Predicting Credit Risk: A Two-Stage Hybrid Model For Reducing False Alarm Rate.



**Yash Sinojia**

**A00268852**

A dissertation submitted in partial fulfilment of the requirements of Athlone Institute of Technology for the degree of

M.Sc. in Data Analytics

##### **Declaration**

I hereby certify that the material, which is submitted in this thesis towards the award of M.Sc. in Data Analytics, is entirely my own work and has not been submitted for any academic assessment other than part fulfilment of the above-named award.

Future students may use the material contained in this thesis provided that the source is acknowledged in full.

Signed\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_ **17th August 2020**\_\_\_\_\_\_\_\_

##### **Abstract**

Credit risk analysis is the foremost and most sensitive issue in any financial institution. The credit unions in Germany have been tackling the issues with several approaches but lack any standard approach which works for a general credit scenario as a set of financial organizations. A credit risk model implemented on machine learning techniques can help credit unions thrive in the current financial market dominated by banks. Moreover, systematic credit management can help credit unions attract more members, enhance their business, and market their brand to help certain communities striving to seek low rate loans. It might even help to build a bridge of trust between the lenders and the borrowers.

This research aims at predicting high credit risks from German Credit Data using a two-stage hybrid model of one clustering and one classification stage implemented on the most modern machine learning approaches. In the first stage, the numerical variables are utilized for clustering the data into more labels. These new labels stored in a new variable would be utilized with all the other features to classify the credit risk. Certain pre-processing methods such as scalar transform and PCA are employed to extract the most essential patterns from the features. After the analysis, the results would be evaluated by confusion matrices, ROC curves, and Precision-Recall curves. The authenticity of the results would be further clarified by cross-validation techniques.

The prime focus of this research lies in reducing the false alarm rate from the classification outcomes. It further expresses the issues inherent in the nature of data and addresses the specificity of certain machine learning models that may render less accuracy but still would be highly efficient in classifying each label and enhance the precision rate.

**Key words:** *Credit Risk, Credit Union, Classification, Clustering, Hybrid Model, False Alarm Rate, Evaluation.*

##### **Acknowledgements**

A foremost thanks to my supervisor Barraí Hennebry for his support, guidance, and making me maintain strict deadlines to complete my dissertation.

Secondly, many thanks to Louise for all her motivation and encouragement throughout the dissertation process and beyond.

**Table of Contents**

[Chapter 1 - Introduction 1](#_Toc46740149)

[**1.1** **Introduction** 1](#_Toc46740150)

[**1.1.1** **Background** 1](#_Toc46740151)

[**1.2** **Research Aim and Objectives** 2](#_Toc46740152)

[**1.2.1** **Research Question** 2](#_Toc46740153)

[**1.2.2** **Aim and Objectives** 3](#_Toc46740154)

[**1.3** **Research Scope and Limitations** 3](#_Toc46740155)

[Chapter 2 – Literature Review 4](#_Toc46740156)

[**2.1** **Introduction** 4](#_Toc46740157)

[**2.2** **Credit Risks, An Overview** 4](#_Toc46740158)

[**2.3** **Credit Scoring** 5](#_Toc46740159)

[**2.5** **Demographic And Socio-economical Factors Influencing Credit Risks** 6](#_Toc46740160)

[**2.6** **Classification On Credit Risk** 9](#_Toc46740161)

[**2.6.1** **Researches Performed** 9](#_Toc46740162)

[**2.7** **Clustering On Credit Risk** 15](#_Toc46740163)

[**2.7.1** **Research Performed** 15](#_Toc46740164)

[**2.8** **Evaluation of Credit Risk Model** 17](#_Toc46740165)

[**2.8.1** **Evaluation of Classification Models** 17](#_Toc46740166)

[**2.8.2** **Evaluation of Clustering Methods** 18](#_Toc46740167)

[**2.9** **Hybridisation** 20](#_Toc46740168)

[**2.9.1** **Clustering and Neural Network Hybrid** 20](#_Toc46740169)

[**2.9.2** **Clustering and Classification Hybrid** 23](#_Toc46740170)

[**2.10** **Conclusion** 24](#_Toc46740171)

[Chapter 3 – Research Methodology 25](#_Toc46740172)

[**3.1** **Introduction** 25](#_Toc46740173)

[**3.2** **Methodology Implimentation** 25](#_Toc46740174)

[**3.3** **German Credit Risk Dataset** 26](#_Toc46740175)

[**3.4** **Tools Used** 28](#_Toc46740176)

[**3.4.1** **Programming Platform** 28](#_Toc46740177)

[**3.4.2** **Data Analytics Software** 29](#_Toc46740178)

[**3.5** **Data Exploration** 29](#_Toc46740179)

[**3.6** **Data Preparation** 29](#_Toc46740180)

[**3.5.1** **Feature Engineering** 30](#_Toc46740181)

[**3.7** **Data Partioning** 31](#_Toc46740182)

[**3.8** **Machine Learning Techniques Utilized** 31](#_Toc46740183)

[**3.8.1** **K Means Clustering** 31](#_Toc46740184)

[**3.8.2** **Random Forest Classifier** 32](#_Toc46740185)

[**3.8.3** **Extreme Gradient Boost Classifier** 32](#_Toc46740186)

[**3.8.4** **Support Vector Classifier** 33](#_Toc46740187)

[**3.8.5** **Logistic Regression Classifier** 34](#_Toc46740188)

[**3.9** **Evaluation Metrics** 34](#_Toc46740189)

[**3.9.1** **Evaluation on Clustering** 34](#_Toc46740190)

[**3.9.2** **Evaluation on Classification** 34](#_Toc46740191)

[**3.10** **Conclusion** 35](#_Toc46740192)

[Chapter 4 – Findings and Analysis 36](#_Toc46740193)

[**4.1** **Introduction** 36](#_Toc46740194)

[**4.2** **Data Exploration** 36](#_Toc46740195)

[**4.2.1** **Dataset Variables** 36](#_Toc46740196)

[**4.3** **Clustering** 47](#_Toc46740197)

[**4.3.1** **Feature Extraction** 47](#_Toc46740198)

[**4.3.2** **K Means** 47](#_Toc46740199)

[**4.4** **Classification** 50](#_Toc46740200)

[**4.4.1** **Clustering Labels as Feature** 50](#_Toc46740201)

[**4.4.2** **Feature Extraction** 50](#_Toc46740202)

[**4.4.3** **Random Forest** 52](#_Toc46740203)

[**4.4.4** **Extreme Gradient Boosting** 54](#_Toc46740204)

[**4.4.5** **Support Vector** 56](#_Toc46740205)

[**4.4.6** **Logistic Regression** 58](#_Toc46740206)

[**4.5** **Analysis of Hybrid Models** 61](#_Toc46740207)

[**4.5.1** **ROC Curves** 61](#_Toc46740208)

[**4.5.2** **Precision-Recall Curves** 62](#_Toc46740209)

[**4.6** **Conclusion** 65](#_Toc46740210)

[Chapter 5 – Discussion and Conclusion 66](#_Toc46740211)

[**5.1** **Introduction** 66](#_Toc46740212)

[**5.2** **Discussion and Conlusion** 66](#_Toc46740213)

[**5.3** **Recommendations** 71](#_Toc46740214)

[References 72](#_Toc46740215)

[**Glossary of terms** 75](#_Toc46740216)

[**List of Abbreviations** 76](#_Toc46740217)

[Appendix 77](#_Toc46740218)

[**Machine Learning Libraries** 77](#_Toc46740219)

##### **List of Tables**

Table 2.4.1: Outcomes of causes upon indicators, reliability, and communality

Table 2.4.2: Confusion matrix for discriminant model accuracy

Table 2.4.3: LDA model estimation outcomes for assessing individual credit risk

Table 2.5.1: Distance-based classifiers - manual parameter selection

Table 2.5.2: Datasets used and their description

Table 2.5.3: Number of features selected for each dataset

Table 2.5.4: Mean time elapsed for each type of classifier

Table 2.6.1: Outcome Table for performance metrics on German credit dataset

Table 2.6.2: Outcome Table for performance metrics on Australian credit dataset

Table 2.6.3: Outcome Table for performance metrics on Korean credit dataset

Table 2.6.4: MCDM rankings for German credit dataset

Table 2.6.5: MCDM rankings for Australian credit dataset

Table 2.6.6: MCDM rankings for Korean credit dataset

Table 2.8.1: Clustering outcomes of German and Australian credit datasets

Table 2.8.2: Classification results of German credit dataset

Table 2.8.3: Classification results of Australian credit dataset

Table 2.8.4: Sensitivity analysis of German credit dataset

Table 2.8.5: Cluster centres (Left) and classification accuracy (Right) built on different attributes

Table 3.2.1: CRISP-DM tasks and their structure

Table 3.3.1: Description for German credit risk dataset

Table 3.3.2: Variable and nature of data view for German credit risk dataset

Table 3.3.3: Tables describing all the qualitative attributes with their symbols

Table 3.6.1: Snapshot of first ten instances of manipulated data

Table 4.3.1: Custer centroids Table

Table 4.4.1: Random Forest classification report

Table 4.4.2: Jupyter Notebook output for 10x cross validation score on Random Forest model

Table 4.4.3: XGBoost classification report

Table 4.4.4: Jupyter Notebook output for 10x cross validation score on XGBoost model

Table 4.4.5: Support Vector classification report

Table 4.4.6: Jupyter Notebook output for 10x cross validation score on Support Vector model

Table 4.4.7: Logistic Regression classification report

Table 4.4.8: Jupyter Notebook output for 10x cross validation score on Logistic Regression model

Table 4.5.1: Classification results

Table 5.2.1: Classification Report on Random Forest (Left) and XGBoost (Right)

Table 5.2.2: Classification Report on SVC (Left) and Logistic Regression (Right)

Table 5.2.3: Time complexity of classifiers

##### **List of Figures**

Figure 2.3.1: Credit scores for a potential individual borrower

Figure 2.3.2: Bond ratings on loan for a business borrower

Figure 2.4.1: Relationship between social and economic factors and credit risk management

Figure 2.4.2: Linear discriminant equation for socio-economic variables

Figure 2.5.1: Robustness of AAM, 1 < α < x

Figure 2.5.2: Robustness of AWM, 1 < γ < x

Figure 2.5.3: Robustness of TPM, 1 < β < x

Figure 2.5.4: Robustness of CCM, 1 < ω < x

Figure 2.5.5: (Left) E-coli dataset, features selected by SBC = 4; (Right) German Credit dataset, features selected by SBC = 6

Figure 2.5.6: (Left) Kr-vs-Kp dataset, features selected by SBC = 4; (Right) Pima-Indians dataset, features selected by SBC = 5

Figure 2.5.7: (Left) Monk dataset, feature selected by SBC = 4; (Right) Gene Promoter dataset, feature selected by SBC = 5

Figure 2.5.8: (Left) Mushroom dataset, feature selected by SBC = 6; (Right) Soybean dataset, feature selected by SBC = 12

Figure 2.5.9: (Left) Wisconsin Breast Cancer dataset, feature selected by SBC = 4; (Right) Congressional Voting Records dataset, feature selected by SBC = 3

Figure 2.7.1: Confusion matrix

Figure 2.7.2: ROC curve example

Figure 2.7.3: Precision-Recall curve example

Figure 2.7.4: Plot of Sum of within cluster distance vs number of clusters

Figure 2.7.5: Optimal gap statistic cluster value k

Figure: 2.8.1: Hybrid learning credit scoring system

Figure 2.8.2: (Left) Borrower distribution map of German and of Australian credit dataset (Right)

Figure 2.8.3: Integrated system architecture

Figure 3.2.1: Stages of experimental design for a methodology

Figure 3.4.1: Frameworks and libraries preinstalled with Anaconda distribution

Figure 3.8.1: K-means clustering example in python

Figure 3.8.2: Decision tree example in python

Figure 3.8.3: Gradient boosting decision trees example in python

Figure 3.8.4: Support vector classifier example in python

Figure 3.8.5: Logistic regression classifier example in python

Figure 4.2.1: Risk variable distribution

Figure 4.2.2: Checking account variable distribution

Figure 4.2.3: Duration variable distribution

Figure 4.2.4: Credit history details variable distribution

Figure 4.2.5: Purpose variable distribution

Figure 4.2.6: Credit amount variable distribution

Figure 4.2.7: Savings amount/bonds variable distribution

Figure 4.2.8: Present employment since variable distribution

Figure 4.2.9: Instalment rate variable distribution

Figure 4.2.10: Sex variable distribution

Figure 4.2.11: Other debtors variable distribution

Figure 4.2.12: Present residence since variable distribution

Figure 4.2.13: Property variable distribution

Figure 4.2.14: Age variable distribution

Figure 4.2.15: Other instalment plans variable distribution

Figure 4.2.16: Housing variable distribution

Figure 4.2.17: Number of credits variable distribution

Figure 4.2.18: Job variable distribution

Figure 4.2.19: Number of liable people variable distribution

Figure 4.2.20: Telephone variable distribution

Figure 4.2.21: Foreign worker variable distribution

Figure 4.3.1: (Left) Age distribution pre-transform, (Right) Age distribution post-transform

Figure 4.3.2: (Left) Duration distribution pre-transform, (Right) Duration distribution post-transform

Figure 4.3.3: (Left) Amount distribution pre-transform, (Right) Amount distribution post-transform

Figure 4.3.4: Sum of squared distances vs k plot

Figure 4.3.5: K-means output clusters

Figure 4.3.6: Silhouette analysis plot

Figure 4.3.7: Heatmap of clusters on random seeds

Figure 4.4.1: Cluster variable distribution

Figure 4.4.2: Heatmap on Pearson correlation coefficients among all variables

Figure 4.4.3: 2-Component PCA scatter plot

Figure 4.4.4: Cumulative explained variance ratio vs. number of components plot

Figure 4.4.5: Confusion matrix for Random Forest classifier model

Figure 4.4.6: Confusion matrix for XGBoost classifier model

Figure 4.4.7: Confusion matrix for Support Vector classifier model

Figure 4.4.8: Confusion matrix for Logistic Regression classifier model

Figure 4.5.1: ROC Curve for Random Forest Classifier

Figure 4.5.2: ROC Curve for XGBoost Classifier

Figure 4.5.3: ROC Curve for Support Vector Classifier

Figure 4.5.4: ROC Curve for Logistic Regression Classifier

Figure 4.5.5: Precision-Recall Curve for Random Forest Classifier

Figure 4.5.6: Precision-Recall Curve for XGBoost Classifier

Figure 4.5.7: Precision-Recall Curve for Support Vector Classifier

Figure 4.5.8: Precision-Recall Curve for Logistic Regression Classifier

Figure 4.5.9: Precision-Recall Curve for Random Forest Classifier (Optimal Precision)

Figure 4.5.10: Precision-Recall Curve for XGBoost Classifier (Optimal Precision)

Figure 4.5.11: Precision-Recall Curve for Support Vector Classifier (Optimal Precision)

Figure 4.5.12: Precision-Recall Curve for Logistic Regression Classifier (Optimal Precision)  
  
Figure 5.2.1: Confusion matrices for classification retrospection

Figure 5.2.2: Precision-Recall Curve for Random Forest Classifier (Optimal ROC\_AUC)

Figure 5.2.3: Precision-Recall Curve for Random Forest Classifier (Optimal Precision)

Figure 5.2.4: Precision-Recall Curve for Logistic Regression Classifier (Optimal ROC\_AUC/Precision)

Figure 5.2.5: Bar plot on feature importance for Random Forest classifier

Figure 5.2.6: Bar plot on feature importance for XGBoost classifier

Figure 5.2.7: Bar plot on feature importance for Logistic Regression classifier

# **Chapter 1 - Introduction**

## **1.1 Introduction**

The major causes of serious banking problems as determined by Basel Committee on Banking Supervision (2000) in any financial institutions are directly related to lower credit standards, poor credit risk management, or a deficiency of stance towards modification in the economy or the society that may cause a degeneration in the credit steadiness of a financial institution’s counterparties. These factors are quite similar and notable in both G-10 and non-G-10 countries.

Credit risk is a broad term generally overused in various financial institutions and organizations. To break it down, it is essential to understand a default risk. A default risk, also used in finance, is a probability that a counterparty under a financial agreement may not be able to meet a contractual promise to fulfill his/her indebtedness stated in the agreement. The occurrence of this event can be stated as that the party has defaulted, or that the default event occurred. Moreover, the risk associated with any sort of credit linked events can be defined as credit risks, such as changes in credit quality i.e. variations of credit deviations, rise, and fall on credit ratings and the event as default (Bielecki and Rutkowski, 2013).

For most financial institutions including credit unions, loans are the most discernible and highest cause of credit risk. Since credit risks to this date persist to be the major reason of troubles in financial institutions around the globe, some necessary lessons can be learned from the past experiences and sound practices can be addressed as reported by Basel Committee on Banking Supervision (2000) that an extensive credit risk management approach should handle the following four areas:

1. Setting up a relevant credit risk context.
2. Functioning under a robust credit approval approach.
3. Sustaining a pertinent credit control, evaluation, and supervising approach.
4. Assuring sufficient administration over credit risk.

### **Background**

A credit union is a member-owned financial cooperative organization, moderated by its autonomous members. Credit unions function with the aim of people helping people, distributing its credit resources to the members at competitive rates as well as other financial services. In other words, credit unions are financial institutions, like banks. However, credit unions are not-for-profit institutions whose purpose is to serve their members rather than banks whose primary obligation is towards their shareholders. Often, credit unions offer diminished transaction fees, lower loan rates, and better savings rates as contrasted with any other financial entities1.

When a person deposits funds in a credit union, he becomes a member-owner of that particular credit union. That person is both an owner and a client. The credit union utilizes those funds that the members deposit to distribute loans to other members in that credit union, much like a bank. Moreover, the profit earned from such a transaction is returned to their owner-members in terms of reduced fees, higher saving rates, and reduced loan terms.

Due to the huge rise in income inequalities in the late capitalist era of the 21st century, more and more people are developing anxieties and mistrust in banks. A study undertaken by Chicago Booth / Kellogg School reveals that according to Financial Trust Index, some 60% of respondents find credit unions trustworthy; only 30% say they trust big, national banks, which tend to invest in financial entities that are unfamiliar to most Americans2.

1 https://www.investopedia.com/terms/c/creditrisk.asp

Primarily credit unions implement credit scores based on a credit description, which is generated by the credit bureaus. A credit bureau is a data collection agency that collects account details from a range of creditors and renders those details to several credit agencies according to the region of the world it is based upon. A consumer/credit reporting agency is an institute rendering an individual’s account status, loan borrowing, and bill-paying habits. Such information can be a powerful means to predict the customer’s future behavior.

Credit unions are different from the banks in the following ways:

1. Credit unions are not-for-profit, member-owned, democratic financial institutions whereas banks are profit-centered financial institutions.
2. Any surplus financial gains are utilized to either develop new and existing services or are distributed amongst the members.
3. Loans and saving accounts are insured at no direct liabilities or costs.
4. Generally, no hidden fees i.e. registration, administration, or transaction fees for the members.
5. Lots of loan repayment flexibilities and conveniences offered to the members.
6. A credit union is involved in their corresponding local community activities e.g. elderly care, youth initiatives, charities, sporting, and cultural events.
7. Credit unions have a much lesser number of working employees than banks. In fact, some credit unions are managed by their member-owners entirely.

The foremost accomplished credit union was established in Germany during 1852 under cooperative pioneer Hermann Schulze-Delitzsch. The values of the credit unions until this date remains quite similar adhering to elemental facets of the co-operative integrity based on the values of solidarity, self-responsibility, self-help, democracy, equality, and equity. The credit union movement was formed with a thought that any society could attain a better standard of living for themselves by collaborating with their savings and distributing loans to neighbors and co-workers (Brooks, n.d.). Today, Germany has 70% community public institutions (including credit unions) which are on an average 200 years old and never demanded a cent of taxpayers’ funds. Whereas commercial banks hold only 12% of the financial market in Germany (Allen, 2010).

## **1.2 Research Aim and Objectives**

Already a substantial amount of research has been done on credit risk classification as described in Baesens *et al.* (2003) and Zekic-Susac *et al.* (2004) and on clustering or credit customer segmentation as described in Lundy (1993) and Chi *et al.* (2001). Some research has been also done on building a multi-staged hybrid model to carry classification tasks as described in Hsieh (2005) and Zakrzewska (2007). However, there is no hybrid model customization that would suit the credit risk scenario or anything similar. Moreover, the studies on hybrid modeling do not wholeheartedly reveal the implementation aspects of their researches. There has been a lot of focus on evaluating models with higher accuracy but there is a lack of information on what connects the two stages of the hybrid model and there is next to no information about the model parameters and other aspects of implementation that may suit a credit risk scenario.

2 http://www.financialtrustindex.org/resultswave22.htm

### **1.2.1 Research Question**

Pertaining to all the points, the research question stands as:

*“Which type of clustering-classification hybrid model can provide the best implementation to minimize the false alarm rate for the analysis of a given credit risk scenario?”*

### **1.2.2 Aim and Objectives**

As no specific method has been adopted by credit unions to model the credit risk analysis, the key aim of this research is to identify a clustering and a classification model that best implements the given credit scenario as a two-staged hybrid model, which minimizes the false alarm rate, and which leads to efficiency in predicting individual credit risks. The research objectives are as described below:

1. To gather the required data on individual socio-economical and demographical factors that may lead to credit risk.
2. Understanding the data, rectifying any issues, and utilizing it for machine learning models.
3. Preparing the features and data required for machine learning analyses.
4. Building two-staged clustering and classification hybrid models feasible for the credit risk scenario.
5. Identifying the best hybrid model by evaluating the model outcomes.

## **1.3 Research Scope and Limitations**

The outlook of this dissertation is to develop the best clustering-classification hybrid model to be implemented on a credit risk scenario for credit unions. Moreover, this research also covers the factors underlying in minimizing the false alarm rate which has not been found in any other previous research on the credit risk field.

One limitation of this research is that it does not involve any deep learning classifiers for the design of the hybrid model. Another limitation being the nature of target labels in the dataset is imbalanced and hence the classification stage would be subjected to fewer data labels stating ‘Bad Risk’ as compared to the labels stating ‘Good Risk’. This factor may affect the accuracy in predicting the ‘Bad Risk’ labels.

# **Chapter 2 – Literature Review**

## **2.1 Introduction**

This chapter revolves around current literature relating to credit risks, machine learning, and prediction methods influential to this research. The chapter begins by elaborating on some essential terminologies i.e. credit risk and credit scoring. Then the focus is centered on some less noticeable but highly impactful aspects that can influence credit risk. The influence of these aspects can then lead to implement some suitable machine learning techniques to analyze the German credit risk data. Moreover, this research in particular focuses on creating a hybrid of multiple machine learning techniques arranged in stages to render more effective analyses outcomes.

In section 2.5 and 2.6 focused on machine learning, two separate types of techniques as classification and clustering have been described as stages of the final model. The prominent past research which serves as a prelude to this research has been explored. Later these stages are combined to explore some case scenarios that delved into utilizing the stages to develop a hybrid technique that rendered better results. Techniques to determine the accuracy of prediction have been examined within each stage and overall is showcased in section 2.7.

## **2.2 Credit Risks, An Overview**

Credit risk is any probability of investment loss occurring from a loan recipient’s decline to repay those funds or to meet agreement indebtedness. Moreover, traditionally, it mentions the risk that a provider may not receive the owed capital of principal and interest, which tends towards a hindrance of monetary flows and raised costs for collection. This leads towards a higher coupon rate from the lender, resulting in greater monetary flows which may trigger additional cover for credit risk3.

Credit risk, or the risk associated with owed money not being repaid, has been prevailing in banking and every other financing history. Although it’s impossible to find out the exact extent of default on obligations, and even to specifically detect an individual surpassing the thresholds, optimized assessment and management of credit risk can actually decrease the severity of capital losses. And at the end, the interest payments from the credit recipient of the debt agreement are the lender’s or the investor’s accolade for assessing and managing the risks.

It is indeed a sensitive and foremost risk type that has been existing in monetary transactions from ancient cultures until today. Various small and large disasters further enhanced the relevance of credit risk management throughout the existence of time. Credit risk management is a procedure that includes the identification of potential risks, threats, the measurement of the extent of those risks, the viable solutions, and implementing those solutions as models. Efficient credit risk management solutions have been essential in approving the phenomenal rise in customer credit since the last century. Without accurate automated solutions, credit lending would be a very slow, inefficient, and tedious process. These days, effective credit risk management has been recognized by almost all organizations or institutions involving finance. All financial organizations and institutes, including credit unions, need to efficiently distribute capital in terms of the selective investments made. Henceforth, optimal implementation of methods for risk evaluation is some crucial foundation of good credit risk management (Van Gestel and Baesens, 2009).

3 https://www.investopedia.com/terms/c/creditrisk.asp

Losses can originate from several situations, as follows:

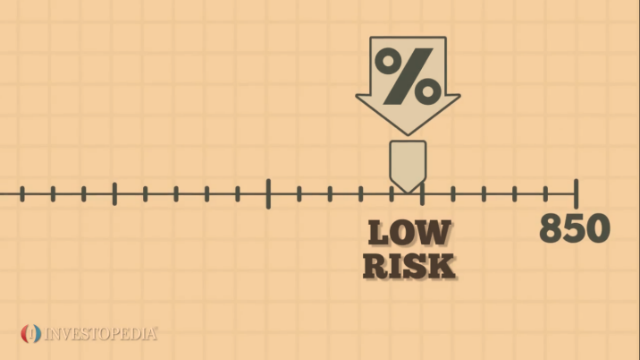
* Failing to make a payback by a borrower due on a loan, credit card or any other line of credit.
* Failure to repay asset-secured by a company, fixed or floating charge debt.
* A firm’s failure to pay a trade invoice when due.
* A firm’s failure to pay an employee’s deserved wages when due.
* Failure to pay for a coupon or principal by a bond issuer when due.
* An insurance company going insolvent and cannot pay back a policy agreement.
* A bank not returning funds to a depositor (Cole *et al.*, 2000).

## **2.3 Credit Scoring**

Many of the major small or large financial institutes or organizations already have special departments that focus solely on assessing and evaluating the credit risks of their current and potential borrowers. With advanced technologies, it is now quicker to analyze data used to evaluate a credit borrower’s credit risk profile. To evaluate credit risk on a customer applying for a loan, lenders look at the five Cs3:

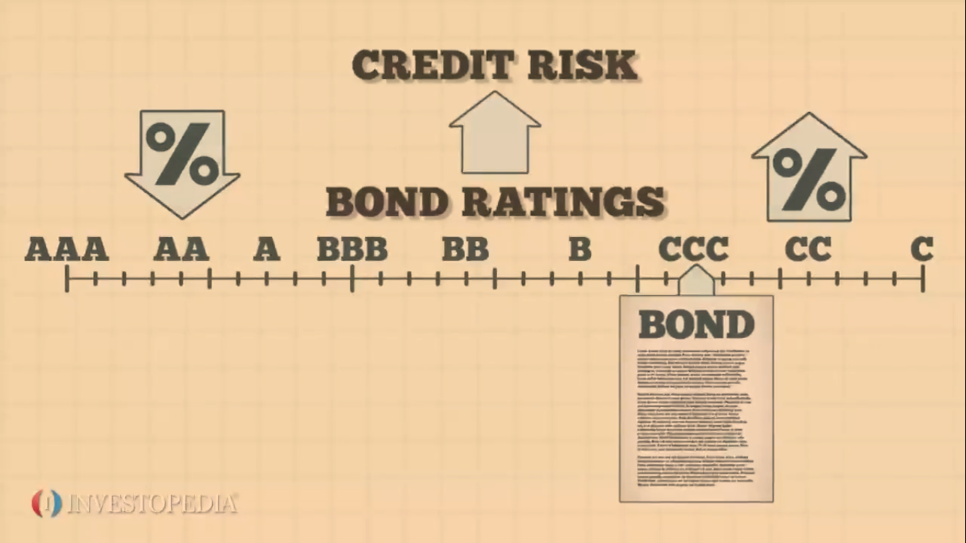
1. Credit History
2. Capacity to Repay
3. Capital
4. Conditions on Loan
5. Collateral Associations

A more general approach for loaners to determine a loan applicant’s credit risk is by assessing their corresponding credit score. The creditworthiness of an applicant comprises of his/her credit files which consist of a numerical expression widely known as a credit score. If an applicant is assessed with a higher credit score, he/she corresponds to lower credit risk to a loan provider and has the potential to be offered a loan with a lower interest rate. If an applicant is assessed with a lower credit score, he/she corresponds to higher credit risk to a loan provider and is less likely to be granted with a loan, and if they do they may be subjected to a higher interest rate.



