Mohammad Farhad Hossain

STUDY OF VARIOUS MACHINE LEARNING APPROACHES TO PREDICT DEFAULT BEHAVIOR OF A BORROWER BASED ON TRANSACTIONAL DATASET

UNIVERSITY OF JYVÄSKYLÄ
FACULTY OF INFORMATION TECHNOLOGY
2021

ABSTRACT

Hossain, Mohammad Farhad
Study of Various Machine Learning Approaches to Predict Default Behavior of a Borrower Based on Transactional Dataset
Jyväskylä: University of Jyväskylä, 2021.
Cognitive Computing and Collective Intelligence, Master's Thesis
Supervisor(s): Oleksiy Khriyenko, Rahima Karimova (Data Analyst) and Ashkan Fredström (Project Developer, Credit Scoring)

Predicting 'default' behavior of borrowers is quite challenging and time consuming, although financial institutions require faster and more reliable decision on loan applications to survive in the competitive market. Availability of huge amount of data makes the work of current credit scoring system harder. To deal with such situation machine learning engineers are trying to build a system that can predict default behavior of a borrower by analyzing application and transaction data. In our current study we applied different machine learning models such as decision tree, logistic regression, gradient boosting, XGBoosting, support vector machine and KNeighbors on transactional dataset to find which model performed better. We also applied deep neural network on the datasets. To further extend the study, we created new features by using manual process and unsupervised machine learning to observe whether they boost the performance or not. In addition to that, we used feature selection to see how it affected the prediction. Due to small dataset, we achieved 70% accuracy with 72% AUC on aggregated dataset from Random Forest. The dataset created by using unsupervised machine learning showed 62% accuracy with 68% AUC value. Manually created ratio-based features and feature selection could not yield any significant difference in results. Deep learning also performed lower than others probably due to small dataset.

Keywords: machine learning, deep learning, credit scoring, transaction data, default behavior, loan application

FIGURES

TABLES

EQUATIONS

TABLE OF CONTENTS

1 INTRODUCTION

Loan is an important instrument in finance. It accelerates economic growth, increases purchase power, and provides support in difficult situations. Financial institutions, such as credit unions and banks provide loans to personal borrowers and other institutions.

The competition of disbursing loans to borrowers has increased among different financial institutions as the number of lending companies is growing rapidly. These lending companies are trying to attract borrowers by providing them loans faster and with less hassle and intervention. This situation throws new challenges to credit risk analysts. Because they need to process loan applications within short period of time maintaining quality of analysis. On the other hand, if the quality of analysis is poor then the number of default loans will increase and ultimately increase the risk of the institution.

For over five decades (Thomas et al., 2017), rule-based credit scoring has been used to maintain the quality of credit risk analysis and speed up the process. This credit scoring system can be compared to decision tree. It takes decision based on applicant's income, age, marital status, and other information.

However, rule-based credit system also has disadvantages. For example, some applicants may systematically hide or manipulate some data to get advantage while getting loans. In addition to that, rule-based credit scoring cannot deal with huge and complex data.

According to revised payment service directive of European union, also known as PSD2, companies are bound to provide data to third party – if the customer requires. This law opens the door for all financial institutions to gain access to large amount of transactional data of a customer. Here, transactional data means the bank statement that we receive from different banks consisting customer's transactions for specific period. The data contains time, amount and description of each transaction record. Thus, in the current context, we need to implement a system that can process huge amount of data and provide better insights and patterns.

Machine learning is famous for processing huge amount of data and discovering valuable insights from unstructured and structured data that we never

thought before. Different machine learning algorithms are used in different fields. Some algorithms are suitable for image recognition and others for time-series and tabular data analysis.

In such situation we can use machine learning to process this large amount of data or in other words big data. In fact, several studies were recently conducted to improve prediction accuracy of machine learning to identify possible `default` or `non-payment` borrowers based on different financial datasets. For example, Wang et al. (2020), assessed five machine learning models including decision tree, logistic regression and K-Nearest Neighbors on bank loan data and compared their performance, strengths and weaknesses.

Shen et al. (2020), used unsupervised machine learning to predict credit scoring and proposed a three-stage reject interface framework. They used a Chinese personal credit dataset to verify generality and applicability of their proposed learning framework. Golbayani et al. (2020), used neural network along with decision tree and support vector machines to forecast corporate credit rating. In addition to using general evaluation metrices, they introduced a new measure of accuracy that they called "Notch Distance".

In our study we also used different machine learning models to identify default borrowers based on their transactional data of 6 to 24 months.

1.1 Research Questions

In our current thesis, we shall be addressing the following questions:

How accurately machine learning models predict non-payment behavior of borrower? Which machine learning model perform on top of other models?

The accuracy of machine learning models differ from each other based on their configuration, type and volume of dataset. Even, with almost identical configuration, different models demonstrate different results. In our study we built decision tree, logistic regression, random forest, gradient boosting, XGBoost and Support vector classification, gaussian naïve bayes classification and K-neighbours models and find out how accurately machine learning models predict default behaviour.

Does feature creation on transactional data improve accuracy?

Machine learning engineers use feature creation technique to improve model accuracy, specially, when the dataset is very small and imbalanced. Here, they create new features from existing ones. For example, dividing total income by total days. However, it does not always guarantee the improvement. In our study, we created new features by using different formulas and unsupervised machine learning techniques and tried to find out whether they improve accuracy or not.

In our case, dataset was very small and imbalanced. Delegating feature creation task to deep learning could not help us to improve the result. Thus, we decided to create new features and explore the impact of them on model accuracy.

Does feature selection on transactional data improve accuracy?
Feature selection is another technique to improve model accuracy. In this technique, instead of supplying all features, we supplied only selected features that might improve accuracy.

In general, we know that deep learning does not require us to give extra effort for feature selection, as it handles feature selection itself. Thus, we tried deep learning model. However, due to small dataset, deep learning could not help us to get a good accuracy.

1.2 Structure and organization

We arranged this thesis paper into six main chapters including introduction. It introduces the topic and necessity of it with a brief background information.

Chapter 2 – literature review focused on previous literature, how they were related to current topic and how the topic was different from others.

In chapter 3 – methodology stated all the methods and techniques used in this study including data preprocessing, feature engineering, feature selection, splitting data and model selection.

Chapter 4 – Result began with summary and important findings of the results and gradually moves towards other results and findings.

Chapter 5 – interpreted and analyzed the results, provided answers to the research questions and possible future research.

Finally, Chapter 6 – conclusion summarizes the study and mentioned the recommendations to further improve the results.

It is worth to mention that we listed all the abbreviations used in this paper in `APPENDIX 1: Abbreviation` for convenience.

2 LITERATURE REVIEW

A set of decision models used by lenders to assess borrowers' ability to pay back loans are known as credit scoring. This scoring system determines how much credit can be disbursed to which borrower and what are the strategies to increase profitability. This model has become mainstream for all banks and credit unions to calculate credit risks. The credit scoring system is also used to assess the probability of loan defaults in each loan portfolio to meet banking regulations such as the 'Basel Accord'. (Thomas et al., 2017:1.)

The latest credit scoring trend hugely relies on different operational or statistical research methods that include decision trees, linear and logistic regressions (LC, 2000). Recently experts have started using artificial intelligence-based models, such as neural networks, nearest neighbor, genetic algorithm, etc (LC, 2000). They use a single model or a combination of them (LC, 2000). In the rest of this section, we shall discuss some studies that focus on these models to improve credit scoring, preceded by some related basic concepts.

Before diving deep into these models, it is worth mentioning that financial institutions gather two types of data while processing a loan application (Dastile et al., 2020). These are application data and behavioral data. Borrower's age, employment status, marital status, number of children or dependents, residence address and other information related to borrowers' demography are considered as application data (Dastile et al., 2020). On the other hand, borrowers' last twelve month's financial transaction data that reflects their average balance, missed payments, purchase history, etc are known as behavioral data (LC, 2000). Behavioral data not only helps financial institution to take decision about current loan application, but also to unveil new products to a particular segment of clients (LC, 2000). The behavioral data analysis can be done based on customer's own behavioral dataset or other past clients' dataset (LC, 2000).

2.1 Expert system and Machine learning

Ben-David & Frank (2009), coined credit scoring 'expert system' as rule based computerized system that is built on top of a collection of interviews of the experts of a particular field whereas machine learning models depend on past data without any further human involvement.

However, based on hit ratio and Kappa statistics Ben-David & Frank (2009), argued that, machine learning based classification models cannot significantly outperform expert systems, although regression results have advantages over expert systems. Thus, Ben-David & Frank (2009), suggested that by spending several man-years, machine learning model could be improved and made better than expert systems as the latter one took several years to come to its current acceptable position.

Khandani et al., (2010) constructed nonparametric and nonlinear models that forecast the credit risks of consumers. They combined data from the credit bureau and customer's transaction history categorized in different categories such as commodity or leisure expenditure, and account balance of over four years (Khandani et al., 2010). Before pouring the data into the machine learning model, Khandani et al., (2010) feature engineered by computing total deposit and withdrawal, number of transactions per month, the channels of transactions (e.g. ATM cash withdrawal, credit card payment), etc. With feature engineered data, they were able to forecast the monthly late payment or default behavior of customers 85% correctly by using linear regression R2 (Khandani et al., 2010).

Tsai & Chen (2010) combined different machine learning models and applied them on a real-world dataset of a bank in Taiwan. A combination of logistic regression-based classification and neural network classification (Classification + Classification) models has shown promising results (Tsai & Chen, 2010). In their study, they used three variations of dataset and three other variations of hybrid system - 'Clustering + Clustering', 'Clustering + Classification' and 'Classification + Clustering' (Tsai & Chen, 2010).

Trustorff et al., (2011) analyzed the performance between logistic regression and support vector machine models to classify and estimate 'the probability of default' - based on a dataset of financial ratios of more than seventy thousand financial statements collected between 2000 and 2006. They focused on small training dataset and high variance of the input data (Trustorff et al., 2011). Their calculation lead to a conclusion that, the performance of support vector machine model is significantly higher than logistic regression models (Trustorff et al., 2011).

However, there are some limitations of Support vector machines. To overcome these limitations, S. Li et al., (2012) for the first time examined the relevance vector machine (RVM) to analyze credit risks. Relevance vector machine is a ML model that exploits Bayesian inference to provide probabilistic classification and other benefits over SVM (Tipping, 2001). S. Li et al., (2012) applied

ensemble learning to further improve the result of RVM and obtained 98.5% testing accuracy on Australian credit dataset and 88% testing accuracy on Japanese credit dataset.

On the other hand, due to their performance, simplicity and speed, Kruppa et al., (2013) have chosen to implement K-nearest neighbors (kNN), Random Forests (RF) and bagged k-nearest neighbors (bNN) on a dataset consists of over 64 thousand short-term installment purchases. Kruppa et al., (2013) found some interesting correlations in their dataset. For example, The people who purchases in the afternoon are most likely employed and hence has a lower chance of becoming default than low-income young purchasers (Kruppa et al., 2013). Their study establishes that, Random forests using probability estimation trees (RF-PET) outperforms kNN, bNN and optimized logistic regression by demonstrating AUC value of 0.959.

2.2 Neural Network

In 2017 Luo et al., (2017) used one variant of neural network called deep belief network (DBN) on credit default swap (CDS) dataset and found that the performance of DBN is the best by comparing the result with some popular credit scoring models - such as - support vector machines, logistic regression and multilayer perceptron. They claim that DBN yields 100% accuracy on that dataset, though in general it is quite impossible and might have overfitting issue.

Addo et al., (2018) studied credit risk scoring on enterprise level by using four deep learning models, random forests and a gradient boosting machine. Their analysis illustrates that, random forests beat deep learning. The record set contains over one hundred thousand records of enterprise. Each record consists of 235 variables with labels derived from company's balance, financial statements and cash flows etc. (Addo et al., 2018).

The government of brazil took an initiative to finance low-income population to purchase home under the program of "My Home, My Life" program (Programa ''Minha Casa, Minha Vida'' — PMCMV), which is one of the largest home loan initiative in the world (de Castro Vieira et al., 2019). A database of PMCMV loans of 2.24 million contracts were anlyzed to predict default behavior of borrowers using Bagging, Random Forest and boosting models by de Castro Vieira et al., (2019). In the study de Castro Vieira et al., (2019) also examined the result of the models by removing discriminatory variables (age, gender, marital status). They drew a conclusion of the study that, default rate could be reduced by using these models from 11.80% to 2.95%.

Bao et al., (2019) has stepped forward and planned a strategy of combining unsupervised machine learning with supervised machine learning and apply the model to three different credit datasets: German, Australian and Chinese. They used four different strategies: individual models, individual models + consensus model, clustering + individual models, clustering + individual models + consensus model (Bao et al., 2019). Their result claims that the integration

of supervised and unsupervised machine learning algorithms achieve better performance than individual models (Bao et al., 2019).

The result of above study is furthered strengthened by a literature survey conducted by Dastile et al., (2020) based on 74 journals and articles published from 2010 to 2018. The survey indicates that ensemble of classifiers performs better than individual or single classifiers (Dastile et al., 2020). In addition to that, they also found that deep learning models show promising results, although these models are not extensively applied in credit scoring literature yet (Dastile et al., 2020).

From the above discussion we could easily figure it out that, most of the studies used application datasets rather than behavioral dataset (Khandani et al., 2010). Usage of Neural network-based models just started to roll in this field with promising results.

Thus, in our study we decided to explore the credit scoring with transactional dataset and fine tune the machine learning models to see how further they go hand in hand in terms of forecasting default behavior of a borrower. Although, at the beginning we wanted to explore deep neural network-based models, however, due to shortage of records, we mainly focused on general machine learning models.

3 METHODOLOGY

3.1 Summary

We applied different machine learning and deep learning models to borrowers' transactional dataset to predict or forecast their default behavior. We measured the performance of these models by using different evaluation metrics including AUC, Type 1, Type 2 error, recall and specificity. We discussed more detail about these key components (dataset, models, monitoring tools) and their selection criteria in this section.

3.2 Dataset and Data Preprocessing

3.2.1 Primary transaction Dataset

We used the transactional dataset in the current study. We received this data from one P2P lending financial institution. This dataset is collected from borrowers under the 'PSD2' guideline. Before providing this dataset, they anonymized and categorized this data.

Table 1 Distribution of classes

Class	Number of records	Description of the class
Default	99	Borrower did not pay their due in time
Non-Default	1024	Borrowers paid all of their dues in time

The dataset contained transactional data of 1,123 borrowers. Each transactional data of a borrower was provided by separate excel files. Thus, we re-

ceived 1,123 excel files of transactional data. Class distribution of this data is provided in Table 1.

Each excel file of transaction data contained CaseId, TransactionDate, Sum and Category column. Description of these columns are mentioned in Table 2.

Table 2 Details of transaction files

Column	Description
CaseId	This column contained numerical value unique to each case or customer loan application.
TransactionDate	This column contained the date when the transaction took place. The format of the date is yyyy-mm-dd. Here yyyy represents the year in four digits, mm represents month in two digits and dd means the day of the month in two dig-its. The actual time (i.e. hours, minutes and seconds) is removed before providing it to us to maintain anonymity.
Sum	The amount of transaction in numbers with a maximum of two decimal places. This number can be positive or negative. The positive number indicated cred-its to the account, whereas a negative number indicated debits from the ac-count.
Category	This column contained the type or category of transaction. All these categories are mentioned in Table 3.

Names of these csv files were constructed by using *tranactions_**CASE_ID**.xls.* Here **CASE_ID** corresponds to each loan application's unique ID that matched the CaseId column of excel file. In addition to that, one more excel file was provided that contained all customer's unique id (CaseId) and a column named `default` contained whether the customer became default or not. The name of this file was: *targets.xls*

Table 3 List of all transaction categories.

