CSE343/ECE3
Credit Score Classification Using Machine Learning Algorithms

Aman Kumar	Pritish Poswal	Vibhu Jain	Karan Prasad Gupta
aman20279@iiitd.ac.in	pritish20321@iiitd.ac.in	vibhu20151@iiitd.ac.in	karan20439@iiitd.ac.in
2020279	2020321	2020151	2020439

Abstract

*"The world is one big data problem." ~**Andrew McAfee.***
These are true words of a wise man, meaning big problems can be solved using big data. We also set out to solve some significant issues in Today's world. After spending a lot of time brainstorming, we came across the problem of Credit score classification.
Credit cards have become an integral part of our lives and a massive fraction of young and medium age people use them. The usage of credit cards has also increased over the years, and with the emergence of companies like Cred, people are incentivized to use credit cards. As the usage of credit cards increases, so does the need for a system of validation of the user so that a lender can determine if the user can repay the loan. A credit score also benefits the user by allowing the deserving person to get loans, credit cards, and more. It also helps in determining which loans would be difficult to pay off.
We found this problem exciting and essential. Thus, we decided to solve this using ML.
Github Link-
https://github.com/karanprasadgupta/Credit-Score-Classification-Using-ML

1. Introduction

Credit score classification is a complex problem as it depends upon many parameters and factors.
Payment history, debt-to-credit ratio, length of credit history, new credit, and the amount of credit you have all play a role in your credit report and credit score. All this data is readily available as details of transactions are recorded for all users, and the majority of the youth also use credit cards.
In this project, we want to classify the credit score as good, bad, and standard by using these factors and achieving reasonable accuracy.
Occupation is also an essential factor for determining the capacity of the user, and many lenders use it for the same. In this project, we are classifying credit scores without occupation as this leads to discrimination, and occupations should be treated equally and with respect. This will also help people with undervalued occupations to get loans if they are eligible.
We will use various preprocessing techniques and different models to achieve maximum accuracy for the classification of credit scores.

2. Literature Review

In this section, we go through some research-related work in Credit Score Classification using Machine Learning Models.
The Paper **Credit Risk Scoring Analysis [1]** reports efforts in using feature engineering and machine learning models for credit Score modeling and reporting their AUC scores. The steps used in Data Pre-processing in the paper were (1) Anomalies and contradiction detection, (2) Missing Data Imputation, (3) Nominal Data Pre-processing, (4) Data Integration, (5) Feature Selection, (6) Feature Construction. A total of three models were trained, namely (1) Logistic Regression, (2) Random Forest (3) Light GBM. The paper used four datasets, including one original dataset and three constructed datasets. The results show that the Random Forest model performs best when trained on the original dataset. The Logistic Regression performed best when trained on a dataset generated through the polynomic approach.

The paper **Credit scoring using machine learning algorithms[2]** used the AUROC approach to analyze machine learning classification methods by doing 10-fold-Cross-validation for the dataset German Credit Data Set. This paper aimed to develop and evaluate the classification data mining techniques.
The models used for training were: Random Forests; Lasso regression; Support Vector Machine; Logistic Regression. The dataset in this paper has the output as binary type, due to which ROC curves were used in evaluating this classification problem. The result of this paper was that the Lasso Regression model had an accuracy of 80 %, which implies that Regression is a suitable model for classifying the credit score.

3. Dataset

3.1 Description of Dataset

The dataset has been taken from Kaggle and comprises 28 attributes, including the target class (Credit_Score) and 27 other variables. A total of 1,00,000 records are present in the dataset. The first field in the dataset is an ID that uniquely identifies the records.

The Schema of the dataset is as follows :
ID (string), **Customer_ID** (string), **Month** (string), **Name**

(string), **Age** (string), **SSN** (string) , **Occupation** (string), **Annual_Income** (string), **Monthly_Inhand_Salary** (float64), **Num_of_Bank_Accounts** (int64), **Num_Credit_Card** (int64), **Interest_Rate** (int64), **Num_of_Loans** (string), **Type_of_Loan** (string), **Delay_from_due_date** (string), **Num_of_delayed_payment** (string), **Changed_Credit_Limit** (string), **Num_Credit_Inquiries** (float64), **Credit_Mix** (string), **Outstanding_Debt** (string), **Credit_Utilization_Ratio** (float64), **Credit_history_age** (string), **Payment_of_min_account** (string), **Total_EMI_per_month** (float64), **Amount_Invested_monthly** (string), **Payment_Behaviour** (string), **Monthly_Balance** (string), **Credit_Score** (string) .

