International Journal of Smart Business and Technology
Vol. 6, No. 1 (2018), pp.49-68
http://dx.doi.org/10.21742/ijsbt.2018.6.1.05

Fintech Credit Scoring Techniques for Evaluating P2P Loan Applications – A Python Machine Learning Ensemble Approach

Rekha Ramesh Shenoy[1], Sabah Mohammed[2] and Jinan Fiaidhi[3]

[1,2,3]*Department of Computer Science, Lakehead University, ON, Canada*
[1]*rshenoy@lakeheadu.ca*, [2]*mohammed@lakeheadu.ca*, [3]*jfiaidhi@lakeheadu.ca*

Abstract

Financial Technology (Fintech) has been widely recognized as one of the most important innovations in the financial industry and is seen to evolve rapidly. It promises to reshape the financial sector by creating a diverse financial landscape, providing stability, improving quality, and, most importantly, reducing costs. One such fintech tool is "Peer to Peer Lending" (also known as "P2P Lending"), which refers to companies that match lenders and borrowers without the use of traditional banking systems. They are intermediaries, usually online investment platforms that offer identity verification, proprietary credit models, loan approval, loan servicing, and legal and compliance. This can be an attractive alternative for a borrower as loans can be applied online, anonymously, and timely. It is also beneficial for borrowers with no previous credit history to be shown. Fintech develops a credit scoring model based on the credit risk evaluation. This model establishes itself by using online data sources, alternative credit models, and a variety of machine learning and data analytics techniques to estimate risks involved in the lending process and minimize operating costs. This paper proposes a stacking ensemble of machine learning classifiers that combines data preprocessing with different learning algorithms. We then compare the results of the bare-bone classifiers with our stacking ensemble classifier. The ensemble model developed performs better than each of the single classifiers that constitute the credit scoring process.

Keywords: *Fintech tools, Credit scoring, Machine learning algorithms, Feature reduction, Outliers, Scikit-learn, Regression, Clustering, Bayesian, Neural networks, Forests, ensembles, Bagging, Boosting, Stacking*

1. Introduction

Managing customer credit is an important issue for each commercial bank; therefore, banks take great care when dealing with customer loans to avoid any improper decisions that can lead to loss of opportunity or financial losses. Manually estimating customer creditworthiness has become both time- and resource-consuming. Moreover, a manual approach is subjective (dependable on the bank employee who gives this estimation), so devising and implementing programming models that provide loan estimations is the only way to eradicate the 'human factor' in this problem [9].

The current computerized credit scoring systems are based on classical statistical theories that are widely used. However, these models are less resilient when it comes to large amounts of data input; consequently, some of the assumptions in the classical statistics analysis fail

Article history:
Received (March 5, 2018), Review Result (April 7, 2018), Accepted (May 6, 2018)

ISSN: 2288-8969 eISSN: 2207-516X IJSBT

[6]. In all types of business startups and established small businesses, many of these businesses are seeking some additional funding that is too small for an angel investor to get a return for their effort. Banks also think it's not worth their time. However, the necessary amount may be too much to finance on a credit card, or the entrepreneur may not want to use that method. That's where Peer-to-Peer (P2P) lending is working to fill that lending gap and why we are considering this lending alternative and to evaluate the credit scoring for such a lending system. This leading credit scoring evaluation may be a solution for many small businesses struggling to tap smaller funding amounts. Peer-to-peer lending involves borrowing money from your peers, including other businesspeople and investors interested in relatively small financing amounts

As I researched various news and reports on Fintech, it was observed that these startups are mainly based on business models that target eminent services in demand, such as Wealth Management, Payments, Lending, Crowdfunding, Capital Markets, and Insurance. Therefore, constructing credit scoring models for startup loans requires data mining techniques. This process may use various data bins, including demographic characteristics, historical payment data, and statistical techniques. While building a machine learning model for credit scoring based on a "bin" of characteristics with value and ranges is much better than the legacy statistical methods, the bins are meant to maximize the separation between known good cases and known bad cases, which largely depend on the dataset selected and machine learning model used in the training stage. However, ensemble methods have been called the most influential development in Data Mining and Machine Learning in the past decade [9]. They combine multiple models into one, usually more accurate than the best of its components. Ensembles can provide a critical boost to industrial challenges -- from investment timing to drug discovery and fraud detection to recommendation systems -- where predictive accuracy is more vital than model interpretability.

Thus, much emphasis is placed on choosing an efficient ensemble algorithm in fintech organizations. Different classifier methods have varied over time, wherein single classifiers were used in the initial period. However, each classifier showed deficiencies in generating a good result given different datasets. Hence, the fintech organizations and researchers started experimenting with complex models and introducing newer techniques, resulting in hybrid / ensemble methods. The only difference between hybrid and ensemble methods was that hybrid methods introduced data preprocessing and filtering on the datasets before training the model and later. In contrast, ensemble methods focused on the classifier learnings of different base classifiers. Our paper aims to build a hybrid system based on clustering and classification. Then, it feeds this processed data to an ensemble to predict the classification with the best results given any dataset.

The remainder of the paper is organized as follows. Section 2 describes the problem definition, related literature survey, and the datasets used for the analysis. Section 3 presents the details of the methodology, and the relevant machine learning algorithms are presented. Section 4 provides a complete report on the implementation. Based on the observations and results of these experiments, Section 5 draws conclusions and future research directions.

2. Research motivations

The credit scoring model results from a statistical model that evaluates the borrower's information and calculates an estimate for the probability of the borrower defaulting on his loan. With the advent of newer techniques and algorithms, efficient systems have been built to estimate the likelihood that a borrower may default. The credit scoring models have

improved the task of estimating the risk of default as they also include other aspects of credit risk management. The entire risk evaluation is divided into stages, which are:

Pre-application stage: to identify potential applicants.

Application stage: to identify the acceptable applicants and collect their information

Performance stage: to identify the possible behavior of the current customers based on customers with similar profiles.

2.1. Problem statement

For P2P lending in Fintech, the startups are not directly involved in the lending, they do not provide the actual capital that influences the lending amount. They are only instrumental in matching the borrowers with the lenders interested in the borrower's credit purpose. Evaluating the objective for which the borrower intends to take the credit is very important. At the same time, the lender also evaluates his intent to benefit from lending the credit to the borrower. Since the fintech companies must evaluate the real requirements and meet the right aspects to match the lender efficiently, there might be different objectives/aspects that need to be considered from both the lender's and borrowers' points of view, such as:

1. What factors explain a loan default in Fintech's P2P lending?
2. How do the P2P lenders associate the borrowers with their risk association?
3. What are the noteworthy attributes/characteristics indicative of defaults in P2P lending?
4. How does machine learning prove beneficial in P2P lending?