**Figure 2.3.1: Credit scores for a potential individual borrower3**

Credit risk is also a major factor when bonds are considered. Companies do apply for a loan from their investors to raise money when they’re selling bonds to them. The bond is the agreement to pay back the loan. A more general approach for bond buyers to access the financial steadiness of these companies is through checking their bond ratings. A, AA or AAA rated bonds correspond to a company with strong financial steadiness. B or C rated bonds corresponds to a company with weaker financial steadiness. The company’s credit risk can be measured by these bond ratings. Companies with higher bond ratings represent lower credit risks and can be subject to lower interest rates. Companies with lower bond ratings represent higher credit risks and can be subject to higher interest rates to counterbalance investors for the extra risk.

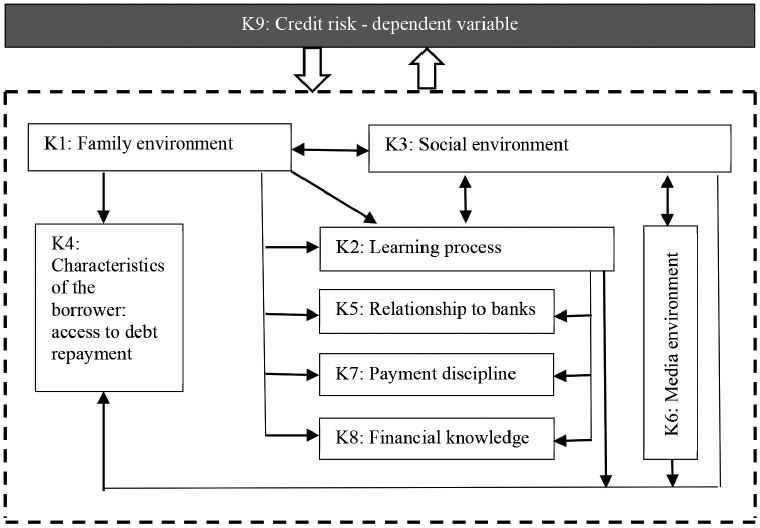


**Figure 2.3.2: Bond ratings on loan for a business borrower3**

## **2.4 Demographic And Socio-economical Factors Influencing Credit Risks**

Based on their specific target groups of customer research for credit risk management, there have been several selective studies. Earlier, bulk researches on the customer group as the Small to Medium sized Enterprises (SME) have been dominant on the economic impacts influencing credit risks in the SME whilst the non-economic factors have been shaping the studies to minimal, in steadfast forms, or are totally ignored. However, those non-economic factors do have notable psychological and sociological effects on a borrower’s behavior and management approaches in SMEs. The research area for these factors with a definite angle of subjectivism is of a qualitative nature which is essential in eliminating by implementing some advanced techniques. Therefore, in later studies, the dominant factors are the age and education of the loan borrowers, which plausibly determines the behavior of the business to credit risk and their connections are definitive, whether social, family, or media. The above statement has also been elaborated in the studies by Vos *et al.* (2007) and Kljucnikov *et al.* (2018) focusing on the relationship among age, level of business education, and the interest rate of external sources of funding. Younger and less educated business people rely more on external financing for their development as older business people are more educated. German experts on factors influencing credit risks Neuberger and Räthke-Döppner (2015) also based on their research the influencing impact of complex demographic and social factors and their outcomes confirm the inclination towards young business people to pay higher interest rates than their older counterparts. While the family background is directly proportional to its cause, it is also caused by the potential borrower in its pre-mature age, the social and the media with a more mature age (Wolfbein, 2017).

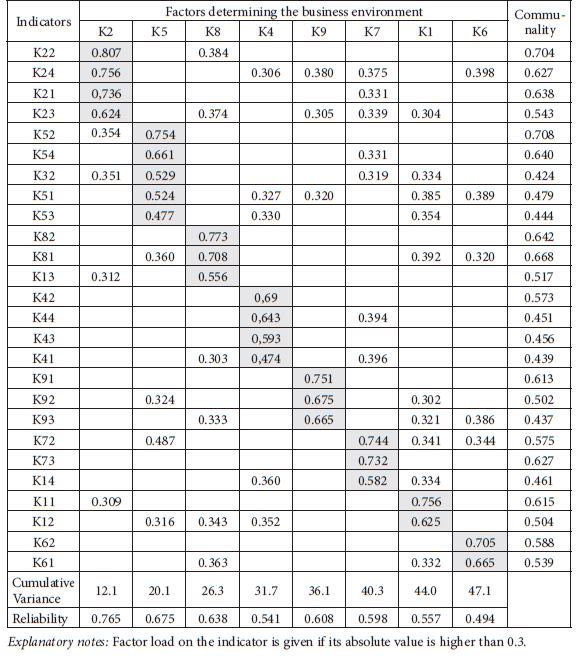
The proportionality of interconnectedness between socio-economic factors and credit risk management is described in Figure 2.4.1:



**Figure 2.4.1: Relationship between social and economic factors and credit risk management**

Source: (Kozubíková *et al.*, 2015)

The survey questionnaire by (Kozubíková *et al.*, 2015) for business people in SMEs consisted of 6 socio-demographic features of their business which consisted of the business sector, size of their enterprise, period of entrepreneurship, gender, highest business education achieved, and of 36 claims related to credit risk. Its practical reward lies above all in its implementation with the capacity to identify and accurately depict proportionality between causes that lay latent and unobserved and that do not explain the same cause analyzed in the first place.



**Table 2.4.1: Outcomes of causes upon indicators, reliability, and communality**

Source: (Wolfbein, 2017)

When surveying the outcome issues, the resulting groups of relational equations allows forming a hierarchical structure of causes ascertaining the overall sense of credit risk and also to ascertain the proportionality among causes with the support of several indicators (Olsson *et al.*, 2000). On top of that, a GUI was installed, rendering grants for both tabular and graphical outcomes. By setting up a statistical hypothesis that if a theoretical model is subjected to a chi-square test, the region of the confidence interval would determine with 95% accuracy (alpha = 0.025, two-tailed test) that the new model works or not? Does it have statistically significant results to prove the change (Hancock and Mueller, 2007)?

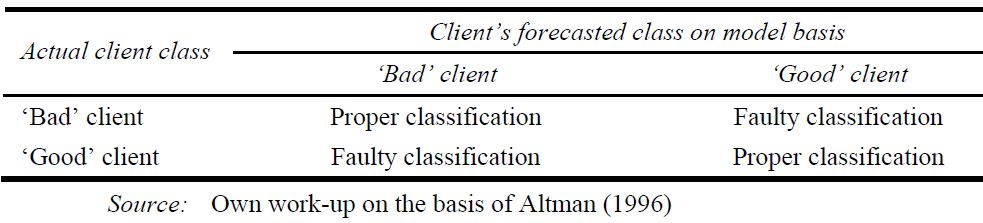
Here, kn is an nth coefficient of a social or a demographic variable. The x-variable is termed as the status where the loan borrower is in default of the conditions of a loan contract. In a scenario where the potential borrower has paid-off a loan, the borrower receives a value of 1; if not, a value of 0. This coefficient list is utilized in the consecutive analysis process (Balina and Nowak, 2017). This study utilized some econometric methods of processing data which enhances the construction of a model to forecast the risk causes linked in approving a loan to a potential borrower of a credit union. This model was implemented for discriminant analysis to classify sets of various features (Kolonko and Schäl, 1979). Thus, the linear discriminant equation takes on the form as shown in Figure 2.4.2:



**Figure 2.4.2: Linear discriminant equation for socio-economic variables**

Source: own processing

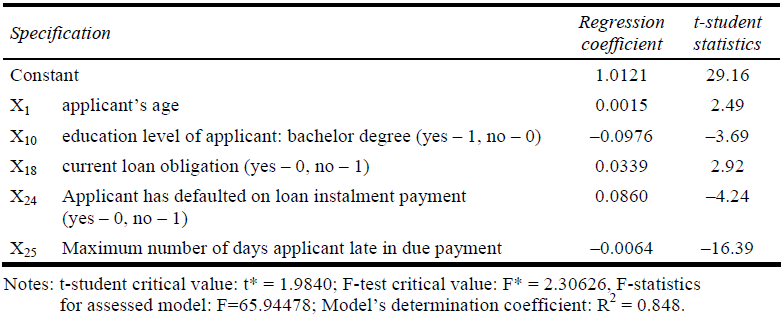
Based on the linear discriminant equation in the previous section, a linear discriminant analysis (LDA) model as a supervised learning technique has been implemented and tested. The linear discriminant function of the LDA’s structural parameter estimation was done with the smallest square method. The function of this model is to accurately predict the insolvency risk of a potential borrower from a credit union. The reverse regression model was utilized. This model eliminates a number of variables until the optimum number remains (Balina and Nowak, 2017). After testing and model verification, the outcomes were subjected to an assessment with regards to the output model’s capabilities in assessing individual credit risk. The assessment was how effectively the borrower group classification was carried out as ‘good’ or ‘bad’. A ‘k x k’ confusion matrix was applied to check the outcomes; the structure of the matrix can be simplified as shown in Table 2.4.2:



**Table 2.4.2: Confusion matrix for discriminant model accuracy**

Source: (Balina and Nowak, 2017)

The limit value separating the good clients from the bad was fixed at 0.5. Five variables remained uneliminated in the final model as shown in Table 2.4.3:



**Table 2.4.3: LDA model estimation outcomes for assessing individual credit risk**

Source: (Balina and Nowak, 2017)

These variables with their corresponding firm coefficients render a credit risk indicator that is highly efficient in the determination of potentially insolvent borrowers. The average efficiency of the tested model on the control group was 100% whilst for the test group 96% (Balina and Nowak, 2017).

## **2.5 Classification On Credit Risk**

The technique implemented in the previous section is a supervised learning technique called classification. Any supervised learning model needs an input of labeled training data consisting of actual training examples the model learns from the assigned labels in training data and then the model can be utilized for a different set of testing data. Classification is the approach of predicting the labels of those testing data points. These predicted labels also known as classes can be also referred to as targets or categories. There are a number of classification algorithms, each known for its distinctive applications and hence it is not possible to pick the superior one. Even the accuracy of prediction within the same model can change with regards to the dataset and various parameters. The prediction is dependent on the application and nature of the available dataset.

### **2.5.1 Research Performed**

#### **2.5.1.1 Distance Based Classification Methods**

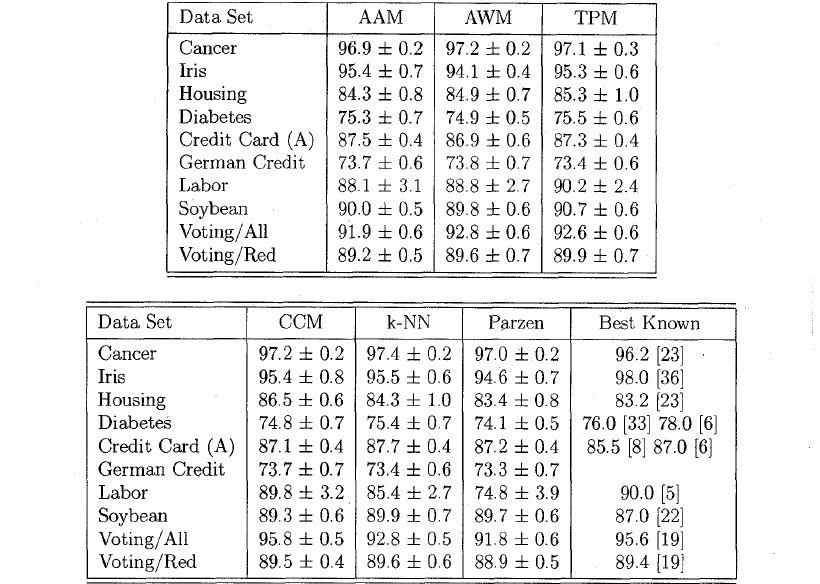
By focusing on probability and statistics, a new set of distance-based classifiers came into existence as researched by Ekin *et al.* (1999). The below methods were shortlisted as:

1. Adjusted Averaging Method (AAM)
2. Adjusted Weighting Method (AWM)
3. Truncated Potentials Method (TPM)
4. Convex Containment Method (CCM)

The datasets that were used belong to the University of California, Irvine Repository. These proved to be quite extensive for the research. The nominated datasets were:

1. Breast Cancer Diagnosis (Wisconsin)
2. Classifying Irises
3. Housing Costs in Boston
4. Diabetes Diagnosis (Pima Indians)
5. Credit Cards (Australian)
6. **German Credit (Statlog)**
7. Labour Negotiations
8. Soybean
9. Voting

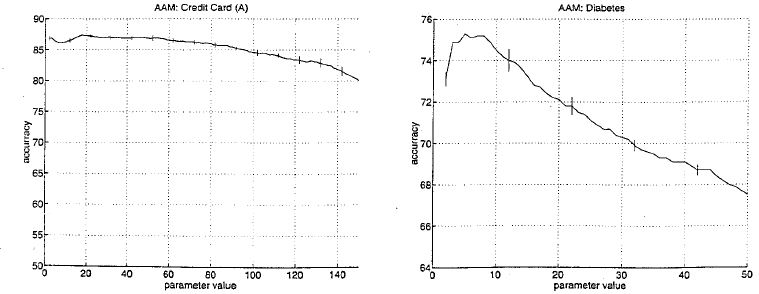
A cross-validation approach was utilized to evaluate these distance-based methods. The ratio of the units of true classification on query points to the sum of query points, which can be described the approximation of the accuracy of the method under evaluation (Ekin *et al.*, 1999). For an effective outcome, a k-fold cross-validation procedure has been reused as described in Table 2.5.1:



**Table 2.5.1: Distance-based classifiers - manual parameter selection**

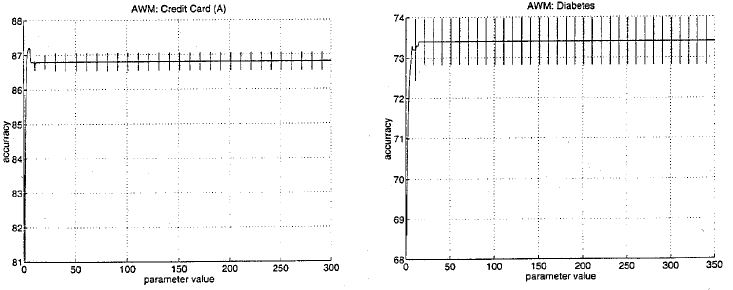
Source: (Ekin *et al.*, 1999)

Moreover, this k-fold cross-validation can be applied as a number of iterations, each iteration with a varied and randomized partition of those datasets. The mean accuracy is then summarized as the outcome accuracy. The results visualized how the method acts when parameters are calibrated (Ekin *et al.*, 1999). Each record as a mean is over 150 runs (5x cross-validation \* 30 runs). The upper bounds of some methods can be visualized as in Figure 2.5.1, 2.5.2, 2.5.3, and 2.5.4:



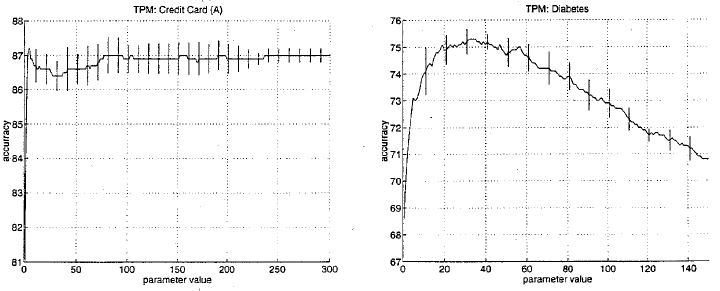
**Figure 2.5.1: Robustness of AAM, 1 < α < x**

Source: (Ekin *et al.*, 1999)



**Figure 2.5.2: Robustness of AWM, 1 < γ < x**

Source:(Ekin *et al.*, 1999)



**Figure 2.5.3: Robustness of TPM, 1 < β < x**

Source: (Ekin *et al.*, 1999)



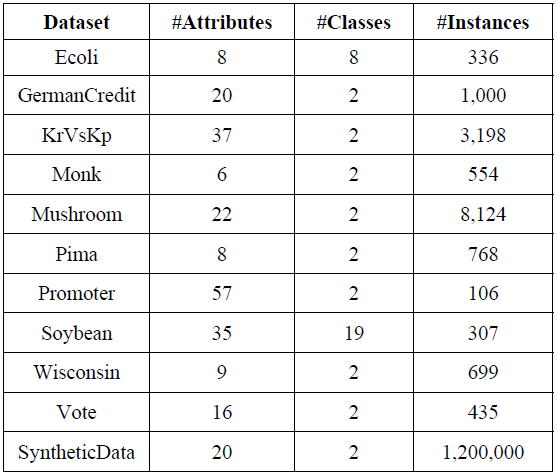
**Figure 2.5.4: Robustness of CCM, 1 < ω < x**

Source: (Ekin *et al.*, 1999)

Even though all these methods determine proximity using measures, the terminology of proximities varies substantially among each method. In the development of these proximity measures, the distances from the query point to the record points from the training data are not utilized definitively. A notable outcome that can be taken from the study by Ekin *et al.* (1999) is the matter that despite the substantially similar approaches in which the proximities are characterized, the achievements, as visualized in Figure 2.5.1, 2.5.2, 2.5.3, and 2.5.4, are indeed similar.

#### **2.5.1.2 Naïve Bayes Classifier for Feature Selection**

Bayesian classifiers work better on some areas, and poorly on the others, i.e. areas that include correlated features; where decision trees perform a lot better. One study by Ratanamahatana and Gunopulos (2003) explores a Selective Bayesian classifier that actually imitates those aspects that C4.5, an algorithm based on decision trees classification would use. This experiment used 10 datasets from the University of California, Irvine Repository. 5 of which Naïve Bayes classifier outperforms C4.5 and that C4.5 outperforms on the other 5 as shown in Table 2.5.2.



**Table 2.5.2: Datasets used and their description**

Source: (Ratanamahatana and Gunopulos, 2003)

* The experimental training and testing set in the scenario were formed as follows:

1. 10% of training and 90% of test data
2. 20% of training and 80% of test data

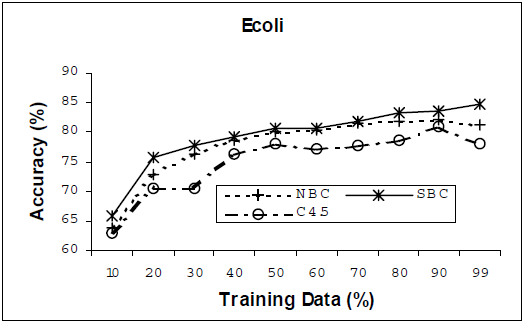
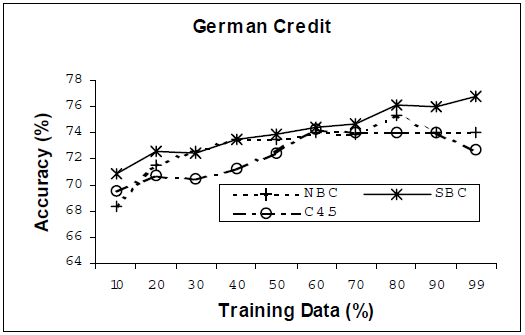
…

1. 90% of training and 10% of test data

* For each group of training and test set, run

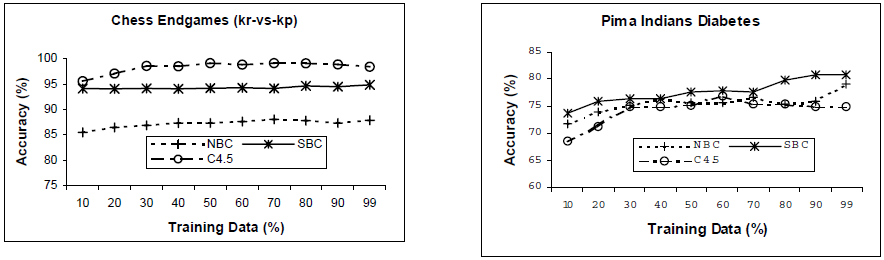
1. NBC
2. C4.5
3. SBC

* Repeat 10 times



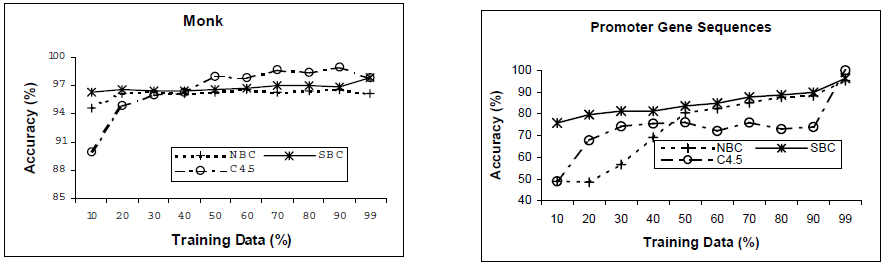
**Figure 2.5.5: (Left) E-coli dataset, features selected by SBC = 4; (Right) German Credit dataset, features selected by SBC = 6**

Source: (Ratanamahatana and Gunopulos, 2003)



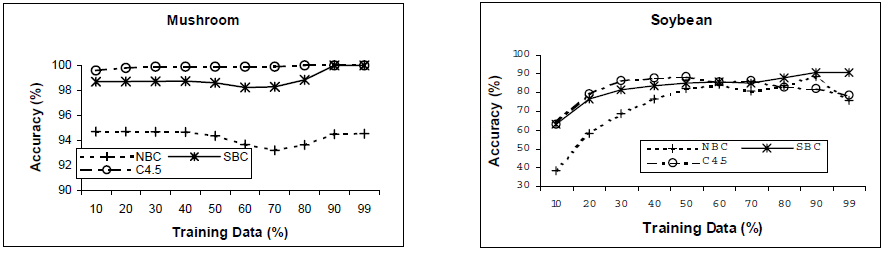
**Figure 2.5.6: (Left) Kr-vs-Kp dataset, features selected by SBC = 4; (Right) Pima-Indians dataset, features selected by SBC = 5**

Source: (Ratanamahatana and Gunopulos, 2003)



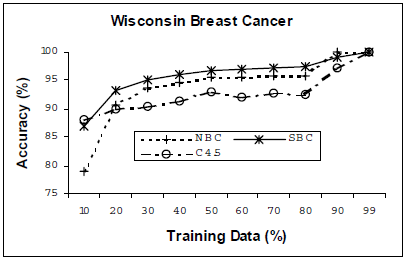
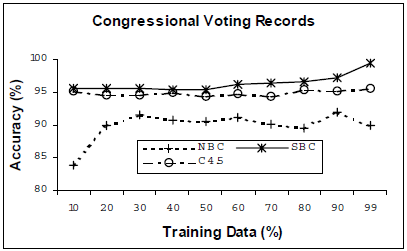
**Figure 2.5.7: (Left) Monk dataset, feature selected by SBC = 4; (Right) Gene Promoter dataset, feature selected by SBC = 5**

Source: (Ratanamahatana and Gunopulos, 2003)



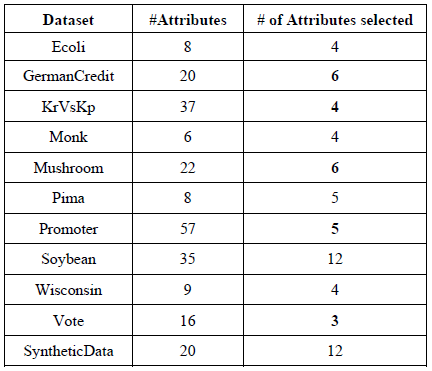
**Figure 2.5.8: (Left) Mushroom dataset, feature selected by SBC = 6; (Right) Soybean dataset, feature selected by SBC = 12**

Source: (Ratanamahatana and Gunopulos, 2003)



**Figure 2.5.9: (Left) Wisconsin Breast Cancer dataset, feature selected by SBC = 4; (Right) Congressional Voting Records dataset, feature selected by SBC = 3**

Source: (Ratanamahatana and Gunopulos, 2003)



**Table 2.5.3: Number of features selected for each dataset**

Source: (Ratanamahatana and Gunopulos, 2003)



**Table 2.5.4: Mean time elapsed for each type of classifier**

Source: (Ratanamahatana and Gunopulos, 2003)

These results prove that:

1. C4.5 does nominate better features for its classification, which in turn enhances the accuracy of NB when only those particular features are selected in the training.
2. The time complexity of SBC is best. Moreover, SBC shortlists less than half of the features. However, SBC trains rapidly because of fewer features involved.
3. Better features can be nominated on small samples of the dataset, only 10% of training data from UCI datasets proved to be quite sufficient (Ratanamahatana and Gunopulos, 2003).

## **2.6 Clustering On Credit Risk**

An unsupervised learning method is a kind of machine learning method that trains a model without taking into consideration the labels associated with the dataset. This task groups unsorted information according to similarities, patterns, and differences without any prior information of any prior labels on the training data. Clustering is such an unsupervised learning technique that divides the data points or records into groups or clusters. The data points in the same cluster have more similarities and have dissimilarities with the data points in other clusters (Priy, n.d.). Clustering has also applications in dimensionality reduction of the dataset, to avoid redundant variables.

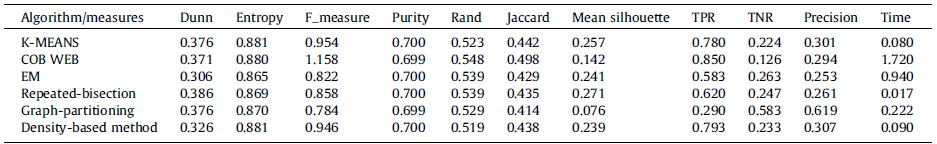
### **2.6.1 Research Performed**

#### **2.6.1.1 Multiple CRITERIA DECISION-MAKING Clustering Methods**

Although supervised learning models can render high accuracy in prediction, they are inapplicable when the credit risk dataset has no predefined target labels. Moreover, unsupervised learning models can even create labels for supervised models. A study by Kou *et al.* (2014) nominates six clustering techniques: k-means, COBWEB repeated-bisection approach, expectation-maximization (EM), density-based method, and graph-partitioning algorithm. For evaluating these algorithms, three multiple criteria decision-making (MCDM) algorithms have been shortlisted. According to the most verified ranking by multiple MCDM methods. The evaluation methods are:

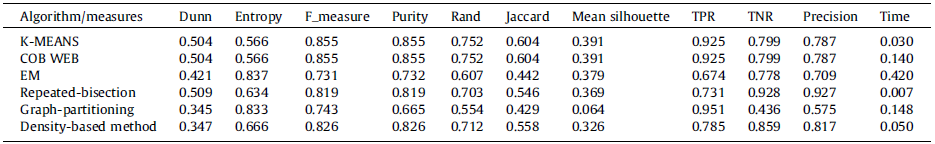
* *Technique for order preference by similarity to ideal solution (TOPSIS)* method is utilized to provide the best substitute by minimizing the measures to the positive optimal outcome and negative optimal outcome.
* *Data envelopment analysis (DEA)* evaluates the capabilities of decision-making groups. It has a relative benefit to evaluate, with few prior assumptions.
* *Multi-criteria optimization and compromise solution (\*translated) (VIKOR)* for multicriterial enhancement of complex systems. It ranks substitutes subjected to conflicting criteria from the multicriteria ranking index, based on the solution of distance to optimal substitute (Kou *et al.*, 2014).