Categories			
debt-collection	relatives	City	travel
gambling	restaurants	news-media	pension
gas-station	secured-loan	Support	housing
groceries	self	furniture-utility	beauty
income	shopping	movie	investments
insurance	social-benefit	online-shop	sole-proprietorship
loan	exp-travel	energy	reading
medical-care	tax	cars-maint	cash-withdraw
none	unknown	credit-cards	car-purchase
parking	gaming	transport	
payment-provider	education	alcohol	
person	phone-internet	outdoors	

3.2.2 Aggregated value dataset (AG)

We derived a secondary dataset from the first one by aggregating values of 'SUM' column of each loan applicant's transaction data grouped by categories. By doing so, we converted all transaction data of each customer into a single row. This row contained the columns mentioned in Table 3. In addition to that, following columns were also added to each row:

default: the value of this column came from *targets.xls file* and represents whether the borrower paid the loan in time or not. If the borrower paid the loan in time, then the value was 0. On the other hand, if the borrower did not pay the loan in time, then the value was 1.

total-days: period of bank statements were different for different borrowers. This period ranged from six months to 24 months. We calculated days of each transaction period and insert them in total-days column.

case-id: unique id of each loan application.

Figure 1 (a) demonstrates the process of creating aggregated value (AG) dataset.

3.2.3 Feature creation by ratio dataset (FCR)

We created new features from existing 'aggregated value' (AG) dataset. In this dataset we created new features by calculating ratio of different expense features and income per day. We used following procedure to create new features:

per day income = total income / total-days

new feature = abs (expense feature) / per day income

Expense features were the debit accounts, and their amounts were usually negative. To avoid negative numbers, we used abs() method of python that returns absolute value of given number. We added 32 features in this dataset that are listed in Table 4.

Table 4 List of features created in FCR dataset.

Generated features		
alcohol_income_ratio	groceries_income_ratio	travel_income_ratio
beauty_income_ratio	housing_income_ratio	person_income_ratio
car_purchase_income_ratio	insurance_income_ratio	reading_income_ratio
cars_maint_income_ratio	investments_income_ratio	relatives_income_ratio
cash_withdraw_income_ratio	loan_income_ratio	restaurants_income_ratio
credi_cards_income_ratio	medical_care_income_ratio	secured_loan_income_ratio
energy_income_ratio	movie_income_ratio	self_income_ratio
exp_travel_income_ratio	online_shop_income_ratio	shopping_income_ratio
furniture_utility_income_ratio	outdoors_income_ratio	tax_income_ratio
gambling_income_ratio	parking_income_ratio	transport_income_ratio
gas_station_income_ratio	payment_provider_income_ratio	

In addition to above features, the dataset also contained per day income, default and case_id features which values were coming from 'aggregated value dataset' without any modification. Default and case_id features were described in Aggregated value dataset. Figure 1 (b) demonstrates the process of feature creation by ratio (FCR) dataset.

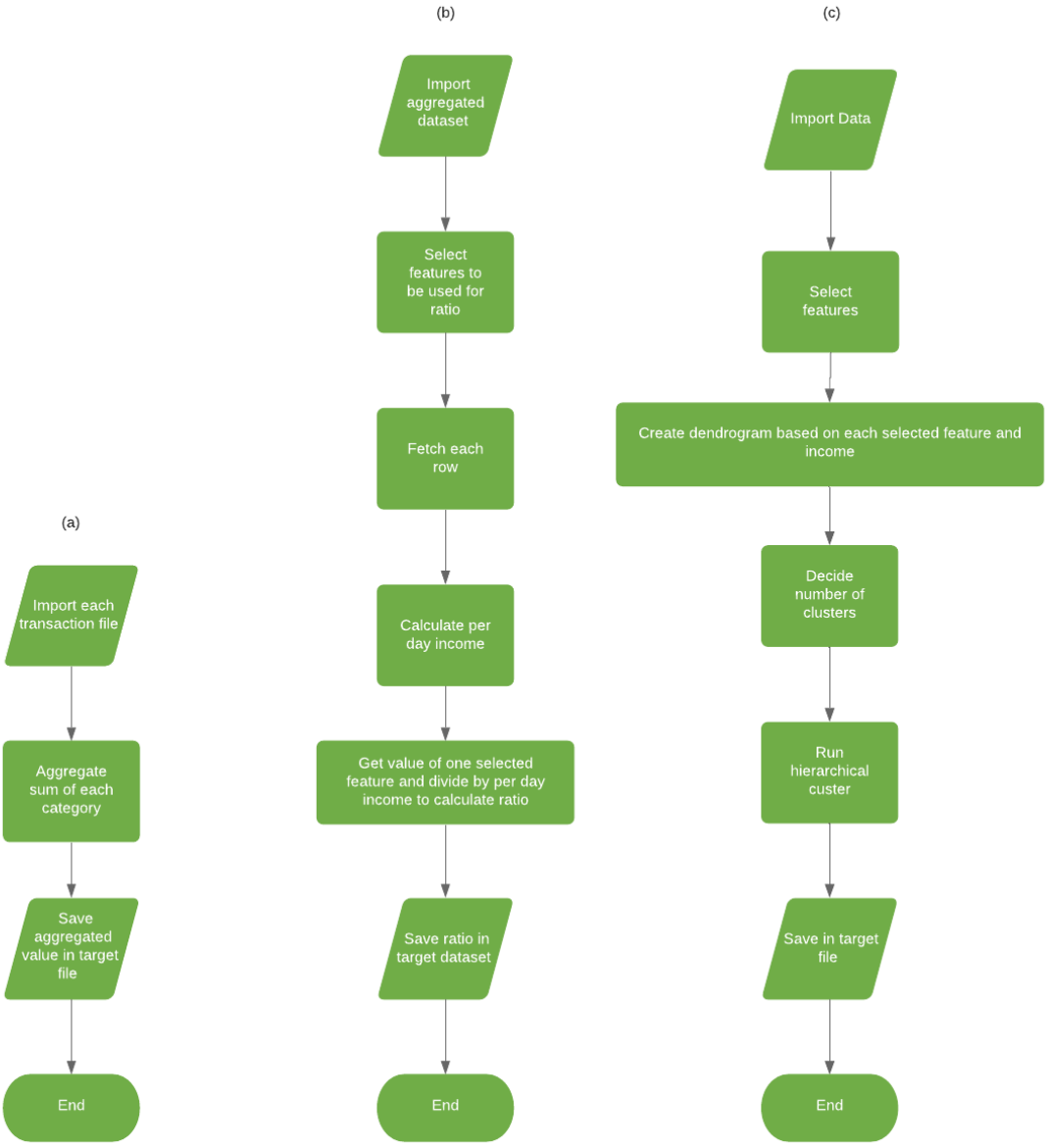

Figure 1 (a) Process of generating aggregated value dataset, (b) Process of creating feature generated ratio dataset, (c) Process of feature generation by using hierarchical clustering.

3.2.4 Feature creation by clustering: unsupervised machine learning (FCRC)

We created a new dataset by using unsupervised machine learning, more specifically hierarchical clustering with agglomerative approach. The dataset contained all features of 'feature creation by ratio' (FCR) dataset and 29 new features. These new features were generated by using following steps:

1. Decided which features should we select for clustering purpose, as all features are not feasible. For example, we did not select furniture-utility expense as it is not always related to borrower's loan payment capability.
2. Created dendrograms of income and 29 other features taken from 'aggregated value' dataset. These other features are listed in Table 5.
3. Based on dendrograms we decided the number of clusters. Most of the cases we decided to use a greater number of clusters than suggested by dendrograms, as we wanted to find hidden cluster that might have hidden correlation with credit scoring.
4. We applied hierarchical clustering on income and 29 other selected features of aggregated value dataset and appended those resulting clusters in FCR dataset to create new dataset named FCRC.
5. We repeated the above processes to see which features were creating good clusters. Finally, we repeated the clustering with only finalized features and exported the resulting clusters in csv file.

Table 5 List of clusters used in FCRC dataset

Clusters	
income_debt-collection_cluster	income_education_cluster
income_gambling_cluster	income_phone-internet_cluster
income_gas-station_cluster	income_city_cluster
income_groceries_cluster	income_furniture-utility_cluster
income_insurance_cluster	income_online-shop_cluster
income_loan_cluster	income_energy_cluster
income_medical-care_cluster	income_transport_cluster
income_payment-provider_cluster	income_alcohol_cluster
income_person_cluster	income_pension_cluster
income_relatives_cluster	income_housing_cluster
income_restaurants_cluster	income_investments_cluster
income_secured-loan_cluster	income_sole-proprietorship_cluster
income_self_cluster	income_reading_cluster
income_shopping_cluster	income_car-purchase_cluster
income_social-benefit_cluster	

Our system generated clusters in numbers, such as 1, 2, 3. To avoid ranking of numbers, we converted these clusters to categorical features by concatenating *`cluster_`* string before these numbers. Later we converted these clusters

to dummy variables before training the models. Figure 1 (c) demonstrates the process of feature clustering (FCRC) dataset.

3.3 Resampling the datasets

At the very beginning, we applied machine learning algorithms on aggregated value (AG) dataset. Surprisingly we noticed that most of the prediction accuracy of models were 87.23%. Then we took a closer look at confusion matrix of these test results. The confusion matrix showed that we were facing accuracy paradox. All predictions were only one class and that was 'non-default' class.

The main reason behind this accuracy paradox was imbalanced data. The data contained 1024 non-default rows and only 99 default rows. That is, only 9.67% data belonged to 'default' class and rest were 'non-default' class.

Machine learning engineers use resampling method to overcome the issue imposed by imbalanced dataset. There are two types of resampling method. They are up-sampling and down-sampling. We used both methods.

3.3.1 Up-sampling

In up-sampling, we duplicated the records that belonged to minor class to match the number of major class. For example, if there were 100 minor classes and 900 major classes, then we duplicated 100 minor records 8 times to become 900. Thus the resulting dataset contains 900 records of each class. In our case, we used sklearn's resample method to automate the upsampling process. Thus, all of our datasets had this upsampling step by default.

3.3.2 Down-sampling

In down-sampling, we removed the records of major class to match the number of minor class. We did this manually by following the steps mentioned below:

1. Copied all records of minor class in 8 new csv files.
2. Copied 100 records of major class of main dataset to one of these new csv files. We repeated this process for all 8 csv files. Note that, we copied different records for each csv file so major records were not repeated in any of these files.
3. We repeated above process for each main dataset – AG, FCR and FCRC and ended up creating 24 more datasets.

3.4 Feature selection

First, we provided data to train models without any selection. That is, we input all data of each dataset for training the model. In the second round, we omitted some features to observe, whether the accuracy improved or not.

3.4.1 Process of feature selection

We made the feature selection automatic by using scikit learn's SelectFromModel class of feature_selection package. We used Random Forest as its estimator model.

The SelectFromModel class run Random Forest model to predict the class. Then the class took the best features from estimator. We setup a pipeline to automate feature selection and then use those features to train models. All the datasets used in this study are listed in Table 6

Table 6 List of all datasets used in this study.

Dataset Name	Total records	Default class	Non-default class	Number of features	Resample technique	Description
AG	1123	99	1024	47	Up-sample	Prepared from transactional dataset
FCR	1123	99	1024	34	Up-sample	Prepared from AG dataset
FCRC	1123	99	1024	63	Up-sample	Prepared from FCR by using clustering method.
AG_1 to AG_7	200	99	100	47	Down-sample	Each dataset derived from AG dataset. Manually down-sampled to solve imbalanced data. Here 99 records belong to default class. 100 unique records are picked from major non-default class
FCR_1 to FCR_7	200	99	100	34	Down-sample	Each dataset derived from FCR dataset. Manually down-sampled. Here 99 records belong to default class. 100 unique records are picked from major non-default class
FCRC_1 to FCRC_7	200	99	100	63	Down-sample	Each dataset derived from FCRC dataset. Manually down-sampled. Here 99 records belong to default class. 100 unique records are picked from major non-default class

3.5 Splitting training set and test set.

While training the model we divided datasets into two sets. They were training set and test set. Test set contained 20% of the whole data. Training set contained the rest. Sklear's train_test_split method was used to automate the process. We repeated the same splitting process before training a model by each dataset. Thus, no training and test datasets overlapped with each other. Note that, we used **`123` as value of random parameter** of the method. If anybody wants to get the same training and test dataset, then they have to use the same random value.

3.6 Model selection

We applied logistic regression, decision tree, random forests, extreme gradient boosting, gradient boosting, support vector classifier, Gaussian Naïve Bayes and K Neighbors on the transaction dataset to predict borrowers' default behavior. We also trained dataset by using deep neural network and analyzed the prediction result.

3.6.1 Logistic regression

Logistic regression is a popular statistical model to solve binary or classification problem (Logistic Regression - Wikipedia, n.d.). Primarily it is used when the number of classes is only two. That's why it better suits in credit scoring to classify a borrower as good or bad. The logistic regression can be mathematically expressed as follows:

$$\log [p (1 - p)] = \beta 0 + \beta 1 X 1 + \beta 2 X 2 + \ldots + \beta n X n \quad (1)$$

here, p is the default probability; βi are the coefficients of independent variables and X i are independent variables.

Table 7 shows the parameters used in Logistic regression model which were obtained from Grid search hyperparameter tuning method.

Table 7 Logistic regression parameters

Parameter name	Configuration
C (Inverse of regularization strength)	100
max_iter	100
penalty	l2
solver	lbfgs

3.6.2 Decision tree

In its simplest form, decision tree is a 'question–answer' or 'if-else' statement-based model that is used in solving both classification and regression problems. If we combine both classification and regression in a decision tree then it is called Classification and regression trees (CART) (Decision Tree Learning - Wikipedia, n.d.). It is a non-parametric classification and widely used on credit scoring (Lee et al., 2006). Table 8 depicts the main configuration of decision tree that is used to train the model:

Table 8 Decision Tree parameters

Parameter name	Configuration
criterion	gini
splitter	best
max_depth	4
min_samples_split	2
min_samples_leaf	0
max_features	None
random_state	None
max_leaf_nodes	None
min_impurity_decrease	0
min_impurity_split	0
class_weight	None
ccp_alpha	0

3.6.3 Random Forest

Multiple decision tree predictors are combined to form random forests (Breiman, 2001). Here each decision tree depends on random vector values that are independently sampled, and the same distribution is used in all trees in the forest (Breiman, 2001).

Table 9 Random forest parameters

Parameter name	Configuration
n_estimators	50
criterion	gini
max_depth	None
min_samples_split	50
min_samples_leaf	3
min_weight_fraction_leaf	0
bootstrap	TRUE
random_state	None

Random forest is one of the top tree-based machine learning models (Wallis et al., 2019). That is why we decided to use random forests in our study. Table 9 presents the main parameters and their values of random forest classifier that was used to train the model.

3.6.4 Extreme Gradient Boosting

Also known as 'XGBoost' - is a scalable end-to-end tree boosting system (Chen & Guestrin, 2016) that builds decision trees in parallel (Nobre & Neves, 2019). It is famous for its performance and processing speed (Nobre & Neves, 2019). Only a few latest credit scoring studies focused on XGBoost (Xia et al., (2018), Chang et al., (2018), Li et al., (2018), Cao et al., (2018)). Thus, we decided to explore XGBoost as it has already shown promising results.

Table 10 shows the parameters used in XGBoost model which were obtained from Grid search hyperparameter tuning method.

Table 10 XGBoost parameters

Parameter name	Configuration
colsample_bytree	0.94
learning_rate	0.1
n_estimators	100
subsample	0.83

3.6.5 Gradient boosting

Gradient boosting is a machine learning algorithm that is used for regression, classification and ranking. Here weak learning models are combined to create a strong model.

