ISSN 2278-3091

Volume 9, No.4, July – August 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse328942020.pdf

https://doi.org/10.30534/ijatcse/2020/328942020

Machine Learning in Consumer Credit Risk Analysis: A Review

Jae Young Seo[1], Samuel Mahatmaputra Tedjojuwono[2]

[1]Business Information Systems Program, Information Systems Department, Faculty of Computing and Media, Bina Nusantara University, Jakarta, Indonesia 11480

[2] Computer Science Department, Faculty of Computing and Media, Bina Nusantara University, Jakarta, Indonesia 11480, e-mail: samuelmt@binus.ac.id

ABSTRACT

Consumer Credit Risk Analysis is an important factor in financial institution as it would be possible to lend credit to only to consumers that have good credit. Besides using traditional methods such as credit scoring, machine learning can also be used as a tool to classify applications for credit. Many research papers had tested and concluded of the ability of machine learning methods to classify good and bad credit. From the previous papers it was concluded that Ensemble Method, Support Vector Machine and MLP (Neural Network) showed the best accuracy in credit risk. However those previous papers compared only a few algorithms so it is difficult to determine which algorithm has the best performance. Therefore, this paper will compare the algorithms from previous paper by applying the German dataset to the algorithms as it is the most common dataset used for credit risk analysis. The algorithms compared are Logistic Regression, Linear Discriminant Analysis, Support Vector Machine, Naïve Bayes, K-Nearest Neighbors, Decision Tree, Ensemble Methods (random forest, bagging, boosting) and MLP. The performance will be measured using accuracy, average accuracy from 10 fold cross validation, AUC, Sensitivity, Specificity, Precision rate, F1. This paper will focus to find the most accurate algorithm for credit risk analysis. From the result it was found out that Logistic Regression, Support Vector Machine and Bagging had a good result. However, since the scores were dispersed, it was difficult to conclude which algorithm had the best accuracy. At the end SVM was chosen as it was deemed to be the most accurate algorithm for credit risk analysis

Key words : credit risk analysis, logistic regression, support vector machine, ensemble method

1. INTRODUCTION

Financial institutions deems consumer credit risk analysis as a crucial factor as it helps to eradicate consumers that are unsuitable for lending. As a traditional method, consumers are 'scored' using the credit scorecards. These score cards statistically analyzes a person's credit worthiness and helps financial institution to extend or deny credit. With this score that is obtained, consumers will be classified as 'good' as in that the consumer will be able to pay off the loan or 'bad' as the consumer does not have the ability to return the loan.

Machine Learning and Deep learning are approaches for data analysis and prediction that automate conventional analytical models. It is a sub domain of Artificial Intelligence based on fact that machines can learn from data, machines can identify patterns, extract key features for decision making with a minimal human intervention [1]. Machine Learning includes the investigations which will find data which is systematic and purposeful data structures acquired from disordered information and summarizing it into helpful data and it very well may be utilized for smart business decision [2]. Machine learning methods can also use for credit risk analysis. Binary classification methods can be used to classify consumers that has the ability to return the loan and consumers that doesn't have the ability to return the loan. Many authors had experimented machine learning techniques on various real credit risk datasets. The algorithms were feed of the datasets and measured of their performances. The algorithms that were tested were logistic regression, linear discriminant analysis (LDA), support vector machine (SVM), naïve bayes, k-nearest neighbors (KNN), decision tree, ensemble methods and neural networks. These authors only compared 3-4 algorithms and some only showcased one algorithm for credit risk analysis. Therefore, this paper will review all the algorithm above to find out the best performing algorithm in credit risk context.

2. RELATED WORK

2.1 Credit Risk Analysis

Machine learning methods has been widely proposed and used in credit risk analysis. Many authors have tested, analyzed and compared different types of machine learning algorithms. From those, some papers were able to find out ensemble method had the better performance comparing to other methods that were tested. This research tested big data with deep learning models and ensemble method (random forest and boosting). It was concluded that the two ensemble methods showed better performance [3]. This is also evident in another research where seven different datasets were analyzed using nine different algorithms. Logistic Regression, Bayesian network and ensemble method showed the best results [4]. Another research compared ensemble

methods with neural networks on a Taiwan bank credit dataset. It was evident that boosting, an ensemble method, showed the best performance [5]. It was also evident in another research compared few techniques that can be suitable for imbalanced credit scoring data which were Benelux institution dataset, German credit dataset and Australian credit dataset. By comparing the techniques, it was concluded that random forest and gradient boosting performed well in imbalanced dataset [6].
In other cases where ensemble method was not tested, SVM was found to be the best performing algorithm. This research compared Logistic Regression, LDA, SVM and KNN of their performance to process a dataset that had 25,000 data. It was concluded that SVM showed better result in comparison to other methods [7]. In another research paper that compared Logistic Regression, SVM, MLP (neural network) and decision tree, it was found out that SVM had the best performance in classifying the German credit dataset [8].