Three financial datasets were analyzed in the experiment and the outcome Tables are as follows:



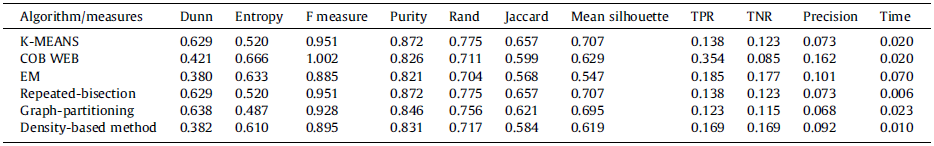
**Table 2.6.1: Outcome Table for performance metrics on German credit dataset**

Source: (Kou *et al.*, 2014)



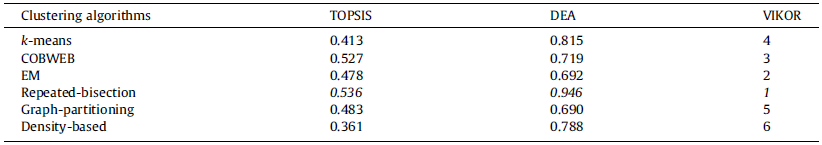
**Table 2.6.2: Outcome Table for performance metrics on Australian credit dataset**

Source: (Kou *et al.*, 2014)



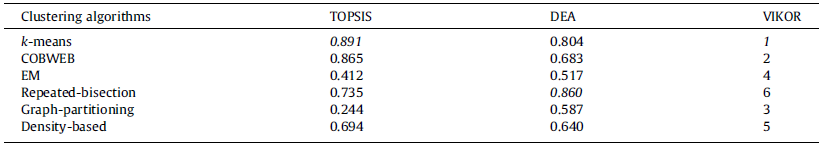
**Table 2.6.3: Outcome Table for performance metrics on Korean credit dataset**

Source: (Kou *et al.*, 2014)



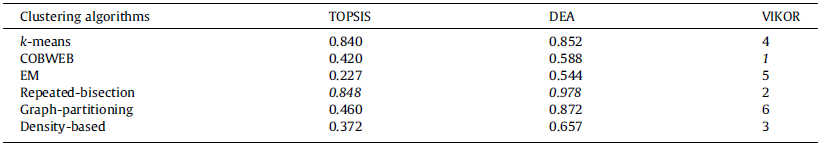
**Table 2.6.4: MCDM rankings for German credit dataset**

Source: (Kou *et al.*, 2014)



**Table 2.6.5: MCDM rankings for Australian credit dataset**

Source: (Kou *et al.*, 2014)



**Table 2.6.6: MCDM rankings for Korean credit dataset**

Source: (Kou *et al.*, 2014)

The Tables 2.6.1, 2.6.2, 2.6.3, 2.6.4, 2.6.5, and 2.6.6 convey that no method can gain the best achievement on all measurements for any of the three datasets. The experiment scenario proves that the repeated-bisection algorithm outperforms on all three datasets. However, the rankings resulting from these three MCDM methods are not similar. But these methods in general agree on the top-ranked repeated-bisection algorithm (Kou *et al.*, 2014).

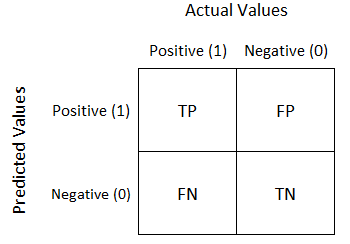
## **2.7 Evaluation of Credit Risk Model**

Evaluation of any machine learning model is to verify its applicability.

### **2.7.1 Evaluation of Classification Models**

#### **2.7.1.1 Classification Report and Confusion Matrix**

The classification report describes the metrics: precision, recall, and f1-score on per class basis. The classes of the target labels are ‘Good Risk’ and ‘Bad Risk’. The classification report describes the quality of predictions from a classification model. The metrics are calculated by using true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). False positives and false negatives are also called type 1 error and type 2 error respectively. These four bases are arranged in the confusion matrix as described in Figure 2.7.1:



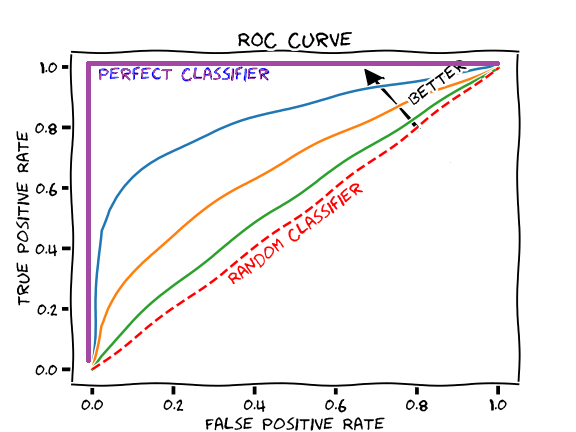
**Figure 2.7.1: Confusion matrix**

Source: (Narkhede, 2018)

Precision is the capability of a classifier to correctly predict the positive results. In this case, it stands for the ratio of records identified with a risk to the correctly identified records. Recall is the capability of a classifier to find all positive instances. In this case, it stands for the ratio of records correctly identified with a risk to all correctly identified records. F1-score is a weighted harmonic mean of precision and recall.

#### **2.7.1.2 Receiver Operating Characteristics Plot**

Receiver Operating Characteristics (ROC) curve is a visualized alternative for classification models that describes the variation of true positive rate with change in the false positive rate. The area under the ROC curve (AUC) describes the accuracy of the model (Asiri, 2018). A sample ROC curve can be shown as in Figure 2.7.2:

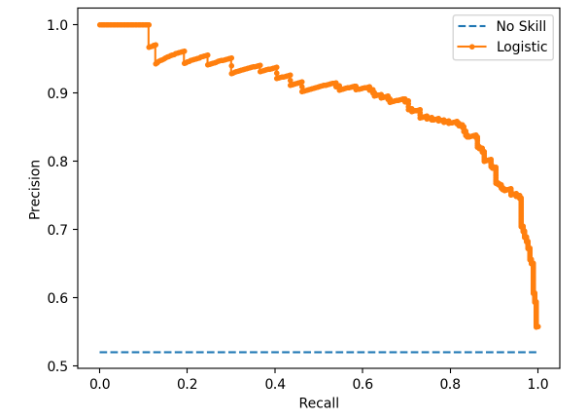


**Figure 2.7.2: ROC curve example**

Source: (Draelos, 2019)

#### **2.7.1.3 Precision and Recall Plot**

The precision and recall curves describe the variation of precision within all possible values of recall in the range of 0 to 1. These types of plots are especially essential for the classification of target variables with imbalanced classes. It is essential to do this analysis for precision-recall trade-offs pertaining to the aim of the study and the priorities of results. An example precision-recall curve can be shown as in Figure 2.7.3:



**Figure 2.7.3: Precision-Recall curve example**

Source: (Brownlee, 2020)

### **2.7.2 Evaluation of Clustering Methods**

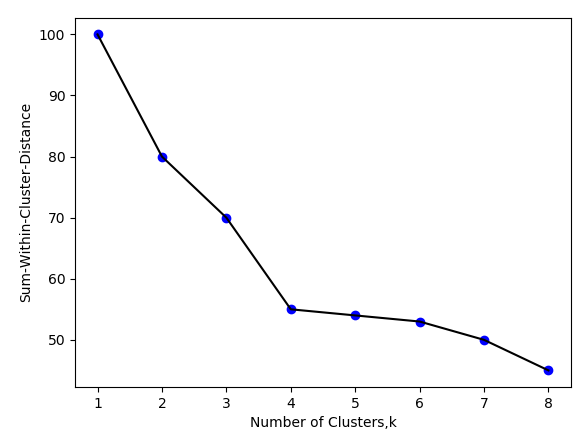
#### **2.7.2.1 Clustering Tendency**

As a starting point, it is essential to make sure that the dataset to be worked upon has a clustering tendency and does not consist of uniformly distributed points. This evaluation can help identify any redundancy that might be relevant to the further evaluation process. To solve the problem, a statistical hypothesis test called the Hopkins test can be applied to test for special randomness of a variable or the probability of data points generated by a uniform distribution (Manimaran, 2019).

#### **2.7.2.2 Number of Optimal Clusters**

Some of the clustering algorithms, i.e. k-means, requires the numerical quantity of final clusters whereas some clustering algorithms, i.e. hierarchical, require the optimum threshold to ascertain the minimum number of final clusters. If the number is too high, each individual data point tends to represent a cluster. Whereas if the number is too reduced, then the data points are prone to faulty clustering (Manimaran, 2019).

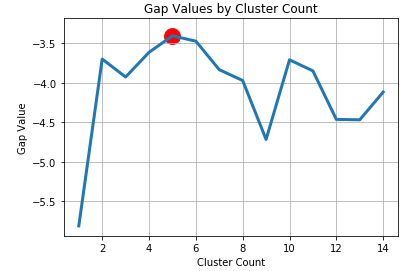
There are several techniques to control the number of clusters. One is the general empirical method to find the square root of N/2 where N is the total number of record points in data. Elbow method calibrates the within-cluster variance and adjusts the compactness within each cluster, accordingly.



**Figure 2.7.4: Plot of Sum of within cluster distance vs number of clusters**

Source: (Manimaran, 2019)

Another such technique is gap statistics which also checks the sum of intra-cluster variance for different k. Then records from data picked randomly from a reference null distribution is created and again a sum of within-cluster variance is calculated for the grouping done for distinct values of k. The corresponding values of k are compared to find the ideal k value where the gap is highest. The number of clusters with a maximum gap statistic value corresponds to optimal k (Manimaran, 2019).



**Figure 2.7.5: Optimal gap statistic cluster value k**

Source: (Manimaran, 2019)

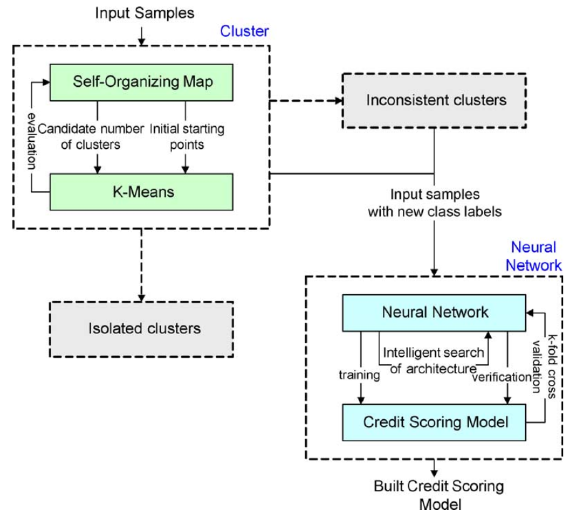
#### **2.7.2.3 Clustering Quality**

The performance of clustering can be measured with respect to the optimal clustering which is distinguished by maximal inter-cluster distance and minimal intra-cluster distance. Extrinsic measures require base point labels, i.e. adjusted rand index, mutual information scores, completeness, homogeneity, etc. Intrinsic measures are not based on any base points, in particular, i.e. silhouette coefficient, Davies-Bouldin index, etc.

## **2.8 Hybridisation**

### **2.8.1 Clustering and Neural Network Hybrid**

One study by Hsieh (2005) utilizes clustering and neural network algorithms to unveil a hybrid mining process designed to handle credit scoring analysis effectively. Clustering has been utilized to pre-treat the input records with the aim of separating unrepresentative records into isolated and inconsistent clusters, on top of that a neural network layer to encapsulate it into a credit scoring model. A self-organizing map clustering algorithm has been implemented to trace the number of clusters and the initial location of each cluster. Then, the K-means clustering algorithm has been implemented to make clusters of representative samples and remove the unrepresentative samples from each group. In the neural network stage, records with new target records were finally incorporated in the formulization of the credit scoring model. The system can be visualized as shown in Figure 2.8.1:

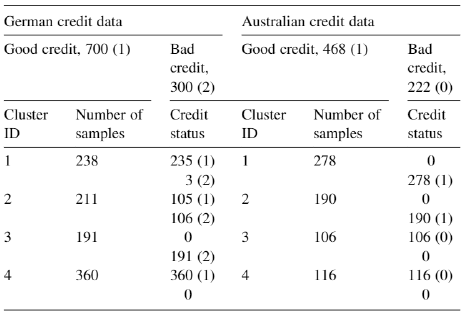


**Figure: 2.8.1: Hybrid learning credit scoring system**

Source: (Hsieh, 2005)

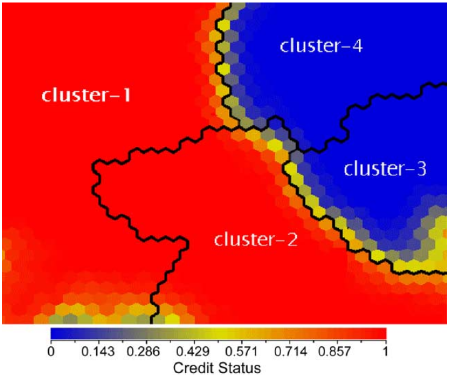
Previous research have described that neural networks render improved performance than statistical algorithms, i.e. linear discriminate analysis, logistic regression, etc as stated by Lacher *et al.* (1995), Desai *et al.* (1996) and Malhotra and Malhotra (2003). These authors realized that the shortcomings in the developed model are due to their design, rather than the inefficiency of a neural network. Other issues such as nature and instances of the training set can be tackled by clustering algorithms. Moreover, misclassification happens due to the lower quality of training samples. The unrepresentative samples can be eradicated as thinly populated clusters and inconsistent class values by clustering techniques (Hsieh, 2005).

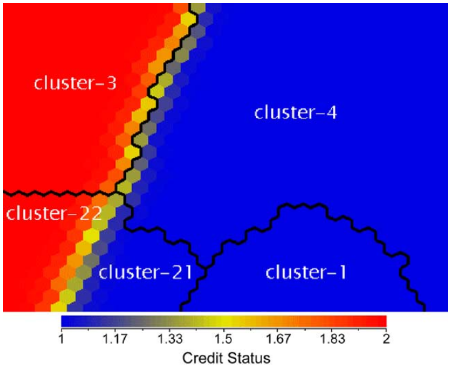
For the unsupervised learning layer, a trial and error approach has been utilized to gain training parameters of those clustering algorithms. After the tests, the best segmentation score of the two credit risk datasets was concluded to be four. The clustering results for both of the datasets are as shown in Table 2.8.1 and Figure 2.8.2:



**Table 2.8.1: Clustering outcomes of German and Australian credit datasets**

Source: (Hsieh, 2005)

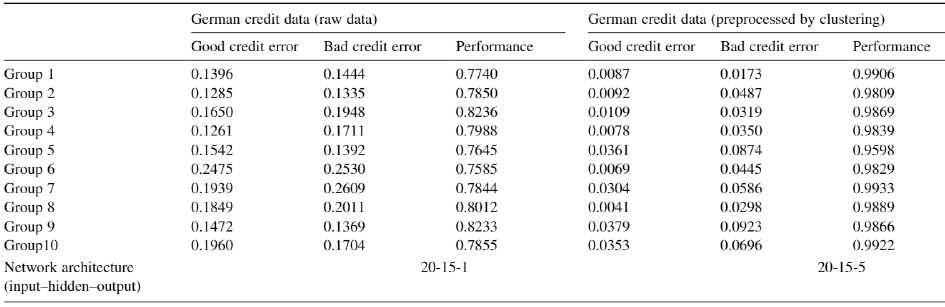




**Figure 2.8.2: (Left) Borrower distribution map of German and of Australian credit dataset (Right)**

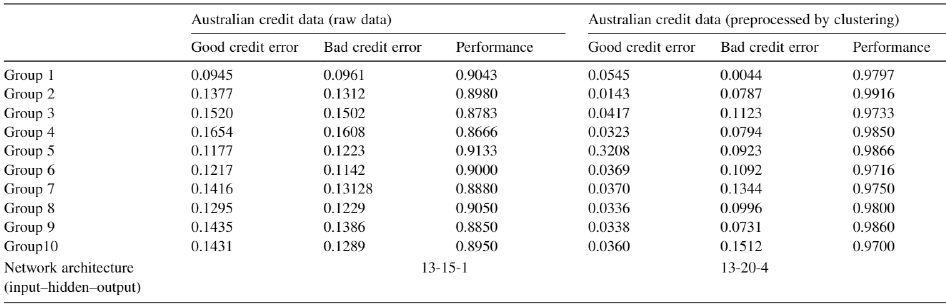
Source: (Hsieh, 2005)

In building the credit scoring model, the dataset was partitioned into 10 mutually exclusive groups. The neural network was trained to utilize the initial 9 groups and was tested utilizing the 10th group, and iterated 10 times as a 10-fold cross-validation technique. The supervised feedforward MLP trained by back-propagation and gradient descent neural network was implemented as the model. The results of the classification are as follows in Table 2.8.2, 2.8.3 and 2.8.4:



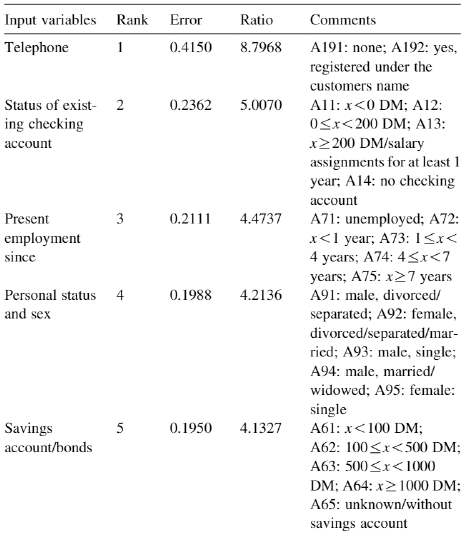
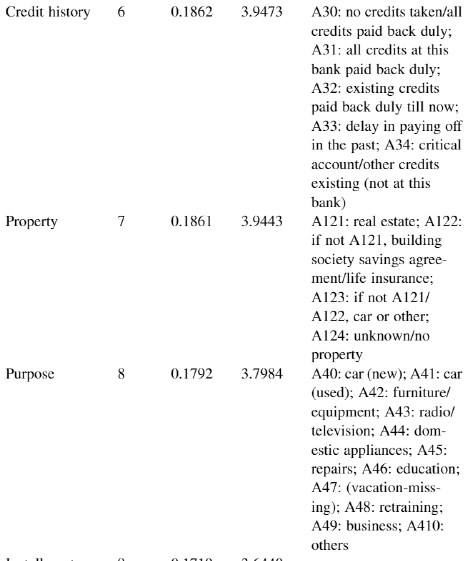
**Table 2.8.2: Classification results of German credit dataset**

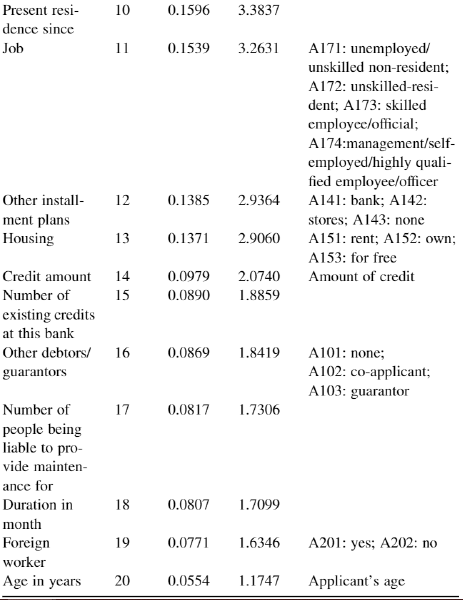
Source: (Hsieh, 2005)



**Table 2.8.3: Classification results of Australian credit dataset**

Source: (Hsieh, 2005)



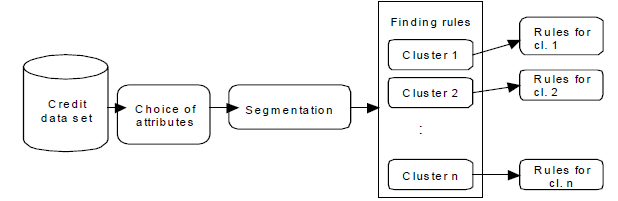


**Table 2.8.4: Sensitivity analysis of German credit dataset**

Source: (Hsieh, 2005)

### **2.8.2 Clustering and Classification Hybrid**

A study by Zakrzewska (2007) attempts to describe an integrated system of clustering and classification methods. In the first stage, by using clustering techniques, potential borrowers are disjointed into clusters with identical attributes. In the second stage, decision trees are grown and pruned for each cluster. This process enhances the use of distinguishing rules within the same dataset, and for pointing out more effectively the records with high risk. The architecture is shown in Figure 2.8.3:

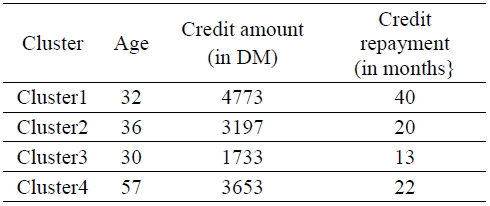


**Figure 2.8.3: Integrated system architecture**

Source: (Zakrzewska, 2007)

Clustering has been well known in the customer segmentation area. The advantage is that a clustering technique never assumes any particular arrangement on the data, so it is highly recommendable for credit risk analysis. For this stage, the k-means algorithm has been nominated due to its clarity and efficiency on big multi-dimensional datasets. Some attributes in the segmentation module have been modified to numerical attributes and normalized.

C4.5 algorithm based on ID3 decision tree induction was implemented for the classification stage with a tree pruning feature. The induction is a subject to every cluster found in the previous stage. Decision rules were built on the German credit dataset with all 17 aspects by utilizing different aspects for each cluster. For each cluster, at least one aspect is essential to formulate the rule (Zakrzewska, 2007). The cluster centers and classification accuracy can be shown as follows:



**Table 2.8.5: Cluster centres (Left) and classification accuracy (Right) built on different attributes**

Source: (Zakrzewska, 2007)

In the first stage, four clusters were chosen as optimal and Euclidean functions were utilized for measures. In the second stage, the nominated case models were made on the full training set by implementing 10x cross-validation, C4.5 technique. Results obtained in Table 2.8.5 shows notable high precisions and clarity of rules incurred for each cluster than for rules linked with the whole dataset (Zakrzewska, 2007).

## **2.9 Conclusion**

This chapter focused on the current literature in the area of credit risk. Furthermore, the emphasis on socio-economic factors in determining credit risk was discussed. It has been verified in this chapter that the factors which contribute to high-level credit risk are multifaceted and difficult to quantify and can be considered quite non-linear. To solve this problem, various machine learning techniques have been suggested by the elaborated past research. However, an attempt has been made to improve the accuracy of those models by introducing a classification-clustering hybrid model. The resultant model would be contrasted with basic models for verification and various evaluation techniques described in Chapter 2 can be utilized.

In a nutshell, this research centers on the efficient building of a hybrid model with two machine learning stages which can provide more accurate evaluation outcomes than any other single-stage model. In addition, this study has been exclusively based on credit risk predictions for credit unions in Germany, which was further explored by studying some generic trends in such a specific financial environment.

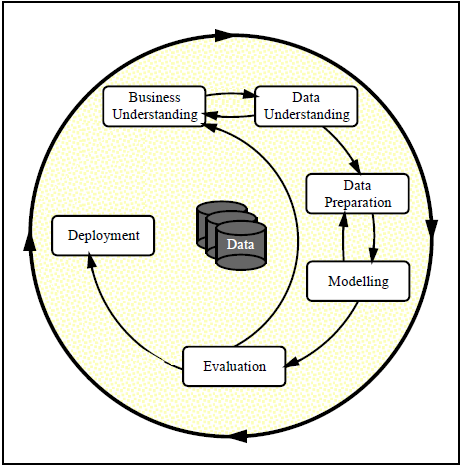
# **Chapter 3 – Research Methodology**

## **3.1 Introduction**

Assimilating all the knowledge obtained from the literature in Chapter 2, Chapter 3 revolves around determining the methods to carry out the analysis of this study. Foremostly, it is essential to select the best methodology model that suits this research in particular. Then, the German credit risk dataset is unveiled with all its attributes and datatypes. After that, the means to perform the analysis i.e. tools and platform are explored. Before beginning the central analysis, it is of utmost importance to discuss how the data is prepared, explored for all its relationships and features, and then partitioned to make it more feasible for the algorithm to run. Finally, the nominated machine learning algorithms are described with the reasons for their nominations. Then, at last, a brief on the evaluation of these algorithms is provided.

## **3.2 Methodology Implimentation**

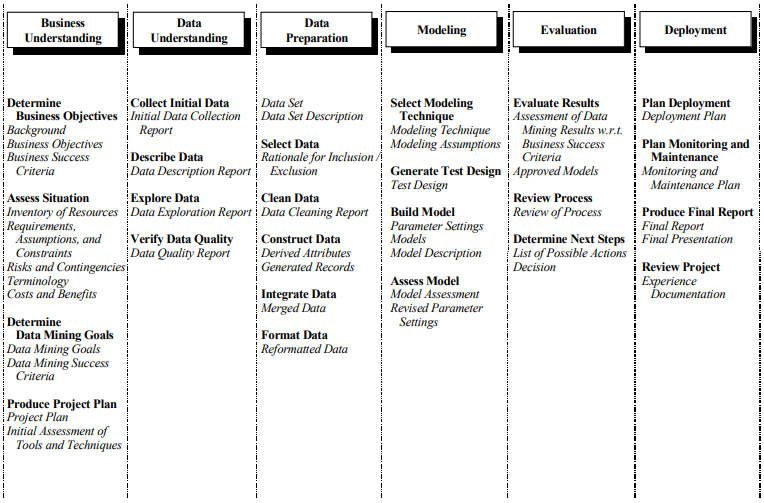
The impact of a sound methodology and efficient project management is high when it comes to the success of any data mining technology. The CRISP-DM (Cross Industry Standard Process for Data Mining) methodology tackles this problem by building a process model that renders a plan for implementing data mining scenarios that are autonomous of both the technology used and the sector of the industry. CRISP-DM model makes an ambitious attempt to make all sized data mining scenarios, cost-effective, more scalable, more extensive, more rapid and, more manageable. An experimental study by Wirth and Hipp (2000) describes a need for a standard methodology for the data mining industry. The researchers studied the market for a system that can benefit a firm’s customers, vendors, and analysts, most effectively. And thus, proposed experimental design. In the research an experiment was designed, the stages of this design were as shown in Figure 3.2.1:



**Figure 3.2.1: Stages of experimental design for a methodology**

Source: (Wirth and Hipp, 2000)

The CRISP-DM model favorable for outlining, the transmission of information, and representation described some very similar flows and was henceforth qualified as a standard industrial approach in data mining industries (Wirth and Hipp, 2000). The CRISP-DM tasks and their outcomes can be visualized as in Table 3.2.1:



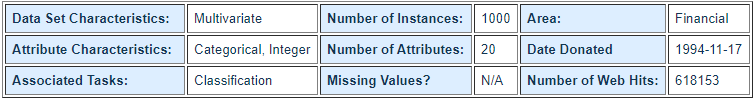
**Table 3.2.1: CRISP-DM tasks and their structure**

Source: (Wirth and Hipp, 2000)

In this dissertation, the beginning stage was to establish an understanding of credit risk background, key data characteristics, and relevant financial industry-wide significant factors worth considering. The aim and objective of this research were thus defined. Furthermore, the next stages in this methodology would be understanding of the data in four substages and then the data preparation into two stages. The approach in building this model includes a description of the modeling techniques utilized, factors of experimental design, the stages in the model build and the final step of model evaluation, retrospection of any improvements, and the presentation of key outcomes.