Table 11 Gradient Boosting parameters

Parameter name	Configuration
n_estimators	100
learning_rate	0.1
max_depth	5
random_state	None

3.6.6 Support Vector Classifier

Support vector machine is a supervised machine learning that separates the classes by using hyperplane (decision boundary) in high dimensional feature space. (Cortes & Vapnik, 1995). This model can be used in both regression and classification problems. We used SVC in our study with the parameters mentioned in Table 12

Table 12 Support vector classifier parameters

Parameter name	Configuration
C	0.1
gamma	0.1
kernel	sigmoid

3.6.7 Gaussian Naïve Bayes

Based on baye's theorem, simple probabilistic classifiers were created, which are known as Naïve Baye's classifier. Different methodologies were used to implement this classification. For example, Gaussian naïve Bayes, Multinomial naïve Bayes, Bernoulli naïve Bayes. In our study, we used Gaussian naïve Bayes that is also suitable for continuous data.

Table 13 Gaussian Naive Bayes parameters

Parameter name	Configuration
var_smoothing	0.000284804

3.6.8 K Neighbors Classifier

We also used K Neighbors classifier which is a non-parametric model used in both classification and regression (Fix & Hodges, 1951). The classifier forms groups or clusters based on provided two-dimensional array of dataset.

Table 14 KNeighbors parameters

Parameter name	Configuration
metric	manhattan
n_neighbors	17
weights	uniform

3.6.9 Deep neural network

Deep neural network, which mimics structure of biological neurons were also used in our study to predict credit scoring. We used Tensorflow to implement the deep neural network by using the following configuration:

Model: Sequential

Layer (Type)	Output shape	Param #
dense (Dense)	(None, 142)	20306
dense_1 (Dense)	(None, 512)	73216
dense_2 (Dense)	(None, 512)	262656
dense_3 (Dense)	(None, 1)	1026

3.7 Hyperparameter optimization

Each machine learning model have their own set of parameters. We can set different values to each parameter that yields different accuracy. The process of searching for parameters that produces the best accuracy is called hyperparameter optimization. Different approaches are used for this purpose. In our study we used Grid Search approach.

In this approach, we set different set of parameters for ML models. Then we try each parameter set to train model and find the best accuracy. Scikit-learn's GridSearchCV was used to automate the whole process.

3.8 Performance measurements

The first thing that we check after training a model is its accuracy. The main goal of machine learning engineers is to improve the accuracy. However, accuracy of a model does not always mean that the model's performance is also high. For example, if any model of binary classification predicts only one class (i.e. 0 or 1) and if majority of the records of test set contain that particular class, then the accuracy is always high, although in real world implementation that model would perform the worst. These types of errors are known as accuracy paradox.

To avoid such issue and find out underlying real performance of a model, we used five popular evaluation metrics. These metrics were i. Area under curve (AUC), ii. Type I Error, iii. Type II Error, iv. Recall and vi. Specificity. Before introducing these metrics, it is worth to mention the abbreviation of few terms. They are as TP = true positive, TN = True negative, FP = False positive and FN = False negative.

Area Under Curve (AUC) measures the capability of a model to distinguish different classes. Higher AUC value represents the better performance of a model. Equation 1 presents the mathematical formula of AUC.

Equation 1 AUC

$$AUC = \frac{1}{2}\left(1 + \frac{TP}{TP+FN} - \frac{FP}{TN+FP}\right)$$

Type I (Equation 2) and Type II (Equation 3) errors deal with wrongly identified classes. Type I focuses on incorrectly identified positive class and Type II incorrectly identified negative classes. A model performs better when the value of these two evaluation metrices are lower.

Equation 2 Type I error

$$Type\ I = \frac{FP}{TN + FP}$$

Equation 3 Type II error

$$Type\ II = \frac{FN}{TP + FN}$$

Recall calculates how many positive cases were correctly identified. On the other hand, specifity calcuates how many negative cases were correctly identified.

Equation 4 and Equation 5 represents these two evaluation metrices. In our case, recall metrix is more important. Because, we want to know how many borrowers became default, which class is represented by 1 or positive.

Equation 4 Recall

$$Recall = \frac{TP}{TP + FN}$$

Equation 5 Specificity

$$Specificity = \frac{TN}{TN + FP}$$

Equation 6 illustrates the accuracy of a machine learning model. Also known as Percentage Correctly Classified (PCC) is a simple metrics that presents correctly identified classes out of total test samples.

Equation 6 Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Figure 2 illustrates the over-all process of training model and evaluation of the results.

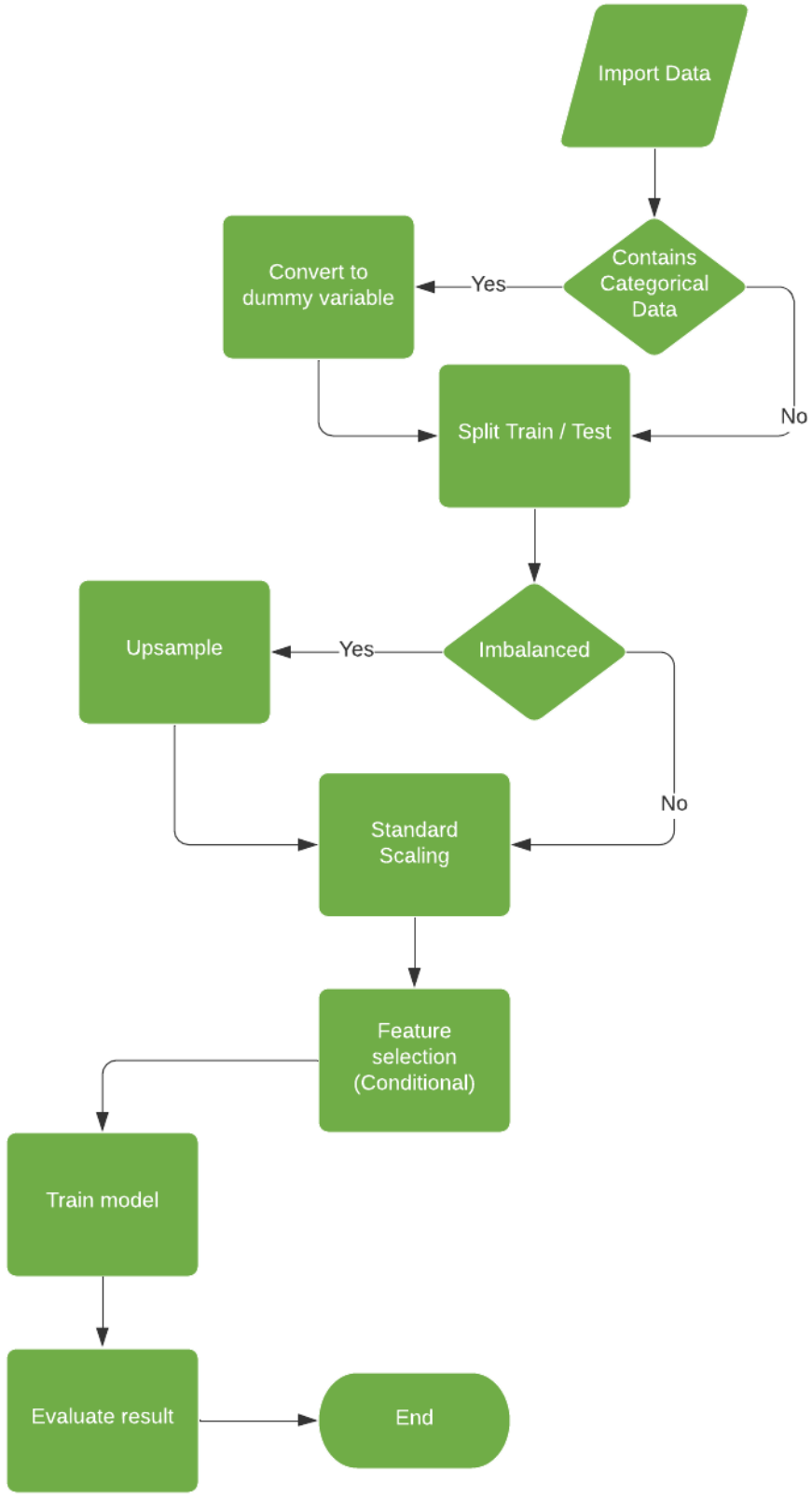

Figure 2 Training model and evaluate performance.

4 Result

In this section, we included top ten results of prediction based on AUC and accuracy including other evaluation metrics. Then we illustrated all the individual metrics and accuracy in graphs. In the third section, we presented confusion metrics of these predictions to better understand the results.

4.1 Summary of top 10 results

We created and trained total 408 models (including 24 deep learning models) with combination of different classifiers, datasets and feature selections. Among them we selected top 10 results based on AUC and accuracy. We summarized those 10 results and illustrated them with configurations and evaluation matrices – AUC, Type I error, Type II error, Recall, and Specificity.

Table 15 Top 10 classifiers with configurations and aggregated results

Dataset	Subset	Classifier Name	Feature Selection	AUC	Type 1	Type 2	Recall	Specificity	Accuracy
AG	Downsampled	RandomForest	TRUE	72%	31%	25%	75%	69%	70%
FCRC	Downsampled	XGB	TRUE	68%	39%	25%	75%	61%	62%
FCRC	Downsampled	GradientBoosting	FALSE	66%	44%	25%	75%	56%	58%
FCR	Downsampled	GradientBoosting	TRUE	66%	44%	25%	75%	56%	58%
AG	Downsampled	LogisticRegression	FALSE	65%	45%	25%	75%	55%	56%
FCRC	Downsampled	RandomForest	FALSE	65%	45%	25%	75%	55%	56%
FCRC	Downsampled	XGB	FALSE	65%	39%	31%	69%	61%	61%
AG	Downsampled	LogisticRegression	TRUE	64%	34%	38%	63%	66%	66%
FCR	Downsampled	GradientBoosting	FALSE	64%	41%	31%	69%	59%	60%
AG	Upsampled	KNeighbors	TRUE	64%	35%	38%	63%	65%	65%

Table 16 Top 10 classifiers with configurations based on AUC and accuracy.

Dataset	Subset	Classifier Name	Feature Selection	AUC	Type 1	Type 2	Recall	Spec-ificity	Accuracy
FCR_5	Downsampled	RandomForest	TRUE	73%	48%	6%	94%	52%	55%
AG_4	Downsampled	RandomForest	TRUE	72%	36%	19%	81%	64%	65%
FCRC_4	Downsampled	RandomForest	FALSE	70%	41%	19%	81%	59%	60%
AG_3	Downsampled	RandomForest	FALSE	70%	35%	25%	75%	65%	66%
AG_7	Downsampled	RandomForest	FALSE	68%	33%	31%	69%	67%	68%
AG_7	Downsampled	RandomForest	TRUE	68%	33%	31%	69%	67%	67%
AG_5	Downsampled	XGB	FALSE	68%	33%	31%	69%	67%	67%
FCRC_7	Downsampled	RandomForest	FALSE	67%	47%	19%	81%	53%	55%
FCR_3	Downsampled	XGB	FALSE	67%	48%	19%	81%	52%	54%
FCRC_7	Downsampled	XGB	FALSE	67%	42%	25%	75%	58%	60%

Table 15 illustrates results of up-sampled datasets and aggregated results of down-sampled subsets. Table 16 contains top 10 results of up-sampled datasets and non-aggregated results of down-sampled subsets. Detailed results are provided in 'APPENDIX 2: All Results Without Aggregation)' and '

APPENDIX 3: All Results (Downsampled Datasets Aggregated)'.

From Table 15 we can see, Random Forest model performed better than all other models. Aggregated value dataset with down-sampled dataset helped models to perform better than up-sample technique. We did not see any major difference by turning on or off the feature selection. Clustered dataset FCRC also achieved better results than manually created features by different ratio dataset FCR. From the table we can see that, AG dataset appeared in the top 10 four times, FCRC four times and FCR two times.

Table 15 - that consists aggregated results of down-sampled datasets and up-sampled datasets - shows that Random Forest model trained on AG dataset with feature selection achieved 70% accuracy with AUC value 72%. XGB model based on FCRC model and feature selection technique attained 62% accuracy with AUC value 68%. Other good performing models were Gradient Boosting, Logistic Regression and KNeighbours on both aggregated datasets (AG) and cluster-based feature created datasets (FCRC). Although Gradient Boosting model trained on FCR dataset by turning on the feature selection positioned 4th place due to higher AUC, its accuracy is only 58%. However, FCR based models performed better while we turned off the feature selection technique (60% accuracy with 64% AUC). Decision tree, GaussianNB and even SVC models could not demonstrate good results. According to Table 16, smaller and down-sampled datasets obtained maximum 73% AUC.

We also trained and applied deep neural network to see how accurately it could identify default behavior of borrowers. However, we could not achieve any better result than other generic machine learning models. The results of deep learning models were shown in Table 17. Here we can see that FCRC based down-sampled sequential model achieved 57% accuracy with 57% AUC.

Table 17 Results of deep neural network model (Classifier name: Sequential)

Dataset	Subset	Feature Selection	AUC	Type 1	Type 2	Recall	Specificity	Accuracy
FCRC	Downsampled	FALSE	57%	43%	44%	56%	57%	57%
AG	Downsampled	FALSE	53%	44%	50%	50%	56%	56%
AG	Upsampled	FALSE	53%	7%	88%	13%	93%	87%
FCRC	Upsampled	FALSE	49%	15%	88%	13%	85%	80%
FCR	Downsampled	FALSE	43%	70%	44%	56%	30%	32%
FCR	Upsampled	FALSE	41%	30%	88%	13%	70%	66%

4.2 Relation between accuracy and evaluation matrices

This section represents combo charts of accuracy and other evaluation matrices. All the charts contain accuracy as line in orange color. Other evaluation metrics were shown as bars in different colors.

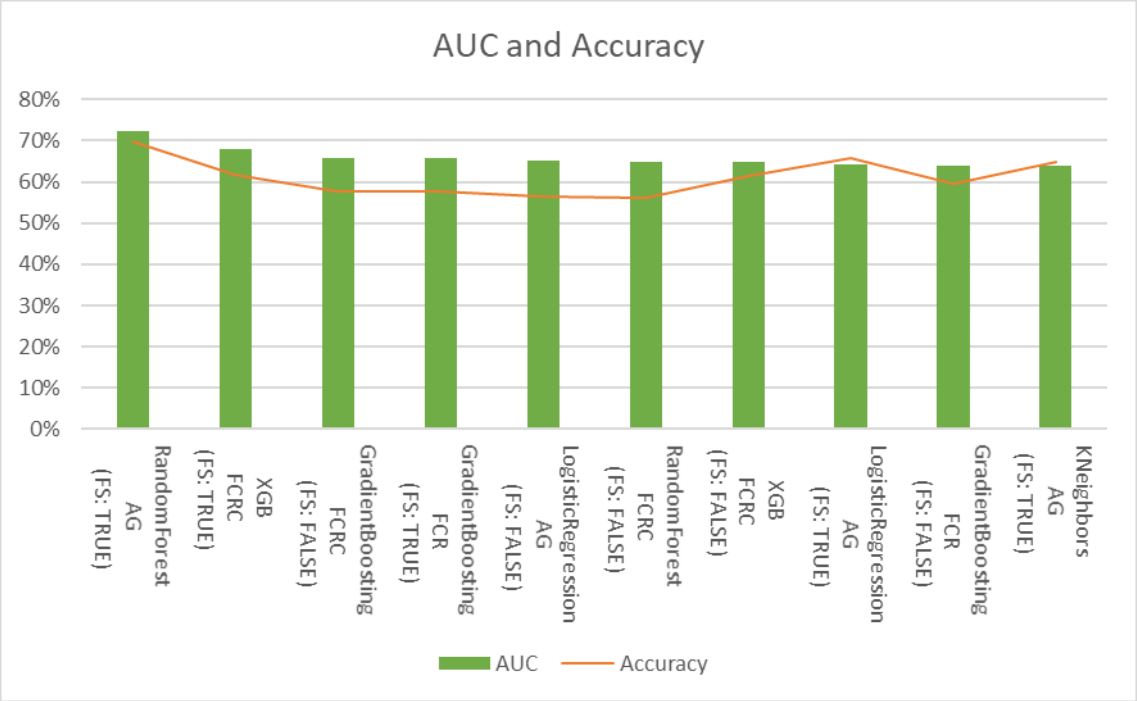

Figure 3 shows AUC and Accuracy.