We can see that some fields like Annual_income, Num_of_loan, and Num_of_delayed_paymet should be float/int, but they are strings due to junk string values in the database.
Null values also exist along with the junk values mentioned above.

Class Distribution Graph :

Most of the records are that of **standard** credit score, followed by **Poor** and **Good**, but overall we say that the dataset is not skewed as all classes have ample representation.

3.2 Cleaning the Dataset

3.2.1 Removing Redundant columns

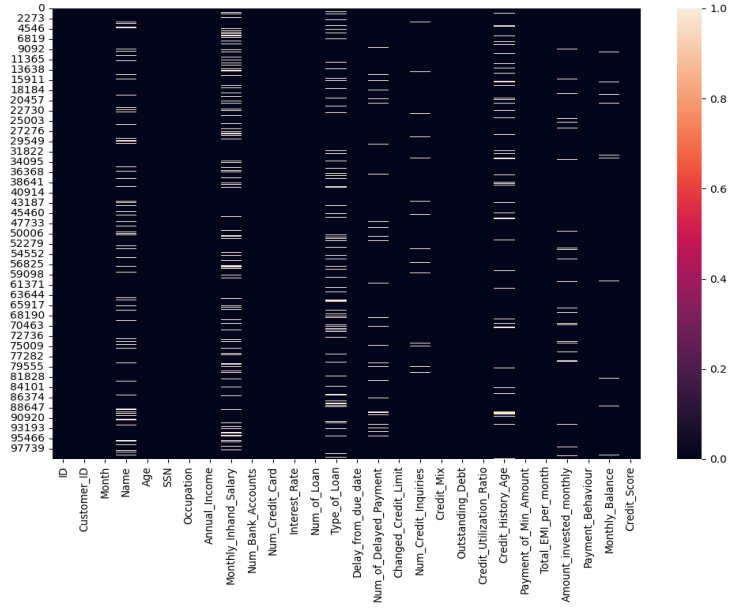
First, we dropped all the columns which did not affect the credit score classification.
Columns dropped - ID, Customer_ID, Name, SSN, Occupation.
ID, **Customer_ID**, **Name**, and **SSN**(social security number) represent a user's personal (account-related) data, and this does not affect the credit score classification.
Occupation is dropped due to reasons mentioned in the intro.

3.2.2 Checking Duplicate Records

We also checked for duplicate records but found none.

3.2.3 Finding and handling Null Values

To find the null values in various fields, we used heatmap with the parameter as null and got the following plot :



It can be seen that there are eight fields in total which contain null values, and Name, monthly_inhand_salary, Type_of_loan, and credit_history_age are the fields which contain most of the null values.
We handled these as follows :
Name - didn't handle as the Name column has to be dropped.
Monthly_inhand_salary -We used the annual salary to calculate the approx monthly salary.
Type of loan - We put the type as 'Not specified. '
Num_of_delayed_payments , and num_credit_enquires Were replaced with 0.
Credit_history_Age - We removed the records with null values.
Amount_invested_monthly, the monthly balance is replaced by the mean of the respective fields.

3.2.4 Handling Junk values

Some fields had junk values such as '_' added after the value in entry, and other junk like '__10000__', "__-333333333333333333333333333__" were present randomly. In some cases where we find that the data is not that relevant, we remove that record, and in other cases, we remove the extra character and typecast the field from string to int/float to get the numeric data.

