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/ijatcse194942020.pdf
https://doi.org/10.30534/ijatcse/2020/194942020

Performance Assessment of Combination in Stacking Ensemble Model for Credit Default Classification

Brenda Sylviasyah[1], Sani M.Isa[2]
[1]Bina Nusantara University, Indonesia, Brenda.Sylviasyah@binus.ac.com
[2]Bina Nusantara University, Indonesia, Sani.m.isa@binus.ac.id

ABSTRACT

Credit Default is one of the most discussed and reviewed problems in a financial institution. The ever-changing factors and variables towards the consideration of credit grant remains a challenge to prevent loss caused by non-performing loans. In the light of the machine learning era, one of the methods that can be applied to solve this problem is by using classification models. Ensemble Method is known for improving better model performance. This paper would focus on assessing the performance of various combination of 7 well-known classification algorithm such as SVC, Decision Tree (CART), Naïve Bayes, Logistic Regression, Random Forest, Extra Trees and XG-Boost.Out of using total 848combination of 1-7 algorithm with 7 meta classifier, this experiment shows that models from stacking ensemble does indeed generally perform better compared to single base-classifier. Assessing the performance between two credit datasets with different product type, the experiment concludes that the most ideal iteration is between 2-5 base-learner combination using SVC as the meta-classifier for this case. This study also suggests the usage of cost function for assessing credit classification problem for its ability to simulate a projection of loss and gain by implementation.

Key words : Stacking Ensemble, Credit Scoring, Cost Function

1. INTRODUCTION

A Credit by definition is a contractual agreement where the borrower receives something of value ‘now’, and agrees to repay the lender in the future with interest. This period gap results in a certain risks for the lender and prior assessment of grant needs to be thoroughly analysed in order to suppress the risk of the default.

The history of Credit assessment went back starting from the 50’s. Where the principal of 5C (Character, Capital, Collateral, Capacity and Condition) is introduced to become a standard assessment for credit approval. Credit score cards generated from various statistical models such as Kolmogorov-Smirnov to the well-known Logistic Regression are often used by various institution due to the simplicity and transparency. The only constraints on this era is only the large amounts of applicants and the assessment resource[1].

In searching for a higher accuracy and faster processing, current studies started to explore Machine Learning models which focus on credit scoring. Credit scoring itself is basically a supervised learning of binary classification problem to predict which applicant has the probability of becoming default. Classification algorithms such as Support Vector Machine Classification (SVC), Naïve Bayes (NB), Tree Models (Decision Tree[2], Random Forest, etc), to Neural Networks [3]such as MLP (Multi-layer Perceptron) has been reviewed and optimized by many researchers for a better performance.

In result, each method would have its own weakness and strengths, for example Logistic Regression is known to be the easiest method to apply, but also having risk of under fitting. SVM is also a very robust type of algorithm with the various kernel availability, yet it is known to be very sensitive for optimization and tuning[4]. Trees on the other hand is known for the most interpretable models but known for its instability for any changes of the data content.

Currently, the research is directed towards hybrid method and ensemble models. Hybrid method [5]is referring to optimization of a single algorithm combined with a feature selection to raise the performance. Ensemble models on the other hand is combining the results for a better result to raise performance and overcoming the weakness of the single algorithm result.

Even though there are many studies of comparing the performance between homogenous [6][7]and heterogeneous[8] algorithm combination in an ensemble[9][10], there are limited literatures which explore the relation of which base-learner types works the best or how many is the ideal amount of base-learner ratio for a good stacking ensemble. For that purpose, this experiment would focus on assessing the impact of base-learner type selection and number of base-learners used towards the stacking ensemble.

According to [4]there are three stages of modelling which can help raise the model credit scoring model performance, which is: Data Pre-processing, Classifier Selection and Classifier Ensemble which will also be applied to this experiment.

The dataset that would be used is a real financial data obtained from a credit institution. Credit scoring dataset are very well known for its imbalanced class ratio. To address this issue, data pre-processing steps that covers the data selection would be applied before moving on to the modelling phase.