## **3.3 German Credit Risk Dataset**

German credit risk dataset in Hofmann (1994) views real data of credit records from a database of potential loan borrowers from credit unions in Germany. This dataset group records elaborated by a set of attributes as good or bad credit risks. The description of the dataset can be unraveled as shown in Table 3.3.1:



**Table 3.3.1: Description for German credit risk dataset**

Source: (Hofmann, 1994)

The original dataset was donated by Prof. Hofmann. Some attributes needed to be converted into numerical types. Henceforth, Strathclyde university edited and added various features hinting to make it feasible for machine learning methods that are not designed to handle categorical features. Many of the attributes that were ordered categorical has been converted to integers. Adhering to the rules for a recommended format by StatLog4. The attributes are presented in Table 3.3.2:

|  |  |  |
| --- | --- | --- |
| **Index** | **Variable** | **Nature of Data** |
| 1 | Status of existing checking account | Qualitative |
| 2 | Duration in months | Numerical |
| 3 | Credit history | Qualitative |
| 4 | Purpose | Qualitative |
| 5 | Credit amount | Numerical |
| 6 | Savings account/bonds | Qualitative |
| 7 | Present employment since | Qualitative |
| 8 | Instalment rate in % of disposable income | Numerical |
| 9 | Personal status and sex | Qualitative |
| 10 | Other debtors/guarantors | Qualitative |
| 11 | Present residence since | Numerical |
| 12 | Property | Qualitative |
| 13 | Age in years | Numerical |
| 14 | Other instalment plans | Qualitative |
| 15 | Housing | Qualitative |
| 16 | Number of existing credits at this credit union | Numerical |
| 17 | Job | Qualitative |
| 18 | Number of people being liable to provide maintenance for | Numerical |
| 19 | Telephone | Qualitative |
| 20 | Foreign worker | Qualitative |

**Table 3.3.2: Variable and nature of data view for German credit risk dataset**

Source: own processing based on (Hofmann, 1994)

4 https://statlogeconometrics.com/en/

|  |  |
| --- | --- |
| **Credit history** | |
| A30 | no credits taken / all credits paid back duly |
| A31 | all credits at this credit union paid back duly |
| A32 | existing credits paid back duly till now |
| A33 | delay in paying off in the past |
| A34 | critical account / other credits existing (not at this credit union) |

|  |  |
| --- | --- |
| **Personal status and sex** | |
| A91 | male: divorced/separated |
| A92 | female: divorced/separated/married |
| A93 | male: single |
| A94 | male: married/widowed |
| A95 | female: single |

|  |  |
| --- | --- |
| **Present employment since** | |
| A71 | Unemployed |
| A72 | < 1 year |
| A73 | 1 <= ... < 4 year |
| A74 | 4 <= ... < 7 year |
| A75 | >= 7 year |

All the qualitative attributes in the dataset are represented symbolically because of the description being too large for a record. The qualitative attribute information is as follows in Tables 3.3.3:

|  |  |
| --- | --- |
| **Status of existing checking account** | |
| A11 | < 0 DM |
| A12 | 0 <= ... < 200 DM |
| A13 | >= 200 DM |
| A14 | No account |

|  |  |
| --- | --- |
| **Savings account/bond** | |
| A61 | < 100 DM |
| A62 | 100 <= ... < 500 DM |
| A63 | 500 <= ... < 1000 DM |
| A64 | >= 1000 DM |
| A65 | unknown / no savings account |

|  |  |
| --- | --- |
| **Purpose** | |
| A40 | car (new) |
| A41 | car (used) |
| A42 | furniture/equipment |
| A43 | radio/television |
| A44 | domestic appliances |
| A45 | repairs |
| A46 | education |
| A47 | vacation |
| A48 | retraining |
| A49 | business |
| A410 | others |

|  |  |
| --- | --- |
| **Property** | |
| A121 | real estate |
| A122 | if not A121: building society savings agreement / life insurance |
| A123 | if not A121/A122: car or other |
| A124 | unknown / no property |

|  |  |
| --- | --- |
| **Other instalment plans** | |
| A141 | bank |
| A142 | stores |
| A143 | none |

|  |  |
| --- | --- |
| **Job** | |
| A171 | unemployed / unskilled - non-resident |
| A172 | unskilled – resident |
| A173 | skilled employee / official |
| A174 | management / self-employed /  highly qualified employee / officer |

|  |  |
| --- | --- |
| **Foreign worker** | |
| A201 | Yes |
| A202 | No |

|  |  |
| --- | --- |
| **Housing** | |
| A151 | rent |
| A152 | own |
| A153 | for free |

|  |  |
| --- | --- |
| **Telephone** | |
| A191 | none |
| A192 | yes, registered under the customer’s name |

|  |  |
| --- | --- |
| **Other debtors / guarantor** | |
| A101 | none |
| A102 | co-applicant |
| A103 | guarantor |

**Table 3.3.3: Tables describing all the qualitative attributes with their symbols**

Source: own processing based on (Hofmann, 1994)

## **3.4 Tools Used**

### **3.4.1 Programming Platform**

The preferred programming language to carry out all the analytics would be Python 3+ due to its feasibility in carrying out any data analytics task. It also comes with a diverse choice of machine learning libraries and frameworks. The nominated platform or python distributor would be Anaconda, which is an open-source and costless distribution of Python for scientific computing, machine learning, predictive analytics, etc. Anaconda distribution has over 250 packages pre-installed and over 7500 additional open-source installations from Conda virtual environment manager5. This distribution is heavily preferred in financial services industries that use open-source data science and machine learning tools to detect fraud, improve credit scoring, evaluate loan applications, and more. It is effective in implementing credit-scoring systems using datasets6.



**Figure 3.4.1: Frameworks and libraries preinstalled with Anaconda distribution7**

### **3.4.2 Data Analytics Software**

Anaconda distribution foremost framework is a web-based interactive environment called Jupyter Notebook. A Jupyter Notebook document is a JSON document consisting of an ordered list of input/output record cells on a predefined schema. A python notebook is a localhost-based desktop python application alternative. A Jupyter Notebook can be converted to various standard output formats i.e. HTML, LaTeX, PDF, Markdown, PPT, Python, and Restructured Text8.

5 https://docs.anaconda.com/

6 https://www.anaconda.com/industries

The following are some salient features of the Jupyter environment:

* Preinstalled Numpy, Matlplotlib, Pandas, SciPy and all the other important python libraries, and frameworks.
* Easy to use live code notebook interface.
* A web application that runs on localhost, no internet connection required.
* Embedded sharing functionalities via email, Dropbox, GitHub, and NbViewer.
* Diverse options to produce outputs: HTML, images, videos, LaTeX, etc.
* Comes with integration for big data tools such as Apache Spark, Scikit-learn, TensorFlow, etc.
* Autosave feature
* Markdown reporting
* Save notebook with widget state information for static rendering, i.e. JSON format.

## **3.5 Data Exploration**

This task aims at gaining a better understanding of data by summarizing the descriptive statistics of each variable and then plotting visualizations elaborating their distribution and nature. Exploration also aims at handling outliers and understanding data capabilities for machine learning techniques. Furthermore, graphical visualizations can render information about patterns and interesting relationships among features which leads to an advanced understanding of deeper aspects of data. Python provides several libraries that render statistical summarizing methods and several types of spectacular visualizations to enhance better interpretation of data to an observer.

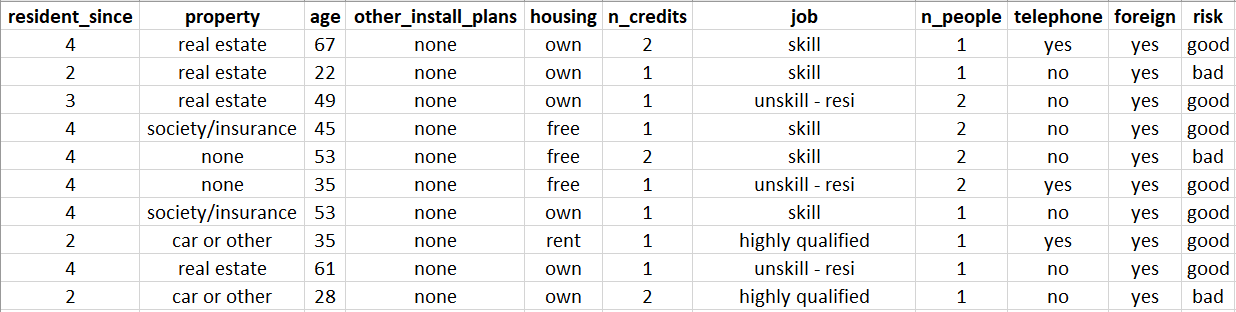
7 https://www.anaconda.com/products/individual

8 https://jupyter.org/about

## **3.6 Data Preparation**

As described in Section 3.3, all the categorical variables in the dataset are coded with symbols. These symbols have been converted to more meaningful short strings so that later when these variables are plotted on a graph, it is easier for a viewer to identify and distinguish each from the other. Besides that, there have been variable headers designated by numbers that have been converted to actual short strings as the header name for that column. Furthermore, the target variable, i.e. risk column had binary entries as ‘0’ and ‘1’ which has been converted to ‘good’ and ‘bad’ to avoid confusion. A snapshot of the first ten entries reflecting the above-mentioned manipulations on the dataset saved as a comma-separated value file in excel can be viewed as in Table3.6.1:





**Table 3.6.1: Snapshot of first ten instances of manipulated data**

Source: own processing

### **3.6.1 Feature Engineering**

#### **3.6.1.1 For Clustering**

To adjust the nature of data that would make it more feasible for clustering, it is essential to check any hint of skewness in the distribution of all the variables utilized for the clustering purpose. General skewness in a distribution can be avoided by logarithmic transformation. Furthermore, clustering techniques like K-means are center oriented algorithms which needs to be subjected to a fit transform to a centre and scaled accordingly around the center.

#### **3.6.1.2 For Classification**

One of the most essential technique for feature extraction is Principal Component Analysis (PCA). PCA combines the input variable in such a way that results in dropping of the least important variables yet retaining the essential parts of all the variables. Moreover, the extracted variables post-PCA are independent of each other which makes it easier for linear parts of a model in prediction tasks (Brems, 2017). However, PCA works only on numerical variables and thus the categorical variables need to be converted to dummy numerical variables for the PCA feature extraction task.

## **3.7 Data Partioning**

Before employing any machine learning technique on a dataset, it is indeed essential to partition the data into training and test data. The training partition is utilized to train the data by a machine learning algorithm and the efficiency in the prediction of that trained model can be tested on test data. Reusing the same data for both training and testing is a bad way since it is not known before how the approach will work on data it was not trained on.

A general partitioning method is using 75% of the data for training purposes and the remaining 25% of the data for testing purposes. However, the question arises that which part of the data would be the best fit for each purpose. The solution is k-fold cross-validation, which shuffles the training and test data in blocks and utilizes every possible combination to summarize the results at the end. For the general partitioning case, the cross-validation method divides the dataset into four blocks and utilizes every combination of three blocks (75%) for training and one block (25%) for testing purposes. This is also known as 4x cross-validation (Starmer, 2018). However, practically with the dataset that has been used for this research has 1000 instances, it is best to apply 10x cross-validation with every combination of nine blocks for training (90%) and one block for testing (10%). Although the partition is more distributed to the training data, the combinations make it more possible for the testing data to test it on a variety of data chunks which makes this technique more diverse and efficient for this study in particular.

Another important technique as described by Hsu *et al.* (2003) is Grid Search cross validation. It is a parameter sweep by exhaustive searching through a user fed subset of a learning algorithm as the hyperparameter space. As explained by Chicco (2017) user-fed hyperparameters eliminate the possibilities of learning in unbounded value spaces and hence are mandatory before utilizing Grid Search. The technique trains a classifier model and measures their performance by inherent cross-validation on the training data, trained per set. As a result, Grid Search concludes the calibrations that rendered the highest rank in the approach.

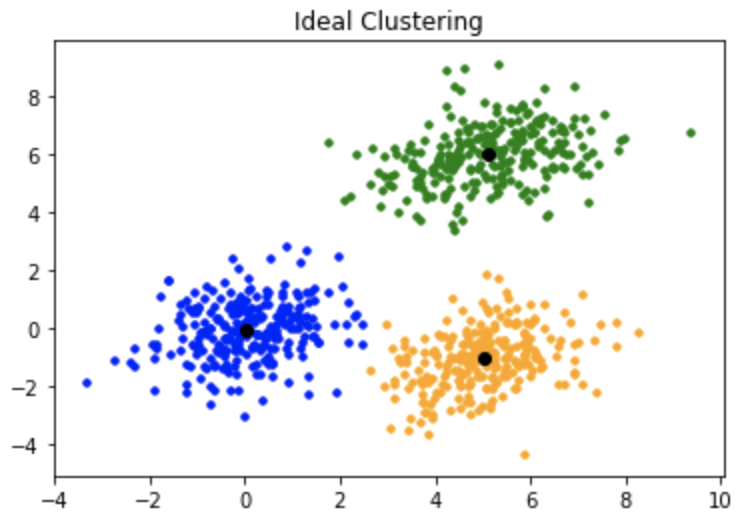
Both of these techniques can be utilized for different tasks. Grid Search technique can be utilized to train the model with the most suitable hyperparameters. Then the quality of the training set shuffle undertaken by combinations of all the data set permutations can be verified by 10x cross-validation.

## **3.8 Machine Learning Techniques Utilized**

### **3.8.1 K Means Clustering**

Clustering is the first stage of the hybrid model as described in Chapter 2. The research elaborated in Chapter 2 described K-means clustering as a highly efficient algorithm to train the model that ignores the risk variable in the German credit risk dataset. This stage is essential in determining the patterns and relations without relying on the prior mentioned response variables. That research is still relevant in our times as K-means clustering is still the most favored clustering technique and they prove that it is even highly suitable for this dataset in particular.

In K means, a target k addresses to the number of output centroids centering the clustered data records of the dataset. Every data point is distributed among these clusters that yield the minimum intra-cluster sum of squares and the maximum inter-cluster sum of squares. The centroids are determined by averaging the data. The algorithm initially assumes k random centroids from the data points and clusters all the other data points around these centroids. Further, new means are calculated within the groups, and centroids are redetermined. The iterations are repeated until the goal is reached.

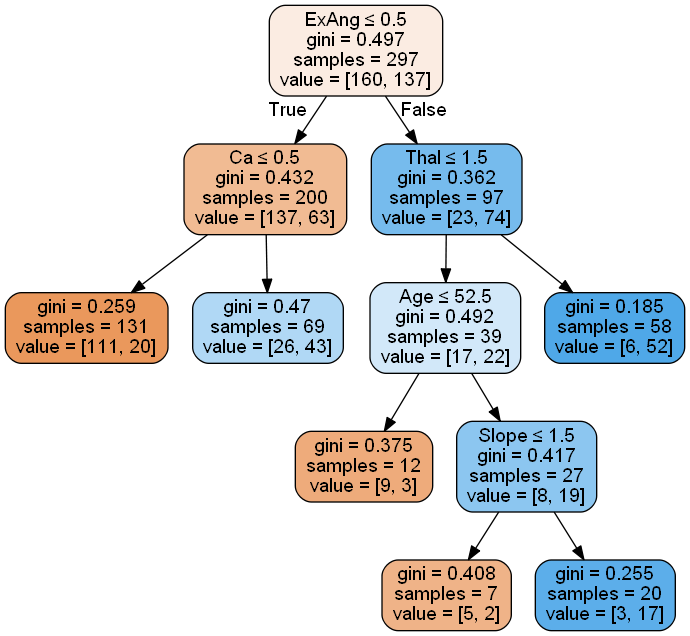


**Figure 3.8.1: K-means clustering example in python**

Source: (Khosla, n.d.)

### **3.8.2 Random Forest Classifier**

The second stage of the hybrid model is classification. In chapter 2, several kinds of research on classification techniques have been elaborated. However, these techniques are a bit outdated and this study aims at utilizing more recent techniques that are proven to render even better results. Among them Random Forest, a classic technique still stands due to its effective performance on credit risk datasets.

 A decision tree is a classification tree model where the response feature takes a discontinuous set of inputs. In this tree structure, branches denote a logical association of features that lead to labels as classes denoted by leaves. The aim of a decision tree is to attain a technique that predicts the value of an outcome target based on the provided input features (Wu *et al.*, 2008). A tree is constructed by splitting the root node into subsets of successor children relying on a group of rules determined by classification variables. This process is iterated recursively till the subgroup at a leaf has all the similar values as in the outcome target, or when splitting stops adding any value to the predictions (Shalev-Shwartz and Ben-David, 2014).

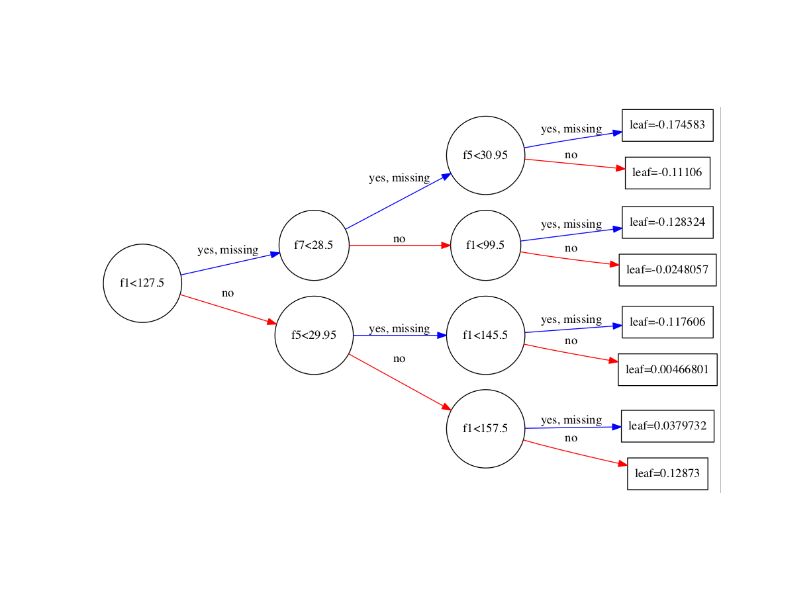
**Figure 3.8.2: Decision tree example in python9**

Random forests are an ensemble learning technique for classification that combines a plethora of decision trees built during training time rendering the mode of the classes as the resulting class. Random forests reduce the general decision tree trait of overfitting the training data. Particularly, random forests average many deep decision trees trained on different sets of training data by introducing some bias which reduces the high variance of individual trees. Though it comes with small bias and some loss of interpretability, it boosts the performance of the overall model (Hastie *et al.*, 2017).

9 https://jss367.github.io/Exploring-Decision-Trees-in-Python.html

### **3.8.3 Extreme Gradient Boost Classifier**

Boosting is an ensemble supervised learning technique for eliminating bias and then variance. It consists of a group of algorithms that modifies weak learners to a single strong learner, primarily based on the studies undertaken by Kearns and Valiant (1989). The boosting problem puts forward a hypothesis question that whether an optimal learning algorithm whose performance evaluated by a hypothesis stating it can be slightly better than random guessing (weak learner) implies the tendency of a competent technique whose hypothesis states its discretionary accuracy (strong learner). Boosting algorithms subsists of repetitively learning weak classifiers concerning a distribution and then combining them to a single strong classifier. Every time a weak learner is appended, the data weights are adjusted according to its learning capabilities. Correctly classified data points lose weight and misclassified ones get higher weights.



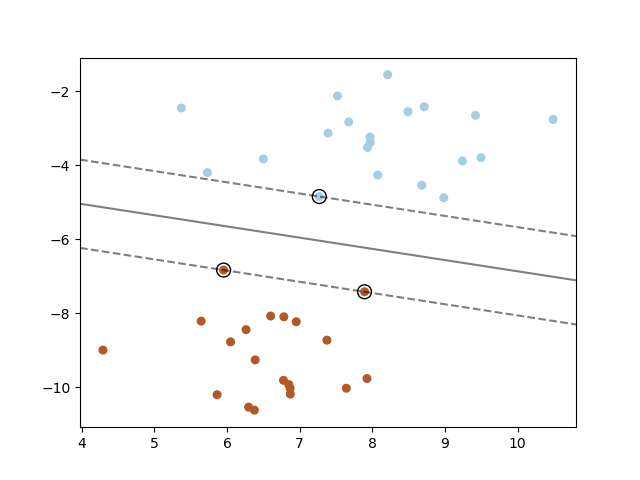
**Figure 3.8.3: Gradient boosting decision trees example in python**

Source: (Brownlee, 2016)

Gradient boosting is one such technique where weak learners are a set of decision trees. The algorithm is based on a gradient descent approach where an iterative functional descent optimizes a cost function by choosing a negative gradient direction (Friedman, 2001). Extreme gradient boosting (XGBoost) has been created to thrust the extreme of the computation limits resulting in a scalable, accurate, and portable method. XGBoost renders a regularized model to curb over-fitting and hence giving better performance.

### **3.8.4 Support Vector Classifier**

A support vector machine (SVM) is highly preferable to meet high accuracy with less computation power. A support vector classifier (SVC) aims at finding a hyperplane in an N-dimensional space representing ‘n’ input variables that specifically classifies the data points. The aim is to obtain a plane with maximum distance among data points of both classes. This grants reinforcement for the future data points classification. Hyperplanes are decision boundaries, each separating one class with the other (Gandhi, 2018).

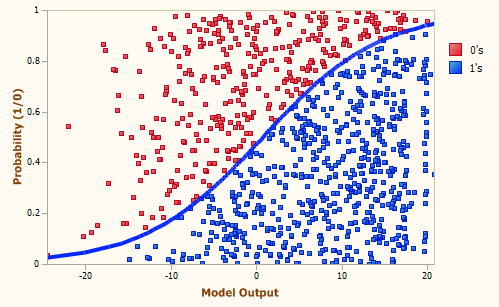


**Figure 3.8.4: Support vector classifier example in python10**

### **3.8.5 Logistic Regression Classifier**

Linear regression is a linear technique utilized to model a linear relationship among a dependent variable with one or more independent variables. A regression analysis aims to detect to what extent a scalar response can explain or influence the target variable. Here, the relationships are modeled by estimating the parameters from the dataset and feeding it into linear predictor functions. These models are termed linear models (Seal, 1967). Logistic regression is a statistical model that utilizes logarithmic functions instead of linear functions as predictors to model a binary dependent variable. These predictor functions convert log-odds to a probability that can vary from 0 to 1 and hence can distinguish the labels for the dependent variable or the target (Cramer, 2005).

The logistic regression model on its own model’s probability of a binary outcome in terms of an input variable and does not execute statistical classification. However, it can be implemented as a classifier design by introducing cut-offs or thresholds. These types of modeling can lead to classifying inputs with higher probability than cut-off as one class and with lower the cut-off as the other class.



**Figure 3.8.5: Logistic regression classifier example in python**

Source: (Nandu, 2019)

All the other classifiers described in Section 3.8 are heavily variance focused and eliminates bias. Whereas, the logistic regression is bias focused and eliminates higher variance. For the credit risk dataset, the outcome variable is binary and hence quite feasible for logistic classification. After all, classification by a regression method is essential to analyze this study from a contrasting angle and cover the spectrum with differing views.

10 https://scikit-learn.org/sTable/modules/svm.html

## **3.9 Evaluation Metrics**

### **3.9.1 Evaluation on Clustering**

As there is a single clustering method to be implemented in this study, i.e. K-means, there’s no need to evaluate clustering tendency. For the number of optimal clusters k, the elbow method is best for K-means as it is founded on the sum of squared distance between data points and their assigned clusters’ centroids. Furthermore, silhouette analysis works best under K-means as it can determine the extent of separation among clusters for each ‘k’ rendering information about contrastable qualities of outcome clusters (Dabbura, 2018).

### **3.9.2 Evaluation on Classification**

In any credit risk scenario, it is not recommendable to tag a risk as good when it is bad, then it is to tag a risk as bad when it is good. Henceforth, the efforts stand in decreasing the False Positive (FP) rates much more than the False Negative (FN) rates. Thus, the x-axis in the ROC curve needs to be minimized and the y-axis maximized. Furthermore, precision has to be prioritized for maximization rather than recall. However, the improved precision should not affect a huge fall on recall. Hence, a careful examination of precision-recall trade-off is essential for the complete evaluation.

## **3.10 Conclusion**

This chapter in a nutshell determined the methods to carry on this research further to analysis. Initially, a proposal for adhering to the standard CRISP-DM methodology was described. The dataset occupied with many symbolical values was converted to suit the visualizations, enhancing its comprehensibility. Along with elaborating the machine learning techniques and their subsequent evaluation to be utilized for this research, a focus was also given on feature selection to make these tasks more efficient and fluid. Moreover, the method for partitioning the data most suitable for machine learning techniques has also been discussed.

Summarizing the previous research from literature, this chapter finds the best means suitable for the credit risk case study in particular and nominated the tools that would work most efficiently for Chapter 4.

# **Chapter 4 – Findings and Analysis**

## **4.1 Introduction**

This chapter gathers the studies and plans from all the previous chapters to implement as a hybrid machine learning model. The models determined in Chapter 3 would be implemented and optimized for each stage in this chapter. The chapter begins with dataset exploration to give a visual depiction of all the data variables. These variables are subjected to refinement by feature extraction techniques such as log transform or PCA on both stages of the final hybrid model. These hybrid models are then analyzed by plots and evaluation metrics. Moreover, the priorities of certain evaluation parameters over others are highlighted and its relevancy is verified by deeper analysis.

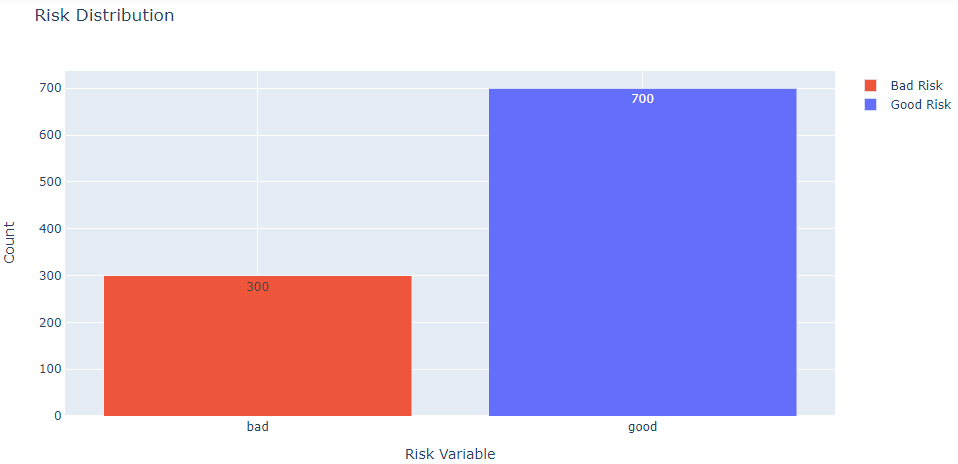
## **4.2 Data Exploration**

German credit risk dataset by Hofmann (1994) has 21 variables and 1000 instances. The target variable is risk. In this section, the nature of each variable is explored.