The acceptability of a model can be verified by AUC value of that model's test result. Figure 3 shows that our top model Random Forest's AUC value was 72%.

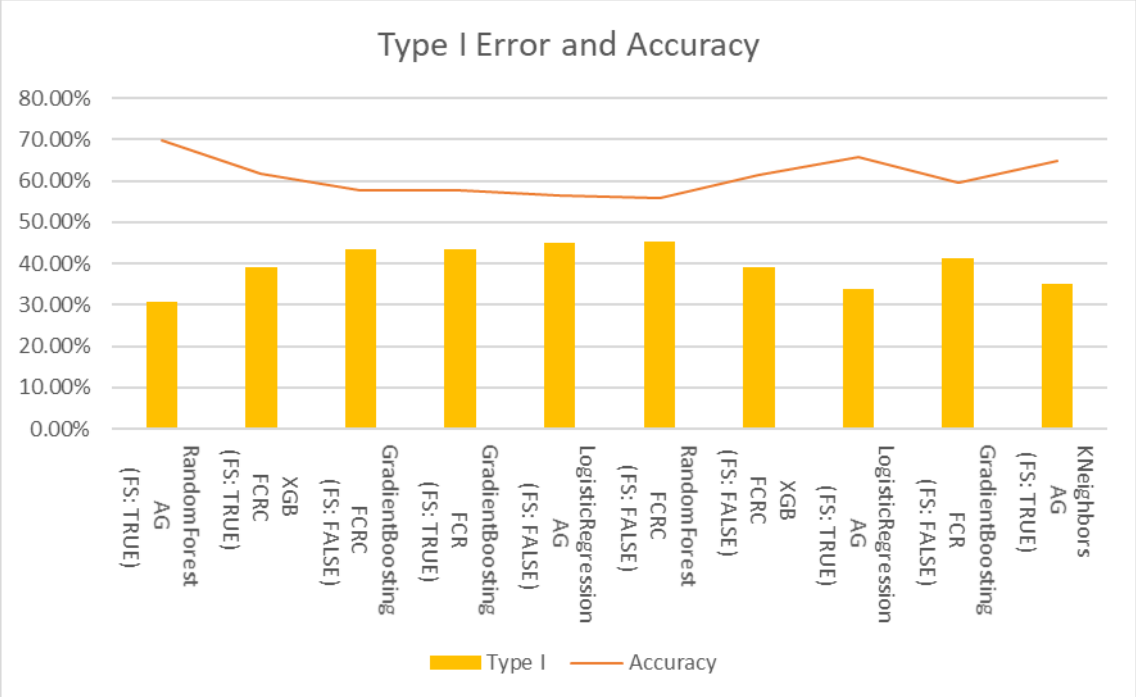

Figure 4 shows Type I error and Accuracy.

From Figure 4 we can see that Type I error was high (45%) in Random Forest model on FCRC dataset, although its AUC was also 65%. Higher value of Type I error means that the model identifies borrowers as defaulters although they paid in time. Wrongly identifying a borrower as defaulter and not giving them loan reduces the total amount of loan disbursement and increases dissatisfaction among potential customers. Our top model - Random forest's Type I error was low and that was 30.26%.

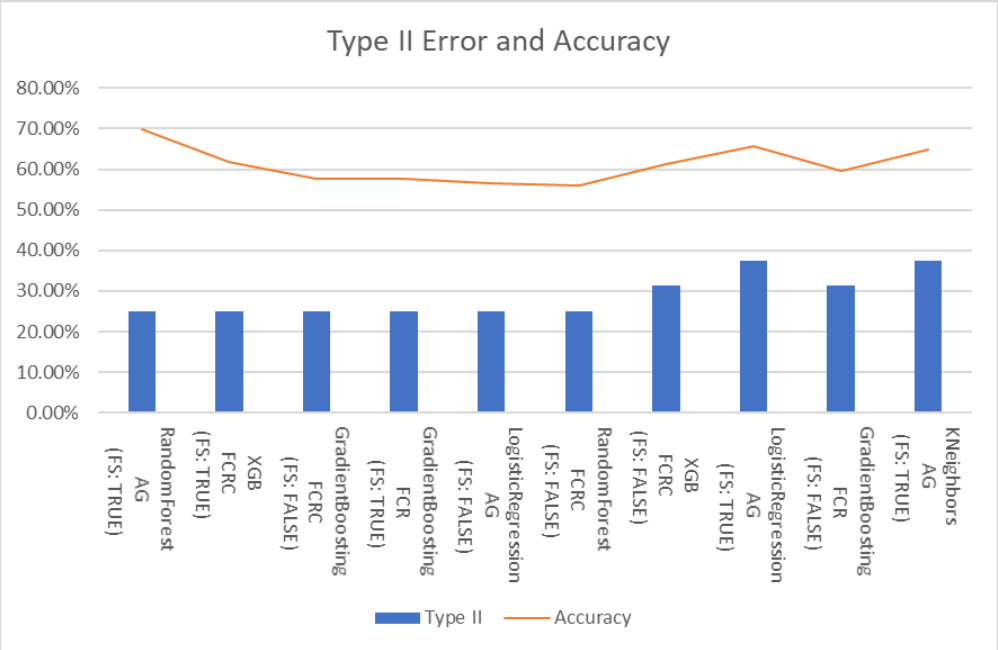

Figure 5 shows Type II error and Accuracy.

Figure 5 shows that, Type II error of Logistic regression and kNeighbors models on AG dataset was 37.50%, which means that it identified 38% borrowers as regular paying good borrower, although they did not pay their installments in time. Higher the Type II error increases the risk of bad loan of a financial institution. On the other hand, top six models including Random Forest exhibited lowest Type II error (25%).

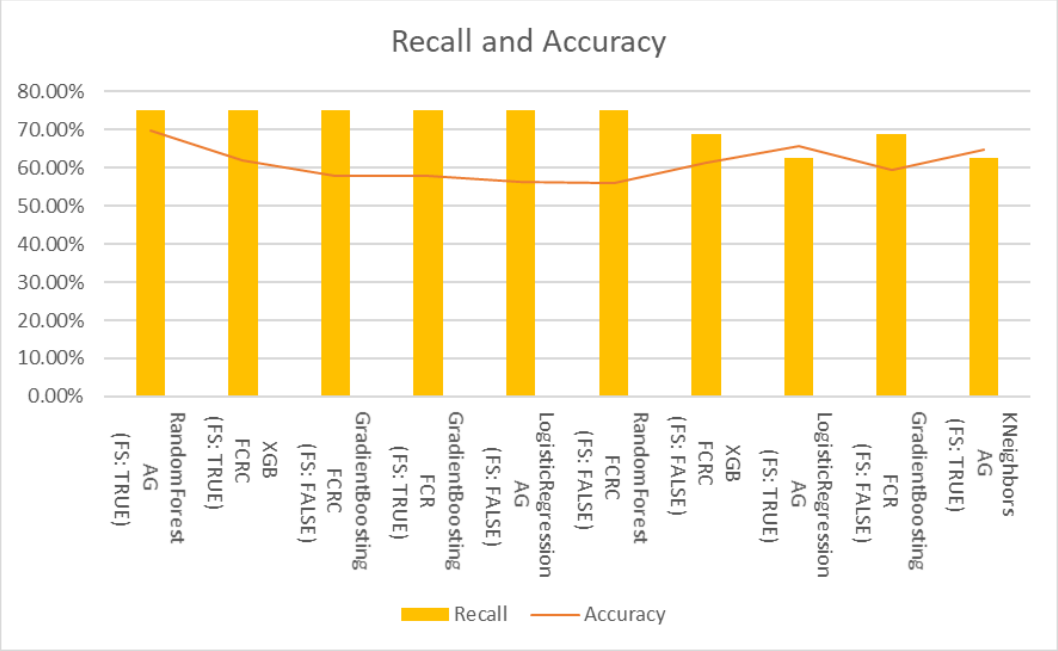

Figure 6 shows Recall and Accuracy.

Figure 6 shows that top six models identified most default borrowers correctly, which is 75%.

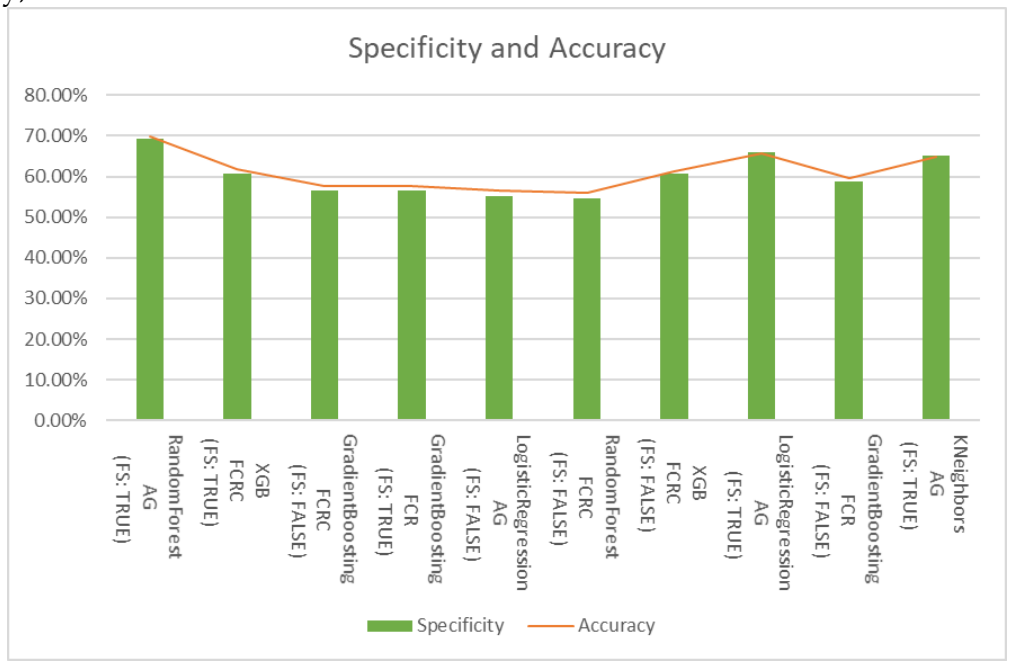

Figure 7 shows Specificity and Accuracy.

Figure 7 shows that Random Forest model on AG dataset were able to correctly identify non-default or in other words good borrowers in 69% cases. The accuracy of this model was 70%.

From above graphs we can see that, our top model, Random forest's best performance was also supported by all other evaluation matrices.

4.3 Confusion matrix of top 10 results

This section houses confusion matrix of top 10 predictors and top one deep learning model.

