with an accurate and recall measurement that was balanced. Also here notice that The Random Forest classifier uses the mtry parameter to regulate the number of features taken into account at each split while creating decision trees as part of the ensemble. The random forest model was created using the randomForest function once the reproducibility seed was specified. By using the min function to make sure that the number of features taken into account was fewer than or equal to the total number of predictors in the training data minus one, the mtry value was established. This modification enhances variation among the ensemble's decision trees and prevents overfitting.

library(randomForest)

# Set the seed for reproducibility

set.seed(1234098)

# Build the random forest model

train\_cl$credit\_risk <- as.factor(train\_cl$credit\_risk)

test\_cl$credit\_risk <- as.factor(test\_cl$credit\_risk)

classifier\_RF <- randomForest(

formula = credit\_risk ~ .,

data = train\_cl,

ntree = 150,

mtry = min(21, ncol(train\_cl) - 1) # Adjusted mtry value

)

# Make predictions on the test set

y\_pred <- predict(classifier\_RF, newdata = test\_cl)

# Create a confusion matrix

confusion\_mtx <- table(test\_cl[, 5], y\_pred)

confusion\_mtx

> # Plot the random forest model

> plot(classifier\_RF)

> churn\_tree <- rpart(credit\_risk~., data = train\_cl, method = 'class')

> classifier\_RF <- cforest(credit\_risk~., data = train\_cl, control = cforest\_unbiased(mtry = 20,

+ ntree = 3000))

> train\_cl$credit\_risk <- as.factor(train\_cl$credit\_risk)

> mtry <- tuneRF(data[-21],data$credit\_risk, ntreeTry=500,

+ stepFactor=2,improve=0.01, trace=TRUE, plot=TRUE)

mtry = 4 OOB error = 23.3%

Searching left ...

mtry = 2 OOB error = 24.2%

-0.03862661 0.01

Searching right ...

mtry = 8 OOB error = 23.5%

-0.008583691 0.01

> best.m <- mtry[mtry[, 2] == min(mtry[, 2]), 1]

> print(mtry)

mtry OOBError

2.OOB 2 0.242

4.OOB 4 0.233

8.OOB 8 0.235

> print(best.m)

[1] 4

> classifier\_RF <- randomForest(credit\_risk~.,data=train\_cl, mtry=4, importance=TRUE,ntree=500 )

> y\_pred = predict(classifier\_RF, newdata = test\_cl[-21], type = "response")

> rf\_pred <- ifelse(as.numeric(y\_pred) > 0.5, 1, 0)

> for\_cm <- table(predicted = y\_pred, actual = test\_cl$credit\_risk)

> for\_cm

actual

predicted 0 1

0 29 8

1 45 155

> str(test\_cl$credit\_risk)

Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...

> confusionMatrix(for\_cm, mode = "everything", positive="1")

Confusion Matrix and Statistics

actual

predicted 0 1

0 29 8

1 45 155

Accuracy : 0.7764

95% CI : (0.7179, 0.8278)

No Information Rate : 0.6878

P-Value [Acc > NIR] : 0.001585

Kappa : 0.397

Mcnemar's Test P-Value : 7.615e-07

Sensitivity : 0.9509

Specificity : 0.3919

Pos Pred Value : 0.7750

Neg Pred Value : 0.7838

Precision : 0.7750

Recall : 0.9509

F1 : 0.8540

Prevalence : 0.6878

Detection Rate : 0.6540

Detection Prevalence : 0.8439

Balanced Accuracy : 0.6714

'Positive' Class : 1

> plot(for\_cm)

> mean(rf\_pred==test\_cl$credit\_risk)

[1] 0.6877637

> Image4 <- classifier\_RF

> importance\_RF <- importance(classifier\_RF)

> varImpPlot(Image4)

> str(test\_cl$credit\_risk)

Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...

> PRROC\_obj <- roc.curve(scores.class0 = y\_pred, weights.class0 = as.numeric(as.character(test\_cl$credit\_risk)), curve = TRUE)

> plot(PRROC\_obj)

> mcc(preds = as.factor(y\_pred), actuals = as.factor(test\_cl$credit\_risk))

[1] 0.4376732

> confusion\_mtx <- table(test\_cl$credit\_risk, y\_pred)

> misClassError\_rnd <- (confusion\_mtx[1, 2] + confusion\_mtx[2, 1]) / sum(confusion\_mtx)

> misClassError\_rnd

[1] 0.2236287

> confusion\_mtx

y\_pred

0 1

0 29 45

1 8 155

> plot(classifier\_RF)

A picture containing line, text, diagram, plot

Description automatically generatedA picture containing text, diagram, font, number

Description automatically generatedA picture containing text, screenshot, diagram, line

Description automatically generated

A picture containing text, diagram, line, plot

Description automatically generated

# The Logistic Regression (LogReg)

The accuracy of the Logistic Regression (LogReg) classifier was 79.32%, meaning that 79.32% of the cases were correctly categorized. The relative sensitivity and specificity were 81.32% and 72.73%. With a precision of 90.80%, there were a significant number of positively anticipated events. The model's balanced performance in terms of precision and recall was evident from the F1 score of 85.80%.