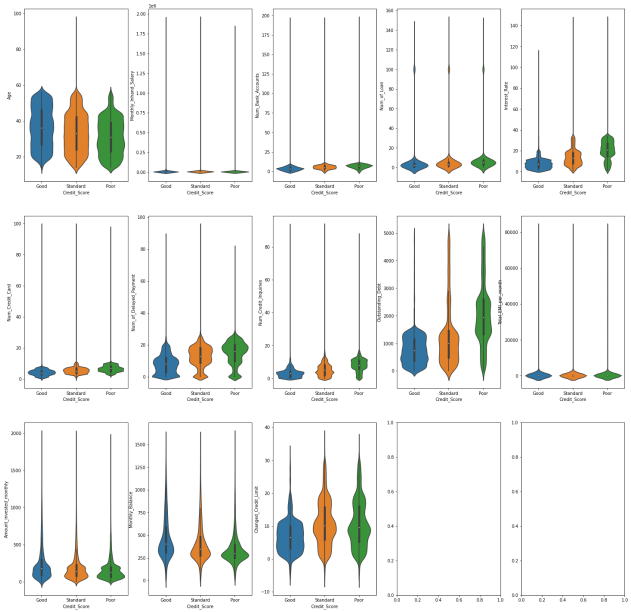
3.2.5 Handling negative Records

Some parameters had negative values(expected to have positive values); we had to take the absolute value of these

records in some cases, and in some cases where there was a negative junk value, we had to remove the record.

3.3 Exploratory Data Analysis (EDA)

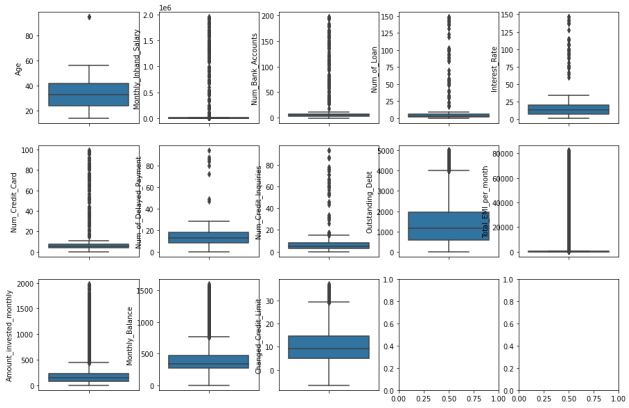
We tried to do EDA on many fields and made many plots which helped us understand the data better. Some of the interesting plots are below :



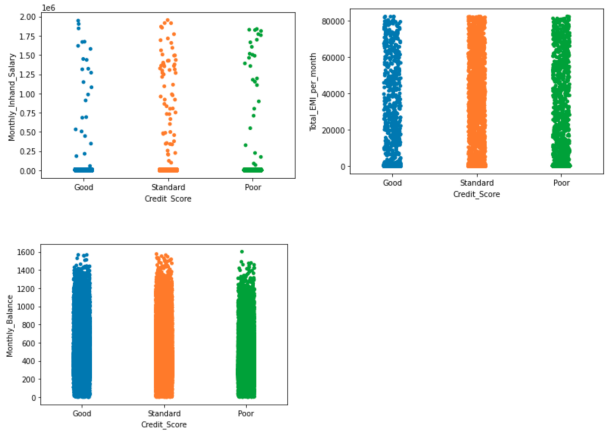
We plotted the violin plot of all the non-string fields, and this was our observation. We can infer the change in the density of various parameters for each credit score.

Age - The distribution is pretty uniform for people with good credit scores. As the quality of credit scores falls, the density of people in the 45-60 age range becomes less. Hence, people above age are more likely to have good credit scores.

Outstanding Debt - As credit quality deteriorates, the bulge in the graph goes higher; this implies people with more outstanding debt are more likely to have poor credit scores.

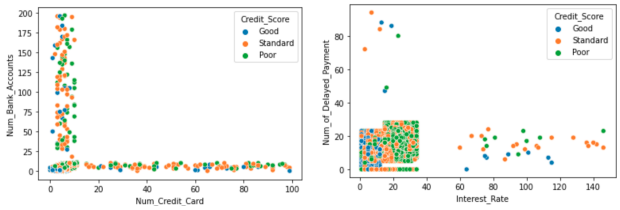


Some fields, like the number of bank accounts, Amount invested monthly, the number of credit cards, etc., are standard throughout. Some fields like Age, Num of Delayed payments, Interest rate, and changed credit limit have outliers that can be removed.



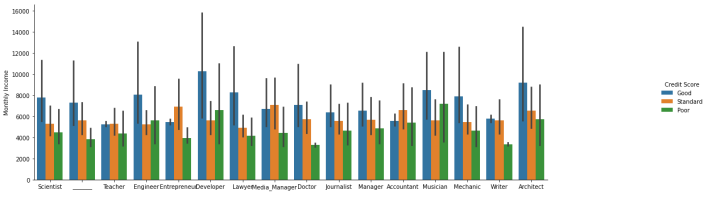
Strip plots for Monthly_Inhand_Salary show most of the records in each credit score category are near the base of the graph, and the outliers extend to a similar upper limit in each category. Standard credit score has the most outliers; it could be because records in this category are more.