Other papers solely tested on the performance of neural networks on credit dataset and was able to conclude that changes in parameters can increase the performance rate. This research found out that Artificial Neural Networks performs efficiently in processing credit applications and are able to improve by altering learning schemes and parameters [9]. Another research paper supported the idea where MLP Neural Network was applied on four different dataset and examined the effect of different ratios of training to testing dataset to be able to find the best ratio [8]. It was also supported in another research paper where it was analyzed that MLP Neural Network shows high accuracy in credit rating and also that increases in hidden unit may decrease the error rate of the model [11]. Similarly, another research paper tested the parameters of MLP but also compared it to Logistic Regression and Decision Trees. It was concluded that MLP performs better than the other algorithms [12]. Table 1 describes the machine learning algorithms that were used in the other papers.

Table 1: Overview of the methods used in other papers

Algorithm	
Logistic Regression	[2],[5],[10],[6]
LDA	[5],[4]
SVM	[2],[5],[6]
KNN	[2],[5]
Naïve Bayes	[2]
Decision Tree	[2], [4], [6]. [10]
Ensemble method	Random forest: [1],[3] Gradient Boosting: [1], [2] Bagging: [3]
NN	ANN: [3] , [7] MLP: [6], [8], [9], [10]

Table 1 describes the machine learning algorithms that were used in the other papers.

Table 2: The performance measurements of other papers

			Accuracy			AUC			
Method		Dataset	training	testing	Precision	training	testing	Sensitivity	Specificity
LR	Addo, Guegan & Hassani (2018) [1]	Bank B				0.84293	0.87628		
	Bellioti & Jonathan (2009) [5]	Bank A				0.779			
	Mohammadi & Zangeneh (2016) [10]	German	75.9			0.792		0.87	0.5
	Peng et. al(2010) [2]	German	77.71		0.6578	0.7919		0.4933	0.89
	Marqués, García, & Sánche (2012) [6]	German	75.70						
LDA	Bellioti & Jonathan (2009) [5]	Bank A				0.781			
SVM	Peng et. al(2010) [2]	German	77.4000		0.6667	0.6938		0.4933	0.8943
	Bellioti & Jonathan (2009) [5]	Bank A				0.783			
	Marqués, García, and Sánche (2012) [6]	German	76						
KNN	Peng et. al(2010) [2]	German	66.9000		0.4485	0.6064		0.45	0.7629
	Bellioti & Jonathan (2009) [5]	Bank A				0.765			
Naïve Bayes	Peng et. al(2010) [2]	German	75.5000		0.6104	0.7888		0.5067	0.8614
Decision Tree	Peng et. al(2010) [2]	German	71.9000		0.5388	0.6607		0.44	0.8386
	Marqués, García, and Sánche (2012) [6]	German	73.3000						
	Mohammadi & Zangeneh (2016) [10]	German	70.7			0.641		0.8429	0.4
Ensemble method	Peng et. al(2010) [2]	German	76.2		0.6476	0.798		0.4533	0.8943
	Random Forest (Addo, Guegan & Hassani (2018) [1]	Bank B				0.99327	0.99306		
	Random Forest (Hamori et.al (2018) [3]	Taiwan	58.39	69.67		0.605			
	Gradient Boosting (Addo, Guegan & Hassani (2018)) [1]	Bank B				0.99420	0.99480		
	Bagging (Hamori et.al (2018) [3][3]	Taiwan	56.15	80.12		0.575			
	Boosting (Hamori et.al (2018) [3]	Taiwan	70.95	71.66		0.769			
NN	NN (Khasman (2010)) [7]	German	73.17	99.25					
	NN (Guotai, Adedin, Moula (2017)) [8]	German	80.46	78.85				0.1821	0.3053
	NN Tanh (Hamori et.al (2018) [3]	Taiwan	69.42	70.64		0.768			
	MLP (Zhao et. al (2015)[9]	German		87					
	MLP Marqués, García, and Sánche (2012) [6]	German	72.4						
	MLP (Mohammadi & Zangeneh (2016) [10]	German	78.4			0.829		0.883	0.5534