For the modelling, this experiment would be using seven types of classifier which has been optimized individually as the base as well as the meta classifier, the list are as follows: Logistic Regression, Support Vector Machine Classifier, Decision Tree Classifier, Naïve Bayes, Random Forest, Extra

Trees and XGBoost. The model then would build a stacking combination which run iteratively between the combination of 1 to 7 selected classifier combination to find the best performance.

Instead of using normal performance metrics (i.e F1-score, Accuracy), the experiment result would later be assessed by using a scoring upon "Cost Function" Gain Ratio, which weights are a pre-consulted cost that can reflect the loss / gain projection for using the model in the company.

2. RELATED WORKS

This chapter is a summary of the results from previous researchers towards the credit classification problem using various ensemble method.

For Ensemble Algorithm,[10] compares the total of 25 individual classifiers in a credit scoring data set to see which algorithm performs better, but at this paper he did not introduce any multi-learners or boosting method. The result of this experiment is that MLP, Logistic Regression, Random Forest and Bagging performs better compared to the other.

There is also a research combining the most popular Logistic Regression algorithm with ensemble for credit scoring, which is combined with either bagging and boosting [6]and it turns out with LR+boosting wins with total of 81% accuracy compared to 78.3% of LR+bagging. Meanwhile LR has only 77% of accuracy.

Another research for credit scoring but in a hybrid model, combines the Decision Tree (C4.5) Algorithm with bagging ensemble [5], it shows with only using DT (C4.5) resulting in 73.1% accuracy, meanwhile with combining bagging ensemble raised it to 75.1% and it also proven the feature split provides quite a different range of accuracy, with split 3 (consisting 8 of 20 variables) proven the most best performing dataset.

Using not only ensemble but also a feature extraction method,[7] proposed a GDBT ensemble with Logistic Regression (LR) while also combined with the end performance of the feature extraction using Auto Encoder. The result is that GDBT_AE_LR and GDBT_LR has a head to head performance difference only ranging around 0.01-0.11 difference depending on the subtrees number.

PSO[11]is also becoming a trend of use for assigning weight on the base classifiers before entering it actually aggregated together to vote the final result. Where in the result he stated that of ensemble models (Table V) is better than that of single base-classifiers (Table III), confirming the benefit of using ensemble models. Among the ensemble models in comparison, Boosting produces the highest accuracy of 0.8462, but their proposed PSO-based ensemble model has the highest F-measure score of 0.8399.

Another approach is self-adaptive classifier ensemble model. [4]Proposes the selection of the base learner is evaluated adaptively with the input data. It used 9 different algorithms, with later picking the top 5 performers to ensemble. Where the result is also being multi-processed further with weight assigned to them performs significantly well compared to the other predecessors.

In conclusion, it was shown in the recent studies that the ensemble model became more popular due to its performing higher for Credit Scoring problems. Various ensembles involving various individual classifiers are experimented to gain a better accuracy[12]in this field. Based on the proposed method, the adaptive model does indeed manage to raise the performance for credit prediction or other fields. Using this as the reference, this experiment would focus on assessing the performance of stacking ensemble by comparing all of the possible combination with the goal of finding an ideal combination that may can be a reference in building credit models in this approach.

3. PROPOSED METHOD

Using reference of the latest research, this study would be carried on in several phase.

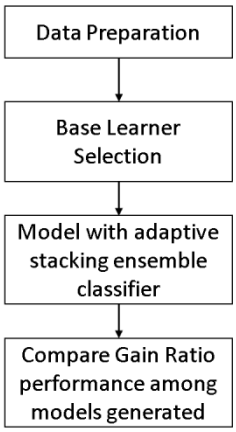

Figure 1: Research Method Stage

3.1 Data Preparation

The dataset is from a certain financial institution in Indonesia. The dataset is taken directly from the SQL database, with features regarding the loan information is selected by the basic coverage that can answer the 5C principal of lending. The dataset feature is such as follows:

Table 1: Feature list on the Credit Dataset

Field	Description
Account	Loan Account Number
Loan Period	Period of Loan
DownPayment	Down Payment %
Installment	Payment per month
Total Loan	Total Loan After Down Payment
Total Original Loan	Total Loan before Down Payment
CustType	Type of Customer
EconomyCode	Economy Code for Customer Job
RS	Existing Credit Checker Recommendation
Gender	Customer Gender
Job	Customer's Job

