Chenghao Liu, V00841275
Zheng Yin, V00915261
Siyi Zhu, V00928181

Final Report

1.Introduction

Lending companies always need to be cautious and consider many factors when judging whether to approve a client's loan application, to maximize their interests. It will greatly facilitate the company if it can predict new loan applications through past approval and rejection cases and realize automated prediction. Our topic is to find and use a suitable model to input the user's detailed information and give a high accuracy output to determine whether the company approves the loan application of the client. We hope that after the clients fill in the personal information online, it will automatically give the loan qualification feedback in real-time so that the company can save human resources while improving accuracy and efficiency.

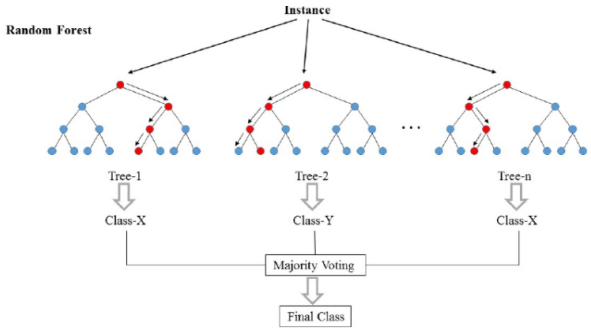
What our project has done is shown below. In the whole project, first of all, we found several different data sets from different sources, in case the preferred data set is not available. After that, the data set was trained on two suitable models to observe the accuracy and optimal parameters of the model. Because we were not satisfied with the accuracy of the resulting model, we reprocessed the data set to obtain a new optimized one. On the same model, the optimized data set was trained again. By comparing the results before and after, finally, get a high-precision result.

We have found 3 CSV dataset files from *Analytics Vidhya* that contain the personal information of previously applied clients and the company's approval results. The dataset has 12 features representing customers' personal information, such as marital status, education level, income, loan amount, etc. And a label that decides whether to approve the application. The first problem we face is data preprocessing. The specific problems are divided into the following three points: 1. Observing that the original data set has null values, we need to consider the method of dealing with missing values. 2. The data set has the features are formed with string and need to be mapped to an integer. 3. The problem of the accuracy of the output result. if the underfitting situation appears, more features need to be created. After completing the data preprocessing, we need to consider model selection. The problem of this project is a typical binary classification problem, so we chose two training models: random forest and logistic regression. At the same time, the impact of different parameters on the accuracy of the model needs to be considered, specifically, the number of decision trees in a random forest, and the degree of regularization in logistic regression. Finally, the methods to evaluate whether the model is ideal need to be considered. Calculating model accuracy is a method, but it is not enough. We decided to make the ROC curve and compute AUC to further evaluate the extent to which we have achieved the goal. Considering the intuitiveness of the results, we need to visualize all the results, that is, use suitable graphs to show all the results.

2. Approach

2.1 Random Forest and Logistic Regression

In this experiment, we are using random forest and logistic regression to experiment on the data.
The first model we apply is random forest. Random forest is an ensemble learning method for classification, regression, and other tasks that operate by generating several decision trees at training. “A large number of relatively uncorrelated medals(trees) operating as a committee will outperform any of the individual constituent models”[12]. The random forest can prevent one tree from having problems that affect the result of training. Random forest models are using two methods to confirm that every individual tree is not related to another tree in the model. First, the bagging method allows every tree in the forest to randomly sample from the dataset with replacement.[12] Secondly, there is a feature randomness function in a random forest, which every tree in the forest can pick only from a random subset of features. In that case, more diversification and lower correlation will be across trees which can make the accuracy higher.

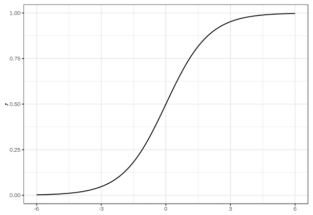


[12]

For these reasons, random forest is the first choice we have for the loan appeal problem.