Strip plot for EMI per month shows a uniform plot throughout, and the Monthly balance also has a similar plot; the only difference is that the latter has some outliers.



We can see a cluster between 0-20 numbers of credit cards showing a linear dependency between **the number of bank accounts and the number of credit cards**. There are a lot of outliers that primarily fall in standard credit.

Num of delayed payments and interest rate also shows a similar plot; the difference is that the number of outliers is less. The ones with poor credit form clusters between 20-40% interest rate, whereas the others occupy lower ranges.



Overall, higher-income people usually have good credit scores, followed by standard and poor ones.

Developers have the highest-income individuals with good

credit scores, whereas writers and doctors have lowest income individuals with poor credit scores.

3.4 Preprocessing

We looked into many techniques for preprocessing the data to improve the accuracy of different models

3.4.1 Handling Outliers

We plotted the boxplots of all numeric parameters to remove absolute outliers, which were very far away from the whiskers of the boxplot.

Then we used the boxplots and violin plots plotted in EDA to remove the outliers closer to the boxplot's whiskers.

We removed the following outliers
Age : less than 0 and greater than 75,
Interest_rate: greater than 115,
Num_of_Loans : greater than 99,
Num_of_delayed_payment : greater than 35,
Num_Credit_Inquiries: greater than 80,
Outstanding_Debt : greater than 4050,
Changed_Credit_Limit: greater than 30,
Total_EMI_per_month: greater than 10,000

3.4.2 Encoding string to int

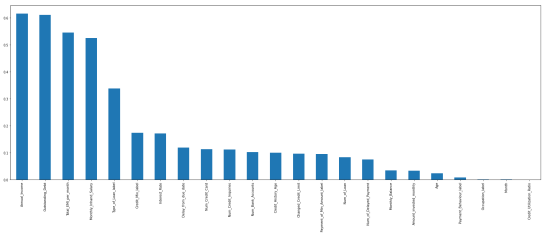
Some fields were important but were strings, so we encoded them as numbers, such as month.
Some fields, like Age, had values in terms of years and months; we typecast this to float by dividing months by 12 and adding to years.

3.5. Feature Selection

We used feature selection to select relevant features for our model, which have a decent amount of role to play in the predictions of our ML model. We used the following two methods for selecting the essential features.

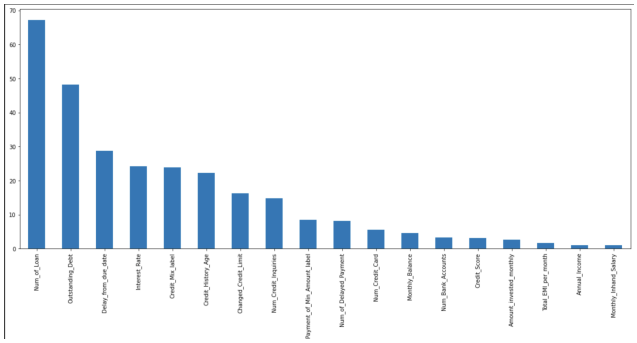
3.5.1 Information Gain

The second criteria used for feature selection is information gain. Information gain is calculated according to the reduction of entropy. The parameters with the highest information gain are the most important and valuable for our ML model, while those with the least information gain can be appropriately dropped. We plotted the Information Gain graph, which came as follows :



We can see the information gain of credit_utilization_ratio , Month, and Payment_behviour_label is significantly less than the others; hence we drop these fields before implementing our ML models.