Job Title	Customer's Job Title
Marital Status	Customer's Marital Status
LOB	Line of Business
ZipCode	Customer Address
Gross Income	Income profile of Customer
Product Type	Collateral Type
Product Year	Collateral Manufactured Year
Dealer Group	Collateral Seller
Dealer Area	Collateral Seller Area
Loan Branch	Branch which gave the loan
Collectibility	Label Customer Current Default Status

This study would test upon two datasets from the same company but different loan products. The difference is only based upon the dealer channel and collateral category (there are no duplicate data among the dataset).

The first dataset would be tagged of as Dataset-U, and the second dataset would be tagged of Dataset-N. The data consists of 2 years credit dataset from 2018 to 2019 with the default position by 2020. By consulting the data analyst, we remove the loan booking date since it can provide bias towards predicting the default traits because the older the booking, it is more likely to show default status compared to the new bookings. Here is the composition of the dataset:

Table 2: Class composition on the Credit Datasets

Dataset Names	Non-Default	Default
Dataset-U	6412	1094
Dataset-N	8518	2128

To address the issue of imbalanced class, the training data selection can be performed with both under sampling or oversampling method. For the reference [13]focused on experimenting on how we can handle the data imbalance for classification purposes. The paper compared between using under sampling the majority class and oversampling the minority class by generating a synthetic sample from the minority class, such as using SMOTE methods. The result of the experiments turns out that unders ampling is much more suited to use in classification as it performs better compared to over sampling. Hence this study would also implement an under sampling method in the pre-processing step by taking randomized data from the non-default class with equal total sample as the default, making it balanced.

For handling the null values, since it was mostly from the job title and Zipcode, for these two features we referenced towards another feature to fill in the null. The job title values would be referenced towards the account's Job and economy code to fill in the values. For zipcode, it would be referenced towards the loan branch location.

The data afterwards is yet again assessed in the correlation matrix to remove two highly correlated features as shown on this figure, in this step, we learned that installment and Total Loan apparently are two highly correlated feature (Installment= Total loan / Loan Period), with correlation of [0.979] so we removed one of the feature.

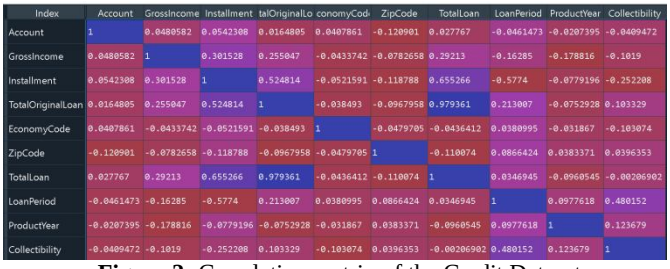

Figure 2: Correlation matrix of the Credit Dataset

As there are many categorical values in this feature list, upon assessment, apparently the variety on the unique category extended to more than a hundred per feature. Hence to convert our data into numerical values, label encoder is chosen instead of one hot encoding so that the number of features would not exponentially increase.

3.2 Stacking Model

Stacking ensemble is different with boosting and bagging which only use a single algorithm to ensemble among the results to gain a better performance. Stacking is an ensemble method which enables to make a prediction out of combination from various algorithm.

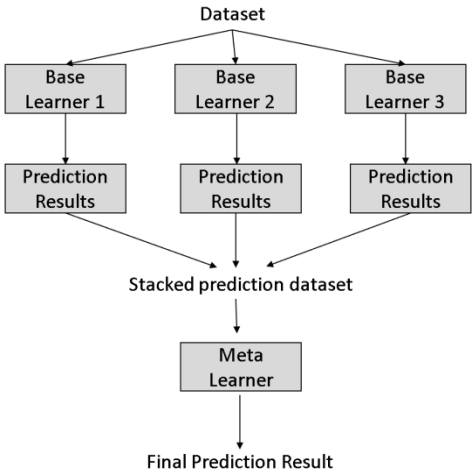

Figure 2: Stacking Heterogenous Ensemble Architecture

Method wise, stacking as heterogenous ensemble would require a new dataset which is comprised of the prediction result of more than a single base-learner, named the stacked prediction dataset from the fig.2.Afterwards they would be classified using the label from the original dataset using another learner, named meta-learner (classifier) to predict the result.