To answer these questions, P2P lending suffers from the problem of information asymmetry wherein the borrowers are better informed than lenders of their ability and willingness to pay. This can cause adverse selection where the lenders cannot distinguish between a good risk / bad risk borrower. The P2P lending platforms evaluate and assign a grade to each loan application associated with an interest rate based on the credit risk. However, it is observed that the higher the interest, the higher the credit risk observed. While evaluating the credit risk, P2P lending considers the loan and borrower characteristics such as Loan characteristics (Loan purpose, loan amount), Borrower characteristics (home ownership, assets, income, and employment length), and credit history (expenses, records, ability, and patterns in which customer paid previous loans). Using machine learning algorithms in P2P lending is important because it helps apply predictive analysis to enormous amounts of data in real-time and produces quick and efficient results. They can also help gather information from various online sources to detect rogue investors working together across multiple accounts.

Hence, for our project, we dive into various machine learning techniques that evaluate different credit scoring datasets available to perform the analysis of comparing different machine learning algorithms. These credit datasets generally use 'historical' data gathered from various customers to build a scorecard based on their previous or current credit status. Of all the data available for the customer, only features that provide valuable information and impact their credit behavior are examined and analyzed.

2.2. Related work

In response to the growth of the fintech sector, especially in lending and managing huge loan portfolios, various models of credit scoring systems have been implemented and adopted successfully. Various financial institutions have diverse ways of collecting credit information, requiring different risk analytics systems. The advantage of fintech tools is that they provide niche and independent analysis systems for client organizations while trying to build an efficient system that addresses multiple clients' demands. Hence, credit risk scoring systems

must implement a model that provides efficient solutions irrespective of the dataset provided to the model.

Table 1. Scoping review

Year →		2006	2010	2012	2013	2015	2016	2016	2017
Machine Learning Method ↓	Authors → Machine Learning Algorithms ↓	[1] Arjun Chandra, Xin Yao	[10]Nan-Chen Hsieh, Lun-Ping Hung	[7]Iain Brown, Christophe Mues,	[8] Jochen Kruppa, Alexandra Schwarz, Gerhard Arminger, Andreas Ziegler	[4] Fatemeh Nemati Koutanaei, Hedieh Sajedi, Mohammad Khanbabaei	[9]Maher Ala'raj, Maysam F. Abbod	[2] Browne, David and Steven Prestwich	[5]Ha, Sang & Nguyen Ha, Nam & Nguyen Thi Bao, Hien.
Data Filtering		√	√	√			√		
Feature Selection		√	√			√	√		√
Dimensionality Reduction	Linear Discriminant Analysis			√					
Bayesian	Naïve Bayes					√	√	√	
	Bayesian Network		√						
	Conditional Decision Trees			√		√	√		
Instance-based	k-Nearest Neighbor			√	√				
Vector machines	Support Vector Machine (SVM)		√	√		√	√	√	
Neural machine	Neural networks		√	√			√		
	Artificial Neural networks (ANN)					√			
Regression	Logistic Regression			√	√			√	
Ensembles	Bagging					√			
	Boosting	√				√			
	Random Forest			√	√			√	
	Gradient Boosting Machines (GBM)			√					√
	Hybrid	√	√				√		

While building efficient credit scoring models, some focus on data preprocessing and stress the importance of handling the data efficiently before training the model. [5] uses the recursive feature elimination approach to evaluate the significance of the features in the classifier by removing each feature step-by-step and assessing the performance. Only the most significant features are retained based on the feature ranking obtained. The accuracy of the classifier using the selected features gave better accuracy and performance results than

other methods. [2] observations also indicate that the feature selection method helped reduce the overfitting problem while improving the accuracy.[7] focused on handling imbalanced data while using traditional classification techniques such as logistic regression(LR), neural networks, and decision trees(NN), the researchers also explore the suitability of gradient boosting(GB), least square support vector machines(LS-SVM) and random forests(RF) for loan default prediction. Over time, the researchers moved from classic algorithms and started experimenting more with complex algorithms and systems integrating multiple base classifiers that brought ensembles into effect.[4] The authors propose a hybrid model of feature selection and ensemble learning classification algorithm based on valuation approaches, such as SVM classification accuracy, AUC, and parameter settings.[10]introduces the concept of class-wise classification to introduce a new class called 'borderline' risk to estimate potential risky borrowers better.

Alborrowersany models have been designed to predict the credit score accurately. However, no ideal or specific classifier exists among the available models, as each model behaves differently with different data sets. It is important to note that each percentage point can affect the scoring system. Hence, choosing the best model is of the utmost importance as it relates to curbing the considerable losses to the financial organization.

Statistical and mathematical formulas have been persistent and used broadly to calculate credit scores; new research studies have proved that Artificial Intelligence (AI), neural networks, support vector machines, and ensemble methods can provide much more accurate analysis than traditional approaches. Further experiments have shown that the hybrid or ensemble approaches, although complex, have proven better performance than individual models.

The usual process in any credit scoring model is to use the previous borrowers' credit history and compute and predict the likelihood of default risk for new clients. The features or attributes of this historical data are thus used to map and predict a client's default probability. The number of features, therefore, forms the feature space. In machine learning, it is considered that the more data there is, the more reliable the prediction analysis is. However, as the dimensionality of the dataset grows, many difficulties arise, too, as there might be a lot of non-meaningful data in the entire dataset. Large datasets usually have many noisy features, significantly impacting machine learning. Hence, noise should be reduced as much as possible to improve the efficiency and accuracy of the machine-learning algorithms.

With the datasets we have used for analysis, our primary observation is that each dataset differs in size, nature of attributes, and the information they hold. Hence, handling such variances and forming an efficient classifier training method is crucial. There is also the problem of imbalanced data in large datasets, especially in credit risk models, where the number of defaulting customer data is far less than that of non-defaulter data. Another thing to consider is that more features in a given dataset will increase computation time, impacting the model accuracy model's prediction.

Our proposed method focuses on building an ensemble model that focuses on the results of a group of classifiers trained on the same dataset and evaluating the best strategies for each dataset. The ensemble built combines the predictions from these different classifiers and gives the final prediction. Through this ensemble, we aim to develop a sturdy system that performs the best with other datasets.

2.3. Datasets

The datasets used in the credit score model are based on the historical credit information of former customers. They are used to predict the risk factor for a new applicant based on similar behavioral/social attributes. The attributes or features of each loan applicant are mapped to the historical loan accounts, and the ideal credit model is built. The only disadvantage of this approach is that these datasets may differ in size, nature, and information, which usually causes discrepancies in the classifier training process, thus missing capturing the real correlation of the information to the desired scoring. They might contain missing values, redundant values, irrelevant features, erratic data, etc., affecting the scoring greatly.