The second model is logistic regression which is suitable for true and false questions that apply to the loan appeal problem. Logistic regression does not export a percentage, it presents the case as numbers (0 and 1) and fit the best hyperplane that minimizes the distance between the points and the hyperplanes.[14] The logistic function is defined as:

$$\text{logistic}(\eta) = \frac{1}{1 + exp(-\eta)}$$



[14]

We can model the relationship between outcome and feature with a linear equation:

$$\hat{y}^{(i)} = \beta_0 + \beta_1 x_1^{(i)} + \ldots + \beta_p x_p^{(i)}$$

[14]

L2 regularization is used in this Logistic Regression method that prevents overfitting. L2 is based on the function below:

$$J(w) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} log(h(x^{(i)}) \; + \; (1 - y^{(i)}) log(1 - h(x^{(i)}) \;]$$

[15]

In conclusion, we have two models applied to the loan applications problem.

2.2 Feature Creation

After experimenting with the models and getting the initial results we started to think of some questions. First, does the predicting system equally inspect every application? In other words, Do we see any discriminations in the dataset? Second, do we have a sufficient amount of features in the dataset? The answer to both questions is no. For the first question, the below charts reveal some facts that the applicants who graduated from college have a much higher approval rate and the male applicants have a higher approval rate than the female applicants. As for the

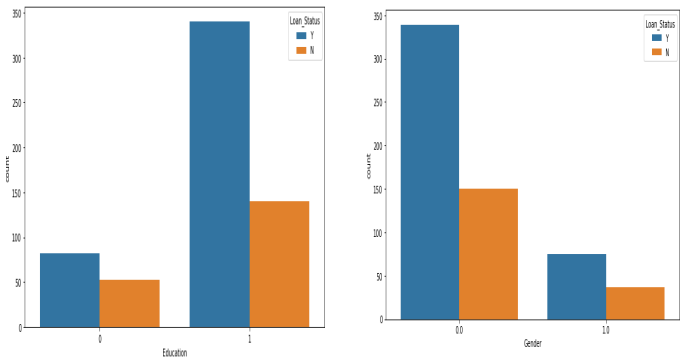


Chart: showing the feature distribution among Education and Gender

The second question, after viewing the charts illustrated above, our curiosity got derived. What if we want to specify the feature further and know the more concrete applicant's characteristics such as a female candidate who did not graduate from college? It turns out we need the strategy called feature creation to solve these issues.

Feature creation is the process of constructing features that do not already exist in the dataset. The goal of this task is to create as many features as possible to let the system have a more detailed view of each applicant. In the meantime, the more detailed features can help the system classify the eligibility of the candidates and reduce discriminations at some points. According to the research[17], we ideally need the size of the dataset / 20 amount of features. Since the size of our dataset is close to 700 we are supposed to expand the number of features to 30-35.

This approach is necessary first because the loss of features is likely to result in bias issues. For example, both male and female applicants who have the same educational background, receive a similar salary, applied for the same amount of loan etc receive different loan outcomes. If without any further features to classify the candidates we can conclude here that this product does not give an applicant the same chance to receive the applied amount. Secondly, this approach saves the costs from collecting the data. It is not realistic to ask all the 1000 applicants one more time to hand in additional information. Instead, we are planning to bind feature attributes to a new feature. For example, the 17th feature can be created by combining "gender" and "education" features.

From the data distribution chart shown below, we can see the features have a different level of impact on the outcome. For example, the applicant who has a good credit history has an apparent higher chance to acquire the financial loan plan. It provides us with a breaking point to think of the features. If we can improve the efficiency of features are the model going to have better accuracy? This question leads us to the feature selection.

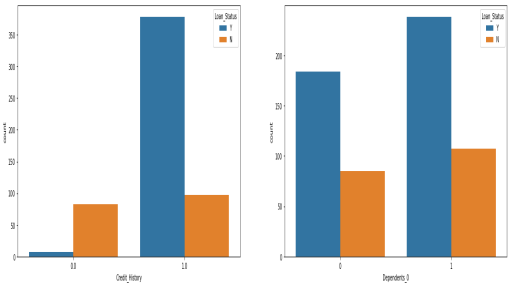


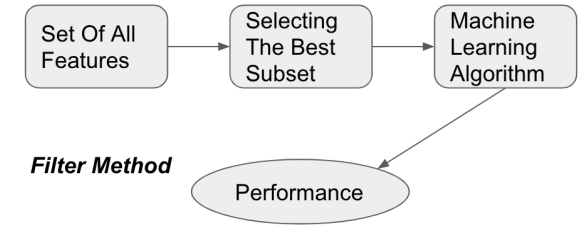
Chart: showing the feature distribution among History Credit and Dependents level

2.3 Feature Selection

Feature Selection refers to the process of reducing the inputs for processing and analysis, or of finding the most meaningful inputs. I raised this idea under Dr.Nishant's advice and developed a further understanding by reading a research article [18]. The feature selection approach in our project collaborates with the feature creation. The feature

selection methods provide us with a reference when we attempt to create more features. And the feature selection method also helps reduce the feature dimensionality. The motivations of dimension reduction are to remove redundant features, irrelevant features and noisy features.