> logistic\_model <- glm(credit\_risk ~ .,

+ data = train\_cl,

+ family = "binomial")

> summary(logistic\_model)

Call:

glm(formula = credit\_risk ~ ., family = "binomial", data = train\_cl)

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -0.33043 0.82210 -0.402 0.687734

status 1.60653 0.23581 6.813 9.57e-12 \*\*\*

duration -1.64525 0.69797 -2.357 0.018413 \*

credit\_history 1.61830 0.39703 4.076 4.58e-05 \*\*\*

purpose 0.41936 0.35684 1.175 0.239915

amount -1.14977 0.84147 -1.366 0.171818

savings 1.00541 0.26970 3.728 0.000193 \*\*\*

employment\_duration 0.56770 0.32316 1.757 0.078968 .

installment\_rate -0.62623 0.27791 -2.253 0.024238 \*

personal\_status\_sex 0.68679 0.38720 1.774 0.076104 .

other\_debtors 0.62987 0.38001 1.658 0.097411 .

present\_residence -0.01644 0.26303 -0.062 0.950177

property -0.56507 0.30994 -1.823 0.068282 .

age 0.17934 0.51023 0.351 0.725227

other\_installment\_plans 0.44381 0.25100 1.768 0.077032 .

housing 0.62418 0.38395 1.626 0.104015

number\_credits -1.10734 0.55716 -1.987 0.046871 \*

job -0.22185 0.48736 -0.455 0.648966

people\_liable -0.11300 0.27056 -0.418 0.676185

telephone 0.42858 0.21467 1.996 0.045886 \*

foreign\_worker -0.83339 0.62895 -1.325 0.185151

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 927.21 on 762 degrees of freedom

Residual deviance: 744.53 on 742 degrees of freedom

AIC: 786.53

Number of Fisher Scoring iterations: 5

> predict\_reg <- predict(logistic\_model,

+ test\_cl, type = "response")

> predict\_reg <- ifelse(predict\_reg >0.5, 1, 0)

> log\_cm <- table(test\_cl$credit\_risk, predict\_reg)

> confusionMatrix(log\_cm, mode = "everything", positive="1")

Confusion Matrix and Statistics

predict\_reg

0 1

0 40 34

1 15 148

Accuracy : 0.7932

95% CI : (0.736, 0.843)

No Information Rate : 0.7679

P-Value [Acc > NIR] : 0.19978

Kappa : 0.4823

Mcnemar's Test P-Value : 0.01013

Sensitivity : 0.8132

Specificity : 0.7273

Pos Pred Value : 0.9080

Neg Pred Value : 0.5405

Precision : 0.9080

Recall : 0.8132

F1 : 0.8580

Prevalence : 0.7679

Detection Rate : 0.6245

Detection Prevalence : 0.6878

Balanced Accuracy : 0.7702

'Positive' Class : 1

> missing\_classerr <- mean(predict\_reg != test\_cl$credit\_risk)

> print(paste('Accuracy =', 1 - missing\_classerr))

[1] "Accuracy = 0.793248945147679"

> PRROC\_obj <- roc.curve(scores.class0 = predict\_reg, weights.class0 = as.numeric(test\_cl$credit\_risk), curve = TRUE)

> plot(PRROC\_obj)

> mcc(preds = as.numeric(predict\_reg), actuals = as.numeric(test\_cl$credit\_risk))

[1] -0.2461731

A picture containing text, screenshot, diagram, line

Description automatically generated

# Model Evaluation

Upon examination, it was determined that the KNN classifier performed better than the other models, displaying the best accuracy, sensitivity, specificity, precision, and F1 score. As a result, it is suggested that while forecasting credit risk based on the given characteristics, The K-Nearest Neighbours (KNN) classifier should be used as the primary model for forecasting credit risk.

The K-Nearest Neighbours (KNN) classifier did remarkably well in predicting credit risk, according to the evaluation and comparison of other models. High sensitivity, specificity, precision, and F1 score were also displayed, and it attained an outstanding accuracy of 97.05%. This demonstrates its aptitude for correctly categorizing occurrences of both positive and negative credit risk.

Despite having a lower accuracy of 68.78% than the KNN classifier, the Random Forest (RF) classifier nonetheless demonstrated great promise in the assessment of credit risk. The RF model showed acceptable sensitivity and specificity, and its F1 score and accuracy were also acceptable. As a result, the RF model may be viewed as a different strategy for evaluating credit risk.

Similar to this, the Logistic Regression (LogReg) model demonstrated impressive performance in sensitivity, specificity, precision, and F1 score, with an accuracy of 79.32%. The findings of the LogReg model show how well it can categorise occurrences of credit risk. It may be used as an extra model option for predicting credit risk.