2.3.1. German credit dataset

The German Credit dataset is a publicly available data set and can be downloaded from the UCI Machine Learning Repository2This dataset contains 1000 entries with 20 categorical attributes that Dr. Hofmann prepared. There are 700 credit-worthy applicants and 300 samples where credit was not extended. This is based on the 20 attributes that describe credit history, account balances, loan purpose, loan amount, employment status, and personal information. These 20 attributes are made of 13 categorical, three continuous, four binary features, and 1 class feature to define good or bad risk.

The cost matrix/status of 1 indicates a good customer, whereas the status of 2 indicates a bad customer. It is based on the principle that "It is worse to class a customer as good when they are bad (5) than it is to class a customer as bad when they are good (1)." 2 Based on the correlation heatmap, the attributes of duration, amount, installment rate, residing since, age, several credits held, and several dependents showed a correlation concerning the credit risk status. No missing values were observed for this dataset.

2.3.2. Australian credit dataset

The Australian Credit dataset is also a publicly available data set and can be downloaded from the UCI Machine Learning Repository2. This dataset contains 690 entries with 15 numerical attributes that Dr. Hofmann prepared. There are 383 samples of credit-worthy applicants and 307 samples where credit was not extended. This is based on the 15 attributes that have been changed to meaningless symbols to protect the confidentiality of data. This dataset is interesting because there is a good mix of attributes -- continuous, nominal with small numbers of values, and nominal with more significant numbers of values2. These 15 attributes comprise eight categorical, six continuous features, and 1 class feature to define good or bad risk. The attribute names have been hidden to maintain the confidentiality of this dataset; hence, they have been named A1, A2, ... and so on. No missing values were observed in this dataset.

[2] UCI Machine Learning Repository is a collection of databases—domain theories and data generators available for the machine learning community. Students, educators, and researchers widely use it as a primary source of machine learning datasets. URL: https://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29 .

2.3.3. Give me some credit

This dataset was a part of a data analysis competition in Kaggle3This dataset consists of 11 features and has 150,000 records. The predictor variable in this dataset is the 'SeriousDlqin2yrs', which has a binary value of 0 or 1. The value of 0 indicates that the borrower is a good customer who repays his loan on time. A value of 1 indicates that the borrower is a bad customer or 'delinquent' and has defaulted on his loans. The data is based on customer evaluation for 2 years. These 11 attributes comprise 10 continuous features and 1 class feature to define good or bad risk. Based on the correlation heatmap, the attributes that showed some correlation to the credit status are RevolvingUtilizationOfUnsecuredLines, age, number of times borrower was past due (30-59 days, 60-89 days, and 90 days), debt ratio, monthly income, open credit, real loans taken and number of dependents. Some missing values were observed for monthly income and number of dependants attributes.

2.3.4. Mock dataset

This dataset is an actual time credit data set that was downloaded from Credit Risk Analytics4 webpage. The original dataset has 887380 entries and 74 attributes. However, there was a lot of missing data for some features. Hence, we took a subset of the original data and built our Mock Dataset. This mock dataset has 884631 samples and 19 attributes. There are 817963 samples of credit-worthy applicants and 66668 samples where credit was not extended. This is based on the 19 attributes that describe credit history, account balances, loan purpose, loan amount, loan reimbursement employment status, and personal information. These 20 attributes are made of 3 categorical, 15 continuous features, and 1 class feature to define good or bad risk. Based on the correlation heatmap, attributes like loan amount, term in months for the loan, interest rate, installment amount, employment length, open credit accounts, income, dti, delinquency observed, revolving balance, revolving utility, total payment including total received principal and interest and last payment played a significant correlation to the credit status. Some missing values for employment length and revolving utilization were observed in this dataset.

The datasets can be summarized as shown in [Table 2] as shown below:

Table 2. Dataset summary

Dataset	No. of instances	No. of numerical features	No. of ordinal features	No. of nominal features	Class 1: Class 2
German	1000	3	5	13	700:300
Australian	690	6	0	8	383:307
Give Me Some Credit	150,000	10	0	0	139974:10026
Mock Data	884631	15	0	3	817963:66668

[3] The Kaggle Public Wiki is a resource for learning statistics, machine learning, and other data science concepts, with a strong focus on the practical application of those skills in a competitive environment. The referenced dataset is available at the URL: https://www.kaggle.com/c/GiveMeSomeCredit

[4] The referenced website is prepared by three professors – Prof. Bart Baesens (KU Leuven, Belgium), Prof Daniel Rösch (Regensburg University, Germany), and Prof. Harald Scheule (Associate Professor at the University of Technology, Sydney, Australia, for credit risk learning and teaching purpose. The given dataset was obtained by registering through the URL: http://www.creditriskanalytics.net/datasets.html

3. Methodology

The proposed model is designed in sequential steps such as data filtering, splitting the dataset into training and testing sets, training the model, generating predictions for a set of algorithms on a particular dataset, building an ensemble that takes in input these set of predictions, combines them to generate the final prediction using cross-validation see Figure. 1.

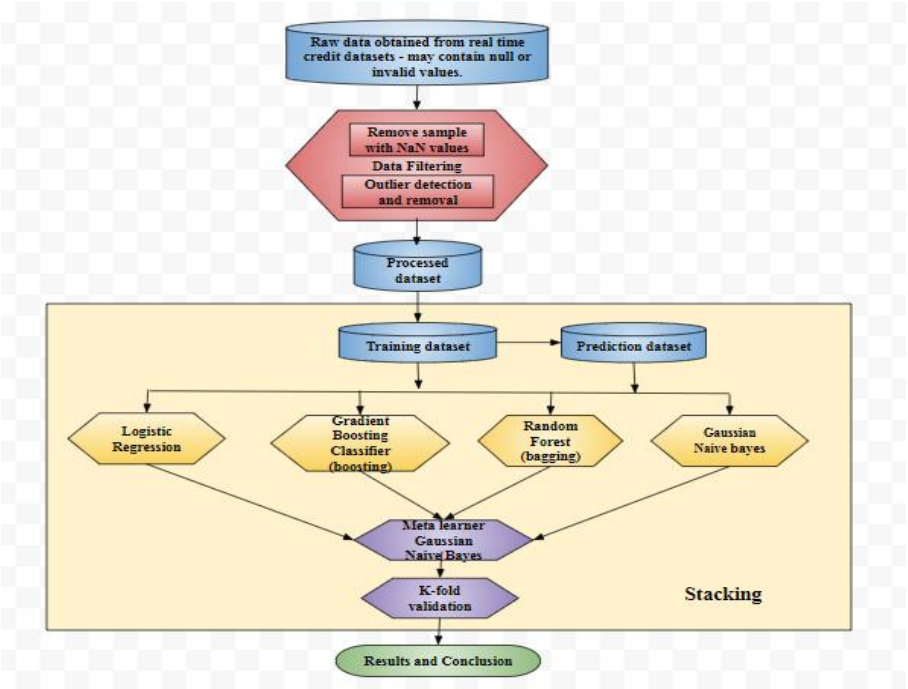


Figure 1. Hybrid ensemble

3.1. Machine learning algorithms used for the hybrid ensemble

This project aims to compare the performances of different classification techniques concerning the credit scoring context. The machine learning algorithms that we used in our credit scoring model are as follows:

3.1.1. Logistic regression

Logistic regression models are usually used to analyze models where the outcome variable is either binary or dichotomous. It follows the same principles as a linear regression, with the difference observed only in the model and its assumptions. In logistic regression, instead of predicting the value of a variable Y based on the predictor variables, like in linear regressions, we calculate the probability of Y being 'Yes / No' based on the given known values of the predictor variables. Our project focuses on the response achieved that determines whether a creditor is good or bad (i.e., non-defaulter or defaulter). The logistic function is thus written as:

$$p(x) = \frac{1}{1+ e^{-(\beta_0+\beta_1 x)}} \quad (1)$$

3.1.2. gradient boosting classifier

Gradient Boosting is an ensemble technique used for regression or classification problems. Since the main idea of boosting is to add new models iteratively to the ensemble, it combines the weak 'learners' into a strong learner in an iterative fashion. The principal idea behind these algorithms is to construct the new base learners to have maximum correlation with the negative gradient of the loss function of the ensemble5. The gradient boosting classifier is thus an additive model that allows for optimizing the arbitrary differentiable loss functions. In each iteration, the regression trees are fit on the negative gradient of the binomial or multinomial loss function6In this case, the regression trees are usually a decision tree where each leaf is given a score.

The objective function for the gradient boosting classifier can be given as:

$$J(\theta) = \sum_{i=1}^{n} l(\mathrm{y_i}, \hat{y_i}) + \sum_k \Omega(f_k), \quad f_k \in \mathcal{F} \qquad (2)$$

Here, yi represents the sum of scores assigned for each leaf in the decision tree, fk is the regression tree, a function that maps attributes to the score, $and\ \mathcal{F}$ is the space of all the regression trees. Thus, the gradient model predicts θ by addictive training. It starts from a constant function and adds new functions fk in each iteration. After the 'K' number of rounds, θ construction is complete. The value of fk is calculated by minimizing the $J(\theta)$; during the minimizations, the gradient of loss of functions is used, thus giving this method 'gradient boosting.'

3.1.3. Random forest

A Random Forest classifier is a supervised learning procedure that operates on the simple principle of "divide and conquer," wherein sample fractions of data are used to generate a randomized tree predictor on a small piece of the dataset, and then these pieces are aggregated together. Once these decision trees are generated and trained, a voting procedure is used to determine the most popular class for each tree. This is selected as the final class determinant for the random forest. Hence, it is also an ensemble method based on bootstrap aggregation or 'bagging.' It uses feature bagging wherein a random subset of features is selected to train the decision trees.

Random Forest is straightforward to use, has proven accurate, and has good prediction results. It also eliminates the concern of overfitting the model as it builds enough trees so that the classifier can divide the data evenly. However, the large number of trees can make the algorithm slower and sometimes ineffective for real-time predictions. They can be said to be fast in train but slow in generating predictions.

[5] Statement as understood in the article of 'Gradient Boosting machines, a tutorial' made available in the research webpage of 'frontiers in Nuerobiotics' referenced at URL: https://www.frontiersin.org/articles/10.3389/fnbot.2013.00021/full

[6] Description as understood from the scikit-learn tutorials available at URL: 1. http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

3.1.4. Gaussian Naïve Bayes Classifier

Naïve Bayes classifiers are probabilistic classifiers based on the Bayes theorem of string (naïve) independence assumption between the features.7The Gaussian Naïve Bayes assumes that the values associated with each class are distributed according to the Gaussian distribution principle for continuous data. Thus, it assumes that all the features may be unrelated.

For Gaussian Naïve Bayes classifier approach, let's assume that an attribute 'x' contains continuous data. Then, the following algorithm segments the data by class and computes the mean μy and variance σ_y^2 for each class as follows:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \quad (3)$$

A Naïve Bayes classifier is used to calculate the class's posterior probability by multiplying the class's prior probability before seeing any likelihood of the data given its class. Thus, the NB classifier analyses the training set to determine the mapping function and the final class.

The most important consideration for combining models is to reduce the probability of misclassification based on any single induced model by increasing the system's area of expertise through different combinations. We use logistic regression as its outputs have a nice probabilistic interpretation, and the algorithm can be regularized to avoid overfitting. Since we are considering different datasets with different attributes and behaviors, we aim to tackle the variance error and implement a parallelized model using the Random Forest algorithm, which is inherently a bagging ensemble technique. We also aim to reduce bias errors through the Gradient Boosting classifier, the boosting ensemble technique. The Gaussian Naïve Bayes assumes that the values associated with each class are distributed according to the Gaussian distribution principle for continuous data. Thus, it assumes that all the features may be unrelated, showcasing powerful knowledge representation and reasoning algorithms under conditions of uncertainty. The main intention behind using all the above-mentioned machine algorithms is to create a robust system that performs consistently.

4. Implementation and analytics

"Ensemble is the art of combining diverse sets of learners (individual models) to improvise on the stability and predictive power of the model." - Analytics Vidhya8Every machine learning algorithm has a limit beyond which it cannot fit the given data, and the accuracy stops. If we try to fit in more data, it leads to a 'data over-fitting problem.' This could be due to differences in population, hypothesis, the given raw data, or the unique modeling tech. Ensembles usually overcome this problem as they use multiple models using different techniques, such as those mentioned below:

Bagging:

Bagging derives its name from Bootstrap Aggregating. It tries to implement similar types of learners on small samples of the data/ training set and then aggregates the model by taking

[7] As learned and understood from the description of the Gaussian Naïve Bayes method on Wikipedia. URL: https://en.wikipedia.org/wiki/Naive_Bayes_classifier

[8] https://www.analyticsvidhya.com/blog/2015/08/introduction-ensemble-learning/

a mean of all the predictions. However, while resampling the data, some instances get represented multiple times, and some are left out. Since the individual base classifiers may not be exposed to the same records, voting on their results is carried out. It is used mainly to reduce the variance error, quantifying how the predictions made on the same observations differ. As the training data size increases, a reduction in the variance is observed, thus making the model predictions more accurate. We use the Random Forest algorithm, a bagging technique, as a base learner for our implementation.