Since there has not been any reference of combination or numbers suggested for the stacking ensemble for credit default classification problem, in this study the proposed model is to make an iteration out of the combination with the following

schema:

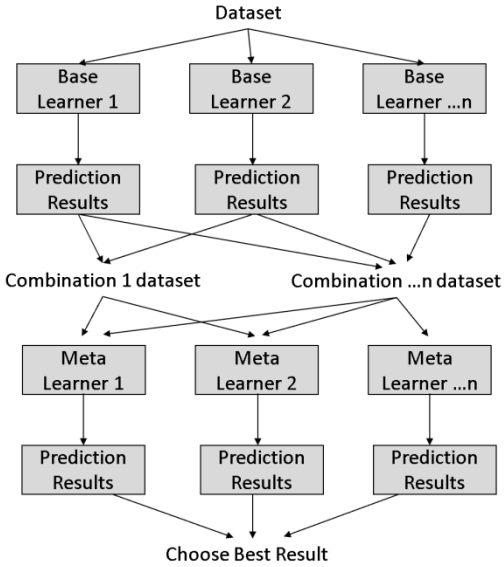

Figure 3: Stacking Ensemble Schema

Depending on how many numbers (n) of algorithm used as the base-learner, the model would iteratively combine the variation between one another in *all available combination*. Each combination would later be classified iteratively with each meta-classifier in the combination until r = max(n) has been reached, and later assess which model regarded as the best result depending on the evaluation metrics.

For this study would use 7 different commonly known algorithm, resulting in 848 mix of combination. We choose 2 base algorithm which based on decision boundary: Logistic Regression and SVM, 1 probabilistic classifier known as Naïve Bayes, 1 Decision Tree Classifier, and a mix of 3 ensemble type algorithm (tree based) which are Random Forests, Extra Tree and XGBoost.

Hereis the brief introduction towards the algorithm list that would be used among the combination:

Logistic Regression - The Logistic Regression (LR) is the most popular statistical model for credit assessment due to its easy interpretability and covers the transparency aspect needed in a credit scoring.

Support Vector Classification - Support Vector Machine (SVM) [14]is basically a decision boundary algorithm that generates a 'hyperplane'which separates classes based on the feature dimensional space.

Decision Tree - (DT) [15]is an algorithm that splits and branch out information based, until it arrives to a certain answer or for classification, we call it label. The most discriminating feature of DT is its root node, meanwhile the classes are known as leaf nodes.

Naïve Bayes - (NB) [15]classifier is based on Bayes decision rule. The adjective ''naive'' comes from the assumption that the features in a dataset are mutually independent. The Naïve Bayes classifier decides for class y=+1 over y=−1

Extra Trees - Extremely Randomized Trees Classifier(ET) is one of the ensemble learning technique which aggregates the results of multiple de-correlated decision trees collected in a "forest" to output it's classification result.

Random Forests - (RF) is another example of ensemble decision trees, i.e. where it generates multiple number of decision trees known as forest. These trees are built on bootstrapped samples with m observations and developed using a subset of randomly chosen k features. Each decision tree will give a class of a new feature vector. Afterwards, based on the trees made in the forest, it would make a 'voting' to decide upon a classification problem.

XGBoost–(XGB) [16] is short for extreme gradient boosting. The XGBoost is famously known for its processing speed and performance. It works similar to gradient boosting but builds the decision trees in parallel instead of building the decision trees in a series.

3.2 Cost Function

For the evaluation method, often other researchers would use between Accuracy, F1-Score or AUC [17]to get a better assessment of the imbalanced class. In this study, we realize a cost function may help as the evaluation method concerning this credit dataset. By assigning weights of loss and gain in each metrics, we can predict how much total profit will the model generate for the company after implementation, which is the background behind the base study of credit scoring.