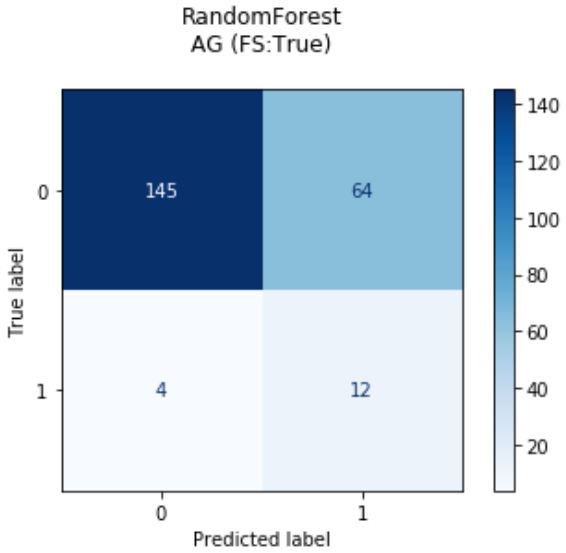

Figure 8 Confusion matrix of RandomForest on AG with feature selection

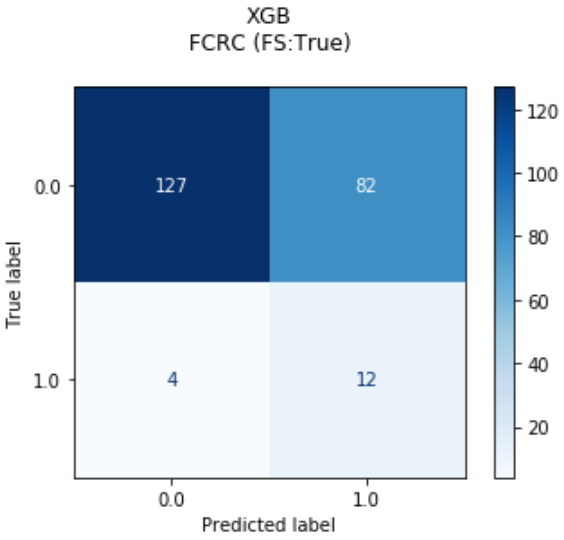

Figure 9 Confusion matrix of XGB on FCRC with feature selection

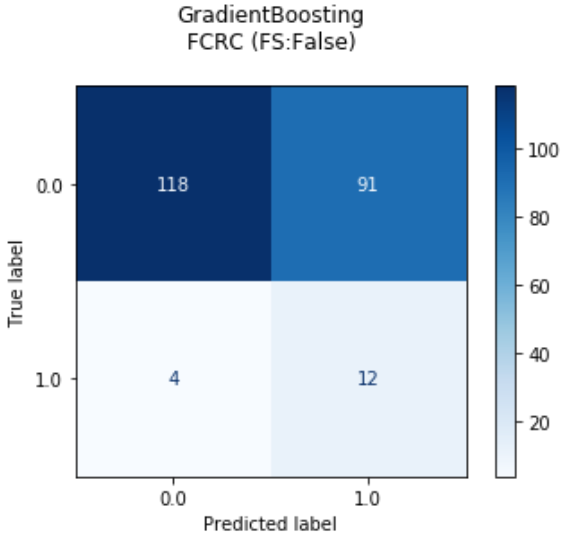

Figure 10 Confusion matrix of GradientBoosting on FCRC without feature selection

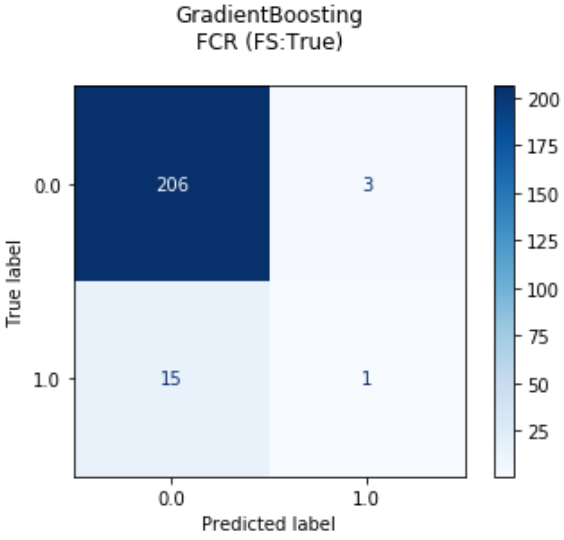

Figure 11 Confusion matrix of GradientBoosting on FCR with feature selection

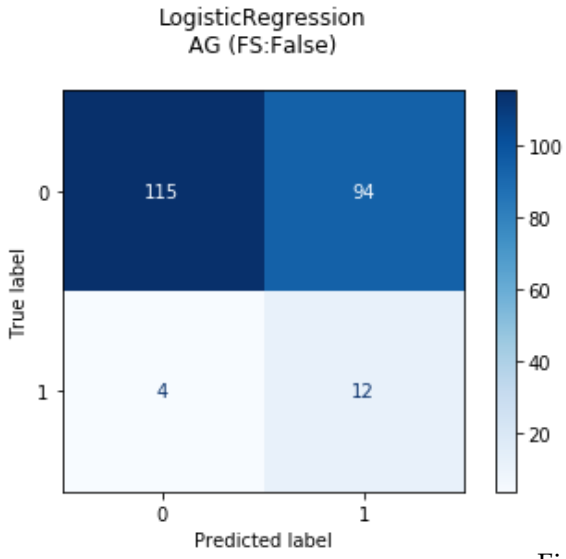

Figure 12 Confusion matrix of LogisticRegression on AG without feature selection

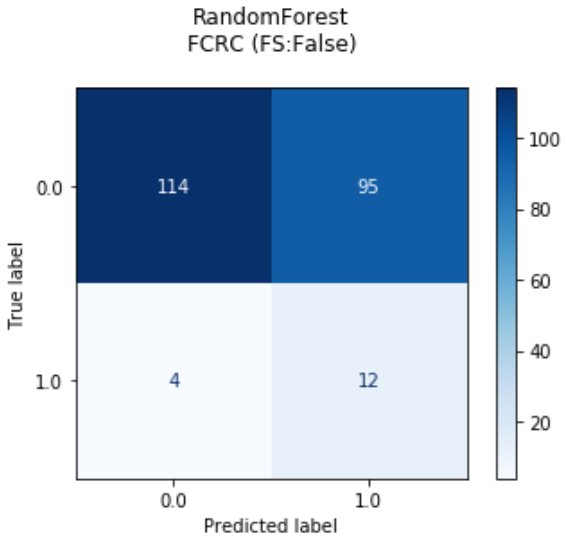

Figure 13 Confusion matrix of RandomForest on FCRC without feature selection

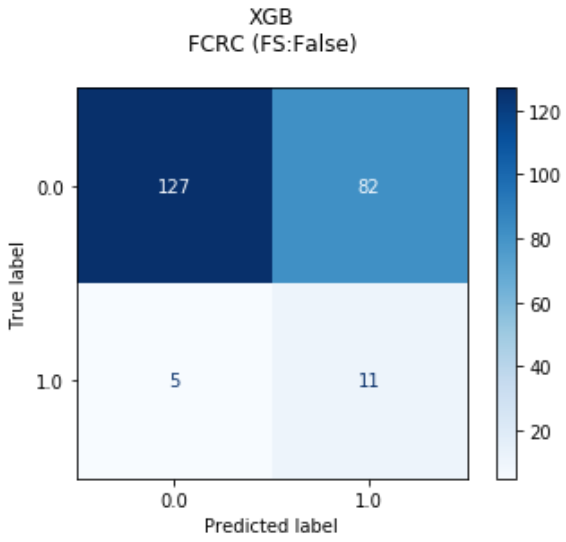

Figure 14 Confusion matrix of XGB on FCRC without feature selection

Figure 15 Confusion matrix of LogisticRegression on AG with feature selection

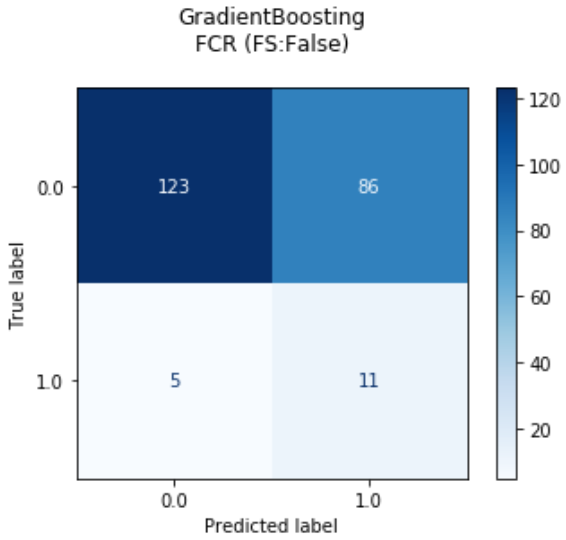

Figure 16 Confusion matrix of GradientBoosting on FCR without feature selection

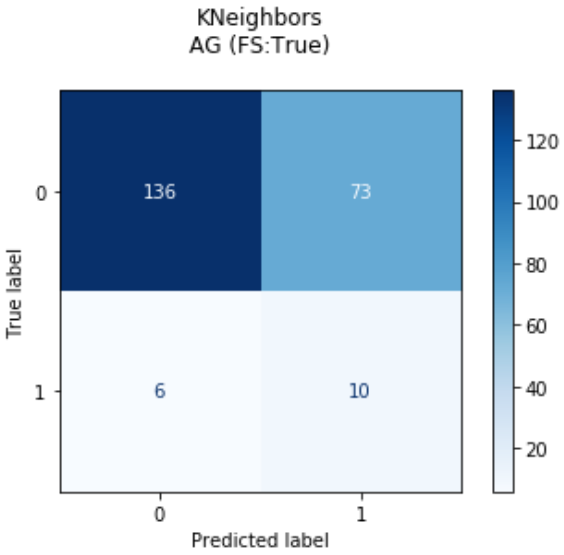

Figure 17 Confusion matrix of KNeighbors on AG with feature selection

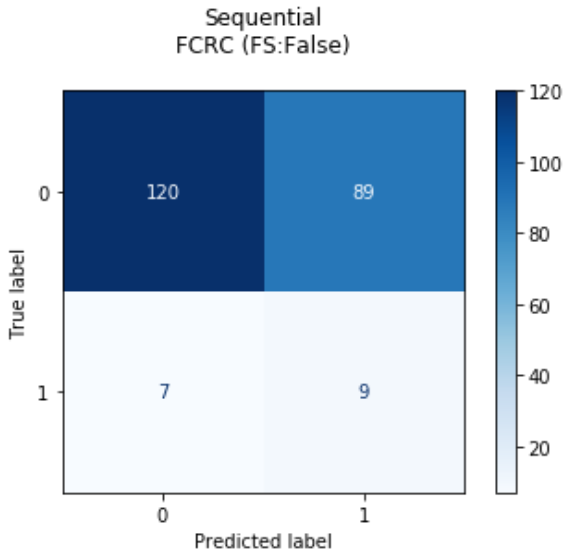

Figure 18 Confusion matrix of Sequential on FCRC without feature selection

5 DISCUSSION

The initial objective of the study was to find how accurately machine learning could predict default behavior of a borrower. Most of the studies conducted earlier used application-based dataset, whereas we used transactional data. Khandani et al., (2010) combined credit bureau data with customer's transaction history and obtained 85% accuracy. They did not mention how many records did their dataset contained.

In our study, we got 70% accuracy with AUC value of 72%. Although these accuracy and AUC value may not be excellent, however, within acceptable range and promizing. We also must consider that our aggregated dataset consisted only about 1,123 records. Among these records, the class ratio was 9:1, which was too much imbalanced. Thus, for this small set of imbalanced data, the result was significant. Random Forest classifier model obtained this accuracy on down-sampled aggregated dataset (AG) and supported by all evaluation matrices. We assume that the accuracy could be improved further with more records.

The second research question was to know whether the feature engineering – more specifically feature creation – improves the accuracy of prediction. We feature engineered the aggregated value (AG) dataset to create new features. Two methods were used to create new features and build datasets.

In the first method we simply calculated the ratio of income and other selected features. The datasets created from this method were feature creation ratio (FCR) and it's down-sampled subsets (FCR_1 to FCR_7). Only two of these datasets appeared in the top ten performer's list.

On the other hand, we used unsupervised machine learning approach (hierarchical clustering) to generate features and create new datasets. These datasets were feature creation ratio cluster (FCRC) and it's down-sampled subsets (FCRC_1 to FCRC_7). The FCRC dataset appeared 5 times in top ten list and in the 2nd position with 62% accuracy and 68% AUC.

With respect to the third research question, we found that feature selection performed slightly better in terms of accuracy result. Because, top two results configured with feature selection technique. In our study we automated the feature selection by using sklearn's SelectFromModel method with Random Forest estimator. We believe that researchers can further extend this study by doing

manual feature selection and see the impact of each feature. Or they may combine both manual and automated feature selection and observe the difference.

5.1 Limitation

The primary dataset that was provided by the company contained 1,350,591 rows of transaction data of 1,123 borrowers. However, after we aggregated all the amounts grouped by transaction categories and created a new dataset containing one row for each borrower loan application, then the total rows became 1,123. This small size of dataset is quite challenging to get better accuracy from machine learning models. Moreover, small sized dataset usually leads to overfitting.

The second limitation of this study was imbalanced data. Non-default class contained 91.18% data and default class contained 8.82% data. This type of imbalanced data created accuracy paradox. Although we tried to balance the classes of dataset by using up-sampling and down-sampling, still it was far from originality.

One borrower might have several accounts. If such borrower provided only one account transaction data and if that data did not contain specific transaction (e.g. gambling, large amount of loan, alcohol), then it was impossible to get full picture of that borrower.

6 CONCLUSION

It is difficult to conclude based on the result of models built on top of 1,123 rows with imbalanced classes. The result shows 70% maximum accuracy with AUC value of 72%. The result also indicates that Random Forest, XGB and Gradient Boosting outperformed all other models. Cluster based feature engineering showed good result. Feature selection performed slightly better than its counterpart. However, feature engineered ratio-based dataset could not assist models to achieve good accuracy results.

Machine learning has many branches and subbranches. For example, supervised machine learning, unsupervised machine learning, reinforcement learning. Exploring all these branches and their sub-branches require huge amount of time and effort which is out of scope of this study. However, we tried to implement eight supervised and one unsupervised machine learning, and one deep learning model. These include decision tree random rainforest, k-nearest neighbour classification, support Vector classification. We implemented one unsupervised machine learning - hierarchical clustering – to create new features. We also implemented deep neural networks, though they could not outperform generic machine learning models, probably because of small dataset.

Due to shortage of time and lack of feasibility reinforcement learning could not be implemented. We hope that future researchers will look at reinforcement learning and try to explore deep learning further on transactional data set for credit scoring.

6.1 Recommendation

At the very end of this paper, we want to suggest that in practical situation, use large dataset to improve the accuracy. However, make sure that the dataset is balanced. If not, then use manual or automatic re-sampling before providing the data to model training process.

There is a saying in machine learning that garbage in – garbage out. It means that, if we train our model with huge unnecessary data, then the result will also affect the accuracy. Thus, in case of general machine learning models pay attention to feature selection.

7 REFERENCES

Addo, P., Guegan, D., & Hassani, B. (2018). Credit Risk Analysis Using Machine and Deep Learning Models. *Risks*, *6*(2), 38. https://doi.org/10.3390/risks6020038

Bao, W., Lianju, N., & Yue, K. (2019). Integration of unsupervised and supervised machine learning algorithms for credit risk assessment. *Expert Systems with Applications*, *128*, 301–315. https://doi.org/10.1016/j.eswa.2019.02.033

Ben-David, A., & Frank, E. (2009). Accuracy of machine learning models versus "hand crafted" expert systems - A credit scoring case study. *Expert Systems with Applications*, *36*(3 PART 1), 5264–5271. https://doi.org/10.1016/j.eswa.2008.06.071

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1023/A:1022627411411

Dastile, X., Celik, T., & Potsane, M. (2020). Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied Soft Computing Journal*, *91*, 106263. https://doi.org/10.1016/j.asoc.2020.106263

de Castro Vieira, J. R., Barboza, F., Sobreiro, V. A., & Kimura, H. (2019). Machine learning models for credit analysis improvements: Predicting low-income families' default. *Applied Soft Computing Journal*, *83*, 105640. https://doi.org/10.1016/j.asoc.2019.105640

Fix, E., & Hodges, J. L. (1951). *Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties*. USAF School of Aviation Medicine, Randolph Field, Texas. https://apps.dtic.mil/dtic/tr/fulltext/u2/a800276.pdf

Golbayani, P., Florescu, I., & Chatterjee, R. (2020). A comparative study of forecasting corporate credit ratings using neural networks, support vector machines, and decision trees. *North American Journal of Economics and Finance*, *54*(April), 101251. https://doi.org/10.1016/j.najef.2020.101251

Khandani, A. E., Kim, A. J., & Lo, A. W. (2010). Consumer credit-risk models via machine-learning algorithms. *Journal of Banking and Finance*, *34*(11), 2767–2787. https://doi.org/10.1016/j.jbankfin.2010.06.001

Kruppa, J., Schwarz, A., Arminger, G., & Ziegler, A. (2013). Consumer credit risk: Individual probability estimates using machine learning. *Expert Systems with Applications*, *40*(13), 5125–5131. https://doi.org/10.1016/j.eswa.2013.03.019

LC, T. (2000). A survey of credit and behavioural scoring: Forecasting financial risk of lending to consumers. *International Journal of Forecasting*, *16*, 149.

Li, S., Tsang, I. W., & Chaudhari, N. S. (2012). Relevance vector machine based infinite decision agent ensemble learning for credit risk analysis. *Expert Systems with Applications*, *39*(5), 4947–4953. https://doi.org/10.1016/j.eswa.2011.10.022

Luo, C., Wu, D., & Wu, D. (2017). A deep learning approach for credit scoring using credit default swaps. *Engineering Applications of Artificial Intelligence*, *65*(October 2016), 465–470. https://doi.org/10.1016/j.engappai.2016.12.002

Shen, F., Zhao, X., & Kou, G. (2020). Three-stage reject inference learning framework for credit scoring using unsupervised transfer learning and three-way decision theory. *Decision Support Systems*, *137*(February), 113366. https://doi.org/10.1016/j.dss.2020.113366

Thomas, L., Crook, J., & Edelman, D. (2017). *Credit scoring and its applications*. https://epubs.siam.org/doi/pdf/10.1137/1.9781611974560.bm

Tipping, M. E. (2001). Sparse Bayesian Learning and the Relevance Vector Machine. *Journal of Machine Learning Research*, *1*, 211–244. http://jmlr.csail.mit.edu/papers/v1/tipping01a.html

Trustorff, J.-H., Konrad, P. M., & Leker, J. (2011). Credit risk prediction using support vector machines. *Review of Quantitative Finance and Accounting*, *36*(4), 565–581. https://doi.org/10.1007/s11156-010-0190-3

Tsai, C. F., & Chen, M. L. (2010). Credit rating by hybrid machine learning techniques. *Applied Soft Computing Journal*, *10*(2), 374–380. https://doi.org/10.1016/j.asoc.2009.08.003

Wang, Y., Zhang, Y., Lu, Y., & Yu, X. (2020). A Comparative Assessment of Credit Risk Model Based on Machine Learning ——a case study of bank loan data. *Procedia Computer Science*, *174*, 141–149. https://doi.org/10.1016/j.procs.2020.06.069

8 APPENDICES

8.1 APPENDIX 1: Abbreviation

XGB	Extreme gradient boosting
TP	True positive
TN	True negative
FP	False positive
FN	False negative
AG	Aggregated value dataset
FCR	Feature creation ratio
FCRC	Feature creation ratio clustering
AUC	Area under curve
PCC	Percentage Correctly Classified
SVC	Support vector classifier
FS	Feature selection