### **4.2.1 Dataset Variables**

#### **4.2.1.1 Risk (Target)**

Risk is a binary variable with values ‘good’ denoting the customer records where credit risk is low and ‘bad’ denoting the high credit risk. The distribution of this variable can be visualized as in Figure 4.2.1:



**Figure 4.2.1: Risk variable distribution**

Noting from Figure 4.2.1, 70% of the target variable denotes good risk and 30% denotes bad risk.

#### **4.2.1.2 Checking Account**

Checking account is a categorical variable that describes the range of checking amount in Deutsche Mark (DM). A checking account grants easier access to funds, generally used to pay one’s bills, and involves most of the financial transactions. The distribution of this variable can be visualized as in Figure 4.2.2:

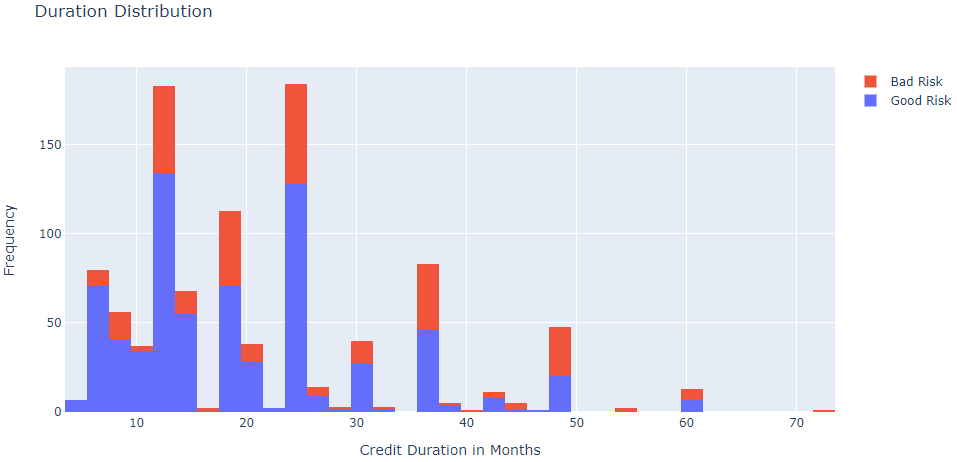


**Figure 4.2.2: Checking account variable distribution**

Noting from Figure 4.2.2, a major set of records don’t have a checking account and a minor set of records have more than 200 DM in their checking account. Moreover, the bad risk is generally present in the accounts having negative or less than 200 DM.

#### **4.2.1.3 Duration**

Duration is a numerical variable describing the credit duration for the loan in months. The distribution of this variable can be visualized as in Figure 4.2.3:

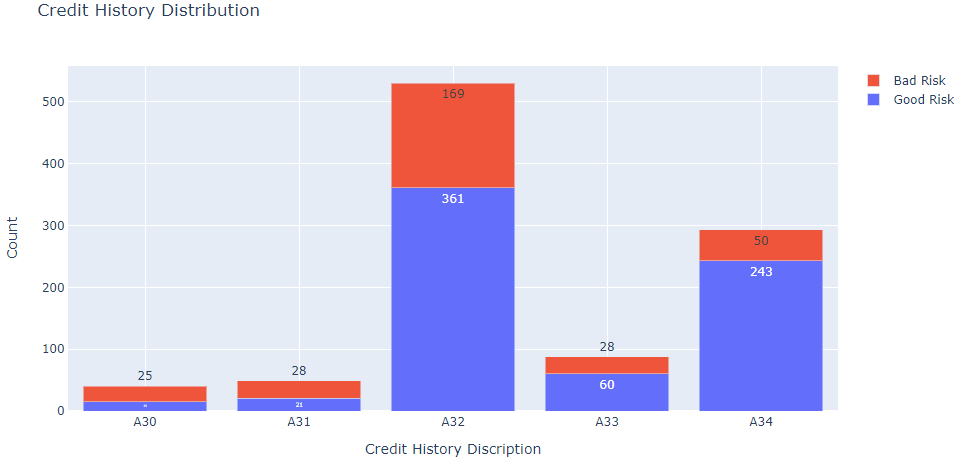


**Figure 4.2.3: Duration variable distribution**

Noting from Figure 4.2.3, major loans were sanctioned in the range of 15 to 25 months.

#### **4.2.1.4 Credit History**

Credit history is a categorical variable elaborating on the current details of a customer’s credit history. The distribution of this variable can be visualized as in Figure 4.2.4:

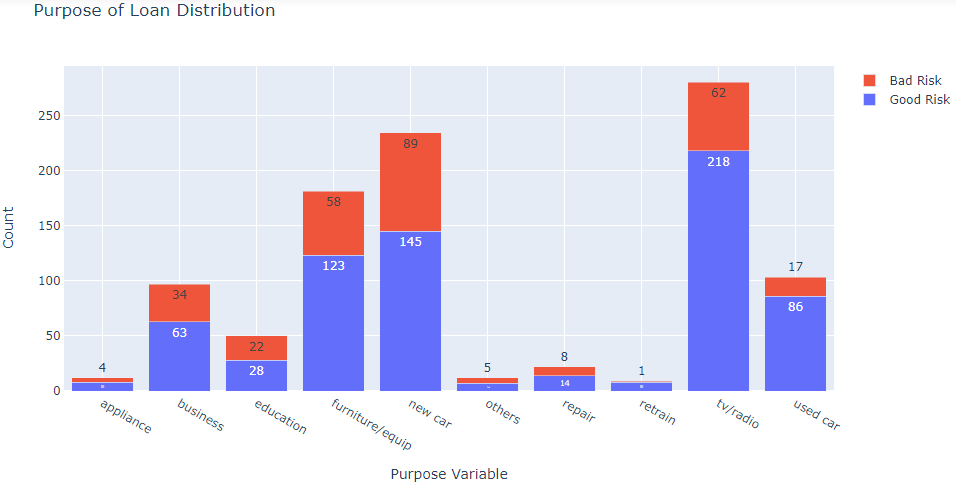


**Figure 4.2.4: Credit history details variable distribution**

Noting from Figure 4.2.4, most of the records have existing credits paid back duly till now (A32) or critical accounts / other credits existing (A34). The major risk is involved with existing credits paid back duly.

#### **4.2.1.5 Purpose**

Purpose is a categorical variable that denotes the purpose for which the funds were borrowed. The distribution of this variable can be visualized as in Figure 4.2.5:

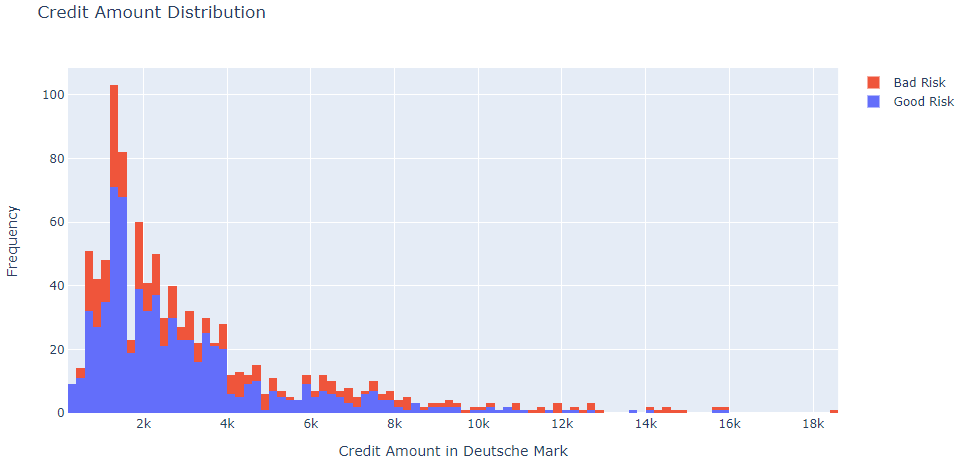


**Figure 4.2.5: Purpose variable distribution**

Noting from Figure 4.2.5, most of the loans were sanctioned for buying television/radio, new car, and furniture. And there’s a high degree of risk involved in these purposes as well.

#### **4.2.1.6 Credit Amount**

Credit amount is a numerical variable describing the total amount of an individual’s credit account in DM. The distribution of this variable can be visualized as in Figure 4.2.6:

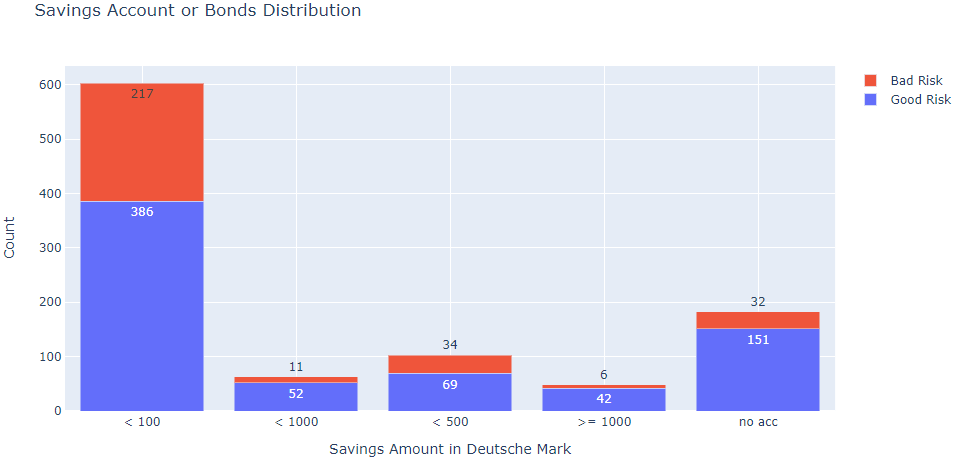


**Figure 4.2.6: Credit amount variable distribution**

Noting from Figure 4.2.6, though distribution in credit amount is vast, most of the credit amount lies in the range of 500 to 4000 DM and the risk lies in this range as well.

#### **4.2.1.7 Savings Amount/Bonds**

Savings amount/bonds is a categorical variable that denotes the range of amount in a holder’s savings account in DM. The distribution of this variable can be visualized as in Figure 4.2.7:

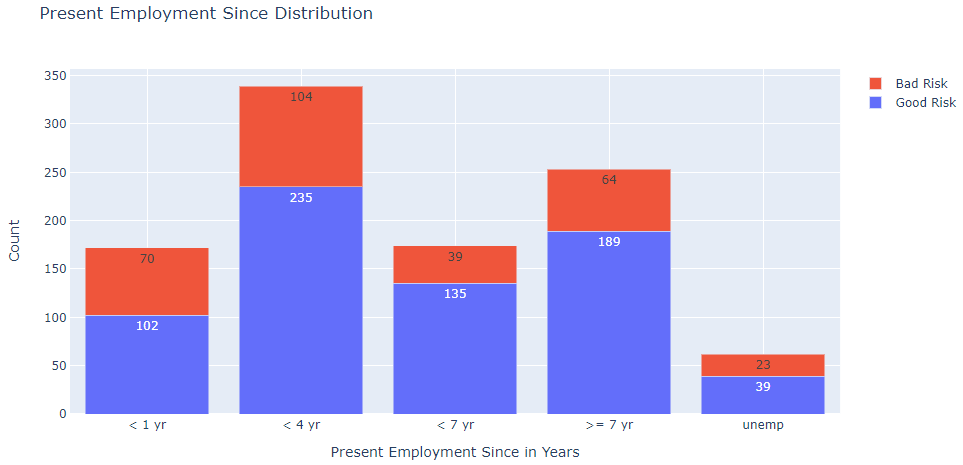


**Figure 4.2.7: Savings amount / bonds variable distribution**

Noting from Figure 4.2.7, more than 50% of the entries have < 100 DM in their savings accounts or bonds. This category involves high risk as well.

#### **4.2.1.8 Present Employment Since**

Present employment since is a categorical variable describing the employment status of a borrower in years. The distribution of this variable can be visualized as in Figure 4.2.8:

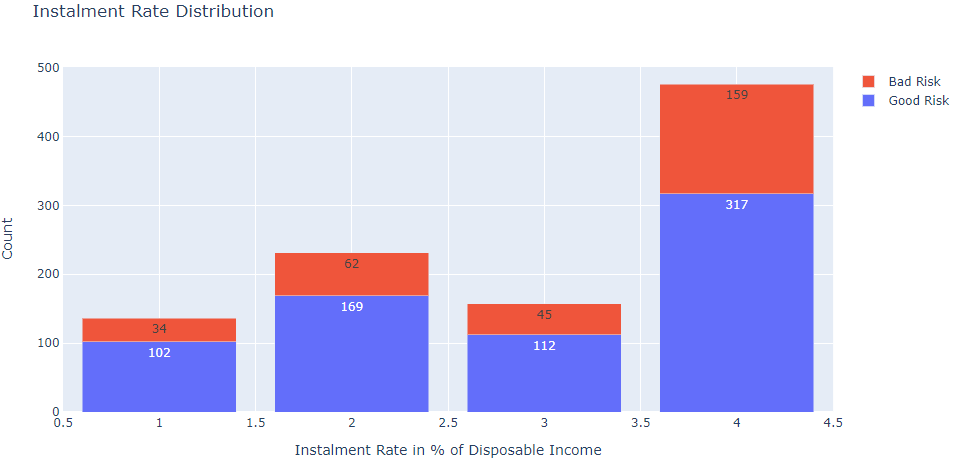


**Figure 4.2.8: Present employment since variable distribution**

Noting from Figure 4.2.8, most of the records were of individuals employed either between 1 to 4 past years or more than 7 years. However, high risk has been noted among the individuals employed for less than 4 years.

#### **4.2.1.9 Instalment Rate**

Instalment rate is a numerical variable describing the instalment rate in terms of percentage of disposable income. The distribution of this variable can be visualized as in Figure 4.2.9:

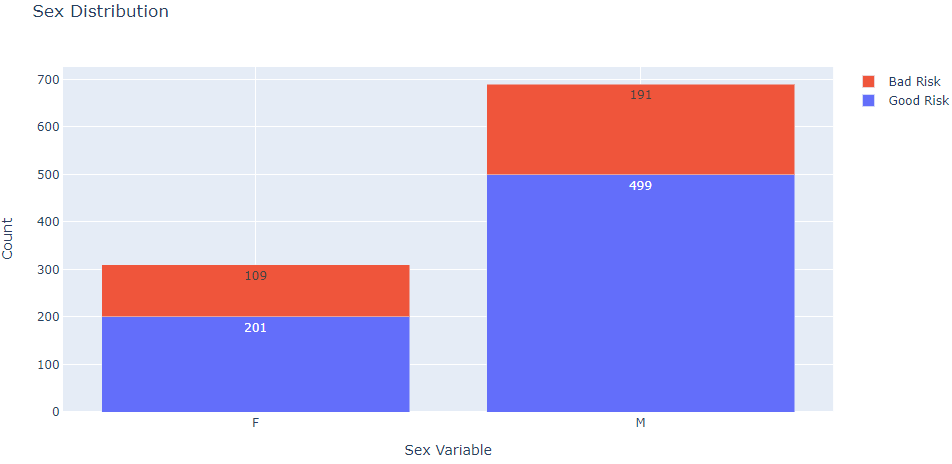


**Figure 4.2.9: Instalment rate variable distribution**

Noting from Figure 4.2.9, approximately 50% of entries have 4% instalment rate and this rate involves the highest risk.

#### **4.2.1.10 Sex**

Sex is a binary variable describing the sex of a borrower. The distribution of this variable can be visualized as in Figure 4.2.10:

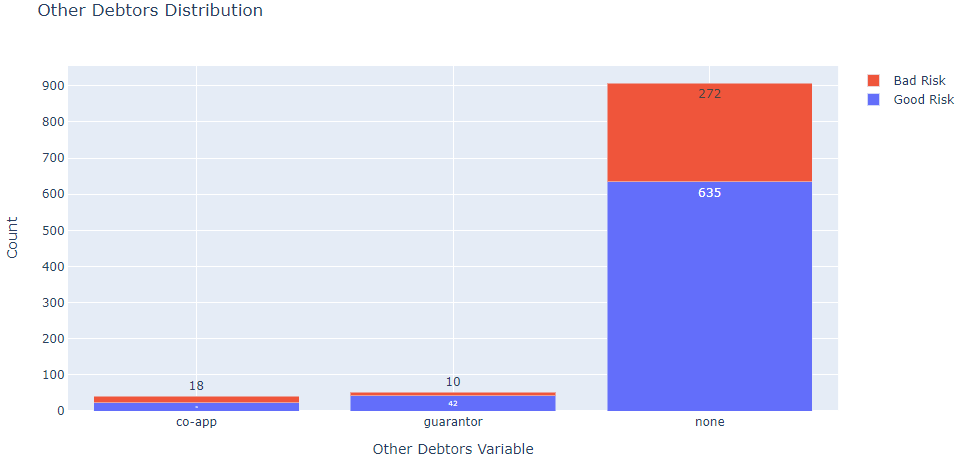


**Figure 4.2.10: Sex variable distribution**

Noting from Figure 4.2.10, around 70% of the entries are male and 30% female. Whilst, the ratio of risk is almost similar.

#### **4.2.1.11 Other Debtors**

Other debtor / guarantor is a categorical variable describing the existence of a co-applicator or a guarantor with an individual borrower. The distribution of this variable can be visualized as in Figure 4.2.11:

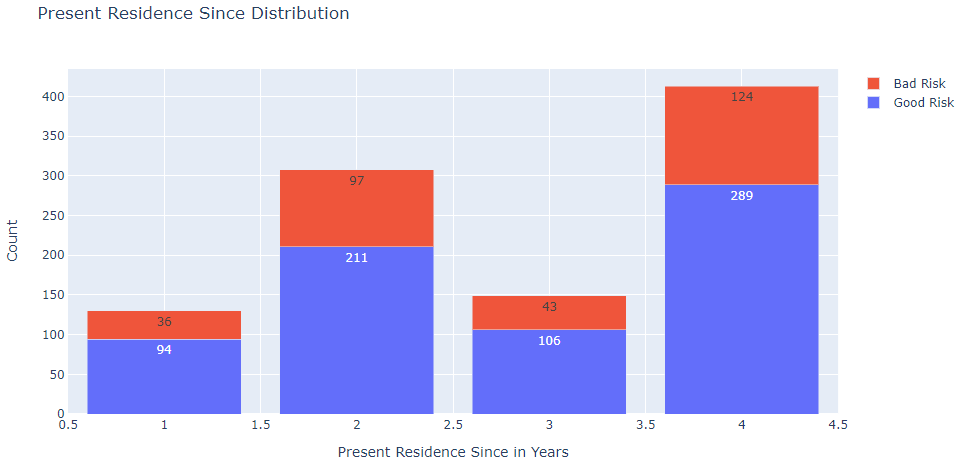


**Figure 4.2.11: Other debtors variable distribution**

Noting from Figure 4.2.11, around 90% of the entries don’t have any co-applicators or guarantors and they also involve the majority of high risk.

#### **4.2.1.12 Present Residence Since**

Present resident since is a numerical variable describing the residence states of an individual borrower in years. The distribution of this variable can be visualized as in Figure 4.2.12:



**Figure 4.2.12: Present residence since variable distribution**

Noting from the above graph, the majority of the individual entries have either 2 or 4 years of current residency and they involve the higher risk as well.

#### **4.2.1.13 Property**

Property is a categorical variable describing the type of property that an individual hold or have mortgaged for the loan. The distribution of this variable can be visualized as in Figure 4.2.13:

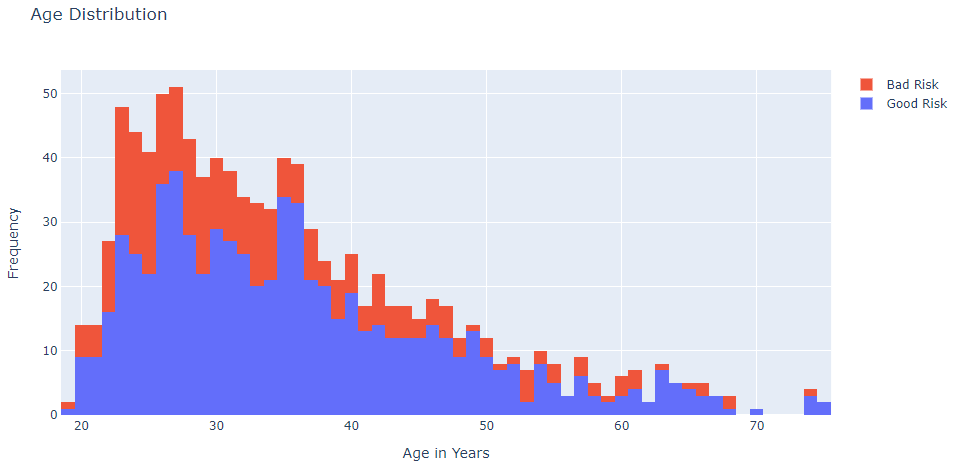


**Figure 4.2.13: Property variable distribution**

Noting from Figure 4.2.13, around 60% of the records holds car or other or real estate as properties. However, high risk has been detected in individuals holding car or other and society/insurance as properties. Moreover, the ratio of risk is high for individuals holding no properties in their loan conditions.

#### **4.2.1.14 Age**

Age is a numerical variable denoting the age of an individual borrower in years. The distribution of this variable can be visualized as in Figure 4.2.14:



**Figure 4.2.14: Age variable distribution**

Noting from Figure 4.2.14, the data is hugely centred around age range of 25 to 35 years and the higher bad risk has been detected in the same range.

#### **4.2.1.15 Other Instalment Plans**

Other instalment plans is a categorical variable describing the type of other instalment plans if an individual borrower holds. The distribution of this variable can be visualized as in Figure 4.2.15:

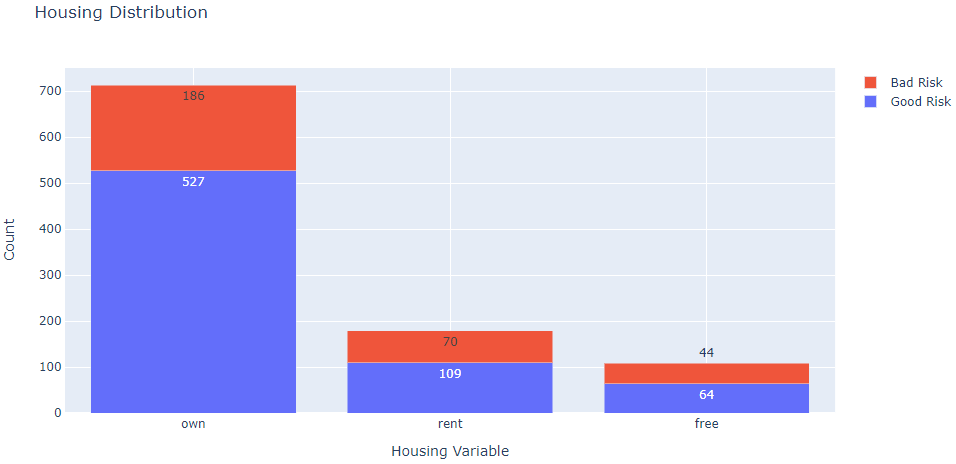


**Figure 4.2.15: Other instalment plans variable distribution**

Noting from Figure 4.2.15, approximately 80% of the records holds no other instalment plans and these records have higher bad risk.

#### **4.2.1.16 Housing**

Housing is a categorical variable describing whether an individual borrower owns or rents his/her house or holds any social housing benefits for free. The distribution of this variable can be visualized as in Figure 4.2.16:

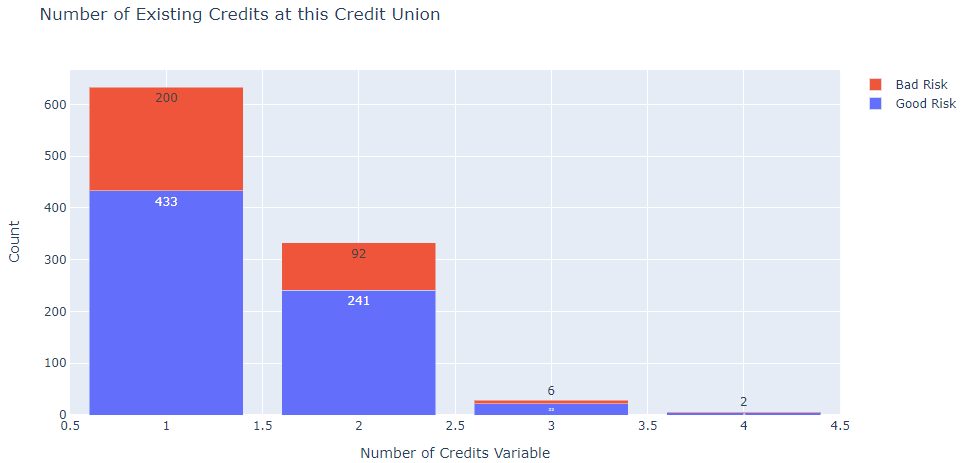


**Figure 4.2.16: Housing variable distribution**

Noting from the above graph, approximately 70% of the records owns their house and these records have higher bad risk.

#### **4.2.1.17 Number of Credits**

This variable describes the existing number of credits for an individual. The distribution of this variable can be visualized as in Figure 4.2.17:

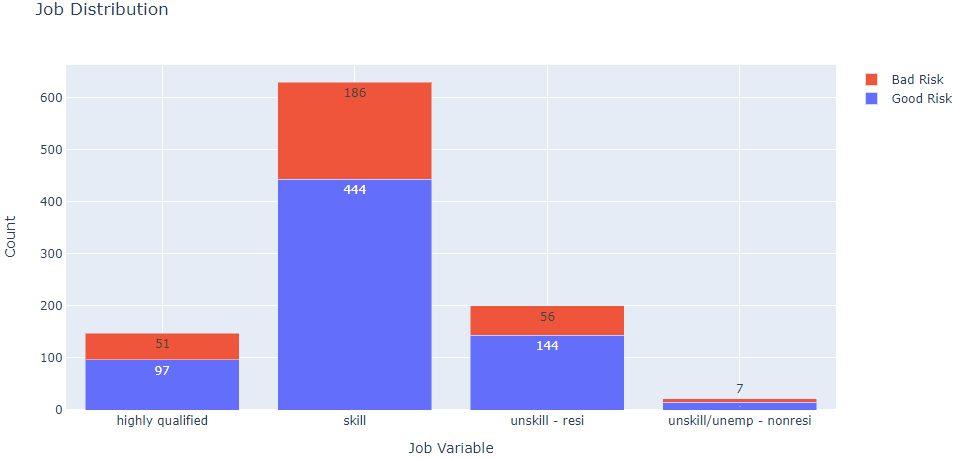


**Figure 4.2.17: Number of credits variable distribution**

Noting from Figure 4.2.17, more than 95% of the records holds 1 or 2 units of credits in their account and these records have most of the bad risks.