Table 3: Cost Function for Credit Dataset

Metrics	Description	Loss/Gain Weight (w)
TP	Credible Customer, Predicted as Credible Customer	40
FP	Granting Default Customer a Loan, loss	-100
TN	Default Customer, Predicted Default Customer	0
FN	Lost Credible Customer	-40

Each result from the confusion matrix would be converted into score. The largest profit summed would be the case if the model managed to predict 100% accuracy. The profit is what the company gain if they enlist all credible customer to the credible class (TP+FN) x TPw, but no profit would be gained even if they managed to 100% predict default customers, hence setting it 0 instead of setting it to minus.

The Equation of the Gain Ratio would be as follow:

$$GR = \frac{((TP \times TPw) + (FP \times FPw) + (FN \times FNw) + (TN \times TNw))}{((TP + FN) \times TPw)}$$

4. EXPERIMENT AND RESULT

The result would be summarized into two separate table. The first table would be the single learner performance summary, while the second one would be the summary of the848-combination performance. The combination summary is grouped by the number of algorithms used, the number of unique combination and the meta-classifier used.

For the result itself it would be sorted based on the highest gain ratio of meta-classifier group, while listing the base-learner (n) in numerical order for easier comparison.F1 score and the accuracy would also be shown in two dimensions (average & max). In the case of two or more combination has the same top spot, the shorter combination would be considered as the leading performer as it is more efficient in terms of computation.

The following is the result of the model used upon the first dataset (Dataset-N):

Table 4: Dataset-N Base-Learners Performance Assesment

Algorithm	Gain Ratio	F1-Score	Acc
RF	**73.19%**	**78.51%**	**89.46%**
ET	72.02%	77.79%	88.98%
XGBoost	63.55%	72.21%	85.81%
DT	**62.35%**	**71.75%**	**85.25%**
SVC	57.70%	69.16%	83.43%
LR	56.00%	68.00%	82.84%
NB	45.69%	63.56%	78.60%

Table 5: Dataset-N Stacking Performance Assessment

Meta-Classifier	Base Learner(n)	Total Combination	Max Gain Ratio	Max F1-Score	Max Acc
SVC	2	21	73.19%	78.51%	89.46%
	3	**35**	**77.37%**	**81.00%**	**91.31%**
	4	35	77.37%	81.00%	91.31%
	5	21	73.64%	78.78%	89.65%
	6	7	77.37%	81.00%	91.31%
	7	1	72.85%	78.31%	89.31%
RF	2	21	73.19%	78.51%	89.46%
	3	35	74.80%	79.53%	90.12%
	4	35	75.65%	79.82%	90.62%
	5	21	73.40%	78.64%	89.55%
	6	7	73.22%	78.53%	89.47%
	7	1	72.85%	78.31%	89.31%
ET	2	21	73.19%	78.51%	89.46%
	3	35	74.25%	79.10%	90.04%
	4	35	75.37%	79.65%	90.49%
	5	21	74.66%	79.43%	90.06%
	6	7	72.85%	78.31%	89.31%
	7	1	72.85%	78.31%	89.31%
NB	2	21	73.19%	78.51%	89.46%
	3	35	73.19%	78.51%	89.46%
	4	35	73.28%	78.58%	89.49%
	5	21	73.47%	78.67%	89.58%
	6	7	73.02%	78.40%	89.39%
	7	1	72.85%	78.31%	89.31%
LR	2	21	73.19%	78.51%	89.46%
	3	35	73.19%	78.51%	89.46%
	4	35	73.19%	78.51%	89.46%
	5	21	73.19%	78.51%	89.46%
	6	7	72.85%	78.31%	89.31%
	7	1	72.85%	78.31%	89.31%
DT	2	21	73.19%	78.51%	89.46%
	3	35	73.19%	78.51%	89.46%
	4	35	73.19%	78.51%	89.46%
	5	21	73.19%	78.51%	89.46%
	6	7	73.19%	78.51%	89.46%
	7	1	73.19%	78.51%	89.46%
XGBoost	2	21	73.19%	78.51%	89.46%
	3	35	73.19%	78.51%	89.46%
	4	35	73.19%	78.51%	89.46%
	5	21	73.19%	78.51%	89.46%
	6	7	73.19%	78.51%	89.46%
	7	1	62.35%	71.75%	85.25%
Base Learner	7	7	73.19%	78.51%	89.46%