8.2 APPENDIX 2: All Results Without Aggregation

Dataset	Subset	Classifier Name	Feature Selection	AUC	Type 1	Type 2	Recall	Specificity	Accuracy
FCR_5	Downsampled	RandomForest	TRUE	73%	48%	6%	94%	52%	55%
AG_4	Downsampled	RandomForest	TRUE	72%	36%	19%	81%	64%	65%
FCRC_4	Downsampled	RandomForest	FALSE	70%	41%	19%	81%	59%	60%
AG_3	Downsampled	RandomForest	FALSE	70%	35%	25%	75%	65%	66%
AG_7	Downsampled	RandomForest	FALSE	68%	33%	31%	69%	67%	68%
AG_7	Downsampled	RandomForest	TRUE	68%	33%	31%	69%	67%	67%
AG_5	Downsampled	XGB	FALSE	68%	33%	31%	69%	67%	67%
FCRC_7	Downsampled	RandomForest	FALSE	67%	47%	19%	81%	53%	55%
FCR_3	Downsampled	XGB	FALSE	67%	48%	19%	81%	52%	54%
FCRC_7	Downsampled	XGB	FALSE	67%	42%	25%	75%	58%	60%
AG_5	Downsampled	RandomForest	TRUE	67%	35%	31%	69%	65%	65%
AG_3	Downsampled	XGB	TRUE	66%	42%	25%	75%	58%	59%
AG_5	Downsampled	LogisticRegression	TRUE	66%	43%	25%	75%	57%	58%
FCRC_5	Downsampled	XGB	TRUE	66%	37%	31%	69%	63%	64%
FCR_6	Downsampled	XGB	TRUE	66%	44%	25%	75%	56%	58%
AG	Downsampled	Sequential	FALSE	66%	44%	25%	75%	56%	58%
AG_2	Downsampled	RandomForest	TRUE	65%	38%	31%	69%	62%	63%
FCRC_5	Downsampled	XGB	FALSE	65%	38%	31%	69%	62%	63%
FCRC_5	Downsampled	GradientBoosting	FALSE	65%	44%	25%	75%	56%	57%
AG_3	Downsampled	XGB	FALSE	65%	38%	31%	69%	62%	62%
AG_6	Downsampled	LogisticRegression	TRUE	65%	39%	31%	69%	61%	62%
AG_4	Downsampled	LogisticRegression	TRUE	65%	27%	44%	56%	73%	72%
FCRC	Downsampled	Sequential	FALSE	65%	40%	31%	69%	60%	61%
FCR_5	Downsampled	XGB	FALSE	64%	40%	31%	69%	60%	60%
AG	Upsampled	KNeighbors	TRUE	64%	35%	38%	63%	65%	65%
FCRC_1	Downsampled	GaussianNB	TRUE	64%	10%	63%	38%	90%	86%
FCRC_5	Downsampled	RandomForest	TRUE	64%	48%	25%	75%	52%	54%
FCR_7	Downsampled	GradientBoosting	TRUE	64%	42%	31%	69%	58%	59%
FCRC_6	Downsampled	RandomForest	FALSE	64%	42%	31%	69%	58%	59%
FCR_2	Downsampled	GradientBoosting	FALSE	63%	48%	25%	75%	52%	53%
FCR_2	Downsampled	GradientBoosting	TRUE	63%	49%	25%	75%	51%	53%
FCR_6	Downsampled	RandomForest	TRUE	63%	49%	25%	75%	51%	53%
FCR_7	Downsampled	XGB	TRUE	63%	43%	31%	69%	57%	58%
FCRC	Downsampled	Sequential	FALSE	63%	43%	31%	69%	57%	58%
AG	Upsampled	KNeighbors	FALSE	63%	30%	44%	56%	70%	69%
FCR_7	Downsampled	XGB	FALSE	63%	43%	31%	69%	57%	58%
AG_4	Downsampled	GradientBoosting	FALSE	63%	37%	38%	63%	63%	63%
FCR_1	Downsampled	GaussianNB	FALSE	63%	18%	56%	44%	82%	79%
FCR_2	Downsampled	DecisionTree	TRUE	63%	56%	19%	81%	44%	47%
AG_6	Downsampled	LogisticRegression	FALSE	63%	44%	31%	69%	56%	57%
FCRC_3	Downsampled	RandomForest	FALSE	62%	56%	19%	81%	44%	46%
FCR_2	Downsampled	DecisionTree	FALSE	62%	56%	19%	81%	44%	46%

Dataset	Subset	Classifier Name	Feature Selection	AUC	Type 1	Type 2	Recall	Specificity	Accuracy
AG_4	Downsampled	XGB	FALSE	62%	38%	38%	63%	62%	62%
FCRC_3	Downsampled	SVC	FALSE	62%	69%	6%	94%	31%	35%
FCR_3	Downsampled	GradientBoosting	FALSE	62%	51%	25%	75%	49%	51%
FCR_4	Downsampled	GradientBoosting	TRUE	62%	51%	25%	75%	49%	51%
FCRC_1	Downsampled	RandomForest	TRUE	62%	44%	31%	69%	56%	56%
AG_1	Downsampled	DecisionTree	TRUE	62%	38%	38%	63%	62%	62%
AG_5	Downsampled	LogisticRegression	FALSE	62%	45%	31%	69%	55%	56%
FCRC_6	Downsampled	XGB	FALSE	62%	45%	31%	69%	55%	56%
FCR_7	Downsampled	RandomForest	TRUE	62%	52%	25%	75%	48%	50%
AG_3	Downsampled	RandomForest	TRUE	62%	45%	31%	69%	55%	56%
AG_5	Downsampled	KNeighbors	FALSE	62%	27%	50%	50%	73%	72%
FCRC_2	Downsampled	XGB	FALSE	61%	46%	31%	69%	54%	55%
FCRC_3	Downsampled	XGB	TRUE	61%	46%	31%	69%	54%	55%
AG_7	Downsampled	KNeighbors	TRUE	61%	33%	44%	56%	67%	66%
AG_6	Downsampled	SVC	TRUE	61%	34%	44%	56%	66%	65%
FCR_5	Downsampled	DecisionTree	TRUE	61%	72%	6%	94%	28%	33%
FCRC_2	Downsampled	GradientBoosting	FALSE	61%	47%	31%	69%	53%	54%
FCRC_7	Downsampled	RandomForest	TRUE	61%	41%	38%	63%	59%	60%
FCR_5	Downsampled	XGB	TRUE	61%	41%	38%	63%	59%	60%
FCRC_7	Downsampled	XGB	TRUE	61%	41%	38%	63%	59%	60%
FCRC_3	Downsampled	XGB	FALSE	61%	41%	38%	63%	59%	60%
FCRC_4	Downsampled	GaussianNB	TRUE	61%	16%	63%	38%	84%	81%
AG_7	Downsampled	LogisticRegression	FALSE	61%	47%	31%	69%	53%	54%
AG_4	Downsampled	KNeighbors	FALSE	61%	47%	31%	69%	53%	54%
FCRC_5	Downsampled	LogisticRegression	FALSE	61%	41%	38%	63%	59%	59%
AG_3	Downsampled	KNeighbors	TRUE	60%	35%	44%	56%	65%	64%
AG_7	Downsampled	DecisionTree	TRUE	60%	23%	56%	44%	77%	75%
FCRC_7	Downsampled	SVC	FALSE	60%	11%	69%	31%	89%	85%
FCR_5	Downsampled	SVC	TRUE	60%	55%	25%	75%	45%	48%
FCR_5	Downsampled	GradientBoosting	FALSE	60%	42%	38%	63%	58%	58%
FCRC_5	Downsampled	DecisionTree	TRUE	60%	36%	44%	56%	64%	64%
AG_1	Downsampled	RandomForest	TRUE	60%	36%	44%	56%	64%	64%
AG_4	Downsampled	RandomForest	FALSE	60%	36%	44%	56%	64%	64%
AG	Downsampled	Sequential	FALSE	60%	36%	44%	56%	64%	64%
FCRC_7	Downsampled	LogisticRegression	TRUE	60%	49%	31%	69%	51%	52%
AG_1	Downsampled	RandomForest	FALSE	60%	43%	38%	63%	57%	58%
FCRC_5	Downsampled	GradientBoosting	TRUE	60%	43%	38%	63%	57%	58%
AG_6	Downsampled	XGB	FALSE	60%	36%	44%	56%	64%	63%
FCR_2	Downsampled	XGB	TRUE	60%	49%	31%	69%	51%	52%
FCR_5	Downsampled	GradientBoosting	TRUE	60%	43%	38%	63%	57%	57%
FCRC_7	Downsampled	LogisticRegression	FALSE	60%	43%	38%	63%	57%	57%
FCR_5	Downsampled	DecisionTree	FALSE	60%	37%	44%	56%	63%	63%
AG_1	Downsampled	DecisionTree	FALSE	60%	37%	44%	56%	63%	63%
FCR_4	Downsampled	SVC	TRUE	60%	68%	13%	88%	32%	36%
FCR_5	Downsampled	LogisticRegression	FALSE	59%	50%	31%	69%	50%	52%

Dataset	Subset	Classifier Name	Feature Selection	AUC	Type 1	Type 2	Recall	Specificity	Accuracy
FCRC	Upsampled	SVC	FALSE	59%	50%	31%	69%	50%	52%
FCRC_4	Downsampled	GradientBoosting	FALSE	59%	44%	38%	63%	56%	57%
FCR_7	Downsampled	GradientBoosting	FALSE	59%	44%	38%	63%	56%	57%
FCRC_6	Downsampled	GradientBoosting	FALSE	59%	44%	38%	63%	56%	57%
FCR_1	Downsampled	XGB	FALSE	59%	37%	44%	56%	63%	62%
AG_6	Downsampled	GradientBoosting	FALSE	59%	37%	44%	56%	63%	62%
FCRC_5	Downsampled	DecisionTree	FALSE	59%	37%	44%	56%	63%	62%
AG_7	Downsampled	KNeighbors	FALSE	59%	25%	56%	44%	75%	73%
AG_4	Downsampled	SVC	FALSE	59%	19%	63%	38%	81%	78%
FCRC_4	Downsampled	SVC	FALSE	59%	12%	69%	31%	88%	84%
FCR_1	Downsampled	SVC	TRUE	59%	81%	0%	100%	19%	24%
FCRC_1	Downsampled	SVC	TRUE	59%	75%	6%	94%	25%	30%
AG_3	Downsampled	DecisionTree	FALSE	59%	50%	31%	69%	50%	51%
FCR_5	Downsampled	RandomForest	FALSE	59%	50%	31%	69%	50%	51%
AG_4	Downsampled	XGB	TRUE	59%	38%	44%	56%	62%	62%
AG_1	Downsampled	GradientBoosting	TRUE	59%	32%	50%	50%	68%	67%
FCR_7	Downsampled	LogisticRegression	TRUE	59%	69%	13%	88%	31%	35%
FCR_3	Downsampled	XGB	TRUE	59%	44%	38%	63%	56%	56%
FCRC_6	Downsampled	SVC	TRUE	59%	26%	56%	44%	74%	72%
FCRC_3	Downsampled	SVC	TRUE	59%	76%	6%	94%	24%	29%
AG_3	Downsampled	LogisticRegression	FALSE	59%	57%	25%	75%	43%	45%
FCRC_4	Downsampled	XGB	FALSE	59%	45%	38%	63%	55%	56%
FCR_6	Downsampled	GradientBoosting	FALSE	59%	39%	44%	56%	61%	61%
AG_6	Downsampled	KNeighbors	TRUE	59%	26%	56%	44%	74%	72%
FCR_6	Downsampled	XGB	FALSE	59%	45%	38%	63%	55%	55%
FCRC_1	Downsampled	LogisticRegression	TRUE	59%	45%	38%	63%	55%	55%
AG_5	Downsampled	XGB	TRUE	59%	39%	44%	56%	61%	60%
AG_7	Downsampled	XGB	TRUE	58%	33%	50%	50%	67%	66%
FCRC_3	Downsampled	LogisticRegression	TRUE	58%	58%	25%	75%	42%	44%
AG	Downsampled	Sequential	FALSE	58%	52%	31%	69%	48%	49%
FCR_5	Downsampled	LogisticRegression	TRUE	58%	46%	38%	63%	54%	55%
AG_5	Downsampled	GradientBoosting	FALSE	58%	40%	44%	56%	60%	60%
AG_7	Downsampled	XGB	FALSE	58%	33%	50%	50%	67%	65%
FCR_6	Downsampled	LogisticRegression	TRUE	58%	59%	25%	75%	41%	44%
FCR_2	Downsampled	XGB	FALSE	58%	46%	38%	63%	54%	54%
AG_6	Downsampled	XGB	TRUE	58%	40%	44%	56%	60%	60%
FCRC_7	Downsampled	GradientBoosting	FALSE	58%	40%	44%	56%	60%	60%
AG_4	Downsampled	SVC	TRUE	58%	15%	69%	31%	85%	81%
FCRC_2	Downsampled	GaussianNB	TRUE	58%	9%	75%	25%	91%	86%
FCR_1	Downsampled	LogisticRegression	TRUE	58%	53%	31%	69%	47%	48%
FCRC_3	Downsampled	GradientBoosting	TRUE	58%	53%	31%	69%	47%	48%
FCR_6	Downsampled	LogisticRegression	FALSE	58%	53%	31%	69%	47%	48%
AG_6	Downsampled	RandomForest	FALSE	58%	47%	38%	63%	53%	54%
AG_6	Downsampled	GradientBoosting	TRUE	58%	41%	44%	56%	59%	59%
FCRC_4	Downsampled	KNeighbors	TRUE	58%	34%	50%	50%	66%	64%

Dataset	Subset	Classifier Name	Feature Selection	AUC	Type 1	Type 2	Recall	Specificity	Accuracy
AG_7	Downsampled	DecisionTree	FALSE	58%	28%	56%	44%	72%	70%
AG_7	Downsampled	SVC	TRUE	58%	22%	63%	38%	78%	75%
FCRC_2	Downsampled	RandomForest	TRUE	58%	47%	38%	63%	53%	53%
FCRC_4	Downsampled	XGB	TRUE	58%	47%	38%	63%	53%	53%
FCR_6	Downsampled	RandomForest	FALSE	58%	47%	38%	63%	53%	53%
AG_4	Downsampled	LogisticRegression	FALSE	58%	41%	44%	56%	59%	59%
AG_1	Downsampled	XGB	TRUE	58%	35%	50%	50%	65%	64%
FCR_4	Downsampled	RandomForest	FALSE	57%	54%	31%	69%	46%	48%
FCR_1	Downsampled	GradientBoosting	TRUE	57%	42%	44%	56%	58%	58%
FCR_6	Downsampled	GradientBoosting	TRUE	57%	42%	44%	56%	58%	58%
AG_7	Downsampled	LogisticRegression	TRUE	57%	29%	56%	44%	71%	69%
FCRC_3	Downsampled	KNeighbors	TRUE	57%	67%	19%	81%	33%	36%
FCR_7	Downsampled	KNeighbors	FALSE	57%	67%	19%	81%	33%	36%
FCR_2	Downsampled	RandomForest	FALSE	57%	61%	25%	75%	39%	41%
FCR	Upsampled	LogisticRegression	TRUE	57%	55%	31%	69%	45%	47%
FCR_4	Downsampled	GradientBoosting	FALSE	57%	49%	38%	63%	51%	52%
FCRC_2	Downsampled	GaussianNB	FALSE	57%	11%	75%	25%	89%	84%
FCRC_2	Downsampled	SVC	TRUE	57%	11%	75%	25%	89%	84%
AG_7	Downsampled	GaussianNB	TRUE	57%	74%	13%	88%	26%	30%
FCRC_1	Downsampled	DecisionTree	TRUE	57%	62%	25%	75%	38%	41%
FCRC_2	Downsampled	KNeighbors	TRUE	57%	62%	25%	75%	38%	41%
FCR_3	Downsampled	GradientBoosting	TRUE	57%	56%	31%	69%	44%	46%
FCRC_5	Downsampled	LogisticRegression	TRUE	57%	43%	44%	56%	57%	57%
AG_5	Downsampled	RandomForest	FALSE	57%	43%	44%	56%	57%	57%
AG_1	Downsampled	GradientBoosting	FALSE	57%	37%	50%	50%	63%	62%
FCRC	Upsampled	LogisticRegression	TRUE	57%	37%	50%	50%	63%	62%
FCRC_5	Downsampled	SVC	TRUE	56%	62%	25%	75%	38%	40%
FCR_5	Downsampled	KNeighbors	FALSE	56%	56%	31%	69%	44%	46%
FCR_3	Downsampled	LogisticRegression	FALSE	56%	56%	31%	69%	44%	46%
AG_1	Downsampled	SVC	TRUE	56%	50%	38%	63%	50%	51%
FCR_4	Downsampled	XGB	TRUE	56%	50%	38%	63%	50%	51%
FCRC_4	Downsampled	RandomForest	TRUE	56%	44%	44%	56%	56%	56%
FCRC_2	Downsampled	LogisticRegression	TRUE	56%	44%	44%	56%	56%	56%
FCRC_6	Downsampled	XGB	TRUE	56%	44%	44%	56%	56%	56%
FCRC_4	Downsampled	LogisticRegression	FALSE	56%	37%	50%	50%	63%	62%
FCR	Upsampled	DecisionTree	TRUE	56%	37%	50%	50%	63%	62%
FCR_7	Downsampled	DecisionTree	TRUE	56%	37%	50%	50%	63%	62%
AG_5	Downsampled	KNeighbors	TRUE	56%	31%	56%	44%	69%	67%
FCRC_6	Downsampled	SVC	FALSE	56%	25%	63%	38%	75%	72%
AG_4	Downsampled	DecisionTree	FALSE	56%	12%	75%	25%	88%	83%
FCR_7	Downsampled	GaussianNB	FALSE	56%	12%	75%	25%	88%	83%
FCRC_4	Downsampled	SVC	TRUE	56%	12%	75%	25%	88%	83%
AG_3	Downsampled	SVC	FALSE	56%	69%	19%	81%	31%	35%
FCR_6	Downsampled	KNeighbors	TRUE	56%	50%	38%	63%	50%	51%
FCR	Upsampled	DecisionTree	FALSE	56%	38%	50%	50%	62%	61%