The flow chart below shows the basic techniques applying for the **Filter Method**. There are three components from the chart. Suppose we have all set of the features. From this set of features, we want to select the best subset where three primary techniques that can apply for selection (chi-square tests, ANOVA test and Correlation coefficient test). These statistical tools can be efficiently used in the small dataset and helps us to select some important features where the features are highly targeted with our outputs. However, one of the limitations of these tests is high time complexity but in our dataset with the size of 700, the techniques work well. After selecting the best subset we are about to run the algorithms(Univariate Selection, Feature Importance and Correlation Matrix with Heatmap) that analyze the features and help us select the best subset.



Flow Chart: Filter Method

We use the following three algorithms to select features.

Univariate Selection
The univariate selection method is used to select the independent features that have one of the strongest relationships with the output variable. Specifically, By the use of DataFrame, we can visualize the outcomes by listing the scores of each independent feature. The higher score stands for the higher importance of the feature is. The scikit-learn library provides the SelectKBest class with us to implement this method. The technique behind this method is the chi-squared (chi^2) statistical test. We can select the top 10 of the best features from the dataset.

Correlation Matrix with Heatmap
Once we completed a combination of the features. The dataset is now composed of both independent and dependent features. As we discussed above, the univariate selection method is about analyzing the independent features. The Correlation Matrix is to state the correlation between the dependent features. The motivation of using this method is to further increase the number of features and increase the dependency of the features. The code implementation is completed by the use of the seaborn library, specifically the sns class from the seaborn. We use the Heatmap to visualize the correlation between each feature. The correlation value ranges from -1 to 1. Any number greater than 0.6 is considered an optimal combination to select and create a new feature.

Feature Importance
We can get the importance of each feature((both independent and dependent) by using the feature importance property of the model. This method is the last step of feature selection. We ran this algorithm after having gotten a bunch of features. The method filters feature based on their importance. The higher score means the more important or relevant is the feature towards our label. The code was done by the use of ExtraTreesClassifier and then we used matplotlib to visualize the outcomes. We used ExtraTreesClassifier to get the top 30 features from the dataset.

2.4 Roc curve and AUC

The performance of the binary classifier task can be evaluated by the ROC curve. The ROC curve compares the relationship between the Recall (true positive rate) and Precision (false positive rate) of the model based on the basis of the confusion matrix*(Table 0)*. [1] The ROC curve x-axis is represented by FPR, and the y-axis is represented by TPR. ROC curve is the curve relationship between the value of TPR and FPR as the threshold increases. TPR as formula*(1)* means among all samples with the true category of 1, the proportion of predicted category of 1, which means that the rate for making the correct prediction. FPR as formula *(2)* means among all samples with a true category of 0, the proportion of predicted category of 1, which means that the rate for making the wrong predictions.

The area under the ROC curve(AUC) can be calculated from *(3)*. AUC range is from 0.5 to 1.0, the larger the AUC value, the better classifier performance.

predicted value / observed value	positive	negative
positive	TP (true positive)	FN (false negative)
negative	FP (false positive)	TN (true negative)

table 0: confusion matrix

$$TPR = \frac{TP}{TP+\mathrm{FN}} \quad (1) \qquad FPR = \frac{FP}{FP+\mathrm{TN}} \quad (2) \qquad AUC = \frac{1}{2}\sum_{i=1}^{m-1}(x_{i+1}-x_i)\bullet(y_i+y_{i+1}) \quad (3)$$

3. Experiment

3.1 Feature Engineering

First, we preprocessed the data. We found three CSV files for the training set, the test set and one file containing the data IDs in the data set and all their corresponding labels. Considering that the ratio of the original training set to the test set is 7 and 3, we want to have a higher proportion of the training set, so we integrate the three datasets into one dataset, and then split it into a new dataset according to the proportions of 8 and 2 for the new training set and test set. To process the data more conveniently, we first map all the features and label values to numeric types. Because there are null values in the data set, we use the mean completer method to fill in the null values. Specifically, if the feature of the null value is numeric, the mean value of the feature in all other objects is used to fill in the missing value. If the null value of a feature is non-numeric, the missing value is filled in with the most frequently occurring value in the feature. For example, we used 0 (for male) to fill in the missing values in the sex feature, because males appear more frequently than females.