Boosting:

Boosting uses an iterative technique that adjusts the observation's weight based on the previous classification. First, it uses the subset of the original data to produce a series of average-performing models. Then, it boosts their performance by combining these models together using some cost function (like voting). If the models in the first step are classified incorrectly, then it tries to increase the weight of the observation and vice versa. It is mainly used to reduce the bias error, which quantifies how much, on average, the predicted values differ from the actual value. Unlike bagging, where random subsets are created, boosting creates sampling based on the performance of the previous models. Thus, every new subset has elements likely misclassified by previous models. We use the Gradient Boosting Classifier algorithm, a boosting technique, as a base learner for our implementation.

Stacking:

In stacking, we use a learner to combine results from other individual learners. Instead of using some empirical formula to calculate weights, we introduce a meta-learner that takes in individual learners' results and uses another approach to estimate the predictions. Hence, an ensemble of individual classifiers is first trained, and the resultant classification output is fed as the input to the meta-learner. It thus reduces bias or variance error depending on which combining meta-learner we have used. Therefore, the role of the meta-learner is to discover the best possible way to combine the prediction of the base learners.

The above three methods can thus be compared as shown in the below table:

Table 2. Comparison between ensemble techniques

Comparison	Bagging	Boosting	Stacking
Method	Parallel ensemble where each model is built independently	A sequential ensemble is used to add new models to perform well in cases where the previous models are lacking.	A meta-learner combines the results of varied base learners to generate the resultant prediction.
Subset creation	Random	Higher preference is given to misclassified samples	Varied
Function to combine into a single model	Weighted average	Weighted majority vote	Logistic regression
Suitable for	Complex models with high variance and low bias	Models with low variance but high bias	Any complex model with low variance or bias
Goals	Reduce variance	Reduce bias	Reduce variance or bias depending on the model
Example	Random Forest	Gradient Boosting	Blending

Ensembles are one of the most popular methods used in machine learning. They combine predictions from different models and generate a final prediction. Any base models can be combined to form the ensemble; the more, the better! Ensembles not only improve prediction, but they also help reduce errors in the prediction. They do so by averaging the irregularities, thus smoothening the decision boundaries.

Datasets usually comprise varied attributes and features that may or may not be correlated to each other but are very important factors in determining the credibility of the expected predictions. However, as suggested earlier, they may sometimes include redundant or irrelevant features that make it challenging to train the models, thereby reducing the accuracy and performance of the given model. Hence, it is very important to determine the nature of these attributes and process them accordingly.

We have used the Python packages Pandas, Numpy, Sci-kit-learn, and Matplotlib, specially designed to help with data analysis and visualization. These packages have also helped get the data ready to build our model.

4.1. Data filtering and preprocessing

Before implementing our training model, we must recognize and remove any data or information that stands out to make our model uniform. "Observation which deviates so much from other observations as to arouse suspicion a different mechanism generated it"— Hawkins (1980). This was defined to explain outliers, which are extreme values that tend to deviate from other observations in a given dataset. Data entry errors, incorrect measurement errors, experimental errors, data processing errors, sampling errors, etc, majorly cause these outliers.

The first step in our credit modeling was to perform data filtering, which reduced the original data set size to retain a meaningful data set while not affecting data integrity. It helps to smoothen the decision boundaries, thereby helping achieve the targeted prediction with much better accuracy and performance. Only essential and most relevant attributes are retained and modeled for training and testing, improving the accuracy and reducing computational costs. The first step is identifying the missing or NaN values in the given datasets. Since the datasets are mainly read in the ‘CSV' file format, the missing symbols are replaced with empty strings," which are interpreted to NaN in the Python packages. The German and Australian datasets did not have any null values. However, the GiveMeSomeCredit and Mock Data datasets showed 29731 and 7165 total null values, respectively. Listwise Deletion for null values was performed.

```
In [ ]: # Cleaning the data to remove all attributes with missing data (Nan/null)
        dataset_df = pd_df.replace('',np.nan)
        dataset_df = dataset_df.dropna(axis="rows", how="any")
```

Figure 2. Code snippet for removing NaN values

4.2. Outlier detection and removal

The quality of the samples plays a critical role in modeling the implemented classifier, as misclassified patterns generally throw many errors in the model, thus affecting its accuracy. Using a scatterplot from the matplotlib package, we identified isolated or inconsistent values based on a clustering approach wherein each feature was classified according to credit status.

The assumption made in this case is that these isolated values tend to be far away from the continuous clusters.

For the outlier removal process, we used the normal distribution and standard deviation approach, especially for numerical attributes, to identify the starting and ending range values of the continuous values. In this approach, we tried to remove the outlier points by removing any points out of the range of (Mean ± 2*SD). The numpy ‘mean’ and ‘std' functions were used to obtain each attribute's mean and standard deviation. Once we got the range of the final list, we removed the rows of values that had values for the characteristics outside the given range.

```
In [22]: # identify the min/max range of attribute values using the mean and standard deviation (MSD)
         import numpy

         for y in list1:
             arr = nd_df[y]
             print(y)

             elements = numpy.array(arr)

             mean = numpy.mean(elements, axis=0)
             sd = numpy.std(elements, axis=0)

             final_list = [x for x in arr if (x > mean - 2 * sd)]
             final_list = [x for x in final_list if (x < mean + 2 * sd)]

             #print(final_list)
             print(min(final_list))
             print(max(final_list))

A1
0
1
A2
15.17
50.75
A3
0.0
12.5
A4
1
2
A5
1
13
A6
1
8
A7
0.0
5.085
A8
0
1
```

Figure 3. Code snippet calculating min max based on mean and standard deviation

The heat map feature shows the correlation between the independent features concerning the credit status.

4.3. Balancing the dataset

Since the number of samples where the credit was extended was significantly larger than the samples where the credit was rejected, there is a huge possibility of the classifier system being biased and tending towards creditworthiness and extension. This could make the model highly unstable, showcasing inaccurate predictions. Hence, we attempted to balance the dataset by randomly choosing a sample of records with class 0, equal to the number of samples belonging to class 1 in the given datasets. This would help the classifier models to learn about each class equally and thus make for a better prediction model see [Figure. 2].


```
In [47]: # Selecting random records with status values of 0
         new_australian_df = pd.DataFrame(australian_df)
         #print (new_gmc_df)
         new_australian_df = new_australian_df.loc[(new_australian_df['status'] == 0)]
         #print (new_gmc_df)

         rows = np.random.choice(new_australian_df.index.values, 104)
         smplastr_df = new_australian_df.loc[rows]

In [48]: # Appending records with status values of 1

         new_australian_df1 = pd.DataFrame(australian_df)
         #print (new_gmc_df)
         new_australian_df1 = new_australian_df1.loc[(new_australian_df1['status'] == 1)]
         #print (new_gmc_df1)

         smplastr_df = smplastr_df.append(new_australian_df1)

In [49]: smplastr_df.status.value_counts()
Out[49]: 1    104
         0    104
         Name: status, dtype: int64
```