Dataset	Subset	Classifier Name	Feature Selection	AUC	Type 1	Type 2	Recall	Specificity	Accuracy
FCRC_7	Downsampled	SVC	TRUE	56%	13%	75%	25%	87%	83%
FCR_5	Downsampled	GaussianNB	FALSE	56%	13%	75%	25%	87%	83%
FCR_2	Downsampled	KNeighbors	FALSE	56%	82%	6%	94%	18%	24%
FCRC_1	Downsampled	SVC	FALSE	56%	76%	13%	88%	24%	29%
AG_3	Downsampled	SVC	TRUE	56%	51%	38%	63%	49%	50%
FCRC_2	Downsampled	XGB	TRUE	56%	44%	44%	56%	56%	56%
FCRC_7	Downsampled	GradientBoosting	TRUE	56%	44%	44%	56%	56%	56%
AG_4	Downsampled	KNeighbors	TRUE	56%	32%	56%	44%	68%	66%
FCR_7	Downsampled	DecisionTree	FALSE	56%	64%	25%	75%	36%	39%
FCR_1	Downsampled	KNeighbors	TRUE	56%	64%	25%	75%	36%	39%
AG	Downsampled	Sequential	FALSE	56%	51%	38%	63%	49%	50%
FCRC_6	Downsampled	GradientBoosting	TRUE	56%	45%	44%	56%	55%	55%
FCR_1	Downsampled	XGB	TRUE	56%	39%	50%	50%	61%	60%
AG_1	Downsampled	KNeighbors	TRUE	56%	33%	56%	44%	67%	66%
AG_2	Downsampled	KNeighbors	TRUE	56%	33%	56%	44%	67%	66%
FCRC_2	Downsampled	GradientBoosting	TRUE	55%	45%	44%	56%	55%	55%
FCR_3	Downsampled	DecisionTree	TRUE	55%	45%	44%	56%	55%	55%
AG_5	Downsampled	GradientBoosting	TRUE	55%	39%	50%	50%	61%	60%
FCRC_1	Downsampled	DecisionTree	FALSE	55%	65%	25%	75%	35%	38%
AG_6	Downsampled	SVC	FALSE	55%	52%	38%	63%	48%	49%
AG	Downsampled	Sequential	FALSE	55%	46%	44%	56%	54%	54%
AG_2	Downsampled	XGB	TRUE	55%	40%	50%	50%	60%	60%
AG_2	Downsampled	XGB	FALSE	55%	33%	56%	44%	67%	65%
AG_3	Downsampled	GaussianNB	TRUE	55%	84%	6%	94%	16%	22%
FCRC_6	Downsampled	GaussianNB	TRUE	55%	65%	25%	75%	35%	38%
FCRC_1	Downsampled	XGB	TRUE	55%	40%	50%	50%	60%	59%
AG_4	Downsampled	GradientBoosting	TRUE	55%	40%	50%	50%	60%	59%
FCR	Upsampled	KNeighbors	TRUE	55%	34%	56%	44%	66%	64%
AG_4	Downsampled	DecisionTree	TRUE	55%	28%	63%	38%	72%	70%
FCRC_5	Downsampled	GaussianNB	TRUE	55%	9%	81%	19%	91%	86%
FCRC	Upsampled	RandomForest	TRUE	55%	3%	88%	13%	97%	91%
AG_3	Downsampled	LogisticRegression	TRUE	55%	53%	38%	63%	47%	48%
FCRC_5	Downsampled	RandomForest	FALSE	55%	53%	38%	63%	47%	48%
FCRC_2	Downsampled	RandomForest	FALSE	55%	47%	44%	56%	53%	53%
AG_3	Downsampled	GradientBoosting	FALSE	55%	41%	50%	50%	59%	59%
AG_4	Downsampled	GaussianNB	FALSE	55%	41%	50%	50%	59%	59%
FCR_2	Downsampled	KNeighbors	TRUE	54%	60%	31%	69%	40%	42%
FCR	Upsampled	SVC	TRUE	54%	47%	44%	56%	53%	53%
FCRC_6	Downsampled	KNeighbors	TRUE	54%	41%	50%	50%	59%	58%
FCRC_7	Downsampled	DecisionTree	TRUE	54%	16%	75%	25%	84%	80%
FCR_7	Downsampled	SVC	FALSE	54%	16%	75%	25%	84%	80%
FCRC	Upsampled	RandomForest	FALSE	54%	4%	88%	13%	96%	90%
AG_1	Downsampled	SVC	FALSE	54%	48%	44%	56%	52%	52%
FCRC_6	Downsampled	LogisticRegression	TRUE	54%	48%	44%	56%	52%	52%
FCR_4	Downsampled	KNeighbors	TRUE	54%	42%	50%	50%	58%	58%

Dataset	Subset	Classifier Name	Feature Selection	AUC	Type 1	Type 2	Recall	Specificity	Accuracy
AG	Upsampled	LogisticRegression	TRUE	54%	35%	56%	44%	65%	63%
AG_2	Downsampled	GradientBoosting	FALSE	54%	36%	56%	44%	64%	63%
FCRC_1	Downsampled	GradientBoosting	TRUE	54%	36%	56%	44%	64%	63%
FCR_4	Downsampled	DecisionTree	FALSE	54%	30%	63%	38%	70%	68%
FCR	Upsampled	GaussianNB	TRUE	54%	92%	0%	100%	8%	14%
FCR	Downsampled	Sequential	FALSE	54%	74%	19%	81%	26%	30%
FCRC_1	Downsampled	KNeighbors	FALSE	54%	67%	25%	75%	33%	36%
FCR_4	Downsampled	LogisticRegression	TRUE	54%	61%	31%	69%	39%	41%
FCR_2	Downsampled	RandomForest	TRUE	54%	55%	38%	63%	45%	46%
AG_1	Downsampled	GaussianNB	TRUE	54%	55%	38%	63%	45%	46%
FCRC_3	Downsampled	GradientBoosting	FALSE	54%	55%	38%	63%	45%	46%
AG_2	Downsampled	DecisionTree	FALSE	54%	49%	44%	56%	51%	52%
AG_6	Downsampled	KNeighbors	FALSE	54%	18%	75%	25%	82%	78%
AG_5	Downsampled	DecisionTree	TRUE	54%	18%	75%	25%	82%	78%
FCRC_7	Downsampled	KNeighbors	FALSE	53%	56%	38%	63%	44%	46%
FCR	Downsampled	Sequential	FALSE	53%	56%	38%	63%	44%	46%
FCR_4	Downsampled	LogisticRegression	FALSE	53%	49%	44%	56%	51%	51%
AG_3	Downsampled	DecisionTree	TRUE	53%	43%	50%	50%	57%	56%
FCRC_7	Downsampled	DecisionTree	FALSE	53%	18%	75%	25%	82%	78%
FCR_6	Downsampled	GaussianNB	TRUE	53%	87%	6%	94%	13%	19%
FCR_3	Downsampled	KNeighbors	FALSE	53%	68%	25%	75%	32%	35%
FCR_7	Downsampled	LogisticRegression	FALSE	53%	62%	31%	69%	38%	40%
AG_6	Downsampled	RandomForest	TRUE	53%	44%	50%	50%	56%	56%
FCRC	Downsampled	Sequential	FALSE	53%	44%	50%	50%	56%	56%
FCR_1	Downsampled	GaussianNB	TRUE	53%	6%	88%	13%	94%	88%
FCR_1	Downsampled	SVC	FALSE	53%	56%	38%	63%	44%	45%
FCR_1	Downsampled	RandomForest	TRUE	53%	38%	56%	44%	62%	61%
FCR_1	Downsampled	RandomForest	FALSE	53%	38%	56%	44%	62%	61%
FCRC	Upsampled	SVC	TRUE	53%	57%	38%	63%	43%	44%
FCR_4	Downsampled	XGB	FALSE	53%	51%	44%	56%	49%	50%
FCRC_4	Downsampled	KNeighbors	FALSE	53%	51%	44%	56%	49%	50%
AG_7	Downsampled	GradientBoosting	TRUE	53%	32%	63%	38%	68%	66%
AG_5	Downsampled	DecisionTree	FALSE	53%	20%	75%	25%	80%	76%
AG	Upsampled	Sequential	FALSE	53%	7%	88%	13%	93%	87%
FCR_2	Downsampled	GaussianNB	FALSE	52%	14%	81%	19%	86%	81%
FCR	Upsampled	GradientBoosting	FALSE	52%	1%	94%	6%	99%	92%
FCR	Upsampled	GradientBoosting	TRUE	52%	1%	94%	6%	99%	92%
FCRC_5	Downsampled	SVC	FALSE	52%	52%	44%	56%	48%	49%
AG_2	Downsampled	GradientBoosting	TRUE	52%	39%	56%	44%	61%	60%
AG_1	Downsampled	LogisticRegression	TRUE	52%	39%	56%	44%	61%	60%
AG_2	Downsampled	KNeighbors	FALSE	52%	27%	69%	31%	73%	70%
FCRC_2	Downsampled	SVC	FALSE	52%	21%	75%	25%	79%	76%
FCRC_6	Downsampled	DecisionTree	FALSE	52%	14%	81%	19%	86%	81%
FCRC	Upsampled	GradientBoosting	FALSE	52%	2%	94%	6%	98%	92%
AG	Upsampled	GradientBoosting	FALSE	52%	2%	94%	6%	98%	92%

Dataset	Subset	Classifier Name	Feature Selection	AUC	Type 1	Type 2	Recall	Specificity	Accuracy
FCRC_2	Downsampled	KNeighbors	FALSE	52%	65%	31%	69%	35%	38%
FCR_4	Downsampled	RandomForest	TRUE	52%	58%	38%	63%	42%	43%
FCRC_7	Downsampled	KNeighbors	TRUE	52%	58%	38%	63%	42%	43%
AG_2	Downsampled	LogisticRegression	FALSE	52%	52%	44%	56%	48%	48%
FCRC_3	Downsampled	LogisticRegression	FALSE	52%	52%	44%	56%	48%	48%
AG	Upsampled	LogisticRegression	FALSE	52%	40%	56%	44%	60%	59%
FCR_4	Downsampled	GaussianNB	FALSE	52%	15%	81%	19%	85%	80%
FCRC	Upsampled	DecisionTree	TRUE	52%	9%	88%	13%	91%	86%
AG	Upsampled	GradientBoosting	TRUE	52%	2%	94%	6%	98%	91%
FCR_6	Downsampled	SVC	TRUE	52%	96%	0%	100%	4%	11%
AG_1	Downsampled	LogisticRegression	FALSE	52%	53%	44%	56%	47%	48%
FCR_7	Downsampled	RandomForest	FALSE	52%	53%	44%	56%	47%	48%
FCRC_6	Downsampled	RandomForest	TRUE	52%	46%	50%	50%	54%	53%
AG_7	Downsampled	GradientBoosting	FALSE	52%	40%	56%	44%	60%	59%
FCRC_4	Downsampled	LogisticRegression	TRUE	52%	41%	56%	44%	59%	58%
FCRC	Downsampled	Sequential	FALSE	52%	34%	63%	38%	66%	64%
FCR_5	Downsampled	GaussianNB	TRUE	51%	10%	88%	13%	90%	85%
FCRC_6	Downsampled	LogisticRegression	FALSE	51%	47%	50%	50%	53%	52%
FCRC_4	Downsampled	GradientBoosting	TRUE	51%	47%	50%	50%	53%	52%
AG	Upsampled	DecisionTree	TRUE	51%	35%	63%	38%	65%	63%
AG_2	Downsampled	GaussianNB	TRUE	51%	29%	69%	31%	71%	68%
AG	Upsampled	RandomForest	TRUE	51%	4%	94%	6%	96%	90%
AG_3	Downsampled	KNeighbors	FALSE	51%	35%	63%	38%	65%	63%
FCR_6	Downsampled	SVC	FALSE	51%	98%	0%	100%	2%	9%
FCR	Upsampled	SVC	FALSE	51%	73%	25%	75%	27%	30%
FCRC_3	Downsampled	RandomForest	TRUE	51%	55%	44%	56%	45%	46%
FCR_3	Downsampled	RandomForest	FALSE	51%	48%	50%	50%	52%	52%
AG	Downsampled	Sequential	FALSE	51%	48%	50%	50%	52%	52%
FCR_1	Downsampled	DecisionTree	TRUE	51%	23%	75%	25%	77%	73%
FCR_7	Downsampled	SVC	TRUE	51%	99%	0%	100%	1%	8%
FCR_2	Downsampled	SVC	FALSE	51%	92%	6%	94%	8%	14%
FCR_4	Downsampled	KNeighbors	FALSE	51%	61%	38%	63%	39%	40%
FCR_4	Downsampled	SVC	FALSE	51%	61%	38%	63%	39%	40%
FCRC_1	Downsampled	XGB	FALSE	51%	36%	63%	38%	64%	62%
FCRC_3	Downsampled	DecisionTree	FALSE	51%	24%	75%	25%	76%	72%
FCRC_6	Downsampled	DecisionTree	TRUE	51%	18%	81%	19%	82%	78%
FCRC_7	Downsampled	GaussianNB	FALSE	50%	87%	13%	88%	13%	19%
FCRC_5	Downsampled	GaussianNB	FALSE	50%	87%	13%	88%	13%	19%
AG_5	Downsampled	SVC	FALSE	50%	74%	25%	75%	26%	29%
FCRC_1	Downsampled	KNeighbors	TRUE	50%	74%	25%	75%	26%	29%
FCRC_5	Downsampled	KNeighbors	TRUE	50%	49%	50%	50%	51%	51%
FCR_1	Downsampled	GradientBoosting	FALSE	50%	43%	56%	44%	57%	56%
FCR_6	Downsampled	DecisionTree	FALSE	50%	24%	75%	25%	76%	72%
AG_4	Downsampled	GaussianNB	TRUE	50%	18%	81%	19%	82%	77%
FCRC_1	Downsampled	GaussianNB	FALSE	50%	87%	13%	88%	13%	18%