3.2 Early Stage Model Results

We use the *sklearn* package in *python* to implement the random forest and logistic regression models of the original data set. As the result has been shown in *figure 2*, the subfigure above shows the effect of different number decision trees on the accuracy of the random forest model. Through comparison, the best result found for the random forest is the highest accuracy of 74% with a decision tree number of 120. The subgraph below shows the accuracy change of the logistic regression result model corresponding to the decrease in the degree of regularization. It can be seen from the plotted graph that when the regularization degrees are large at the beginning, the accuracy of the training set and the test set are both low, so the model is underfitting. As the degree of regularization gradually decreases, the accuracy of the test set reaches the highest value, and then decreases, when the model is overfitting. The optimal accuracy for logistic regression is 74% with the parameter of regularization of 0.22. In terms of logistic regression, we use the **k-fold cross-validation** strategy to handle the **hyperparameter tuning issue.**

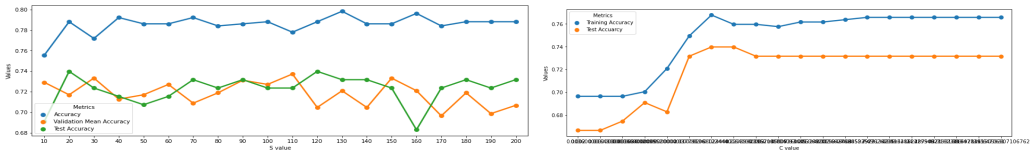


figure 1: the subfigure above shows the result of the random forest model, the subfigure below show the result of the logistic regression model

3.3 Univariate Selection

Before creating any of the features, we took a look at the existing dataset and noticed the existing features are all independent. Therefore, we decided to use univariate selection techniques to list the importance of each feature and then combine some of them based on priority. It turns out the number of features in our dataset is expanded to 35. For instance, in this example, the 17th feature can be created by the combination of feature 6 and feature 7.

	feature	Score
0	1	0.162407
1	2	1.534292
2	3	0.988390
3	4	0.007285
4	5	93.904964
5	6	11342.041603
6	7	3.829885
7	8	0.661295
8	9	26.014804
9	10	0.010509
10	11	0.768400
11	12	1.996446
12	13	0.384200
13	14	0.783946
14	15	7.103093
15	16	4.410584

feature	Score
6	11342.041603
5	93.904964
9	26.014804
15	7.103093
16	4.410584
7	3.829885
12	1.996446
2	1.534292
3	0.988390
14	0.783946

Table1: showing the importance of each feature

3.4 Correlation Matrix with HeatMap

After having created the dependent features we want to explore the correlation of the dependent features. The motivation for doing this is to increase the feature dimension and make the newly created features describe the candidates more detailed and concisely. From the map, we can see that the map illustrates the correlation between two features and gives a correlation value between -1 to 1. Since the bigger number refers to a more closed correlation we hence create more features based on the correlation value. We combine feature 23 with feature 26 and create a second feature. The same idea is applied to the following feature creation. As a result, we further expanded the number of features to 50 (table 2).

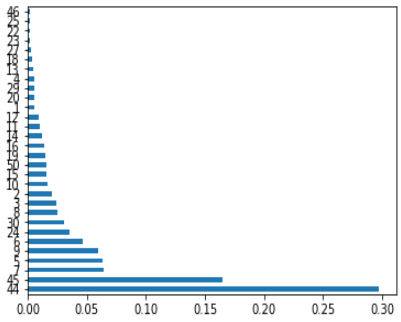
Heatmap shows the correlation between dependent features