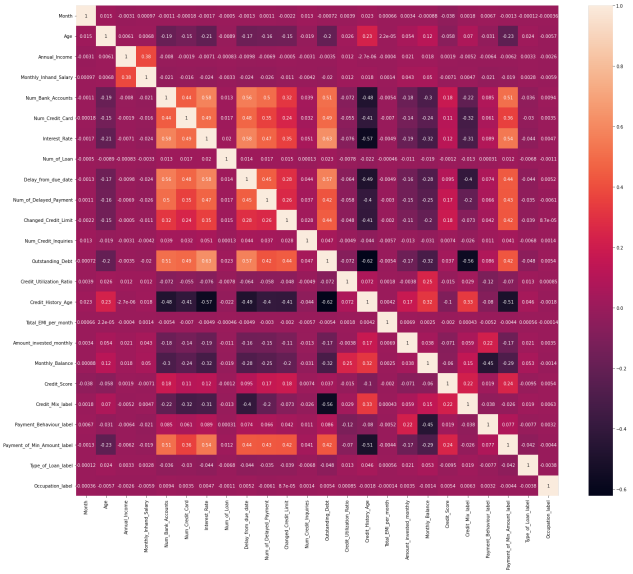
3.5.2 Fischer Score



Fisher score is the method to select the most valuable features for training our model. Features with higher fisher scores are more useful in predicting our target variable. Fisher's score is based on the principle of Maximum likelihood estimation. We take note of the features with the lowest score.

3.5.3 Heatmap

Here we found those columns with high correlation between them, i.e., we ideally looked for columns with a correlation greater than 0.5. Out of the two columns with a high correlation, we removed the column with a lower correlation with the target variable in the same heatmap. We noted features with high correlation from the heatmap to decide which feature to remove after analyzing the heatmap, information gain, and Fischer score.



4. Methodology

In this project, our objective was to classify the credit score of the concerned person as Good, Standard, and Poor from the given data with various related parameters which affect the person's credit score in real life.
The dataset was huge but required a lot of preprocessing as it contained multiple null values and many values which were not suitable logically according to our problem statement. Several values had"_" at the end. Many parameters had outliers that had to be removed to get better results in our ML model. The details of all this preprocessing have been explained in the section above.
A lot of fields had outliers that needed to be removed.
We did a lot of EDA, which helped us understand the data better and determine how to remove the outliers.

After all this preprocessing of data, we applied various machine learning models from the sklearn library to our dataset and measured the accuracy, Precision, Recall, and F1 score for each.
We applied Logistic Regression, Gaussian Naive Bayes, a Decision tree with both the Gini index and entropy criteria, and a Random forest classifier to find the best model for our given dataset.
After training the data with the above models, we tried models like multilayer perceptrons, Support Vector Machines (SVM), and K-nearest neighbors(KNN).
We also had to standardize the data to obtain good results in the SVM classifier.

We used Grid search to fine-tune the Hyperparameters on the models with good accuracies, such as KNN, Random Forest, and Extra Tree Classifier. We chose varied values for parameters such as the number of iterations, depth size, and criterion and compared them using Grid search.

We also built a classifier by training 100 different weak MLP classifiers on a fraction of the dataset, then made a prediction using all 100 weak classifiers and took the final estimate as the majority prediction among the weak classifiers. This is our MLP bagging classifier, and it was an improvement over MLP.

We also applied some ensembling techniques such as XGBoost(Extreme Gradient Boosting) and Extra Trees Classifier, similar to a random forest. Still, it randomly chooses the split point and does not calculate the optimal one. After finding good accuracy on some models, we fine-tuned the hyperparameters and analyzed those models. We also tried combining several models using StackingClassifier to get better results.

5. Results and Analysis

The results for each model were satisfactory at the end of training a couple of models and entirely above the random guess mark of 50% accuracy. The below table shows the performance of all the models used for classification.

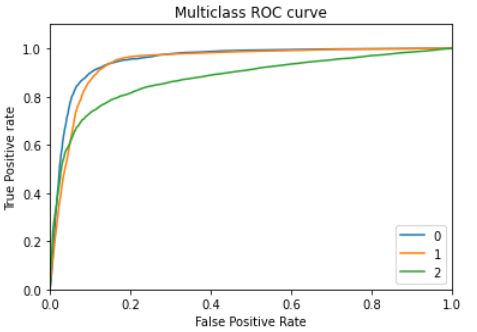
Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	57.65 %	0.49	0.43	0.41
Gaussian Naive Bayes	59.28 %	0.54	0.51	0.52
Decision Tree with Gini	71.86 %	0.69	0.70	0.69
Decision Tree with Entropy	72.68 %	0.70	0.72	0.71
Random Forest	81.14 %	0.80	0.80	0.80
KNN	76.17 %	0.75	0.75	0.75
MLP	57.00 %	0.60	0.38	0.33
MLP bagging	60.40%	0.61	0.62	0.61
XGB (Extreme Gradient Boosting)	73.14 %	0.71	0.72	0.71
Extra Tree Classifier	80.39 %	0.79	0.79	0.79
SVM	72.00 %	0.70	0.71	0.70
StackClassifier(KNN + Random forest)	80.60 %	0.80	0.79	0.80
StackClassifier(XGB+ KNN + Random forest)	80.85 %	0.80	0.80	0.80