Table 2 describes the algorithms that were used in other papers and also their performance that is measured. For papers with multiple data set or only the German dataset, the German dataset was purposely chosen as it was also used in this paper to measure the performance of the algorithms. For the research papers with different dataset is because it did not use German dataset in their paper. Some measurement of accuracy (accuracy, precision, AUC, sensitivity and specificity) are left blank because the papers did not use those methods to measure the performance of the algorithms. In Table 1, in terms of the testing accuracy, neural network and ensemble method shows relatively high accuracy compared to other algorithms.

2.2 Logistic Regression

Logistic regression is a common algorithm that is used for binary classification. In this case y, the response variable, can take one of two possible value where 0 is bad credit and 1 is good credit. X is a column vector of M explanatory variables and $\pi = \Pr(y = 1|x)$ is the responsible probability to be modelled. The number of observations is represented by N. α is the intercept parameter and β^T contains the variable coefficients [13].The logistic regression equation takes the form:

$$\text{logit}(\pi) \equiv \log\left(\frac{\pi}{1-\pi}\right) = \alpha + \beta^T x \text{ ,}(1)$$

2.3 Linear Discriminant Analysis

Logistic Discriminat analysis approaches binary classification problems by assuming that the conditional probability density functions $p(\vec{x}|y = 0)$ and $p(\vec{x}|y = 1)$ are normally distributed with mean and covariance parameters $(\vec{\mu}_0, \Sigma_0)$ and $(\vec{\mu}_1, \Sigma_1)$ [12]. The equation to classify y= 0 becomes:

$$(x - \mu_0)^T \sum\nolimits_0^{-1} (x - \mu_0) - (x - \mu_1)^T \sum\nolimits_0^{-1} (x - \mu_1) < 2(\log(P(y = 0) - log(P(y = 1))) + log|\Sigma_1| - \log|\Sigma_0| \quad (2)$$

Linear discriminant analysis is obtained if the simplifying assumption is made that both covariance matrices are equal.

2.4 Support Vector Machine

Support Vector Machines is a non-probalistic binary linear classifier. In 1963, Vapnik initially proposed a linear SVM classifier. However, in 1992, a nonlinear classifier by applying the kernel trick was proposed by Bernhard E. Boser, Isabelle M. Guyon and Vladimir N. Vapnik [15]. As the dot product is replaced by nonlinear classifier kernel fuction, it allowed the algorithm to fit the maximum-margin hyperplane in transformed feature space [16].

The SVC (Support Vector Classifier) solves the following primal problem:

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=0}^{n} \zeta_i$$
$$subjects\ to\ y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i, \quad (3)$$
$$\zeta_i \geq 0, i = 1, \ldots, n$$

Its dual is:

$$\min_{\alpha} \frac{1}{2} \alpha^T Q\alpha - e^t \alpha$$
$$\text{subject to } y^T \alpha = 0 \quad (4)$$
$$0 \leq \alpha_i \leq C, i = 1, \ldots, n$$

e is the vector of all ones, $C > 0$ is the upper bound, Q is an n by n positive semidefinite matrix, $Q_{ij} \equiv y_i y_j K(x_i, x_j)$, where $K(x_i, x_j) = \phi(x_i)^T \phi(x_i)$ is the kernel. Training vectors are mapped into a higher dimensional space by the function ϕ.

The decision function is

$$(5) \qquad sgn(\sum_{i=0}^{n} y_i \alpha_i K(x_i, x) + \rho)$$

2.5 K-Nearest Neihbours

K-Nearest neighbors is a classification method which takes a majority vote of its k most similar data points to classify a data point[17]. Euclidean distance between the two points was used to measure the similarity. The equation is as follows:

$$d(x_i, x_j) = \|x_i - x_j\| = \left[(x_i - x_j)^T (x_i - x_j)\right]^{\frac{1}{2}} \quad (6)$$

2.6 Naïve Bayes

Naïve Bayes method is an algorithm based on applying Bayes' theorem with the "naïve assumption of independence between every pair of features [18]. As a class variable y and a dependent feature vector x_1 through x_n, Bayes theorem states the following relationship:

$$P(y|x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots, x_n|y)}{P(x_1, \ldots, x_n)} \quad (7)$$

In this paper, Gaussian Naïve Bayes algorithm was used to analyze the data. The equation is as follows:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \quad (8)$$

The parameters σ_y and μ_y are estimated using maximum likelihood.