Figure 2. Code snippet of balancing data

4.4. Implementation of the ensemble

The ideal ensemble consists of highly accurate predictors, which, at the same time, disagree as much as possible.[9]Hybrid ensembles deal with the combination of base learners trained using different algorithms. Their predictions are given as input to another ensemble leaner that generalizes the resulting prediction based on the input probabilities. The ensemble learning method is a commonly used approach by researchers where multiple base classifier outputs are pooled to provide the decision. In our implementation, we focus on the Stacking technique, where the base classifier results are processed and used as input to a meta-classifier that generates the final prediction. It is recommended to use as many different models as possible. Hence, for our hybrid system, we have used the base learner algorithms such as logistic regression, random forest, gradient boosting classifier, and Gaussian Naïve Bayes classification methods. Using these models, we create a prediction matrix that corresponds to the predictions generated by each model. We observe that each dataset's base model with the highest accuracy differs. The logistic regression model performs best for German and Mock datasets, the Gradient Boosting Classifier works best for the GiveMeSomeCredit dataset, and the Random Forest performs best for the Australian dataset. The training and test set data is divided into a 70-30% ratio, and the performances of the base learners are verified based on the ROC-AUC score.

Table 3. Validation results for base classifiers

	Logistic Regression		Gradient Boosting Classifier		Random Forest		Naïve-Bayes	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
German	0.865	0.825	0.991	0.812	1	0.764	0.789	0.768
Australian	0.913	0.886	1	0.89	1	0.894	0.869	0.873
GiveMeSomeCredit	0.85	0.852	0.877	0.858	0.999	0.835	0.843	0.847
Mock Data	0.852	0.847	0.827	0.818	0.999	0.759	0.718	0.714

We then define a meta-learner that will generate the final prediction. We have reused the Gaussian Naïve Bayes as a meta-learner for our research. Bayesian networks are considered to easily model the complex relationships among the different variables, especially if they are discrete. They do not have any requirements on the distribution of the underlying variables as the relationship between variables is explicitly represented by acyclic graphs.

We split the entire training set into training and prediction sets equally for the base learners. Therefore, we have one training set (Xtrain base, ytrain_base) and another prediction

[9] As stated in the research document, reference at URL: https://arxiv.org/pdf/1106.0257.pdf

(Xpred_base ypred_base), which will generate the prediction matrix to be fed as input to the meta learner. Once the base learners are trained and create the prediction matrix, we feed this prediction matrix to the meta learner (Gaussian Naïve Bayes model) and further train this meta learner. The meta learner thus generates the final prediction see Figure 3.

```
In [65]: # defining a procedure to generate new train and test datasets
         # Herein, we use the Blending approach
         # also, training and testing data is split 50-50 of the the previous training set.
         # We now have one training set of the base learners (Xtrain_base,ytrain_base) and one prediction set (Xpred_base,ypred_base) and
         xtrain_base, xpred_base, ytrain_base, ypred_base = train_test_split(
             X_train, y_train, test_size=0.5, random_state=SEED)
```

Figure 3. Code snippet for training the base and meta learners

Until now, we have trained the base learner and the meta learner on only 50% of the dataset; hence, a lot of crucial information may be lost. To overcome this loss, we use the K-fold cross-validation method. In this method, the base learners are trained again, wherein a copy of the base learner is fitted on K-1 folds, thus predicting the left data. For the number of folds specified, the entire process is iterated. Keeping a more significant number of folds is recommended to ensure the whole data is uncaptured. Thus, for each 10-fold cross-validation, the given data set is first partitioned into 10 equal-sized sets, then each set is used as the test set while the classifier trains on the other nine sets. This entire process of fitting an ensemble with cross-validation is called 'stacking .'This process helps the base and meta-learners train on the complete datasets. It is observed that stacking results in a sizeable improvement in performance and generates the best score. We use the ROC-AUC score to measure how well our models perform, which trades off having high precision and high recall, see Figure 4.

```
In [75]: from sklearn.model_selection import KFold

         # Train with stacking
         cv_base_learners, cv_meta_learner = stacking(
             base_models(), clone(meta_learner), X_train.values, y_train.values, KFold(10))

         P_pred, p = predict_ensemble(cv_base_learners, cv_meta_learner, X_test, verbose=False)
         print("\nEnsemble ROC-AUC score: %.3f" % roc_auc_score(y_test, p))

Fitting final base learners...done
Generating cross-validated predictions...
Fold 1 done
Fold 2 done
Fold 3 done
Fold 4 done
Fold 5 done
Fold 6 done
Fold 7 done
Fold 8 done
Fold 9 done
Fold 10 done
CV-predictions done
Fitting meta learner...GaussianNB(priors=None)
done

Ensemble ROC-AUC score: 0.908
```

Figure 4. Code snippet of K-fold validation on ensemble

4.5. Results and discussion:

In a ROC (Reciever Operating Characteristic) curve, the true positive rate (also termed Sensitivity) is plotted as a function of the false positive rates (also termed Specificity). As shown in the figures below, the model's accuracy is measured by the area under the ROC curve referred to as AUC. In a ROC curve, we plot the 'True Positives' on Y-Axis and the 'False-Positives' on the X-axis. The 'True Positives' are the correctly predicted positive values, meaning that the value of the actual class and the expected class are both yes. 'False Positives' are values wherein the actual class is yes, but the predicted class is no. As per definition, an area of 1 represents a perfect test, whereas an area of 0.5 represents a worthless test. The more

a ROC curve is lifted up and away from the diagonal, the better the model is10. In other words, the greater the AUC, the more accurate our test model will be. In our analysis of comparing the base learners to the ensembles, we achieved the AUC for the ensembles to be as close to the top left-hand borders as possible, indicating better accuracy of the model. See Figures 5-8.