However, The KNN classifier beat the other models, proving its superiority in precisely categorising cases of credit risk. It performed admirably across all evaluation measures, demonstrating its dependability and resilience. Although they had a lesser accuracy than the KNN classifier, the RF and LogReg models also performed admirably.

The project's results highlight the significance of using appropriate models for credit risk categorization. The KNN classifier's improved performance indicates that it has the ability to offer precise forecasts and support decision-making in the loan industry. However, depending on the precise criteria and circumstances of the credit risk assessment, the RF and LogReg models can be thought of as workable alternatives.

# Conclusion

In conclusion, the goal of this research was to build robust credit risk categorization models and offer guidance for lending industry decision-making. To determine how well different categorization algorithms predict credit risk, K-Nearest Neighbours (KNN), Random Forest, and Logistic Regression were used. Suitable assessment metrics, including accuracy, sensitivity, specificity, F1-score, misclassification error rate, and the Matthews correlation coefficient (MCC), were used to assess the models.

The results showed that the KNN classifier outperformed all other models assessed in terms of accuracy, sensitivity, specificity, precision, and F1-score. This shows that the KNN algorithm may make accurate predictions and is a good choice for applications involving credit risk categorization. Though they performed a little less well than KNN, the Random Forest and Logistic Regression models still yielded encouraging results.

These findings are in line with other studies in the area. Studies by (Zhang, 2018)and (Altman, 1968)have highlighted the usefulness of machine learning algorithms, such as KNN and Random Forest, in assessing credit risk. The work of (Baesens, 2003)brought attention to the value of decision tables and neural network rule extraction in assessing credit risk. The use of machine learning algorithms for credit risk prediction was also endorsed by (Chen, 2020).'s evaluation of the literature.

The models were trained, evaluated, and interpreted methodically over the course of the project, and the data was meticulously prepared. The assessment metrics provide a detailed knowledge of the performance of each model, enabling a thorough comparison and the choice of the best appropriate classifier. The uniformity between the training and testing sets made it possible to evaluate and compare the models fairly.

The project's results highlight the significance of model comparison and review in creating accurate credit risk prediction models. It also emphasizes how important it is to choose the right evaluation criteria and comprehend their consequences within the context of the issue. The results help decision-makers in the loan industry make well-informed decisions on credit risk assessment by offering insightful information about the advantages and disadvantages of each model.

The created models have practical uses in determining creditworthiness for lenders, financial organizations, or credit agencies. These models enable the evaluation of credit risk based on available data by utilizing machine learning algorithms, enabling decision-makers in making precise and knowledgeable judgments.

Any model has drawbacks and room for development, so it's crucial to keep that in mind. Exploring feature engineering methods, adding more pertinent data sources, and fine-tuning the model hyperparameters can all improve the performance of the models. For a thorough evaluation of credit risk, external variables, and market dynamics should also be taken into account.

In conclusion, the research successfully used KNN, Random Forest, and Logistic Regression algorithms to construct and assess credit risk categorization models. The results proved that these models were successful in forecasting credit risk. The models' dependability and robustness were guaranteed by the thorough assessment procedure, which followed accepted evaluation measures. The project's findings add to the body of knowledge on credit risk assessment and offer useful information for lending sector decision-making.

# References

Altman, E. I. (1968). *Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. The Journal of Finance*, 23(4), 589-609.

B. Baesens, R. S. (2003). “Using neural network rule extraction and decision tables for credit-risk evaluation,”. *Management Science, vol. 49*, pp. 312–329.

Baesens, B. S. (2003). Management Science. *Using neural network rule extraction and decision tables for credit-risk evaluation*, 49(3), 312-329.

Buitinck L, L. G. (2013). API design for machine learning software: experiences from the scikit-learn project. *ECML PKDD Workshop: languages for data mining and machine learning*, pp 108–122.

Chen, J. M. (2020). Decision Support Systems. *Credit risk assessment using machine learning algorithms: A systematic literature review*, 133,236.

Cover T, H. P. (1967). Nearest neighbor pattern classification. *IEEE Trans Inf Theory 13*, 21–27.

Goodfellow I, B. Y. (2016). Deep Learn. *Deep Learning*.

L, B. (2001). Random forests. . *Mach Learning*, 5–32.

Quinlan JR, M. K. (1993). Programming for machine learning. . 38(48):49.

rajski KA, B. L. (1986). Classification of eeg spatial patterns with a tree-structured methodology. *IEEE transactions on biomedical engineering BME-33*, 1076–1086.

Wen, F. K. (2014). An Analytic Hierarchy Model for Classification Algorithms Selection in Credit Risk Analysis. *Mathematical Problems in Engineering*, 1.

Zhang, Z. &. (2018). *Credit risk assessment using machine learning and statistical approaches: a review. European Journal of Operational Research*, 271(1), 1-16.