2.7 Decision Tree

Decision Tree is a non-parametric algorithm that creates a model that predicts the value of a target variable by learning simple decision rules inferred from the data [19]. The tree algorithm used in this paper is CART (Classification and Regression Tree).

If a target is a classification outcome taking on values 0,1,…, K-1, for node m, representing a region R_m with N_m observation, the proportion of class k observations in node m is:

$$\text{Pmk} = 1/N_m \sum_{X_i \in R_m} I(y_i = k) \qquad (9)$$

2.8 Ensemble Method

Bagging

Bagging (Bootstrap Aggregating) creates several test data through the test data sampling (Bootstrap) and makes several weak learner using each test data [20]. Finally, the predictions of each learner are averaged and aggregated. The training algorithm of bagging defines as below:

Given a training set $\mathrm{X} = x_1, \dots, x_n$ with responses $\mathrm{Y} = y_1, \dots, y_n$, bagging repeatedly B times selects a random sample with replacement of the training set and fits trees to the samples:

For $\mathrm{b} = 1, \dots, \mathrm{B}$:

1. Sample with replacement, $\mathbf{n}$ training examples from X, Y; given X_b, Y_b
2. Train a classification tree f_b on X_b, Y_b

After training, by averaging the predictions from all the individual regression trees on $\mathbf{x}'$ or by taking the majority vote in the classification trees the predictions for unseensamples $\mathbf{x}'$ can be made:

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} fb(x') \qquad (10)$$

Random Forest

A random forest in machine learning is a kind of ensemble learning method used for classification and regression analysis. It was first introduced by Breiman in 2001. The model is one of the most effective algorithm to improve the shortcomings of decision trees. The training algorithm for random forests applies the technique of bootstrap aggregating (bagging) to the trees. Random forest applies the training algorithm of bagging but uses modified tree learning algorithm that selects a random subset of the feature [21]. For classification problems with p features, $\sqrt{p}$ features are used in each split.

Boosting

Gradient Boosting is an ensemble method that is a generalization of boosting to arbitrary differentiable loss functions [22]. It is an algorithm that has high predictive power and shows high robustness to outliers in output space. However, it is less scalable as the sequential nature of boosting is hard to be parallelized. The following equation is the additive models that Gradient Boosting considers:

$$F(x) = \sum_{m-1}^{M} \gamma_m h_m(x) \qquad (11)$$

2.9 Neural Network

Multilayer perceptron is a neural network with one or more middle layers (hidden layers) between the input layer and the output layer, and has a hierarchical structure.

A simple notation for a simple MLP is as follows:

$$o^0 = x,\ o^{'} = F^{'}\left(W^{'}\hat{o}^{('-1)}\right) for\ l = 1, \dots, L \qquad (12)$$

Given x, the input vector, that is set as the output of the zeroth layer. $\hat{o}^{('-1)}$ Represents an operation where a number 1 is prepended to a vector, increasing its dimension. This allows the bias terms of layer l to be written as the first column of matrix W'. Given F'as the activation function is applied to all components of a vector.

3. METHODOLOGY

This review paper used the German credit dataset that is publicly available in the uci machine learning repository. The dataset has 1000 real data with 21 features. The first 20 attributes are personal information and financial information that contained both numerical and categorical variables. The last attribute the approval of the applicant which was marked as 1 for approved and 2 for rejected. The dataset is imbalanced, just like other credit risk dataset. 70% of the application was classified as good credit and the rest 30% was classified as bad credit.

As there were categorical variables, attributes that can be ranked/rated from low to high were altered and given numerical values that matched their rank. For categorical values that weren't able to be rank such as gender, was altered to numerical values with one-hot encoding method. For the last attribute the values were changed to 0 and 1. 0 for bad credit risk and 1 for good credit risk.