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
1	0	0	1	0	5849	0	0	360	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
2	0	1	1	0	4583	1508	128	360	1	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
3	0	1	1	1	3000	0	66	360	1	1	0	0	0	1	0	0	1	0	1	0	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	0	0	0	0	1
4	0	1	0	0	2583	2358	120	360	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
5	0	0	1	0	6000	0	141	360	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
610	1	0	1	0	2900	0	71	360	1	1	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
611	0	1	1	0	4106	0	40	180	1	0	0	0	1	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1
612	0	1	1	0	8072	240	253	360	1	0	1	0	0	1	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1
613	0	1	1	0	7583	0	187	360	1	0	0	1	0	1	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1
614	1	0	1	1	4583	0	133	360	0	1	0	0	0	0	1	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0

Table2: showing all the feature after the creation

3.5 Feature Importance Method

This step is to reduce the feature dimension. Since the dataset is composed of dependent and independent features we decided to use the feature importance method to extract the top 30 features and visualize them by the bar graph. The X-axis is the score based on the importance and the y-axis represents each of the features. The importance of the feature is listed from the bottom to the top of the y-axis. Since it will be messy if we expand each feature's full name we decided to use a number to represent each feature. As a result, we get the best subset from the existing dataset features.



The Graph shows the top30 features in the dataset

3.6 Roc curves and Results

Figure 2 shows the model results of the dataset after feature selection and feature creation. For the random forest model, the highest accuracy is 83% with 50 decision trees. It can be seen that the accuracy is improved by 12% compared to the original dataset. For the logistic regression model, and the accuracy is 80 with optimal parameter 78, the accuracy is increased by 8%. In summary, the performance of the two models has been improved.

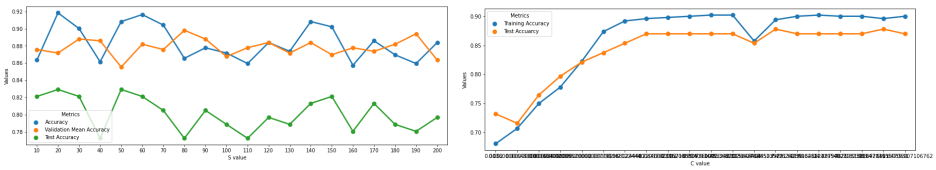


figure 2: the subfigure above shows the result of the random forest model,
the subfigure below show the result of the logistic regression model

Next, we further observe and compare the performance of our model results. Considering that the problem we are solving is a typical binary classification task. We decided to use the receiver operating characteristic curve (ROC curve) to test the quality of the generative model. We used the *sklearn.metrics* package in python to plot the ROC curve of the random forest model and logistic regression model, as shown in *figure 3*. The two graphs above are the results of using the original data set to generate the model (before feature selection and feature creation), and the two

graphs below are the results of the optimized data set (after feature selection and feature creation). To observe the performance of the model more intuitively, we computed the Area Under Curve (AUC) of each ROC curve, which is marked in the lower right corner of each subfigure. It can be seen that the AUC of the random forest model and the logistic regression model based on the original dataset are 0.71 and 0.74, respectively. Compared with the models generated by the optimized data set, which are 0.9 and 0.91 respectively, both models have been greatly improved (26.7% and 22.97%).

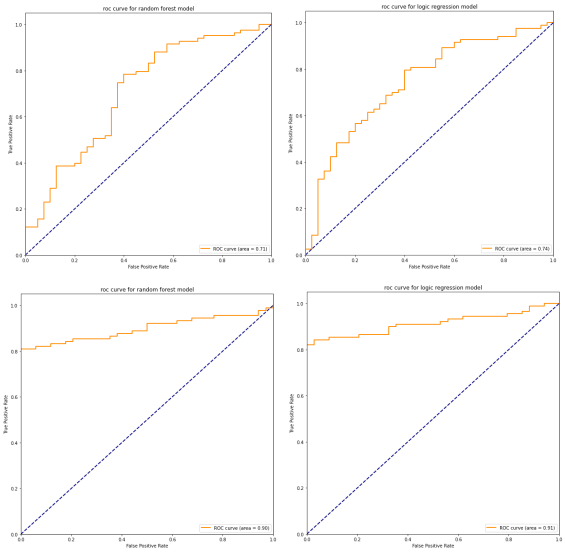


Figure 3: two subfigure above represent roc curve of random forest and logistic regression result using original dataset, subfigures below represent roc curve of random forest and logistic regression result using the optimized dataset

3.7 Discussions and Continued Work

During the experiments of the feature selection, we first observed that the dependent features are less important and relevant than the independent features. Our assumption is independent features that are coming from the dataset have more variety in terms of the output. The combinations of features are like splitting the old feature into pieces of more concrete features. As a result, the creation is less powerful than the base feature. However, the subfeature is more precise and specified so it increases the predicting accuracy. Secondly, our dataset is composed of dependent and independent features. We have not found the issue caused by overlapped features but it does not mean there are no potential issues. This is an open discussion, we want to know whether the overlapped features would create issues in the model training.