#### **4.2.1.18 Job**

This variable describes the category of job in which an individual is currently employed in. The distribution of this variable can be visualized as in Figure 4.2.18:

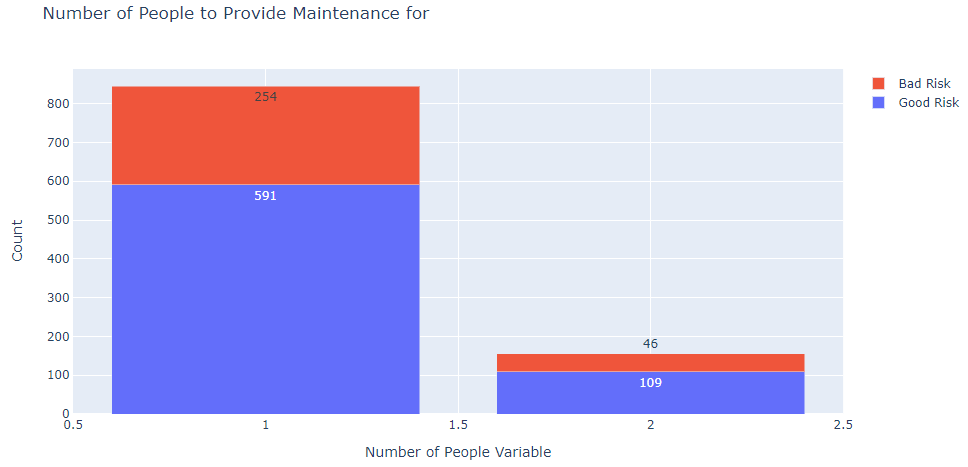


**Figure 4.2.18: Job variable distribution**

Noting from the above graph, approximately 65% of the records are skilled employees and these records have significant bad risks.

#### **4.2.1.19 Number of Liable People**

This variable represents the quantity of people liable for providing preservation for the loan including the borrower. The distribution of this variable can be visualized as in Figure 4.2.19:

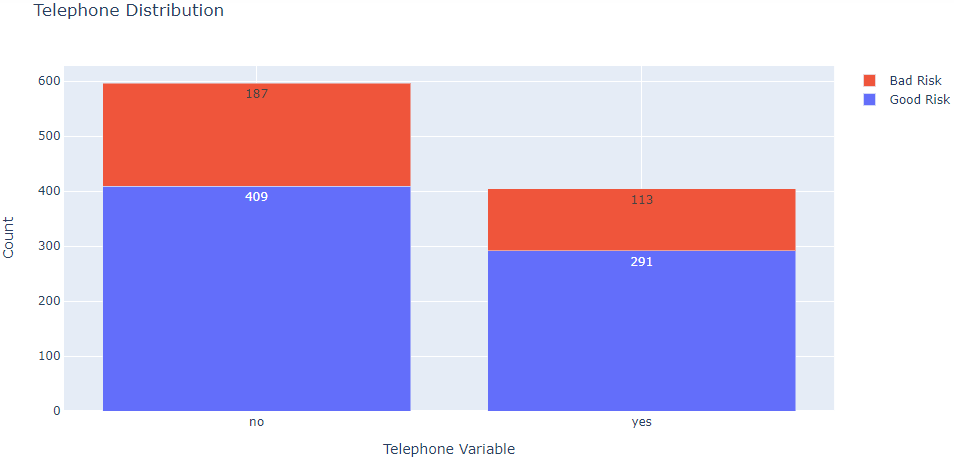


**Figure 4.2.19: Number of liable people variable distribution**

Noting from Figure 4.2.19, only a fraction of the records (15%) have other liable people to provide maintenance for. Most high risk is associated with entries who don’t have any liable people other than themselves.

#### **4.2.1.20 Telephone**

This variable describes whether an individual has a registered telephone number under his/her name. The distribution of this variable can be visualized as in Figure 4.2.20:

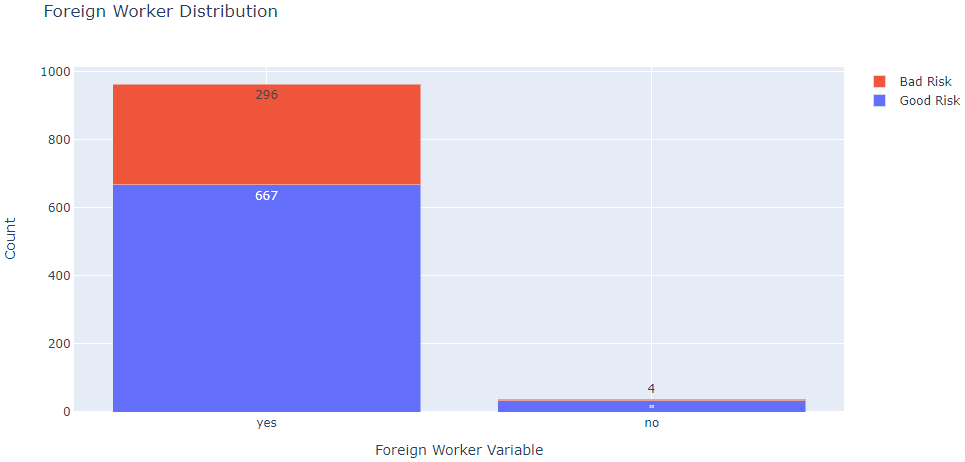


**Figure 4.2.20: Telephone variable distribution**

Noting from Figure 4.2.20, around 60% of the record holders have not registered any telephone number in their corresponding credit union. The ratio of risk involved is nearly similar for both values.

#### **4.2.1.21 Foreign Worker**

This variable describes whether an account holder is a foreign worker or not. The distribution of this variable can be visualized as in Figure 4.2.21:



**Figure 4.2.21: Foreign worker variable distribution**

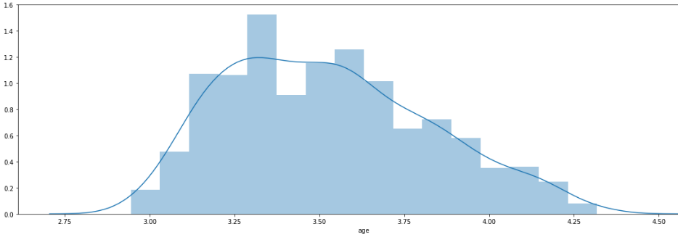
Noting from Figure 4.2.21, more than 95% of the records are foreign workers, and among them approximately 30% of the entries have bad risks.

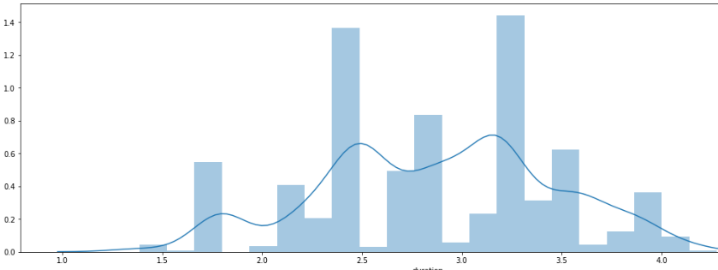
## **4.3 Clustering**

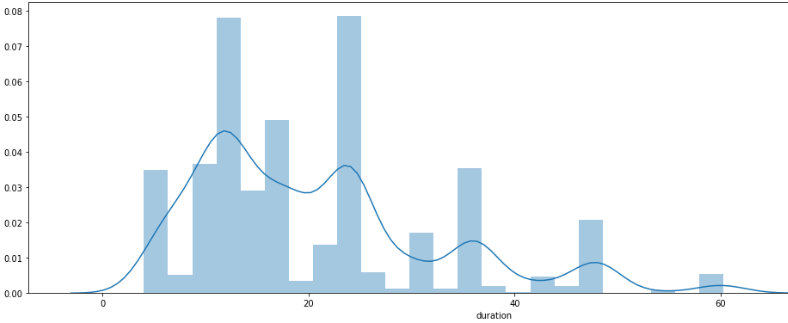
### **4.3.1 Feature Extraction**

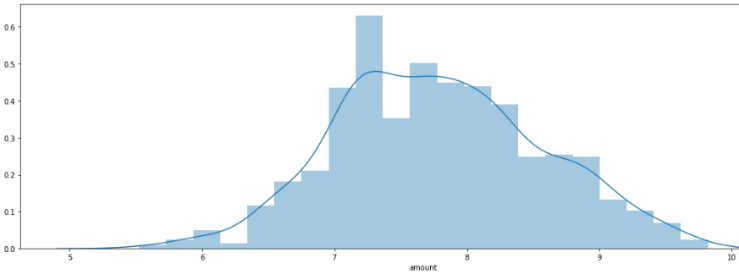
K-means algorithm works on continuous quantitative variables only (Zakrzewska, 2007). In the German credit risk dataset, there are three continuous variables as ‘age’, ‘duration’, and ‘amount’ which would be utilized to perform this task.

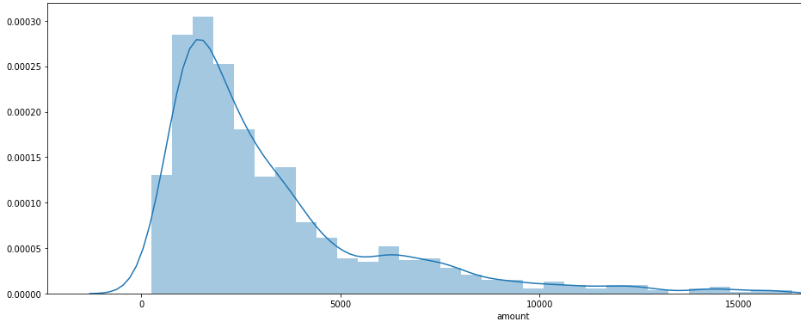
Noting the variable plots of three selected variables in the previous section, it can be noticed that the distributions are right-skewed. To refine the outcomes and reduce the noise from outliers, a logarithmic transformation is required. The distributions before and after transform can be viewed as in Figure 4.3.1, 4.3.2, and 4.3.3:



**Figure 4.3.1: (Left) Age distribution pre-transform, (Right) Age distribution post-transform**



**Figure 4.3.2: (Left) Duration distribution pre-transform, (Right) Duration distribution post-transform**



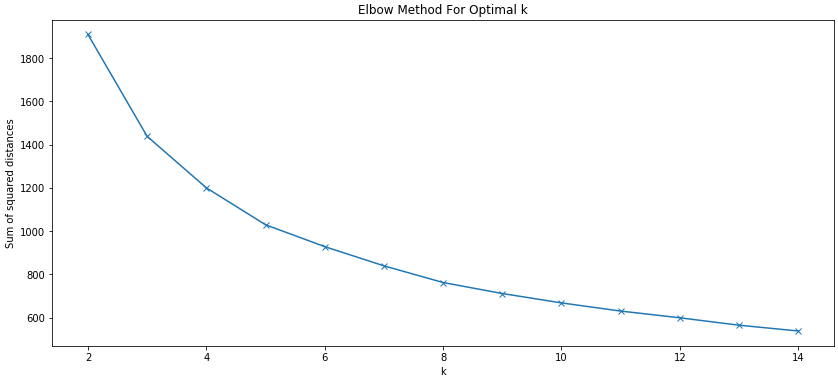
**Figure 4.3.3: (Left) Amount distribution pre-transform, (Right) Amount distribution post-transform**

Further, the distributions are centered to fit its corresponding mean and transformed to scale around the mean.

### **4.3.2 K Means**

#### **4.3.2.1 Number of Clusters**

The most feasible number of clusters ‘k’ can be determined by the variation of residual sum of squared distances with k as visualized in Figure 4.3.4:

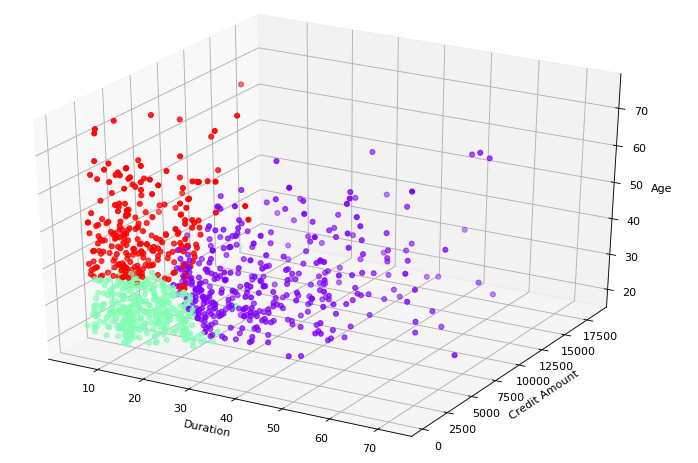


**Figure 4.3.4: Sum of squared distances vs k plot**

The best k value is where SSE in Figure 4.3.4 starts to flatten out the most. Here, that elbow spot is noticeable at k=3. Moreover, it is not quite beneficial to go beyond 14 clusters for efficient data segmentation.

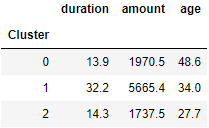
#### **4.3.2.2 Cluster Segmentation**

The segmented outcomes of k-means cluster analysis with three clusters plotted on a 3D axes are visualized as in Figure 4.3.5:



**Figure 4.3.5: K-means output clusters**

It is clearly noticeable that the data is segmented quite well. Furthermore, the centroids for every three clusters are as in Table 4.3.1:



**Table 4.3.1: Custer centroids Table**

From the above Table based on cluster rule, the segmented data can be interpreted as:

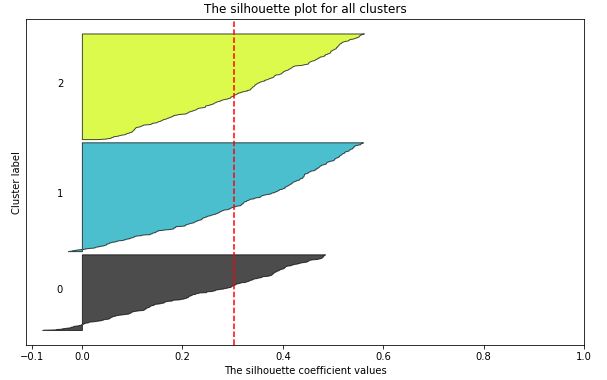
Cluster 0: Older customers, short durations, lower credit amounts.

Cluster 1: Middle-aged customers, long durations, higher credit amounts.

Cluster 2: Younger customers, short durations, lower credit amounts.

#### **4.3.2.3 Cluster Quality**

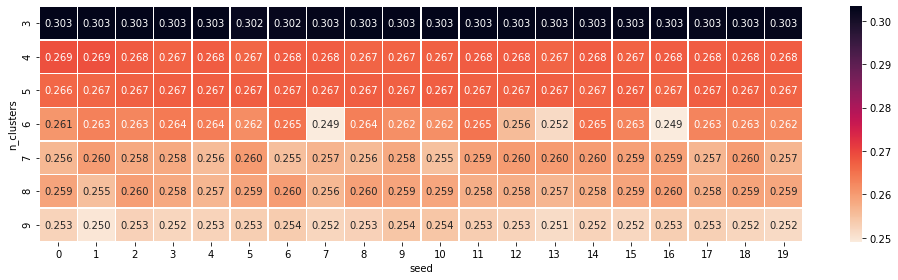
The silhouette analysis undertaken on three clusters can be visualized as in Figure 4.3.6:



**Figure 4.3.6: Silhouette analysis plot**

The plot in Figure 4.3.6 describes k=3 having the best average silhouette score and all three clusters being above the silhouette coefficient proves the clusters to be quite efficient. The thickness of each cluster silhouette describes the size of each cluster and here all three clusters seem quite large. Moreover, it describes that Cluster 1 and Cluster 2 have more data points than Cluster 0. There are some cluster points in Cluster 0 and a few in Cluster 1 having a negative silhouette coefficient describing a hint of misclustering for a small set of data points which reduces the average score a bit. Overall, the clusters are segmented quite well.

It is also important to determine the variation of average silhouette score within a range of iterations for three clusters. This can be determined by applying k-means on the data for a set of random seed values and verification. The heatmap describing clustered data for 20 random seed values as in Figure 4.3.7:



**Figure 4.3.7: Heatmap of clusters on random seeds**

The silhouette coefficient remaining constant as 0.303 with k=3 proves the results verified.

## **4.4 Classification**

### **4.4.1 Clustering Labels as Feature**

As the next step and final step of the hybrid model, in this stage, the results obtained from the previous stage needs to be utilized for enhancing the model. In Section 4.3, the data points were segmented into three clusters. The k-means model stores cluster labels on each individual data point. These labels can be extracted by indices and stored in a new variable which can be added to the dataset as a new feature column. The distribution of this new cluster variable is shown in Figure 4.4.1:



**Figure 4.4.1: Cluster variable distribution**

Noting from Figure 4.4.1, the good risk is distributed in all the clusters in nearly equal proportions. However, more hints of bad risks can be viewed in Cluster A and Cluster C. Note that Cluster A, B, and C are Cluster 0, 1, and 2 respectively as described in Section 4.3. With the addition of this new feature, now the credit risk dataset has 21 feature columns and 1 target column.

### **4.4.2 Feature Extraction**

The German credit risk dataset is a collection of points which can be described in a 21-dimensional space if the feature variables are considered as axes. A line that curtails the average squared distance from a point to itself can be defined as the best fitting line. Similarly, the next consecutive line can be selected from directions perpendicular to the previous best-fitting line. This approach can be repeated iteratively to yield an orthogonal basis. These bases can be expressed as vectors called principal components and this process is known as principal component analysis (PCA). PCA is a dimensionality reduction, noise filtering, feature selection, and feature engineering technique utilized to pre-process the classification algorithm when there are several features in the dataset (VanderPlas, 2016). German credit risk dataset has 21 features and thus it needs to be subjected to PCA. Note that PCA does not eliminate the features but rather eliminates the superfluous variance that can be generated by all the features combined.

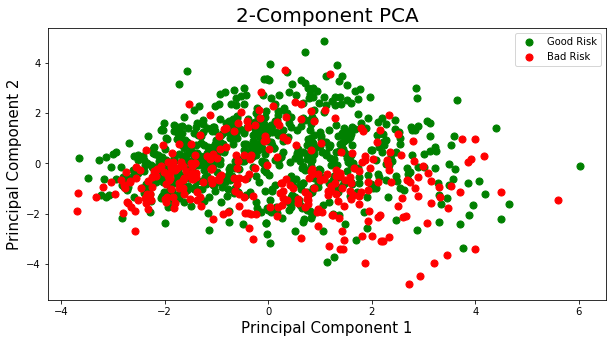
To perform a PCA on a dataset, all the categorical variable needs to be converted into duplicate numerical variables. When all the variables in the dataset are numerical, it is then also possible to perform a Pearson correlation analysis to get a general view of relationships among variables as visualized in Figure 4.4.2:



**Figure 4.4.2: Heatmap on Pearson correlation coefficients among all variables**

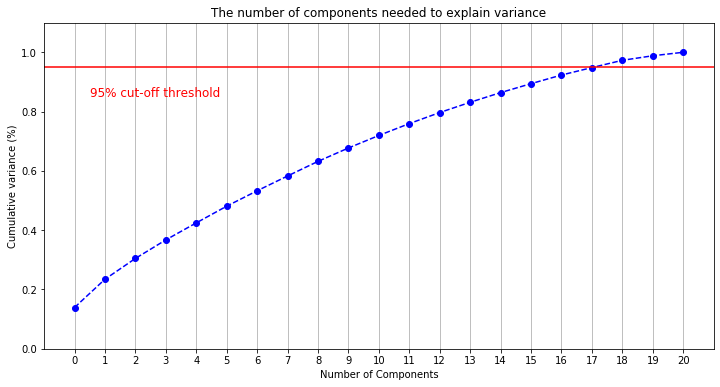
A general view of on what extent the target depends on its features. It can be viewed that the variable ‘Checking account’ has a coefficient of 0.35, which is fair. The other variables describe loose relationships; however, their combined contribution is substantial. ‘Cluster’ the new feature obtained from the k-means clustering task has a coefficient of 0.07 which is still significant. Note that, some variable describes negative correlation which defines their inverse proportionality with the target variable.

PCA identifies the relationship of features with its corresponding target i.e. good and bad risk. This relationship is quantified by finding a list of principle axes or orthogonal eigenvectors in mathematical terms. PCA can reduce the number of principal components to any number between 2 to the number of features but the reduction in components may also lead to a reduction in essential relationships. As a preliminary analysis and for the intent of visualization, the analysis can be fitted in two components to plot it in a 2D scatter plot as in Figure 4.4.3:



**Figure 4.4.3: 2-Component PCA scatter plot**

To find the optimum number of components with the minimum loss of essential relationships, it is essential to retain 95% of the total variance (VanderPlas, n.d.). This can be done by examining the variation of explained variance with the number of components as visualized in Figure 4.4.4:



**Figure 4.4.4: Cumulative explained variance ratio vs. number of components plot**

Noting from Figure 4.4.4, the features are fitted to a PCA which learns its relationship with the target and reduces it into a feature data-frame of 17 principal components.

### **4.4.3 Random Forest**

#### **4.4.3.1 Tuning Hyperparameters**

The values formed after training a model on a dataset and configured according to the data are called model parameters. Whereas model hyperparameters are values passed into a model before training a dataset. Generally, hyperparameters do not learn directly from data. In short, hyperparameters influence model parameters. For every model, there are several hyperparameters. So, the best set of hyperparameters can be obtained by subjecting to all different combinations and verifying from contrasting the results. Grid Search is a technique that explores hyperparameters and evaluates all combinations from an input list fed in its function and nominates the combination that renders the best accuracy to the model (Norena, 2018).

All the classification algorithms have several hyperparameters and within each hyperparameter there are a plethora of possible values. If the Grid Search cross-validation technique has to start from ground zero and explore all the possible permutations of values in each hyperparameter, it occupies a lot of time complexity. To avoid that, it is a good practice to feed a parameter tray consisting of some optimal values for the most important hyperparameters as an input to Grid Search. The prioritized hyperparameters for random forest classifiers can be found as stated in Scikit-learn Random Forest classifier documentation11. Pertaining to it the following parameters can be fed with their corresponding optimum values as stated below:

1. The total number of decision trees in the forest (n\_estimators).
2. Maximal level in each decision tree (max\_depth).
3. The number of random sampling and bootstrapping states (random\_state).
4. The number of jobs running in parallel (n\_jobs).

Furthermore, another additional and the most important input parameter for Grid Search is scoring. The scoring option determines which scoring metrics need to be optimized for the particular classifier model. As elaborated in Section 3.9.2, the preferred evaluation metric for the classifier model would be ROC over overall accuracy because the accuracy is based on one specific cut point, whereas ROC tries all cut points to explore a trade-off between sensitivity and specificity and visualizes the variation. Furthermore, the trade-off between precision and recall will be discussed in Section 4.5.

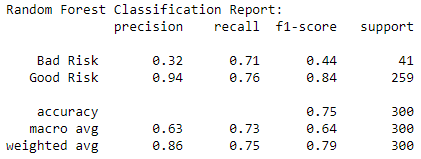
The starting points of the estimator, the parameter grid, and the scoring option as printed by Grid Search CV on the console when trained for the random forest classifier are as described in the appendix.

After finishing the cross-validation, Grid Search nominates the best estimator and hyperparameters respectively as described in the appendix.

11 https://scikit-learn.org/sTable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

#### **4.4.3.2 Classification Report**

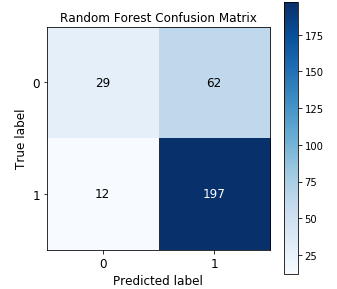
To summarize the results after training the Random Forest classifier by training data with the best estimator and hyperparameters, the following classification report generated by the model can be viewed as in Table 4.4.1:



**Table 4.4.1: Random Forest classification report**

From the report described in Table 4.4.1, it can be interpreted that the precision rate for the prediction of ‘Good Risk’ is quite high, whereas for ‘Bad Risk’ its low. The recall rate for both the risks is nearly similar and good. The f1-score reflects the differences in precision for each class. The overall accuracy of the classifier stands at 75%.

Furthermore, the confusion matrix can be analyzed to give a deeper insight into the classification performed by the model visualized as in Figure 4.4.5:



**Figure 4.4.5: Confusion matrix for Random Forest classifier model**

It can be noted from Figure 4.4.5 that the true positive rates (197) are much higher than false positive rates (12) whereas the true negative rates (29) are lower than false negative rates (62).

Finally, Random Forest models are prone to overfit the training set and, hence it is essential to verify the obtained accuracy by 10x cross-validation. This can be done by calculating the cross-validation score from the trained classifier model with the scoring option set to optimize the ROC curve. The obtained score is as displayed in Table 4.4.2:



**Table 4.4.2: Jupyter Notebook output for 10x cross validation score on Random Forest model**

### **4.4.4 Extreme Gradient Boosting**

#### **4.4.4.1 Tuning Hyperparameters**

The prioritized hyperparameters for XGBoost classifier tuning as stated in XGBoost documentation12 can be fed with a parameter tray as an input. These hyperparameters are as follows:

1. Maximum depth of a tree (max\_depth).
2. Total number of tree estimators (n\_estimator).
3. Learning rate of gradient descent (learning\_rate).

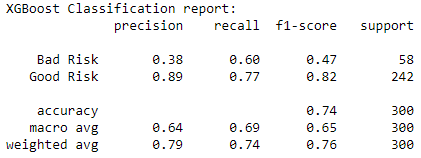
Besides these hyperparameters, the objective option for XGBoost has been set to ‘binary: logistic’ for a log(odds) analysis on a two-class classification problem. And the number of threads has been set to 4 as the algorithm has to be executed on a quad-core processor and hence it can utilize all four cores to work concurrently and minimize time complexity. And the scoring option for Grid Search cross-validation has been set for ROC optimization. Furthermore, the trade-off between precision and recall would be discussed in Section 4.5.

The starting points of the estimator, the parameter grid, and the scoring option as printed by Grid Search CV on the console when trained for the XGBoost classifier are as described in the appendix.

After finishing the cross validation, Grid Search nominates the best estimator and hyperparameters respectively as described in the appendix.

#### **4.4.4.2 Classification Report**

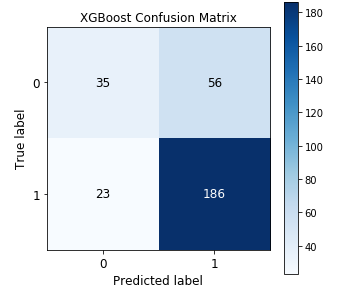
To summarize the results after training the Extreme Gradient Boosting classifier by training data with the best estimator and hyperparameters, the following classification report generated by the model can be viewed as in Table 4.4.3:



**Table 4.4.3: XGBoost classification report**

From the report described in Table 4.4.3, it can be interpreted that the precision rate for the prediction of ‘Good Risk’ is quite high, whereas for ‘Bad Risk’ its low. The recall rate shows an improvement in ‘Bad Risk’ classification and a fair ratio for classification on ‘Good Risk’. The f1-score reflects the differences in precision for each class. The overall accuracy of the classifier stands at 74%.

Furthermore, the confusion matrix can be analyzed to give a deeper insight into the classification performed by the model visualized as in Figure 4.4.6:



**Figure 4.4.6: Confusion matrix for XGBoost classifier model**

It can be noted from Figure 4.4.6 that the true positive rates (186) are much higher than false positive rates (23). Whereas the true negative rates (35) are lower than false negative rates (56).

Finally, XGBoost models are prone to overfit the training set, and hence it is essential to verify the obtained accuracy by 10x cross-validation. This can be done by calculating the cross-validation score from the trained classifier model with the scoring option set to optimize the ROC curve. The obtained score is as displayed in Table 4.4.4:



**Table 4.4.4: Jupyter Notebook output for 10x cross validation score on XGBoost model**

12 https://xgboost.readthedocs.io/en/latest/parameter.html

### **4.4.5 Support Vector**

#### **4.4.5.1 Tuning Hyperparameters**

The prioritized hyperparameters for Support Vector classifier tuning as stated in Scikit-learn SVC documentation13 can be fed with a parameter tray as an input. These hyperparameters are as follows:

1. Strength of the regularisation parameter (C).
2. Support Vector Machine kernel type as ‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’, or a callable (kernel).
3. Degree of polynomial kernel function (degree).