The machine learning methods that were tested were logistic regression, linear discriminant analysis (LDA), support vector machine (SVM), naïve bayes, k-nearest neighbors (KNN), decision tree, ensemble methods (random forest, bagging, boosting) and mlp. The machine learning methods were coded in Jupyter Notebook with python2 using Sci-kit Learn as the library

Dataset was divided into training and testing dataset. As the default setting for splitting dataset in Ski-kit learn, the training dataset was 75% and the testing dataset was 25% of the previous dataset. Some algorithms used scaled dataset since there was it was overfitting. The algorithms that did not use scaled dataset were LR, LDA, and Naïve Bayes, as the accuracy dropped when scaled dataset was used.

In order to measure the performance of each machine learning method, few methods has been chosen. The overall accuracy was measured initially to see the performance of the algorithms. Since accuracy itself is not enough, other methods has also been measured. The average accuracy was obtained by 10 fold cross validation. AUC, Sensitivity/True Positive rate/ Recall, Specificity/ True Negative rate, Precision rate, F1 and RMSE was also measured to aid the measurement of the performance. The sensitivity measured the be the people who are unable to repay correctly specified as ‘bad credit’ and specificity measured be the people who are able to repay correctly classified as ‘good credit’.

Table 3: Overview results of the performance of the algorithms

Method	Training Accuracy	Testing accuracy	Average Accuracy*	AUC	Sensitivity	Specificity	Precision	F1	RMSE
Logistic Regression	77.5	**80.4**	74.4	0.788	0.931	0.507	0.815	**0.869147**	**0.443**
LDA	76.8	79.2	74	0.789	0.92	0.493	0.809	0.860937	0.456
SVM	79	79	**75.6**	**0.794**	**0.937**	0.453	0.8	0.863097	0.456
KNN	77	75	70.7	0.711	0.886	0.44	0.787	0.833571	0.498
Naïve Bayes	73	67	69.6	0.74	0.834	0.52	0.802	0.817687	0.51
Decision Tree	76.7	73.6	74.4	0.747	0.811	0.56	0.811	0.811	0.514
Random Forest	**99.3**	76.4	73.3	0.758	0.857	0.547	0.815	0.835472	0.486
Gradient Boosting	92.4	78.4	75.3	0.773	0.909	0.493	0.807	0.854969	0.465
Bagging	98.9	76.4	72.3	0.754	0.817	**0.627**	**0.836**	0.826390805	0.49
MLP	98.3	74.4	72.3	0.754	0.829	0.613	0.833	0.830995	0.486

4.EXPERIMENTAL RESULTS

It is shown form the data above (Table.3) that is relatively difficult to pick which algorithm gave the best performance. Logistic regression, LDA and Naïve Bayes was able to perform well without the data being scaled as the models are relatively simpler than the other models that were tested. Support Vector Machine and KNN performed relatively bad before the scaled data was feed to the models. More complex methods such as ensemble methods and MLP performed very well in the training but did not perform well in the testing. The scaled dataset relatively increased the measurements but there is still a big gap between the training and testing accuracy.

In terms of testing accuracy, logistic regression was able to achieve the best accuracy. However, in the 10 fold cross validation average accuracy, Support Vector Machine obtained the highest average accuracy of 75.6, followed by gradient boosting (ensemble method) of 75.3 percent. In addition, SVM showed the highest AUC of 0.794, which shows quite a big gap with LDA, the one that achieved the second highest AUC, which is 0.789. The highest sensitivity was achieved by SVM of 0.937, followed by linear regression, achieving a sensitivity of 0.931. This indicates that the model is able to classify the people who are unable to repay as ‘bad credit’, which is very important in financial institutions. However, SVM was not able to perform well in the Specificity as it achieved a score of 0.493 while Bagging was able to achieve the highest specificity which was 0.627 showing a big gap with the second highest scorer, MLP that received 0.613. In terms of precision,

Bagging received the highest ratings of 0.86. For F1 score and RMSE, logistic regression was able to receive the highest numbers of 0.869147 and 0.443 respectively.

5. CONCLUSION

Based on the results and findings, some conclusions could be made. This paper had compared 10 algorithms for credit dataset. In terms of testing accuracy, F1 score and RMSE, logistic regression received the highest numbers. Whereas SVM showed good performance in terms of the average accuracy, AUC and sensitivity which was 75.6, 0.764 and 0.443 respectively. Bagging was able to obtain highest numbers for specificity and precision which was 0.627 and

0.833 respectively. As the scores are dispersed it is difficult to conclude the best algorithm for credit risk analysis. However, in credit risk analysis it is very crucial to find consumers that has ‘bad credit’. In conclusion, SVM is the most suited algorithm for credit risk analysis as it has the highest average accuracy and it also has the highest numbers in Sensitivity.