We think there is future work needed to do to improve the quality of this project. The work is that we are going to include a model selection section to let the probabilistic measure(AIC) tell us how to select the model.

4. Conclusion

From the information we got from experiments, there are several conclusions about the logistic regression and random forest models. In the ROC curve, both models get about 90% accuracy on the dataset, but the random forest is more stable than logistic regression which both early-stage and final stage, random forest model does not need as much data as logistic regression to reach a higher level of accuracy. At the same time, logistic regression always takes more time to train than random forest when we are doing experiments.
On the side of the dataset, the original 12 features are not enough for training in both random forest and logistic regression. Increasing the feature number is essential for our experiments which deeply changes the accuracy of both models.
In conclusion, the final stage of the experiment overestimates the expectation which is about 75% percent. Data training seems to be an important category in the future, and computers can simulate the result before things happen someday.

Reference

[1] Datahack.Analyticsvidhya.Com, 2016, https://datahack.analyticsvidhya.com/contest/practice-problem-loan-prediction-iii/#LeaderBoard.

[2] Data Science Stack Exchange. 2020. *Is feature selection necessary?*. [online] Available at: <https://datascience.stackexchange.com/questions/16062/is-feature-selection-necessary> [Accessed 17 February 2021].

[3] Kaggle.com. 2014. *Loan Default Prediction - Imperial College London | Kaggle*. [online] Available at: <https://www.kaggle.com/c/loan-default-prediction/data> [Accessed 17 February 2021].

[4] Kevin P. Murphy. 2012. Machine Learning: A Probabilistic Perspective. The MIT Press

[5]A study on predicting loan default based on the random forest algorithm, 2019, https://www.sciencedirect.com/science/article/pii/S1877050919320277

[6] Loan Prediction using Decision Tree and Random Forest, 2020 https://towardsdatascience.com/classifying-loans-based-on-the-risk-of-defaulting-using-logistic-regression-9bd9c6b44640

[7] Classifying Loans based on the risk of defaulting, 2019 https://towardsdatascience.com/classifying-loans-based-on-the-risk-of-defaulting-using-logistic-regression-9bd9c6b44640

[8] Loan default prediction using decision trees and random forest: A comparative study , 2021 https://iopscience.iop.org/article/10.1088/1757-899X/1022/1/012042/pdf

[9]LOGISTIC REGRESSION BASED LOAN APPROVAL PREDICTION, 2020 http://www.jctjournal.com/gallery/39-may-2020.pdf

[10]Loan Predictions with Logistic Regression, 2018 https://www.kaggle.com/jsingh93/loan-predictions-with-logistic-regression

[11]Loan Prediction Using selected Machine Learning Algorithms, 2019 https://medium.com/devcareers/loan-prediction-using-selected-machine-learning-algorithms-1bdc00717631

[12]Understanding Random Forest, 2019 https://towardsdatascience.com/understanding-random-forest-58381e0602d2

[13] Logistic Regression — Detailed Overview, 2018 https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc

[14]Interpretable machine learning, 2021 https://christophm.github.io/interpretable-ml-book/logistic.html

[15]implement logistic regression with L2 regularization from scratch in Python, 2020 https://towardsdatascience.com/implement-logistic-regression-with-l2-regularization-from-scratch-in-python-20bd4ee88a59

[16]Kotsiantis, S.B., Zaharakis, I.D. & Pintelas, P.E. Machine learning: a review of classification and combining techniques. *Artif Intell Rev* 26, 159–190 (2006). https://doi.org/10.1007/s10462-007-9052-3

[17] Moradi, S. & Mokhatab Rafiei, F. Financ Innov (2019) 5: 15. https://doi.org/10.1186/s40854 -019-0121-9

[18]Correlation-based Feature Selection for Machine Learni, Mark A. Hall, 1999
https://www.cs.waikato.ac.nz/~ml/publications/1999/99MH-Thesis.pdf