For the Dataset-N, we can conclude that the highest performer is the combination of 3 Base-learner Algorithm which is meta-classified by the SVC with the total of 77.37% of gain ratio, 81% of F1-scoreand 91.31% accuracy, raising total 15% compared to the Decision Treeclassifier and 4% raise from the performance of the Random Forest Ensemble Algorithm. The combination which performs the highest is the ensemble of [SVC-DT-RF].

Following the same format, below is the result of testing the second dataset (Dataset-U):

Table 6: Dataset-U Base-Learners Performance Assesment

Algorithm	Gain Ratio	F1-Score	Acc
RF	**72.44%**	**70.95%**	**88.30%**
DT	**69.87%**	**68.68%**	**87.29%**
ET	69.18%	68.66%	86.89%
XGBoost	68.32%	67.84%	86.57%
LR	63.13%	63.79%	84.48%
SVC	61.80%	63.60%	83.79%
NB	54.69%	59.61%	80.74%

Table 7: Dataset-U Stacking Performance Assessment

Meta-Classifier	Base Learner (n)	Total Combination	Max Gain Ratio	Max F1-Score	Max Acc
SVC	2	21	71.52%	70.23%	87.92%
	3	35	71.87%	70.56%	88.05%
	4	35	74.47%	72.10%	89.28%
	5	**21**	**76.69%**	**73.89%**	**90.23%**
	6	7	71.12%	70.06%	87.72%
	7	1	71.12%	70.06%	87.72%
ET	2	21	76.69%	73.89%	90.23%
	3	35	73.28%	71.27%	88.74%
	4	35	71.52%	70.23%	87.92%
	5	21	74.56%	72.17%	89.32%
	6	7	71.58%	70.28%	87.94%
	7	1	71.12%	70.06%	87.72%
RF	2	21	76.38%	73.72%	90.07%
	3	35	71.52%	70.23%	87.92%
	4	35	74.15%	71.88%	89.13%
	5	21	74.84%	72.40%	89.44%
	6	7	74.55%	72.19%	89.30%
	7	1	71.12%	70.06%	87.72%
NB	2	21	71.52%	70.23%	87.92%
	3	35	71.87%	70.56%	88.05%
	4	35	72.60%	71.06%	88.37%
	5	21	72.54%	71.02%	88.34%
	6	7	72.54%	71.02%	88.34%
	7	1	71.12%	70.06%	87.72%
Base Learner	7	7	72.44%	70.95%	88.30%
LR	2	21	71.52%	70.23%	87.92%
	3	35	71.87%	70.56%	88.05%
	4	35	72.36%	70.90%	88.26%
	5	21	72.06%	70.70%	88.13%
	6	7	72.15%	70.75%	88.17%
	7	1	71.22%	70.13%	87.76%

DT	2	21	71.52%	70.23%	87.92%
	3	35	71.52%	70.23%	87.92%
	4	35	71.52%	70.23%	87.92%
	5	21	71.52%	70.23%	87.92%
	6	7	71.52%	70.23%	87.92%
	7	1	70.00%	68.79%	87.34%
XGBoost	2	21	71.52%	70.23%	87.92%
	3	35	71.52%	70.23%	87.92%
	4	35	71.52%	70.23%	87.92%
	5	21	71.52%	70.23%	87.92%
	6	7	71.52%	70.23%	87.92%
	7	1	70.00%	68.79%	87.34%

For the Dataset-U, the highest performer is the 5 algorithms [LR-SVC-DT-NB-RF-XGB] also with SVC meta-classifier, in total of76.69% gain ratio and 73.89% of F1score, and 90.23% accuracy, raising total performance of 4% from the highest base-learner ensemble classifier Random Forest and 6.69% from the Decision Tree classifier.