We have plotted the ROC-AUC score of the base learners and the hybrid ensemble as follows:

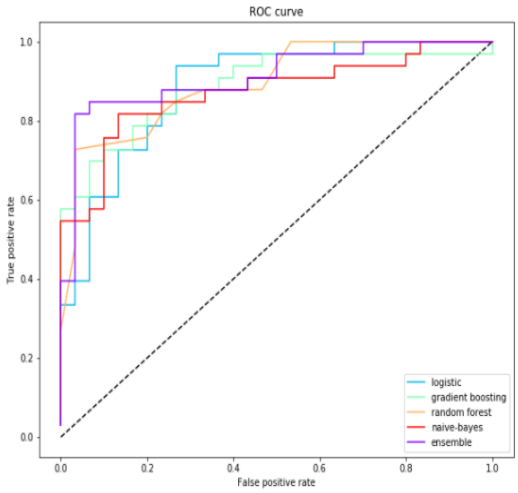


Figure 5. German data ROC

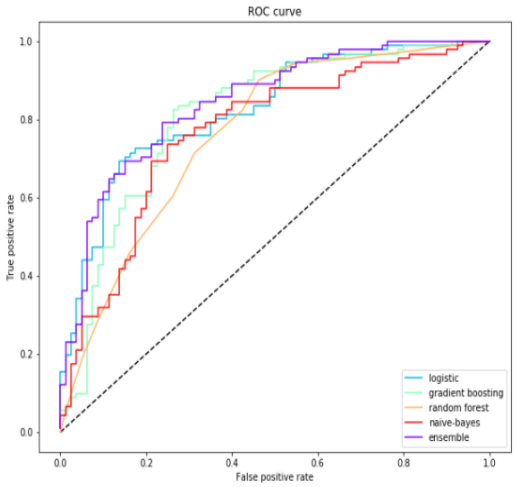


Figure 6. Australian data ROC

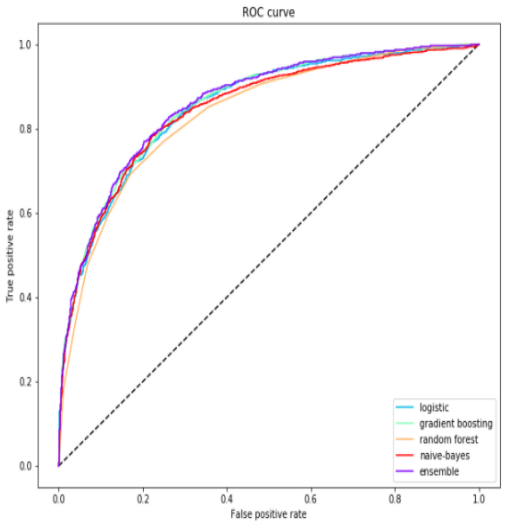


Figure 7. GiveMeSomeData ROC

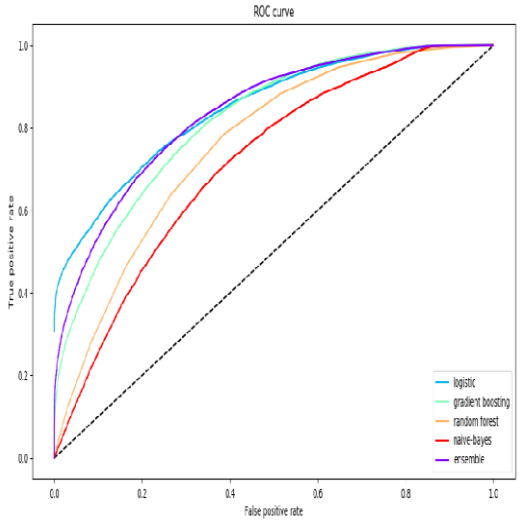
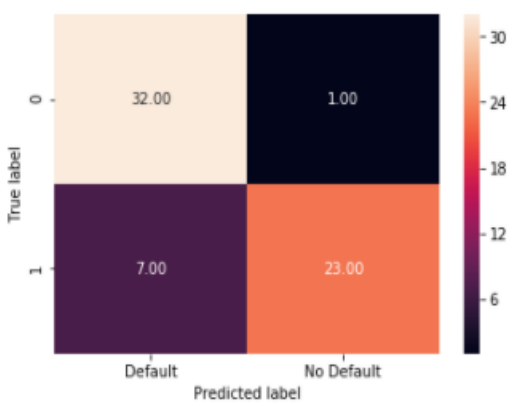
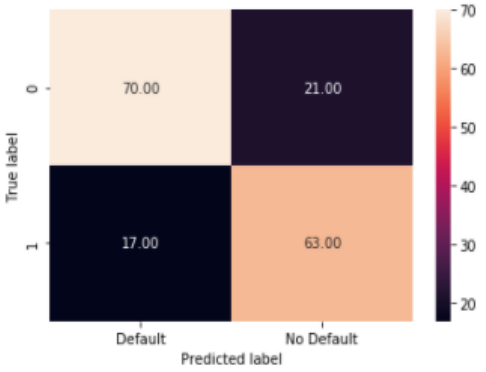


Figure 8. Mock data ROC

[10] https://ashokharnal.wordpress.com/2014/03/14/a-very-simple-explanation-for-auc-or-area-under-the-roc-curve/

Now that we have calculated the ROC-AUC, we verify the accuracy of the final hybrid ensemble model by plotting the confusion matrix (also known as the error matrix). It is a specific table layout that allows the visualization of the performance of the hybrid model. In

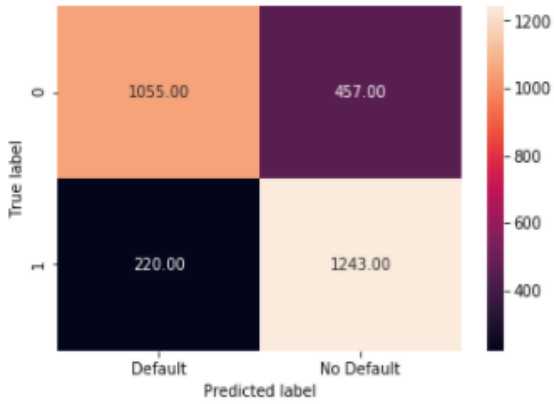




the confusion matrix, each column represents the instances in the predicted class, while each row represents the instances in the actual class. The confusion matrix (CM) for each of the datasets using the hybrid ensemble is shown as follows shown Figures 9-12.

Figure 9. CM German test data

Figure 10. CM Australian test data



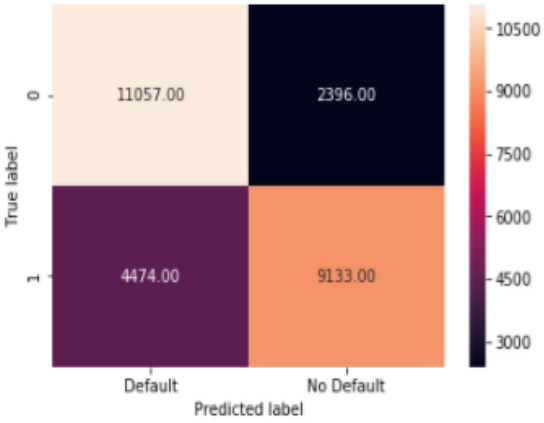


Figure 11. CM GiveMeSomeCredit test data

Figure 12. CM mock test data

Once we have the confusion matrix's visual representation, we calculate the accuracy, precision, and recall values for the predictions. Accuracy is the ratio of correctly predicted observations to the total observations. Although accuracy is one of the greatest measures and is expected to have the highest value, consideration must also be given to the symmetry of the datasets. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations—high precision related to a low false positive rate. Recall is the ratio of correctly predicted positive observations to all the observations in the actual class. Accuracy works best if false positives and false negatives have similar costs. If the price of false positives and false negatives vary, it's better to look at both Precision and Recall. The cumulative results of the hybrid model for these parameters are shown in the following table.