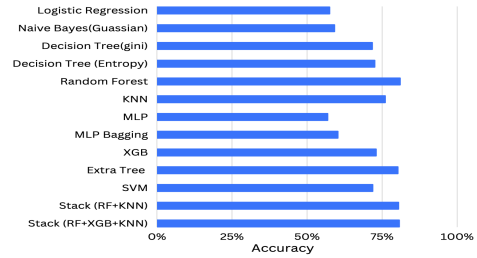
These results were obtained after optimizing and

fine-tuning the hyperparameters for some models to produce the best results.

ROC curve of Random forest

0 - good 1 - standard 2 - poor





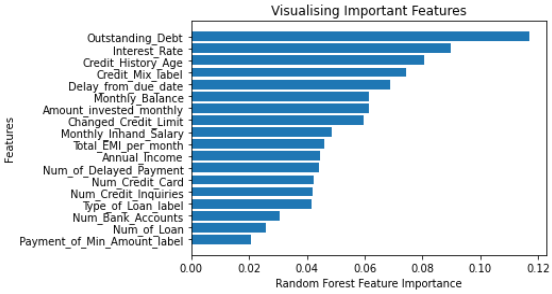
Looking at all the evaluation methods, it is evident that the random forest model outperforms all the other models with maximum accuracy of 81.14%; Extra Tree Classifier Tree being a similar model, gives us an accuracy of 80.39%, which means calculating an optimum split point rather than randomly choosing one produces better results.

We ran the SVM on limited data, i.e., 5-10% of the dataset, as it is challenging to perform SVM classification on a vast dataset. Still, it gives an accuracy of 72%, which is still better than some models.

We stacked models like the random forest, k nearest neighbors, etc. the maximum accuracy achieved was around 80% which was still lower than the random forest model.

We concluded that the random forest model is best for credit score classifications.

We employ the random forests-based plot to evaluate variable importance, as shown in the figure below.



The higher the decrease in the Gini means the variable plays a more significant role in partitioning the data into the probable classes. From the above plot, we found that the feature which affects the credit score most is a person's outstanding debt.

6. Conclusion

6.1 Learnings from Project

This project helped us understand the essence of working with large datasets with various null, improper values, and outliers in the dataset, along with multiple values unsuitable for usage.

We could understand various data visualization, plotting, EDA, and feature selection techniques. We also understood how small things like outliers, selection of proper features, normalization, etc., could significantly improve or decrease the accuracy of the ML models and hence understand the importance of appropriate data for preparing such ML models.

We explored numerous models and learned various model optimization techniques. We got accustomed to varying hyperparameters for better accuracy according to the scenario. Many models and techniques we used were not part of the course syllabus, and hence domains of our knowledge expanded.

6.2. Future Work

We have followed the proposed timeline given in the project proposal and produced an accurate ML model. We did much more work than our initial proposal, but there is still scope for improvement. In the future, we can work and find out a model which is even more accurate, which is a combination of multiple models which we have missed. We also plan to use our knowledge gained in this project to work with a dataset that predicts the exact CIBIL score and optimize it further for accuracy.

6.3. Individual Contributions

Aman Kumar: Preprocessing, Exploratory Data Analysis, Hyperparameter tuning, Data Visualization, Report Writing, Analysis of the performance of the models.

Karan Prasad Gupta: Literature Review, Data Visualisation, Model Training specially stacking of the models, Model Testing, Report Writing

Pritish Poswal: Data Cleaning, Data Preprocessing , Applying ML Models to data, Model Selection, Model Testing, Hyperparameters tuning, Making presentation

Vibhu Jain: Data Preprocessing, Data Visualisation, Exploratory Data Analysis, Literature Review, Feature Selection, Report Writing and making presentation

7. References :

[1] https://www.researchgate.net/publication/338526492_Credit_Risk_Scoring_Analysis_Based_on_Machine_Learning_Models

[2]

https://www.nust.ac.zw/zjst/index.php/volume-13?download=119:credit-scoring-using-machine-learning-algorithims