5. CONCLUSION

This experiment is carried for the motivation of assessing the selection of stacking ensemble adaptive models when performed on a real credit default dataset. By the result of this experiment, it is shown that for this dataset, the ideal number of unique base-learners for the stacking combination is between 2 to 5, as there is no further enhancement on the 6th and the 7$^{th.}$ Combination performance on both datasets as shown in this table.

Table 8: Dataset-N Stacking Combination Summary

Meta-Classifier	Gain Ratio (Max)	F1-Score (Max)	Acc (Max)	Best Combination
SVC	77.37%	81.00%	91.31%	3
RF	75.65%	79.82%	90.62%	4
ET	75.37%	79.65%	90.49%	4
NB	73.47%	78.67%	89.58%	5
LR	73.19%	78.51%	89.46%	2
DT	73.19%	78.51%	89.46%	2
XGBoost	73.19%	78.51%	89.46%	2

Table 9: Dataset-U Stacking Combination Summary

Meta-Classifier	Gain Ratio (Max)	F1-Score (Max)	Acc (Max)	Best Combination
SVC	76.69%	73.89%	90.23%	5
ET	76.69%	73.89%	90.23%	2
RF	76.38%	73.72%	90.07%	2
NB	72.60%	71.06%	88.37%	4
LR	72.36%	70.90%	88.26%	4
DT	71.52%	70.23%	87.92%	2
XGBoost	71.52%	70.23%	87.92%	2

Though there are no direct correlation between the total of base-learners used to the end result, the use of minimum of 1 ensemble method maybe encouraged to the mix, since all of the top 10 performer combination in each dataset always consists at least minimum of 1 *ensemble algorithm*as the base-learner.

For Meta-classifier, SVC is actually the most top performer in both experiment, with it actually placing as the top performer ranked 1 to 10 in the Dataset-N and placing 4 top places out of 10 for Dataset-U, followed by Random Forest (Tree Ensemble). Here is the following top 10 ranking of respective dataset.

Table 10: Dataset-N Top 10 Combination List

Classifier	Combination	Gain Ratio	F1-Score	Acc
SVC	[‘LR’, 'SVC', 'DT', 'NB', 'RF', 'XGBoost']	77.37%	81.00%	91.31%
SVC	['SVC', 'DT', 'RF']	77.37%	81.00%	91.31%
SVC	['SVC', 'DT', 'RF', 'XGBoost']	77.37%	81.00%	91.31%
SVC	['DT', 'RF', 'XGBoost']	77.37%	81.00%	91.31%
SVC	[‘LR’, 'SVC', 'DT', 'NB', 'ET', 'XGBoost']	76.85%	80.64%	91.10%
SVC	['SVC', 'DT', 'ET', 'XGBoost']	76.85%	80.64%	91.10%
SVC	['DT', 'ET', 'XGBoost']	76.85%	80.64%	91.10%
SVC	['SVC', 'DT', 'ET']	76.85%	80.64%	91.10%
SVC	[‘LR’, 'SVC', 'DT', 'RF']	76.04%	80.09%	90.78%
SVC	[‘LR’, 'SVC', 'NB', 'RF', 'ET', 'XGBoost']	75.77%	80.16%	90.51%

Table 11: Dataset-U Top 10 Combination List

Classifier	Combination	Gain Ratio	F1-Score	Acc
SVC	[‘LR’, 'SVC', 'DT', 'NB', 'RF']	76.69%	73.89%	90.23%
SVC	[‘LR’, 'SVC', 'DT', 'RF', 'XGBoost']	76.69%	73.89%	90.23%
ET	['DT', 'RF']	76.69%	73.89%	90.23%
SVC	[‘LR’, 'SVC', 'DT', 'ET', 'XGBoost']	76.38%	73.72%	90.07%
SVC	[‘LR’, 'SVC', 'DT', 'NB', 'ET']	76.38%	73.72%	90.07%
RF	['DT', 'ET']	76.38%	73.72%	90.07%
RF	['SVC', 'DT', 'NB', 'RF', 'XGBoost']	74.84%	72.40%	89.44%
ET	['SVC', 'DT', 'NB', 'RF', 'XGBoost']	74.56%	72.17%	89.32%

RF	[‘LR’, 'SVC', 'DT', 'NB', 'RF', 'XGBoost']	74.55%	72.19%	89.30%
SVC	['SVC', 'DT', 'RF', 'XGBoost']	74.47%	72.10%	89.28%

Using this method, we can also pinpoint which algorithms we should focus on `tuning for better performance.