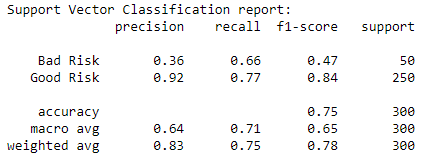
Besides these hyperparameters, the scoring option for Grid Search cross-validation has been set for ROC optimization. Furthermore, the trade-off between precision and recall would be discussed in Section 4.5.

The starting points of the estimator, the parameter grid, and the scoring option as printed by Grid Search CV on the console when trained for the random forest classifier are as described in the appendix.

After finishing the cross-validation, Grid Search nominates the best estimator and hyperparameters respectively as described in the appendix.

#### **4.4.5.2 Classification Report**

To summarize the results after training the Support Vector classifier by training data with the best estimator and hyperparameters, the following classification report generated by the model can be viewed as in Table 4.4.5:

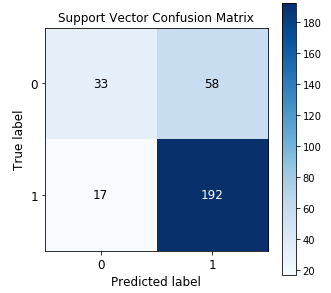


**Table 4.4.5: Support Vector classification report**

From the report described in Table 4.4.5, it can be interpreted that the precision rate for the prediction of ‘Good Risk’ is quite high, whereas for ‘Bad Risk’ its low. The recall rate shows an improvement in classification of ‘Bad Risk’ and a fair ratio for classification of ‘Good Risk’. The f1-score reflects the differences in precision for each class. The overall accuracy of the classifier stands at 75%.

13 https://scikit-learn.org/sTable/modules/generated/sklearn.svm.SVC.html

Furthermore, the confusion matrix can be analyzed to give a deeper insight into the classification performed by the model visualized as in Figure 4.4.7:



**Figure 4.4.7: Confusion matrix for Support Vector classifier model**

It can be noted from Figure 4.4.7 that the true positive rates (192) are much higher than false positive rates (17). Whereas the true negative rates (33) are lower than false negative rates (58).

Finally, support vector models are prone to overfit the training set, and hence it is essential to verify the obtained accuracy by 10x cross validation. This can be done by calculating the cross-validation score from the trained classifier model with scoring option set to optimize the ROC curve. The obtained score is as displayed in Table 4.4.6:



**Table 4.4.6: Jupyter Notebook output for 10x cross validation score on Support Vector model**

### **4.4.6 Logistic Regression**

#### **4.4.6.1 Tuning Hyperparameters**

The prioritized hyperparameters for Logistic Regression classifier tuning as stated in Scikit-learn logistic regression classifier documentation14 can be fed with a parameter tray as an input. These hyperparameters are as follows:

1. Penalisation norm specification type as ‘l1’, ‘l2’, ‘elastic net’, or none (penalty).
2. Regularization strength inverse (C).
3. Weights associated with target labels (class\_weight).
4. Optimization algorithm type as ‘newton-cg’, ‘lbfgs’, ‘liblinear’, ‘sag’, or ‘saga’ (solver).

Besides these hyperparameters, the scoring option for Grid Search cross-validation has been set for ROC optimization. Furthermore, the trade-off between precision and recall would be discussed in the Section 4.5

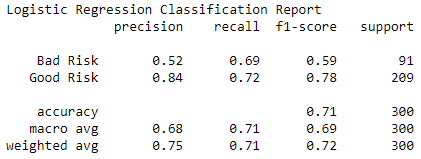
14 https://scikit-learn.org/sTable/modules/generated/sklearn.linear\_model.LogisticRegression.html

The starting points of the estimator, the parameter grid, and the scoring option as printed by Grid Search CV on the console when trained for the logistic regression classifier are as described in the appendix.

After finishing the cross-validation, Grid Search nominates the best estimator and hyperparameters respectively as described in the appendix.

#### **4.4.6.2 Classification Report**

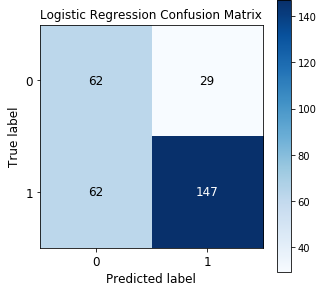
To summarize the results after training the logistic regression classifier by training data with the best estimator and hyperparameters, the following classification report generated by the model can be viewed as in Table 4.4.7:



**Table 4.4.7: Logistic Regression classification report**

From the report described in Table 4.4.7, it can be interpreted that the precision rate for the prediction of ‘Good Risk’ is quite high, whereas for ‘Bad Risk’ its low. The recall rate shows an improvement in the classification of ‘Bad Risk’ and is almost equal to the ‘Good Risk’ classification recall rate. The f1-score reflects the differences in precision for each class. The overall accuracy of the classifier stands at 70%.

Furthermore, the confusion matrix can be analyzed to give a deeper insight into the classification performed by the model visualized as in Figure 4.4.8:



**Figure 4.4.8: Confusion matrix for Logistic Regression classifier model**

It can be noted from Figure 4.4.8 that the true positive rates (147) are much higher than false positive rates (62). Whereas the true negative rates (29) are lower than false negative rates (62).

Finally, logistic regression models are prone to underfit the training set, and hence it is essential to verify the obtained accuracy by 10x cross-validation. This can be done by calculating the cross-validation score from the trained classifier model with the scoring option set to optimize the ROC curve. The obtained score is as displayed in Table 4.4.8:



**Table 4.4.8: Jupyter Notebook output for 10x cross validation score on Logistic Regression model**

## **4.5 Analysis of Hybrid Models**

The following results in Table 4.5.1 were obtained from each classifier model used:

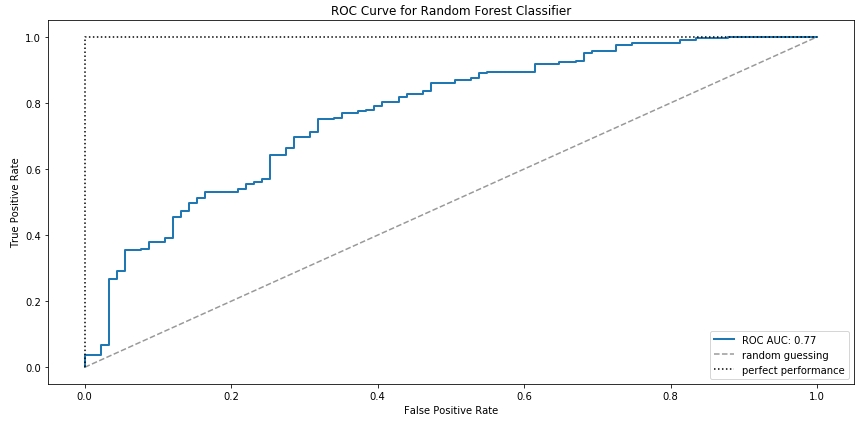
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric** | **Random Forest** | **XGBoost** | **SVC** | **Logistic Regression** |
| Accuracy | 75% | 74% | 75% | 70% |
| Precision | 86% | 79% | 83% | 73% |
| Recall | 75% | 74% | 75% | 70% |
| F1-score | 79% | 76% | 76% | 71% |

**Table 4.5.1: Classification results**

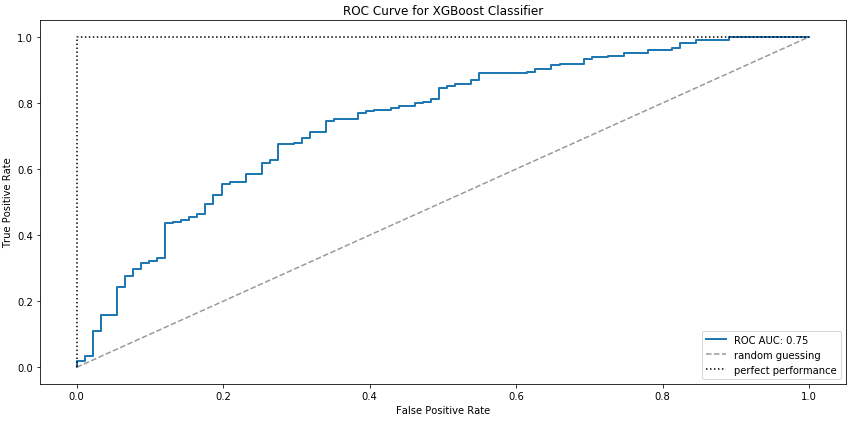
From Table 4.5.1, it is observable that Random Forest performed the finest and Logistic Regression the worst. However, the above results are not sufficient for this analysis as there needs to be a deeper inspection as elaborated in Section 3.9.2.

### **4.5.1 ROC Curves**

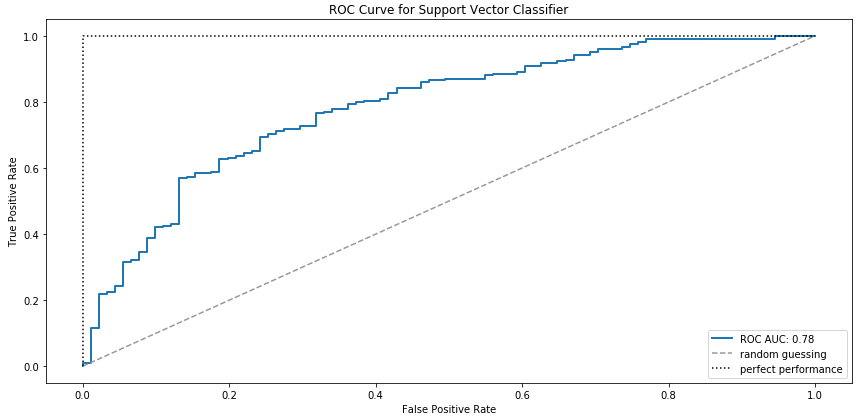
In the credit risk scenario, it is essential to minimize the false alarm rate or false positive rate as for credit unions and other financial institutions, it can be quite risky to tag a customer profile as a good risk when actually it is a bad risk than the other way around. A ROC curve determines this variation of false positive rate with the corresponding change in the true positive rate and hence explores the entire spectrum of accuracy cut points. The AUC evaluates the ROC results. Furthermore, the nomination of the hyperparameters of each model was determined to optimize the ROC analysis. The ROC curves for each classifier model are as shown in Figure 4.5.1, 4.5.2, 4.5.3, and 4.5.4:



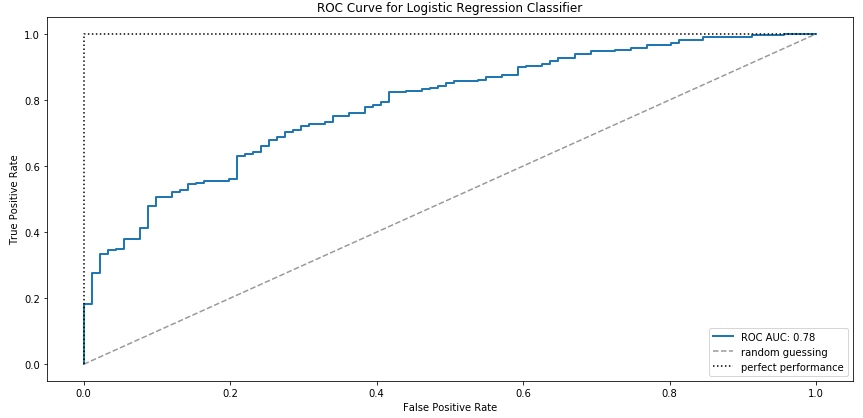
**Figure 4.5.1: ROC Curve for Random Forest Classifier**



**Figure 4.5.2: ROC Curve for XGBoost Classifier**



**Figure 4.5.3: ROC Curve for Support Vector Classifier**

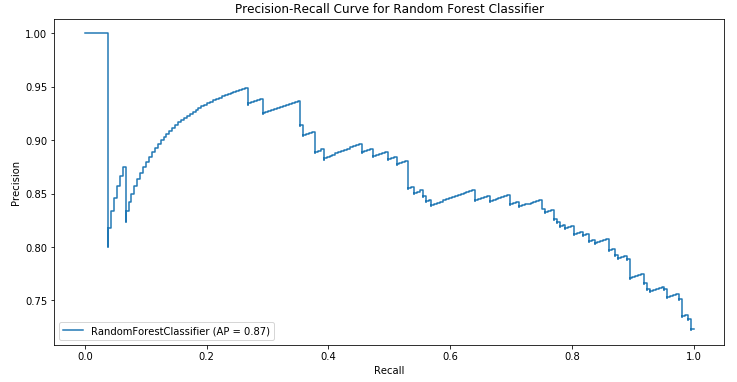


**Figure 4.5.4: ROC Curve for Logistic Regression Classifier**

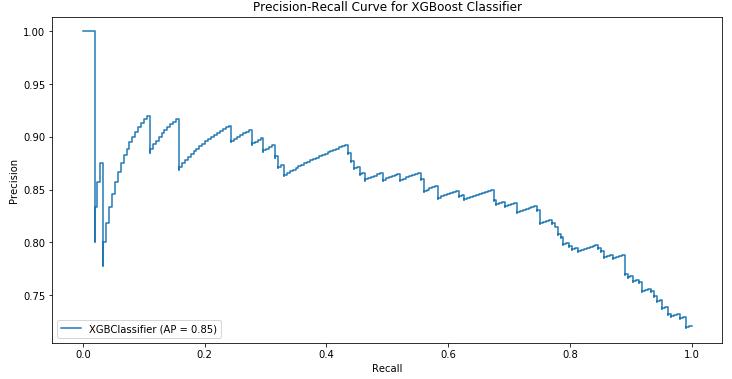
From the above ROC curves, it is quite observable that Logistic Regression and SVC did better than Random Forest and XGBoost which is quite contrary to the accuracy results.

### **4.5.2 Precision-Recall Curves**

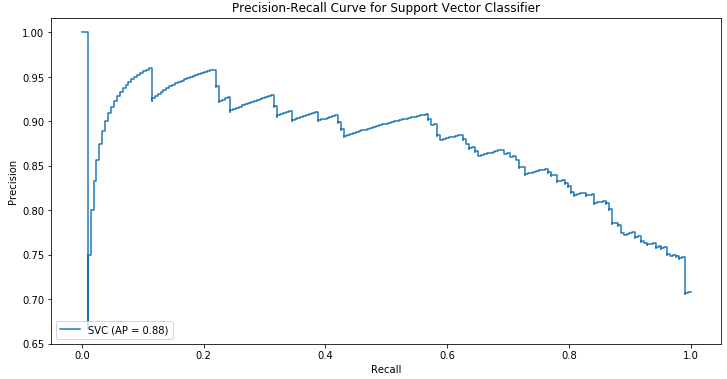
To get an even deeper insight into the minimization of false positive rate and maximization of true positive rate, the trade-off between precision and recall can be analyzed. This analysis takes into consideration the prioritization of the false positive rate minimization over the false negative rate minimization. Henceforth, it is essential to check the maximization of precision over each values of recall. The precision-recall curves for each classifier model are visualized in Figures 4.5.5, 4.5.6, 4.5.7 and 4.5.8:



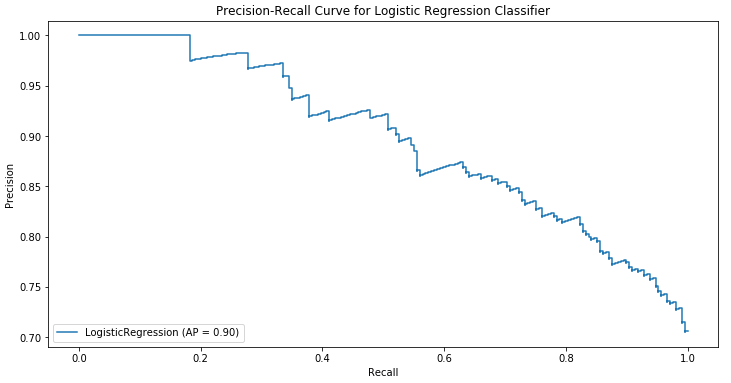
**Figure 4.5.5: Precision-Recall Curve for Random Forest Classifier**



**Figure 4.5.6: Precision-Recall Curve for XGBoost Classifier**



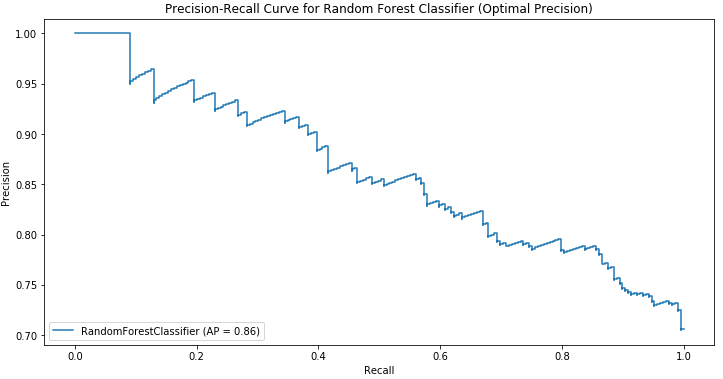
**Figure 4.5.7: Precision-Recall Curve for Support Vector Classifier**



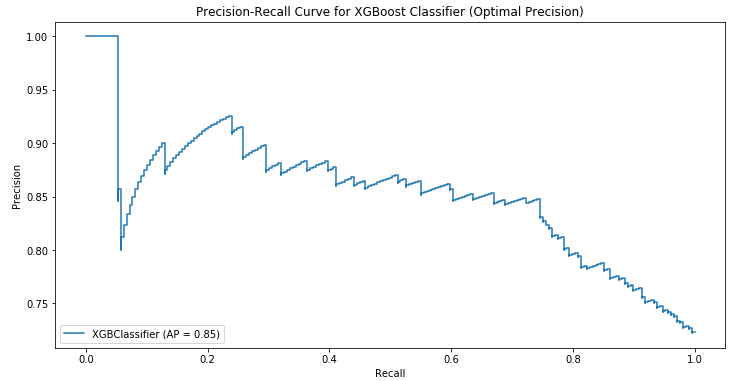
**Figure 4.5.8: Precision-Recall Curve for Logistic Regression Classifier**

These results of Precision-Recall curves back the results of ROC curves. It further explores that Logistic Regression maintains the best average precision rate (0.90) followed by SVC (0.88). Whereas XGBoost (0.85) followed by Random Forest (0.87) lag behind.

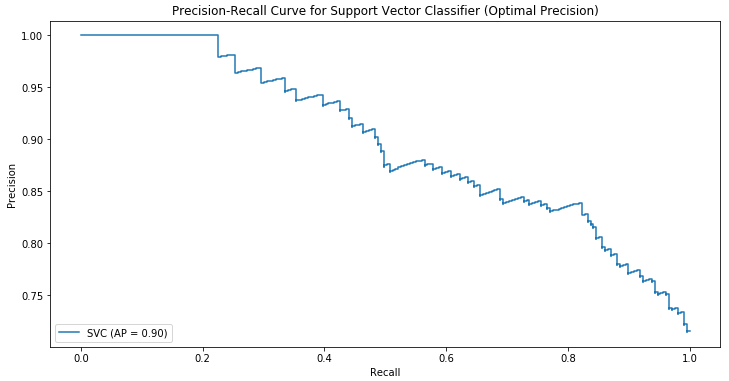
One thing to note that the above results are of a model trained to optimize ROC curves rather than precision. If the Grid Search CV and then the final model is trained by optimizing the precision then these new Precision-Recall curves unfold as in Figure 4.5.9, 4.5.10, 4.5.11, and 4.5.12:



**Figure 4.5.9: Precision-Recall Curve for Random Forest Classifier (Optimal Precision)**



**Figure 4.5.10: Precision-Recall Curve for XGBoost Classifier (Optimal Precision)**



**Figure 4.5.11: Precision-Recall Curve for Support Vector Classifier (Optimal Precision)**



**Figure 4.5.12: Precision-Recall Curve for Logistic Regression Classifier (Optimal Precision)**

The average precision rate of SVC (0.90) improved whilst the precision rate of Random Forest (0.86) reduced and for the others: Logistic Regression (0.90) and XGBoost (0.85) remained the same. There is a change in the curvature of all the curves except for Logistic Regression. However, the ranking remains the same as the previous results.

## **4.6 Conclusion**

This chapter described the application of this research as a set of working clustering-classification hybrid models. Before training the model, the variables were explored and pre-processing was carried out. The results of clustering and classification were verified with metrics and all the insightful similarities and contrasts were mentioned. Moreover, a new feature was utilized to bind the clustering and classification stages into a hybrid model. All the results obtained in this chapter would be examined and discussed in Chapter 5, the final chapter of this research.

# **Chapter 5 – Discussion and Conclusion**

## **5.1 Introduction**

The final chapter is a reflection on the insightful findings and retrospection on the model parameters. The information gathered from unveiling the classification reports is assimilated into a comprehensive knowledge of how a hybrid model functions and the driving forces behind. The class-wise classification is then considered to trace the sources of imbalance in precision. Furthermore, the nature of ROC and Precision-Recall has been examined under two optimizing conditions.

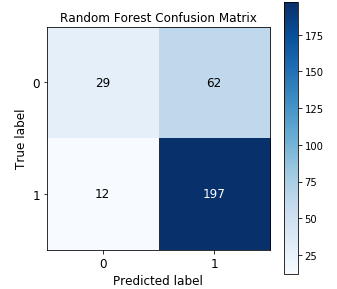
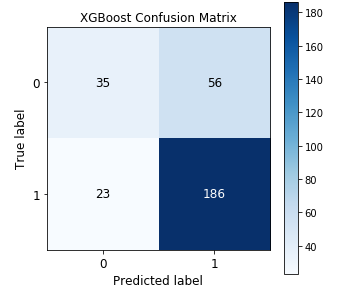
To verify whether the primary aim of this research has been met, the feature importance has been visualized. Besides that, this chapter even explores a secondary point of this research to find the best classifier in terms of computational complexity.

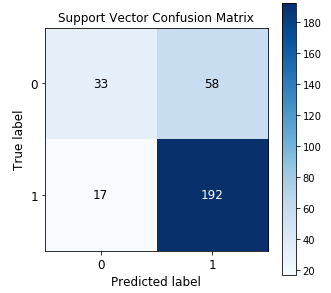
## **5.2 Discussion and Conlusion**

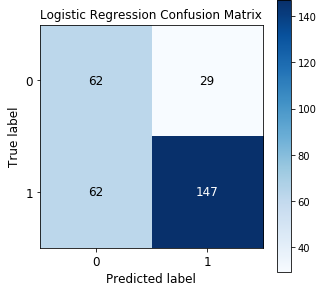
#### **5.2.1 Reflection**

Previous research such as Zakrzewska (2007) utilize the numerical variables solely for the clustering stage and eventually eliminates those variable and utilize categorical variables for the classification stage. However, in this research, it was noted that the numerical variables have the highest influence on the class labels, and hence it would be more fruitful to utilize these variables for the classification stage as well for this research scenario. Henceforth, the clustering labels as well as the numerical variables are included with all the other categorical variables for the classification task.

To reflect the results obtained in Chapter 4, the confusion matrices can render some notable insights as visualized in Figure 5.2.1:

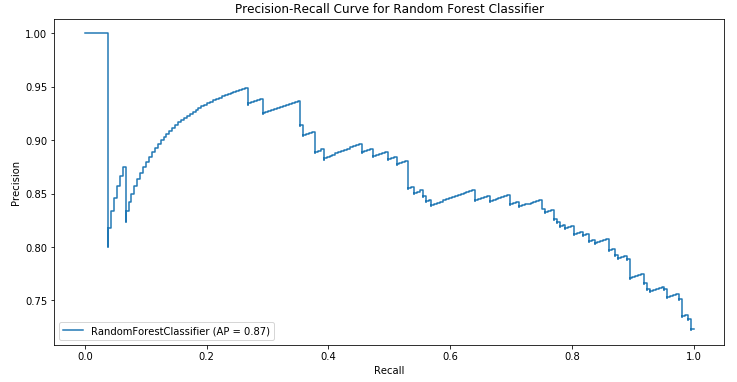




  
  
**Figure 5.2.1: Confusion matrices for classification retrospection**

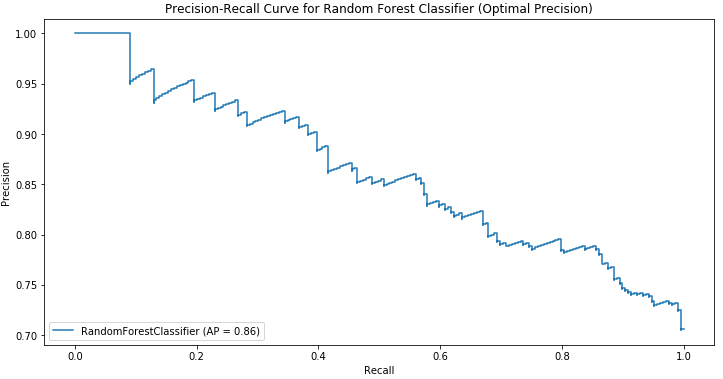
The Random Forest classification model proved out to be best in minimizing the false alarm rate (FP = 12) along with yielding a higher true positive rate. While the SVC stood just closer (FP = 17) by giving a rather better false negative rate (FN = 58) than Random Forest (FN = 62). However, the weight of this research on precision still marks Random Forest performing best in terms of accuracy metric.