Table 4. Performance parameters for the hybrid model

Parameters\Model	German	Australian	GiveMeSomeCredit	Mock Data
Accuracy	0.78	0.87	0.77	0.75
Recall	0.77	0.97	0.7	0.82
Precision	0.8	0.82	0.83	0.71
AUC	0.826	0.898	0.855	0.848

The following table represents the results obtained on different datasets using the base learners and the hybrid ensemble:

Table 5. Comparison of results for base learners and hybrid ensemble

Dataset\Method	Logistic Regression	Gradient Boosting Classifier	Random Forest	Gaussian Naïve Bayes	Hybrid Ensemble	K-fold validation
German	0.825	0.812	0.765	0.768	0.826	0.847
Australian	0.886	0.89	0.894	0.873	0.898	0.908
Give Me Some Credit	0.852	0.858	0.835	0.847	0.855	0.861
Mock Data	0.847	0.818	0.759	0.714	0.848	0.851

As observed, Logistic Regression performs best for German and Mock Datasets, Gradient Boosting Classifier works best for the the GiveMeSomeCredit credit dataset, and Random Forest for the the Australian dataset. The hybrid system outperforms the base learners for all the datasets except GiveMeSomeCredit. However, an important point to note is that the Hybrid system is trained only on partial datasets at this stage. Hence, we might lose many important samples, affecting the performance. To overcome this problem, we implemented the K-fold validation. If we look at the K-fold results, we observe that the hybrid system, when trained over the complete dataset, outperforms the base learners despite the different datasets. Using the bagging (Random forest) and boosting (Gradient Boosting classifier), we make our system robust enough to reduce the variance and the bias error. This is our desired outcome as we are looking for an ideal classifier system that provides the best results despite the variance of datasets, unlike the individual base learner model performance, which varies with every dataset.

5. Conclusions and future work

Our study proposed the complete procedure for designing a hybrid ensemble for efficient credit scoring analysis. Considering that real-world datasets are made up of inconsistent and uncorrelated data, it is necessary to build an efficient credit scoring system that performs the best, irrespective of the nature of the datasets that are provided to them. This was the idea of our study, and we were able to train and build an efficient system that outperforms the individual best base classifier performance. The data filtering approach helps us remove the inaccurate/unrelated samples from our datasets, helping the classifiers to distinguish between the classes efficiently and thus define the decision boundaries to be specific. We surveyed a lot of research that described better ways of preprocessing the data, using different approaches to build the ensemble methods. However, most of them pointed to the concern that although ensembles outperform the base learners, deciding which base learners would build an efficient ensemble is crucial. The proposed hybrid ensemble investigated these

concerns and aimed to use efficient, diverse learning algorithms to provide the best optimum results for each applied dataset.

If observed from the ensemble point of view, the Bagging (Random Forest) and Boosting (Gradient Boosting Classifier) perform the best for different data sets, which are Australian and Give Me Some Credit datasets, respectively, but when applied individually. However, since our stacking ensemble combines these methods, we take advantage of both baggings, which reduces variance, and boosting, which reduces bias error and proves itself to be a 'champion model.' Hence, our hybrid ensemble of preprocessed data with stacking proves to be a more robust and accurate system.

We have built our hybrid ensemble using the stacking approach with cross-validation, allowing both the base and the meta-learners to train on the full dataset. Our study of the stacking process brought forth some concerns, such as computational complexity. The more base learners there are, the more efficient the ensemble is; however, this could also slow down the analysis significantly. Parallel processing is a suitable solution for this issue, but again, we must assign each process its memory allocation in parallel processing. A bigger dataset and more base learners could mean much memory consumption and time complexity. This shortcoming can be addressed in future work.

References

[1] A. Chandra, X. Yao, "Evolving hybrid ensembles of learning machines for better generalization," Neurocomputing, vol.69, no.7-9, pp.686-700, March **(2006)**

[2] D. Browne, C. Manna, and S. Prestwich. "Relevance-redundancy dominance: A threshold-free approach to filter-based feature selection," In 24th Irish Conference on Artificial Intelligence and Cognitive Science, Sun SITE Central Europe/RWTH Aachen University, **(2016)**

[3] C. L. Huang, M. C. Chen, and C. J. Wang, "Credit scoring with a data mining approach based on support vector machines," Expert Systems with Applications, vol.33, no.4, pp.847-856, November **(2007)**

[4] F. N. Koutanaei, H. Sajedi, and M. Khanbabaei, "A hybrid data mining model of feature selection algorithms and ensemble learning classifiers for credit scoring," Journal of Retailing and Consumer Services, vol.27, pp.1-23, November **(2015)**

[5] S. Ha, N. Ha, N. Thi Bao, and Hien, "A hybrid feature selection method for credit scoring," EAI Endorsed Transactions on Context-aware Systems and Applications, vol.4, **(2017)** DOI: 10.4108/eai.6-3-2017.152335

[6] H. Wiryanto, "Credit scoring machine Learning with Keras - R," medium.com blog, February 11, **(2018)**, Available Online: https://medium.com/@heruwiryanto/credit-scoring-machine-learning-with-keras-r-502fc6eb451d

[7] I. Brown and C. Mues, "An experimental comparison of classification algorithms for imbalanced credit scoring data sets," Expert Systems with Applications, vol.39, no.3, pp.3446-3453, February 15 **(2012)**

[8] J. Kruppa, A. Schwarz, G. Arminger, and A. Ziegler, "Consumer credit risk: Individual probability estimates using machine learning," Expert Systems with Applications, vol.40, no.13, pp.5125-5131, October 1, **(2013)**

[9] M. Ala'raj and M. F. Abbod, "A new hybrid ensemble credit scoring model based on classifiers consensus system approach," Expert Systems with Applications, vol.64, pp.36-55, December, 1 **(2016)**

[10] N. C. Hsieh and L. P. Hung, "A data-driven ensemble classifier for credit scoring analysis," Expert Systems with Applications, vol.37, no.1, pp.534-545, January **(2010)**

[11] C. Serrano-Cinca, B. Gutiérrez-Nieto, and L. López-Palacios. "Determinants of default in P2P lending," PloS one 10, no.10 **(2015)**: e0139427

[12] Opitz, W. David, and R. Maclin. "Popular ensemble methods: An empirical study," J. Artif. Intell. Res. (JAIR) 11 **(1999)**: pp.169-198

This page is empty by intention.