For future study, since adaptive models sometimes known for their long iteration that may take toll on computation power and training time when performed on a large dataset. A study of finding the ideal method and sub-sample size from a certain amount of total data population which can project the accuracy when performed on the full dataset is perhaps also one of the possible options.

REFERENCES

[1] X. Dastile, T. Celik and M. Potsane, "Statistical and machine learning models in credit scoring:A systematic literature survey," *Applied Soft Computing Journal*, vol. 91, 2020.

[2] T. Darmawan, A. S. Birawa, E. Eryanto and T. Mauritsius, "Credit Classification Using CRISP-DM Method On Bank," *IJETER*, vol. 8, no. 6, 2020. https://doi.org/10.30534/ijeter/2020/28862020

[3] C. Edmond and A. S. Girsang, "Classification Performance for Credit Scoring using Neural Network," *IJETER*, vol. 8, no. 5, 2020. https://doi.org/10.30534/ijeter/2020/19852020

[4] S. Guo, H. He and X. Huang, "A Multi-Stage Self-Adaptive Classifier Ensemble Model With Application in Credit Scoring," *IEEE Access (Volume: 7)*, pp. 78549 - 78559, 2019.

[5] M. A. Muslim, A. Nurzahputra and B. Prasetiyo, "Improving accuracy of C4. 5 algorithm using split feature reduction model and bagging ensemble for credit card risk prediction," in *2018 International Conference on Information and Communications Technology (ICOIACT)*, Yogyakarta, 2018.

[6] A. Lawi, F. Aziz and S. Syarif, "Ensemble GradientBoost for increasing classification accuracy of credit scoring," in *4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*, Kuta Bali, Indonesia, 2017.

[7] C. Shuhui, Q. Wang and S. Liu, "Credit Risk Prediction in Peer-to-Peer Lending with Ensemble Learning Framework," in *2019 Chinese Control And Decision Conference (CCDC)*, China, 2019.

[8] M. Papouskova and P. Hajek, "Two-stage consumer credit risk modelling using heterogeneous ensemble," *Decision Support System*, vol. 118, pp. 33-45, 2019.

[9] M. Graczyk, T. Lasota, B. Trawiński and K. Trawiński, "Comparison of Bagging, Boosting and Stacking Ensembles Applied to Real Estate Appraisal," in *Asian conference on intelligent information and database systems (pp. 340-350)*, Springer, Berlin, 2010.

[10] P. Singh, "Comparative study of individual and ensemble methods of classification for credit scoring," in *2017 International Conference on Inventive Computing and Informatics (ICICI)*, India, 2017.

[11] C. Wang, Z. Hu, R. Chiong, S. Dhakal, Y. Chen and Y. Bao, "A PSO-Based Ensemble Model for Peer-to-Peer Credit Scoring," in *14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, China, 2018. https://doi.org/10.1109/FSKD.2018.8687154

[12] B. E. R. Singh and . E. S. , "Enhancing Prediction Accuracy of Default of Credit Using Ensemble Techniques," in *First International Conference on Artificial Intelligence and Cognitive Computing (pp. 427-436)*, Springer, Singapore, 2019.

[13] A. Somasundaram and R. S. U., "Data imbalance: Effects and solutions for classification of large and highly imbalanced data," in *Proc. of 1st International Conference on Research in Engineering, Computers and Technology (ICRECT 2016)*, 2016.

[14] C. Cortes and V. Vapnik, "Support-vector networks," *Springer*, vol. 20 (3), no. Machine Learning, p. 273–297, 1995.

[15] R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification, second ed., Wiley, 2001.

[16] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[17] J. M. Tomczak and M. Zieba, "Classification restricted Boltzmann machine for comprehensible credit scoring model," *Expert System in Application*, vol. 42, no. 4, p. 1789–1796, 2015. https://doi.org/10.1016/j.eswa.2014.10.016