REFERENCES

1. S.I. Manzoor, J. Singla. A Comparative Analysis of Machine Learning Techniques for Spam Detection, International Journal of Advanced Trends in Computer Science and Engineering, 8(3), 2019, pp. 810-814, 2019, https://doi.org/10.30534/ijatcse/2019/73832019
2. T. Ketha, S.S. Imambi, Analysis of Road Accidents to Indentify Major Causes and Influencing Factors of Accidents – A Machine Learning Approach, International Journal of Advanced Trends in Computer Science and Engineering, vol. 8, no. 6, pp. 3492 - 3497, 2019, https://doi.org/10.30534/ijatcse/2019/127862019
3. P. M. Addo, D.Guegan and B. Hassani, Credit Risk Analysis Using Machine and Deep Learning Models, Risk, 6(2), 2018, p. 38
4. Y. Peng, G. Wang, G.Kou and Y. Shi, An empirical study of classification algorithm evaluation for financial risk prediction, Applied Soft Computing, 11(2),2011, pp. 2906-2915
5. S. Hamori, M. Kawai, T. Kume, Y. Murakami, and C. Watanabe, Ensemble Learning or Deep Learning? “Application to Default Risk Analysis,” Journal of Risk and Financial Management, 11(1), 12, 2018
6. I. Brown and C. Mues, “An experimental comparison of classification algorithms for imbalanced credit scoring data sets,” Expert Systems with Applications, 39(3), 2012, pp. 3446-3453 https://doi.org/10.1016/j.eswa.2011.09.033
7. T. Bellotti and Crook, J, “Support vector machines for credit scoring and discovery of significant features,” Expert Systems with Applications, 36(2), 2009, pp. 3302-3308
8. A. I. Marqués, G. Vicente, and S. S. Javier, "Exploring the behaviour of base classifiers in credit scoring ensembles," Expert Systems with Applications 39(11), 2012, pp. 10244-10250
9. A. Khashman, "Neural networks for credit risk evaluation: Investigation of different neural models and learning schemes," Expert Systems with Applications 37(9), 2010,pp. 6233-6239.
10. C. Guotai, M. Z. Abedin, F. E. Moula, "Modeling credit approval data with neural networks: an experimental investigation and optimization," Journal of Business Economics and Management 18(2) ,2017, pp. 224-240.
11. Z. Zhao, S. Xu, B.H. Kang, M. M. J. Kabir, Y. Liu, and R. Wasinger, “Investigation and improvement of multi-layer perceptron neural networks for credit scoring,” Expert Systems with Applications, 42(7), . 2015, pp. 3508-3516
12. N. Mohammadi, M. Zangeneh, "Customer Credit Risk Assessment using Artificial Neural Networks," IJ Information Technology and Computer Science 8(3), 2016, pp. 58-66. https://doi.org/10.5815/ijitcs.2016.03.07
13. D.W. Hosmer and L. Stanley, Applied logistic regression 2nd edition, Wiley, Chichester, New York, 2000
14. W.N. Venables, Modern Applied Statistics with S 4th edition, Ripley, B. D, Springer Verlag, 2002, ISBN 0-387-95457-0.
15. A. Ben-Hur, D. Horn, H. T. Siegelmann and V. Vapnik, Support vector clustering. Journal of machine learning research, 2 Dec, 2001, pp. 125-137.
16. C.C. Chang, and C.J Lin, LIBSVM: a library for support vector machines. ACM transactions on intelligent systems and technology (TIST*)*, *2*(3), 27, 2011
17. T. Hastie, R. Tibshirani, and J. Friedman, “The elements of statistical learning, data mining, inference, and prediction,” Springer, New York, 2001
18. H. Zhang, "The optimality of naive Bayes." AA 1(2), 2004, 3.
19. L. Breiman, J. Friedman, R. Olshen, and C. Stone, Classification and Regression Trees, Wadsworth, Belmont, CA, 1984.
20. L. Breiman, "Bagging predictors". Machine Learning. 24 (2), 1996
21. L. Breiman, (2001). Random forests. *Machine learning*, *45*(1), 5-32.
22. J. Friedman, “Greedy Function Approximation: A Gradient Boosting Machine”, The Annals of Statistics, Vol. 29, No. 5, 2001. https://doi.org/10.1214/aos/1013203451