The next part of the analysis relied on evaluation curves (ROC and Precision-Recall) to explore a supplementary angle in this research. The results gathered by this step were quite contrary. The contrast pointed by ROC curves was then evidently backed up by the insights from Precision-Recall curves. It proved Random Forest to be a bit maladjusted to all the variations in positives and hence its accurate results compromise in maintaining a high precision rate. The Precision-Recall rate on Random Forest described a sharp fall in maintaining a good precision rate with an initial rate of recall as visualized in Figure 5.2.2 (AP = 0.87):



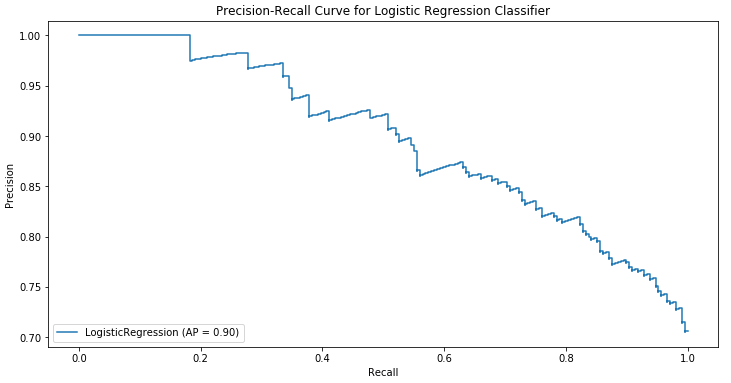
**Figure 5.2.2: Precision-Recall Curve for Random Forest Classifier (Optimal ROC\_AUC)**

When the same model was triggered by a different parameter to optimize precision rather than optimizing the AUC, the results described a bit steeper fall. However, the sharp fall of precision disappear in this version due to the efforts made by Grid Search cross-validation to maintain the precision. However, the fall of precision with rising recall seems inevitable as visualized in Figure 5.2.3 (AP = 0.86):

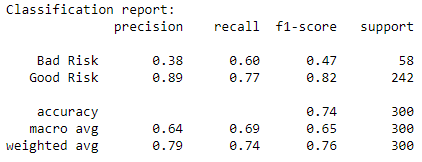


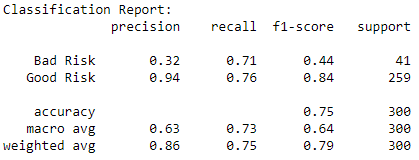
**Figure 5.2.3: Precision-Recall Curve for Random Forest Classifier (Optimal Precision)**

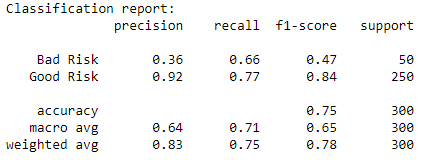
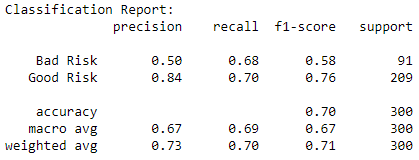
A similar pattern of avoiding the sharp fall of precision has been noted in all the other models except Logistic Regression. Even the Logistic Regression classifier optimized on ROC maintained a high precision rate at all times. The results were the same as visualized in Figure 5.2.4 the same for both cases:



**Figure 5.2.4: Precision-Recall Curve for Logistic Regression Classifier (Optimal ROC\_AUC/Precision)**

 This proved that the model maintained a healthy bias with the dataset and delivered the best results in terms of precision. To verify this a reflection on the classification report stands essential as the target labels were imbalanced in nature. When a look at the precision and recall rates in terms of each label is taken into the consideration noting the Table 5.2.1 and 5.2.2:

  
 **Table 5.2.1: Classification Report on Random Forest (Left) and XGBoost (Right)**



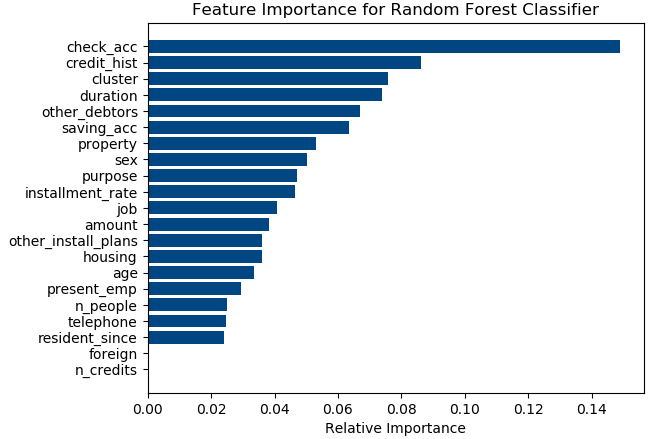
**Table 5.2.2: Classification Report on SVC (Left) and Logistic Regression (Right)**

There is a quite staggering difference in precision rates in terms of labels (Good Risk and Bad Risk) as evident from Table 5.2.1 and 5.2.2. The overall accuracy yields in Random Forest have been identified with its strongest rate of classifying on Good Risk label and the overall accuracy is heavily invested in Good Risk due to its major part in the distribution (70%) whereas the Bad Risk is occupied minorly (30%). Hence Random Forest utilizes this task by focusing on the majority and hence topping the results.

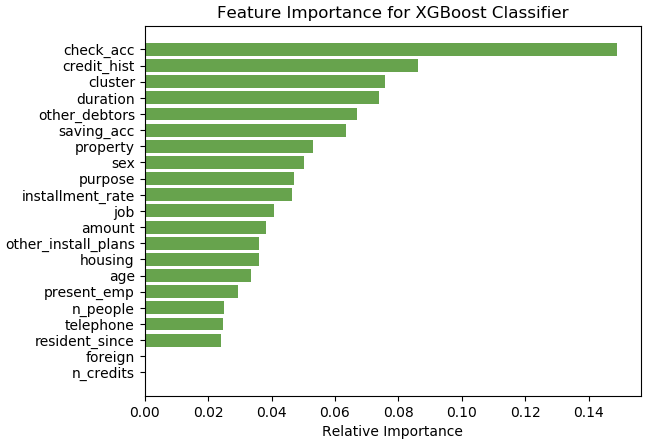
Looking towards the best classifier in classifying the Bad Risk, it stands as Logistic Regression (P = 0.50, R = 0.68). Although, it is the best among all and yet the rate is not very high. However, the low influence of the Bad Risk labels on the entire training model results in lesser accuracy yields. It follows that the precision rate of classifying Bad Risk could not go beyond 50% however the recall rate and accuracy are compromised.

#### 5.2.2 Feature Importance

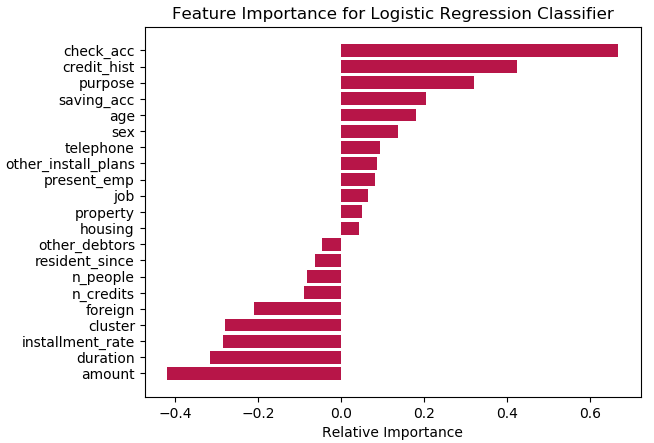
To add to the value of this research, the share of clustering in the output results cannot be overlooked. To verify that statement, the trained classifier algorithm can be looked into. Every trained classification model has a model parameter known as ‘feature importance’, this parameter ranks all the features utilized for training the model in order of their relevancies in the model. Those feature importance results can be converted to column names and then arranged in terms of its relevancy in the classifier. The bar plots describing the feature importance can be visualized as in Figure 5.2.5, 5.2.6, and 5.2.7:



**Figure 5.2.5: Bar plot on feature importance for Random Forest classifier**



**Figure 5.2.6: Bar plot on feature importance for XGBoost classifier**

  
**Figure 5.2.7: Bar plot on feature importance for Logistic Regression classifier**

There is no feature importance plot for SVC. SVC during training converts all the features into eigenvectors which operate in its individual dimension to find a solution and hence the contribution of each single feature cannot be calculated under such geometric conditions. Noting the plot for logistic regression in Figure 5.2.7, it can be seen that there are a set of features having negative importance. It describes the influence on the logistic regression from an opposite angle from the features with positive importance. Random Forest and XGBoost have the same weights on features. From the plots in Figure 5.2.5, 5.2.6, and 5.2.7; it is clear that the ‘cluster’ feature has a significant positive influence on Random Forest and XGBoost as well as a significant negative influence on Logistic Regression. And this proves that the outcomes from the clustering stage have a share of influence on the overall hybrid model.

#### 5.2.3 Computational Complexity of Algorithms

As a secondary study, the factor of computational complexity was taken into consideration. The secondary study focuses on finding a model for the classification stage which does the task in the optimal computational resource. The computational complexities of the classifiers are mentioned in Table 5.2.3:

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Training** | **Prediction** |
| Random Forest | O(n2pntrees) | O(pntrees) |
| XGBoost | O(npntrees) | O(pntrees) |
| SVC | O(n2p+n3) | O(pnsv) |
| Logistic Regression | O(p2n+p3) | O(p) |

**Table 5.2.3: Computational complexity of classifiers15**

From Table 5.2.3, n is the number of training samples, p is the number of features, ntrees isthe number of trees, and nsv is the number of support vectors. The degree of the polynomial for the training time of all algorithms is 3. However, Logistic Regression relies on a cube of the number of features heavily, and as p << n Logistic Regression takes the least time to be trained. On the other hand, in prediction Logistic Regression does best by only relying linearly on the number of features. And SVC leads in the other three models as nsv < ntrees. Overall, Logistic Regression is most efficient in classifying with optimum memory resource constraints.

## **5.3 Recommendations**

The retrospection in the Section 5.2 points out Logistic Regression and SVC are better techniques towards precision. However, due to the maintenance of the efficiency of SVC towards the overall accuracy of the model yet retaining a high AUC yield, SVC stands as the best for a hybrid model on top of k-means clustering. Moreover, the choice of classifier must rely on the aim of the analysis rather than higher yields. This research has accomplished the part of building a two-stage model to analyze the credit risks. Beyond this research scope, lie the efforts towards improving the metrics of the classifiers. A recommendation for balancing the target label distribution can be considered. Because the hybrid outcome model built in this study can practically work much better on datasets with balanced class labels and more balanced overall data distribution.

Another recommendation would be to include deep learning classifiers for the classification stage of model building as these sets of classifiers render a higher accuracy in predicting the labels and have an ability to form a feedback loop with the clustering stage for maximizing the clustering efficiencies as well.

15 https://www.thekerneltrip.com/machine/learning/computational-complexity-learning-algorithms/

# **References**

Allen, G. (2010) *Credit Unions*, *Public Banking Forum Ireland*. Available at: http://republicirelandbank.com/?page\_id=10 (Accessed: 25 July 2020).

Asiri, S. (2018) *Machine Learning Classifiers - Towards Data Science*. Available at: https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623 (Accessed: 16 June 2020).

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J. and Vanthienen, J. (2003) ‘Benchmarking state-of-the-art classification algorithms for credit scoring’, Journal of the Operational Research Society. Nature Publishing Group, 54(6), pp. 627–635. doi: 10.1057/palgrave.jors.2601545.

Balina, R. and Nowak, M. (2017) ‘Assessing individual credit risk on the basis of discriminant analysis by Poland’s cooperative banks’, *International Journal of Business Continuity and Risk Management*, 7(2), p. 103. doi: 10.1504/ijbcrm.2017.086067.

Basel Committee on Banking Supervision (2000) ‘Principles for the Management of Credit Risk’, *Bank for International Settlements*. Available at: https://www.bis.org/publ/bcbs75.htm (Accessed: 25 July 2020).

Bielecki, T. R. and Rutkowski, M. (2013) *Credit Risk: Modeling, Valuation and Hedging*. 1st edn. Berlin: Springer-Verlag.

Brems, M. (2017) *A One-Stop Shop for Principal Component Analysis - Towards Data Science*, *towards data science*. Available at: https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c (Accessed: 30 June 2020).

Brooks, K. (n.d.) *The History of Credit Unions*. Available at: https://vfccu.org/assets/boardtraining/The-History-of-Credit-Unions-2014.pdf (Accessed: 25 July 2020).

Brownlee, J. (2016) *How to Visualize Gradient Boosting Decision Trees With XGBoost in Python*, *Machine Learning Mastery*. Available at: https://machinelearningmastery.com/visualize-gradient-boosting-decision-trees-xgboost-python/ (Accessed: 9 August 2020).

Brownlee, J. (2020) *ROC Curves and Precision-Recall Curves for Imbalanced Classification*, *Machine Learning Mastery*. Available at: https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-imbalanced-classification/ (Accessed: 9 August 2020).

Chi, G., Hao, J., Xiu, Ch. and Zhu, Z. (2001) ‘Cluster analysis for weight of credit risk evaluation index’, Systems Engineering-Theory Methodology, Applications, 10(1), pp. 64–67.

Chicco, D. (2017) ‘Ten quick tips for machine learning in computational biology’, *BioData Mining*. BioMed Central Ltd., 10(1), p. 35. doi: 10.1186/s13040-017-0155-3.

Cole, R., Dupont, A., Meuleman, J., Liepins, A., Prato, O., Hied, M., Neumann, U., Laviola, S., Mori, T., Fujimoto, T., Morinaga, S., Reinard, D., Knot, K., Hedquist, J., Ferenius, C., Sprenger, M., Quick, J., Stephenson, M., Emblow, A., Schmidt, M., Walter, S., Elkes, D., Bailey, K., Strouzas, P., Martino, M., Nash, R. and Garcia, G.R. (2000) Principles for the Management of Credit Risk. Basel.

Cramer, J. S. (2005) ‘The Origins of Logistic Regression’, *SSRN Electronic Journal*. Elsevier BV. doi: 10.2139/ssrn.360300.

Dabbura, I. (2018) *K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks*, *towards data science*. Available at: https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a (Accessed: 1 July 2020).

Desai, V. S., Crook, J. N. and Overstreet, G. A. (1996) ‘A comparison of neural networks and linear scoring models in the credit union environment’, *European Journal of Operational Research*. Elsevier, 95(1), pp. 24–37. doi: 10.1016/0377-2217(95)00246-4.

Draelos, R. L. B. (2019) *Measuring Performance: AUC (AUROC)*, *Glass Box*. Available at: https://glassboxmedicine.com/2019/02/23/measuring-performance-auc-auroc/ (Accessed: 9 August 2020).

Ekin, O., Hammer, P.L., Kogan, A., and Winter, P. (1999) ‘Distance-based classification methods’, INFOR: Information Systems and Operational Research, 37(3), pp. 337–352. doi: 10.1080/03155986.1999.11732388.

Friedman, J. H. (2001) ‘Greedy Function Approximation : A Gradient Boosting Machine 1 Function estimation 2 Numerical optimization in function space’, *The Annals of Statistics*. Institute of Mathematical Statistics, 29(5), pp. 1189–1232. doi: 10.2307/2699986.

Gandhi, R. (2018) *Support Vector Machine*, *towards data science*. Available at: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47 (Accessed: 1 July 2020).

Van Gestel, T. and Baesens, B. (2009) *Credit Risk Management Basic Concepts: financial risk components, rating analysis, models, economic and regulatory capital*. 1st edn, *Oxford University Press*. 1st edn. New York: Oxford University Press Inc. doi: 10.1093/acprof:oso/9780199545117.001.0001.

Hancock, R. and Mueller, R. O. (2007) *Structural Equation Modeling: A Second Course*. 2nd edn, *Information Age Publishing Inc.* 2nd edn. doi: 10.1111/j.1744-6570.2007.00081\_13.x.

Hastie, T., Tibshirani, R. and Friedman, J. (2017) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd edn. Stanford: Springer.

Hofmann, H. (1994) *UCI Machine Learning Repository: Statlog (German Credit Data) Data Set*, *University of California, Irvine Machine Learning Repository*. Available at: https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data) (Accessed: 18 June 2020).

Hsieh, N. C. (2005) ‘Hybrid mining approach in the design of credit scoring models’, *Expert Systems with Applications*, 28(4), pp. 655–665. doi: 10.1016/j.eswa.2004.12.022.

Hsu, C., Chang, C. and Lin, C. (2003) *Evaluating unsupervised and supervised image classification methods for mapping cotton root rot Cotton leafroll dwarf virus: A Practical Guide to Support Vector Classification*. Taiwan. doi: 10.1007/s11119-014-9370-9.

Kearns, M. and Valiant, L. G. (1989) ‘Cryptographic limitations on learning Boolean formulae and finite automata’, *STOC ’89: Proceedings of the twenty-first annual ACM symposium on Theory of computing*. New York: Journal of the ACM, 41(1), pp. 433–444. doi: 10.1145/73007.73049.

Khosla, S. (n.d.) *K-means++ Algorithm*. Available at: https://www.geeksforgeeks.org/ml-k-means-algorithm/ (Accessed: 9 August 2020).

Kljucnikov, A., Majkova, M. S. and Vincurová, Z. (2018) ‘Credit standards and factors affecting their strictness in the segment of SMEs in the Slovakia’, *Article in Journal of International Studies*, 11(1), pp. 163–176. doi: 10.14254/2071-8330.2018/11-1/12.

Kolonko, M. and Schäl, M. (1979) ‘Optimal Control of Semi-Markov Chains under Uncertainty with Applications to Queueing Models’, *Papers of the Annual Meeting, Proceedings in Operations Research*. Physica-Verlag HD, 9, pp. 430–435. doi: 10.1007/978-3-662-00401-2\_76.

Kou, G., Peng, Y. and Wang, G. (2014) ‘Evaluation of clustering algorithms for financial risk analysis using MCDM methods’, *Information Sciences*. Elsevier Inc., 275, pp. 1–12. doi: 10.1016/j.ins.2014.02.137.

Kozubíková, L., Belás, J., Bilan, Y. and Bartoš, P. (2015) ‘Personal characteristics of entrepreneurs in the context of perception and management of business risk in the SME segment’, Economics and Sociology, 8(1), pp. 41–54. doi: 10.14254/2071-789X.2015/8-1/4.

Lacher, R.C., Coats, P.K., Sharma, S.C. and Fant, L.F. (1995) ‘A neural network for classifying the financial health of a firm’, European Journal of Operational Research. North-Holland, 85(1), pp. 53–65. doi: 10.1016/0377-2217(93)E0274-2.

Lundy, M. (1993) ‘Cluster Analysis in Credit Scoring. Credit Scoring and Credit Control.’, *New York: Oxford University Press*.

Malhotra, R. and Malhotra, D. K. (2003) ‘Evaluating consumer loans using neural networks’, *Omega*. Elsevier BV, 31(2), pp. 83–96. doi: 10.1016/S0305-0483(03)00016-1.

Manimaran, N. C. (2019) *Clustering Evaluation strategies - Towards Data Science*, *towards data science*. Available at: https://towardsdatascience.com/clustering-evaluation-strategies-98a4006fcfc (Accessed: 19 June 2020).

Nandu, M. (2019) *Machine Learning using Logistic Regression in Python*, *Good Audience*. Available at: https://blog.goodaudience.com/machine-learning-using-logistic-regression-in-python-with-code-ab3c7f5f3bed (Accessed: 9 August 2020).

Narkhede, S. (2018) *Understanding Confusion Matrix*, *towards data science*. Available at: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62 (Accessed: 9 August 2020).

Neuberger, D. and Räthke-Döppner, S. (2015) ‘The role of demographics in small business loan pricing’, *Small Business Economics*. Kluwer Academic Publishers, 44(2), pp. 411–424. doi: 10.1007/s11187-014-9602-4.

Norena, S. (2018) *Python Model Tuning Methods Using Cross Validation and Grid Search*, *Medium*. Available at: https://medium.com/@sebastiannorena/some-model-tuning-methods-bfef3e6544f0 (Accessed: 17 July 2020).

Olsson, U.H., Foss, Tron, T., Sigurd, V. and Howell, R.D. (2000) ‘The performance of ML, GLS, and WLS estimation in structural equation modeling under conditions of misspecification and nonnormality’, Structural Equation Modeling. Lawrence Erlbaum Associates, Inc. , 7(4), pp. 557–595. doi: 10.1207/S15328007SEM0704\_3.

Priy, S. (n.d.) *Clustering in Machine Learning*, *Geeks for Geeks*. Available at: https://www.geeksforgeeks.org/clustering-in-machine-learning/ (Accessed: 30 June 2020).

Ratanamahatana, C. and Gunopulos, D. (2003) ‘Feature selection for the naive bayesian classifier using decision trees’, *Applied Artificial Intelligence*, 17(5–6), pp. 475–487. doi: 10.1080/713827175.

Seal, H. L. (1967) ‘Studies in the history of probability and statistics XV: The historical velopment of the Gauss linear model’, *Biometrika*. Oxford Academic, 54(1), pp. 1–24. doi: 10.1093/biomet/54.1-2.1.

Shalev-Shwartz, S. and Ben-David, S. (2014) *Understanding Machine Learning: From Theory to Algorithms*. 1st edn. Cambridge: Cambridge University Press. Available at: http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning (Accessed: 30 June 2020).

Starmer, J. (2018) *Machine Learning Fundamentals: Cross Validation*. Available at: https://www.youtube.com/watch?v=fSytzGwwBVw (Accessed: 30 June 2020).

VanderPlas, J. (2016) *Python Data Science Handbook*, *O’Reilly Media, Inc.* Available at: https://jakevdp.github.io/PythonDataScienceHandbook/ (Accessed: 17 July 2020).

VanderPlas, J. (n.d.) *In Depth: Principal Component Analysis*, *O’Reilly Media*. Available at: https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html (Accessed: 17 July 2020).

Vos, E., Yeh, A.J.Y., Carter, S. and Tagg, S. (2007) ‘The happy story of small business financing’, Journal of Banking and Finance. North-Holland, 31(9), pp. 2648–2672. doi: 10.1016/j.jbankfin.2006.09.011.

Wirth, R. and Hipp, J. (2000) *CRISP-DM: Towards a standard process model for data mining*, *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*. London: Springer-Verlag UK. Available at: http://www.cs.unibo.it/~danilo.montesi/CBD/Beatriz/10.1.1.198.5133.pdf (Accessed: 12 August 2020).

Wolfbein, S. L. (2017) ‘The impact of social and economic factors in the credit risk management of SME’, *Technological and Economic Development of Economy*, 24(3), pp. 1215–1230. doi: 10.1002/j.1556-6978.1967.tb01896.x.

Wu, X., Kumar, V., Ross, Q.J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., Zhou, Z.H., Steinbach, M., Hand, D.J. and Steinberg, D. (2008) ‘Top 10 algorithms in data mining’, Knowledge and Information Systems. Springer, 14(1), pp. 1–37. doi: 10.1007/s10115-007-0114-2.

Zakrzewska, D. (2007) ‘On integrating unsupervised and supervised classification for credit risk evaluation’, *Information Technology and Control*, 36(1A), pp. 98–102.

Zekic-Susac, M., Sarlija, N. and Bensic, M. (2004) ‘Small business credit scoring: a comparison of logistic regression, neural network, and decision tree models’, *26th International Conference on Information Technology Interfaces*, 1, pp. 265–270. Available at: https://ieeexplore.ieee.org/abstract/document/1372413 (Accessed: 25 July 2020).

### **List of Abbreviations**

**SME**: Small to Medium Sized Enterprises

**AAM**: Adjusted Averaging Method

**AWN**: Adjusted Weighting Method

**TPM**: Truncated Potentials Method

**CCM**: Convex Containment Method

**NBC**: Naïve Bayes Classifier

**SBC**: Selective Bayesian Classifier

**UCI**: University of California, Irvine

**EM**: Expectation-Maximization

**MCDM**: multiple criteria decision-making

**TOPSIS**: Technique for order preference by similarity to ideal solution

**DEA**: Data envelopment analysis

**VIKOR**: Multi-criteria optimization and compromise solution (\*translated)

**TP**: True Positives

**TN**: True Negatives

**FP**: False Positives

**FN**: False Negatives

**ROC**: Receiver Operating Characteristics

**AUC**: Area Under the Curve

**PCA**: Principle Component Analysis

**XGBoost**: Extreme Gradient Boosting

**SVM**: Support Vector Machine

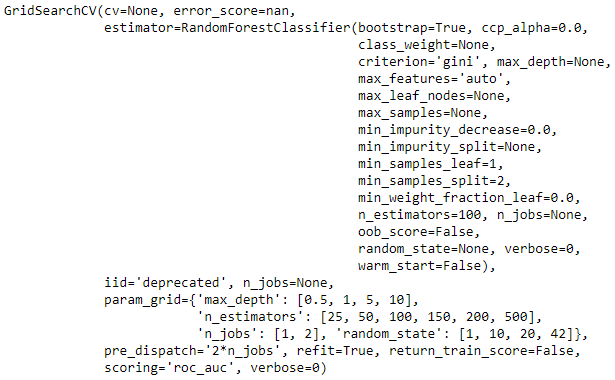
**SVC**: Support Vector Classifier

# **Appendix**

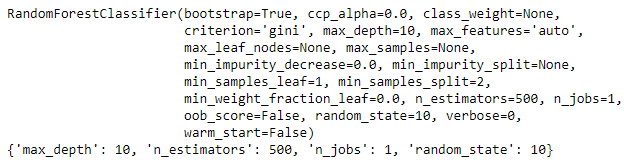
## **Model Hyperparameters Selection**

### **Random Forest Classifier**

**Input:**

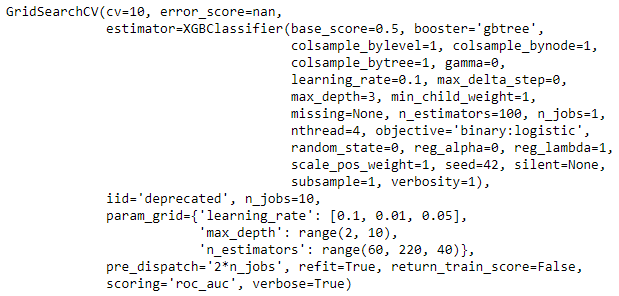


**Output:**

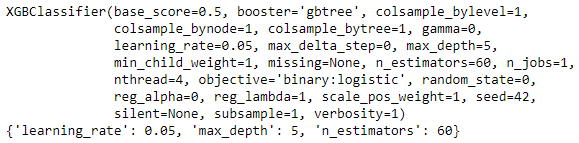


### **Extreme Gradient Boosting Classifier**

**Input:**

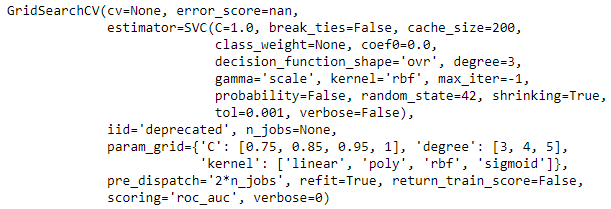


**Output:**

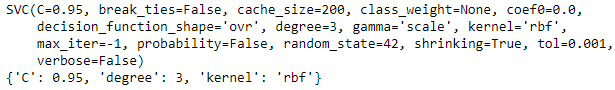


### **Support Vector Classifier**

**Input:**

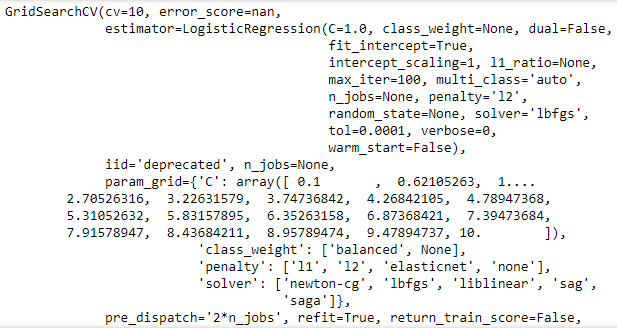


**Output:**

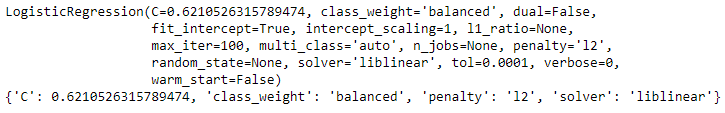


### **Logistic Regression Classifier**

**Input:**



**Output:**



## **Libraries Utilized**

### **Data Manipulation**

**NumPy**: numerical and matrix manipulation.

**Pandas**: data-frame manipulation.

### **Machine Learning**

**Sklearn.preprocessing**: label encoder fit-transform, standard scalar.

**Sklearn.decompostion**: PCA.

**Sklearn.model\_selection**: Grid Search cross validation, cross validation score, train\_test\_split.

**Sklearn.ensemble**: Random Forest classifier model.

**Sklearn.metrics**: confusion matrix, classification report, roc\_curve, auc, precision\_recall\_curve.

**Sklearn.feature\_selection**: select from model.

**Xgboost**: XGBoost classifier model.

**Sklearn.svm**: Support Vector classifier model.

**Sklearn.linear\_model**: Logistic Regression classifier model.

**Sklearn.cluster**: K Means clustering model.

### **Visualization**

**Matplotlib**: general visualizations.

**Seaborn**: adding layers to visualizations.

**Scikitplot**: confusion matrix visualizations.

**Plotly**: interative visualiztions.

**mpl\_toolkits.mplot3d**: 3D plots.