Dataset	Subset	Classifier Name	Feature Selection	AUC	Type 1	Type 2	Recall	Specificity	Accuracy
AG	Upsampled	SVC	TRUE	50%	44%	56%	44%	56%	56%
AG	Downsampled	Sequential	FALSE	50%	44%	56%	44%	56%	56%
AG_1	Downsampled	XGB	FALSE	50%	37%	63%	38%	63%	61%
FCRC_3	Downsampled	DecisionTree	TRUE	50%	25%	75%	25%	75%	72%
FCRC_3	Downsampled	GaussianNB	FALSE	50%	88%	13%	88%	12%	18%
FCR_2	Downsampled	GaussianNB	TRUE	50%	81%	19%	81%	19%	23%
FCRC_3	Downsampled	GaussianNB	TRUE	50%	69%	31%	69%	31%	34%
AG_3	Downsampled	GradientBoosting	TRUE	50%	44%	56%	44%	56%	55%
AG	Upsampled	DecisionTree	FALSE	50%	13%	88%	13%	87%	82%
FCRC_7	Downsampled	GaussianNB	TRUE	50%	7%	94%	6%	93%	87%
FCR	Upsampled	XGB	TRUE	50%	0%	100%	0%	100%	92%
FCR_3	Downsampled	DecisionTree	FALSE	50%	76%	25%	75%	24%	28%
FCRC_4	Downsampled	GaussianNB	FALSE	50%	69%	31%	69%	31%	33%
AG	Upsampled	SVC	FALSE	50%	44%	56%	44%	56%	55%
FCR	Upsampled	XGB	FALSE	50%	1%	100%	0%	99%	92%
FCR	Upsampled	GaussianNB	FALSE	49%	82%	19%	81%	18%	22%
FCRC	Upsampled	GradientBoosting	TRUE	49%	1%	100%	0%	99%	92%
AG	Upsampled	XGB	FALSE	49%	1%	100%	0%	99%	92%
FCR_3	Downsampled	SVC	FALSE	49%	70%	31%	69%	30%	32%
FCR_5	Downsampled	KNeighbors	TRUE	49%	52%	50%	50%	48%	48%
AG	Upsampled	XGB	TRUE	49%	2%	100%	0%	98%	91%
FCR_1	Downsampled	KNeighbors	FALSE	49%	89%	13%	88%	11%	16%
FCRC_6	Downsampled	GaussianNB	FALSE	49%	77%	25%	75%	23%	27%
FCR_7	Downsampled	KNeighbors	TRUE	49%	58%	44%	56%	42%	43%
FCRC_4	Downsampled	DecisionTree	FALSE	49%	27%	75%	25%	73%	69%
FCRC_4	Downsampled	DecisionTree	TRUE	49%	27%	75%	25%	73%	69%
FCRC	Upsampled	Sequential	FALSE	49%	15%	88%	13%	85%	80%
FCRC	Upsampled	XGB	FALSE	49%	2%	100%	0%	98%	91%
FCR_4	Downsampled	GaussianNB	TRUE	49%	2%	100%	0%	98%	91%
FCR	Upsampled	RandomForest	FALSE	49%	2%	100%	0%	98%	91%
FCR	Upsampled	RandomForest	TRUE	49%	2%	100%	0%	98%	91%
FCR_6	Downsampled	DecisionTree	TRUE	49%	28%	75%	25%	72%	69%
AG_2	Downsampled	GaussianNB	FALSE	49%	22%	81%	19%	78%	74%
FCRC	Upsampled	DecisionTree	FALSE	49%	9%	94%	6%	91%	85%
FCR_7	Downsampled	GaussianNB	TRUE	49%	3%	100%	0%	97%	90%
FCRC	Upsampled	XGB	TRUE	49%	3%	100%	0%	97%	90%
AG_5	Downsampled	SVC	TRUE	49%	78%	25%	75%	22%	26%
FCR	Upsampled	KNeighbors	FALSE	48%	47%	56%	44%	53%	52%
FCRC	Upsampled	KNeighbors	FALSE	48%	41%	63%	38%	59%	58%
FCR_3	Downsampled	GaussianNB	TRUE	48%	16%	88%	13%	84%	79%
AG	Upsampled	RandomForest	FALSE	48%	3%	100%	0%	97%	90%
FCR	Downsampled	Sequential	FALSE	48%	66%	38%	63%	34%	36%
FCR	Upsampled	LogisticRegression	FALSE	48%	54%	50%	50%	46%	47%
AG_7	Downsampled	SVC	FALSE	48%	41%	63%	38%	59%	57%
FCRC	Upsampled	LogisticRegression	FALSE	48%	35%	69%	31%	65%	63%

Dataset	Subset	Classifier Name	Feature Selection	AUC	Type 1	Type 2	Recall	Specificity	Accuracy
AG_2	Downsampled	DecisionTree	TRUE	48%	35%	69%	31%	65%	63%
FCR_6	Downsampled	KNeighbors	FALSE	48%	60%	44%	56%	40%	41%
FCRC	Downsampled	Sequential	FALSE	48%	48%	56%	44%	52%	51%
AG_2	Downsampled	RandomForest	FALSE	48%	42%	63%	38%	58%	56%
AG_6	Downsampled	DecisionTree	FALSE	48%	42%	63%	38%	58%	56%
FCR_2	Downsampled	SVC	TRUE	48%	92%	13%	88%	8%	13%
FCRC_3	Downsampled	KNeighbors	FALSE	48%	67%	38%	63%	33%	35%
FCR_3	Downsampled	LogisticRegression	TRUE	48%	67%	38%	63%	33%	35%
FCRC	Upsampled	KNeighbors	TRUE	47%	36%	69%	31%	64%	61%
FCRC	Upsampled	GaussianNB	TRUE	47%	87%	19%	81%	13%	18%
FCRC	Upsampled	GaussianNB	FALSE	47%	87%	19%	81%	13%	18%
FCR	Downsampled	Sequential	FALSE	47%	68%	38%	63%	32%	34%
FCR_5	Downsampled	SVC	FALSE	47%	44%	63%	38%	56%	55%
FCR	Downsampled	Sequential	FALSE	47%	44%	63%	38%	56%	55%
FCR_1	Downsampled	DecisionTree	FALSE	47%	31%	75%	25%	69%	66%
AG_5	Downsampled	GaussianNB	TRUE	47%	81%	25%	75%	19%	23%
AG_1	Downsampled	KNeighbors	FALSE	47%	75%	31%	69%	25%	28%
FCRC	Downsampled	Sequential	FALSE	47%	44%	63%	38%	56%	54%
AG_2	Downsampled	LogisticRegression	TRUE	46%	38%	69%	31%	62%	60%
FCRC	Downsampled	Sequential	FALSE	46%	70%	38%	63%	30%	32%
FCR_3	Downsampled	KNeighbors	TRUE	46%	64%	44%	56%	36%	38%
FCR_3	Downsampled	RandomForest	TRUE	46%	45%	63%	38%	55%	54%
FCRC_2	Downsampled	LogisticRegression	FALSE	46%	45%	63%	38%	55%	54%
AG_2	Downsampled	SVC	FALSE	46%	26%	81%	19%	74%	70%
FCRC_5	Downsampled	KNeighbors	FALSE	46%	45%	63%	38%	55%	53%
FCRC_1	Downsampled	RandomForest	FALSE	46%	45%	63%	38%	55%	53%
AG_2	Downsampled	SVC	TRUE	46%	21%	88%	13%	79%	75%
AG	Upsampled	GaussianNB	TRUE	46%	83%	25%	75%	17%	21%
FCRC_6	Downsampled	KNeighbors	FALSE	46%	52%	56%	44%	48%	48%
FCR_3	Downsampled	GaussianNB	FALSE	46%	21%	88%	13%	79%	74%
AG	Upsampled	GaussianNB	FALSE	46%	78%	31%	69%	22%	26%
FCR	Downsampled	Sequential	FALSE	45%	59%	50%	50%	41%	41%
FCR_2	Downsampled	LogisticRegression	FALSE	45%	53%	56%	44%	47%	47%
FCR_3	Downsampled	SVC	TRUE	45%	54%	56%	44%	46%	46%
AG_6	Downsampled	GaussianNB	TRUE	45%	17%	94%	6%	83%	78%
AG_5	Downsampled	GaussianNB	FALSE	44%	74%	38%	63%	26%	29%
FCR_1	Downsampled	LogisticRegression	FALSE	44%	61%	50%	50%	39%	40%
AG_6	Downsampled	DecisionTree	TRUE	44%	43%	69%	31%	57%	56%
FCR_2	Downsampled	LogisticRegression	TRUE	44%	62%	50%	50%	38%	39%
FCRC_1	Downsampled	LogisticRegression	FALSE	44%	56%	56%	44%	44%	44%
FCRC_2	Downsampled	DecisionTree	TRUE	44%	50%	63%	38%	50%	49%
FCR_6	Downsampled	GaussianNB	FALSE	44%	81%	31%	69%	19%	22%
FCRC_2	Downsampled	DecisionTree	FALSE	44%	50%	63%	38%	50%	49%
AG_7	Downsampled	GaussianNB	FALSE	43%	84%	31%	69%	16%	20%
FCR	Upsampled	Sequential	FALSE	41%	30%	88%	13%	70%	66%

Dataset	Subset	Classifier Name	Feature Selection	AUC	Type 1	Type 2	Recall	Specificity	Accuracy
FCRC_1	Downsampled	GradientBoosting	FALSE	41%	43%	75%	25%	57%	55%
AG_1	Downsampled	GaussianNB	FALSE	41%	75%	44%	56%	25%	28%
FCR_4	Downsampled	DecisionTree	TRUE	40%	45%	75%	25%	55%	53%
AG_6	Downsampled	GaussianNB	FALSE	40%	77%	44%	56%	23%	26%
AG_3	Downsampled	GaussianNB	FALSE	40%	77%	44%	56%	23%	26%
FCR	Downsampled	Sequential	FALSE	38%	61%	63%	38%	39%	39%

8.3 APPENDIX 3: All Results (Downsampled Datasets Aggregated)

Dataset	Subset	Classifier Name	Feature Selection	AUC	Type 1	Type 2	Recall	Specificity	Accuracy
AG	Downsampled	RandomForest	TRUE	72%	31%	25%	75%	69%	70%
FCRC	Downsampled	XGB	TRUE	68%	39%	25%	75%	61%	62%
FCRC	Downsampled	GradientBoosting	FALSE	66%	44%	25%	75%	56%	58%
FCR	Downsampled	GradientBoosting	TRUE	66%	44%	25%	75%	56%	58%
AG	Downsampled	LogisticRegression	FALSE	65%	45%	25%	75%	55%	56%
FCRC	Downsampled	RandomForest	FALSE	65%	45%	25%	75%	55%	56%
FCRC	Downsampled	XGB	FALSE	65%	39%	31%	69%	61%	61%
AG	Downsampled	LogisticRegression	TRUE	64%	34%	38%	63%	66%	66%
FCR	Downsampled	GradientBoosting	FALSE	64%	41%	31%	69%	59%	60%
AG	Upsampled	KNeighbors	TRUE	64%	35%	38%	63%	65%	65%
FCR	Downsampled	XGB	FALSE	63%	43%	31%	69%	57%	58%
AG	Upsampled	KNeighbors	FALSE	63%	30%	44%	56%	70%	69%
FCR	Downsampled	RandomForest	TRUE	63%	50%	25%	75%	50%	52%
FCRC	Downsampled	SVC	FALSE	61%	21%	56%	44%	79%	76%
FCR	Downsampled	RandomForest	FALSE	61%	46%	31%	69%	54%	55%
FCR	Downsampled	XGB	TRUE	61%	40%	38%	63%	60%	60%
AG	Downsampled	RandomForest	FALSE	61%	34%	44%	56%	66%	65%
FCRC	Downsampled	SVC	TRUE	61%	22%	56%	44%	78%	76%
AG	Downsampled	KNeighbors	TRUE	61%	29%	50%	50%	71%	70%
FCRC	Downsampled	RandomForest	TRUE	60%	42%	38%	63%	58%	59%
AG	Downsampled	KNeighbors	FALSE	60%	30%	50%	50%	70%	68%
AG	Downsampled	GradientBoosting	TRUE	60%	31%	50%	50%	69%	68%
FCRC	Upsampled	SVC	FALSE	59%	50%	31%	69%	50%	52%
AG	Downsampled	XGB	FALSE	59%	32%	50%	50%	68%	67%
FCR	Downsampled	KNeighbors	FALSE	58%	77%	6%	94%	23%	28%
FCRC	Downsampled	LogisticRegression	TRUE	58%	46%	38%	63%	54%	54%
AG	Downsampled	GradientBoosting	FALSE	58%	34%	50%	50%	66%	64%
FCRC	Downsampled	GaussianNB	TRUE	58%	10%	75%	25%	90%	86%
FCRC	Downsampled	KNeighbors	TRUE	58%	54%	31%	69%	46%	48%
AG	Downsampled	XGB	TRUE	57%	35%	50%	50%	65%	64%
FCR	Upsampled	LogisticRegression	TRUE	57%	55%	31%	69%	45%	47%
FCRC	Downsampled	Sequential	FALSE	57%	43%	44%	56%	57%	57%
FCRC	Upsampled	LogisticRegression	TRUE	57%	37%	50%	50%	63%	62%
AG	Downsampled	DecisionTree	FALSE	57%	24%	63%	38%	76%	73%
FCRC	Downsampled	DecisionTree	FALSE	57%	18%	69%	31%	82%	78%
FCR	Downsampled	KNeighbors	TRUE	56%	56%	31%	69%	44%	46%
FCR	Upsampled	DecisionTree	TRUE	56%	37%	50%	50%	63%	62%
AG	Downsampled	SVC	TRUE	56%	31%	56%	44%	69%	67%
FCRC	Downsampled	DecisionTree	TRUE	56%	19%	69%	31%	81%	78%

Dataset	Subset	Classifier Name	Feature Selection	AUC	Type 1	Type 2	Recall	Specificity	Accuracy
FCR	Upsampled	DecisionTree	FALSE	56%	38%	50%	50%	62%	61%
FCR	Downsampled	LogisticRegression	FALSE	55%	53%	38%	63%	47%	48%
FCR	Downsampled	DecisionTree	FALSE	55%	40%	50%	50%	60%	59%
FCR	Upsampled	KNeighbors	TRUE	55%	34%	56%	44%	66%	64%
FCRC	Upsampled	RandomForest	TRUE	55%	3%	88%	13%	97%	91%
FCR	Upsampled	SVC	TRUE	54%	47%	44%	56%	53%	53%
FCR	Downsampled	DecisionTree	TRUE	54%	35%	56%	44%	65%	64%
FCRC	Upsampled	RandomForest	FALSE	54%	4%	88%	13%	96%	90%
FCR	Downsampled	LogisticRegression	TRUE	54%	60%	31%	69%	40%	42%
AG	Upsampled	LogisticRegression	TRUE	54%	35%	56%	44%	65%	63%
FCR	Upsampled	GaussianNB	TRUE	54%	92%	0%	100%	8%	14%
FCRC	Downsampled	GaussianNB	FALSE	54%	86%	6%	94%	14%	20%
FCR	Downsampled	GaussianNB	FALSE	54%	11%	81%	19%	89%	84%
FCRC	Downsampled	GradientBoosting	TRUE	53%	44%	50%	50%	56%	56%
AG	Downsampled	Sequential	FALSE	53%	44%	50%	50%	56%	56%
FCRC	Downsampled	KNeighbors	FALSE	53%	56%	38%	63%	44%	45%
FCRC	Downsampled	LogisticRegression	FALSE	53%	44%	50%	50%	56%	56%
FCRC	Upsampled	SVC	TRUE	53%	57%	38%	63%	43%	44%
AG	Upsampled	Sequential	FALSE	53%	7%	88%	13%	93%	87%
FCR	Upsampled	GradientBoosting	FALSE	52%	1%	94%	6%	99%	92%
FCR	Upsampled	GradientBoosting	TRUE	52%	1%	94%	6%	99%	92%
FCRC	Upsampled	GradientBoosting	FALSE	52%	2%	94%	6%	98%	92%
AG	Upsampled	GradientBoosting	FALSE	52%	2%	94%	6%	98%	92%
AG	Upsampled	LogisticRegression	FALSE	52%	40%	56%	44%	60%	59%
FCRC	Upsampled	DecisionTree	TRUE	52%	9%	88%	13%	91%	86%
AG	Upsampled	GradientBoosting	TRUE	52%	2%	94%	6%	98%	91%
FCR	Downsampled	SVC	TRUE	52%	96%	0%	100%	4%	11%
AG	Upsampled	DecisionTree	TRUE	51%	35%	63%	38%	65%	63%
AG	Upsampled	RandomForest	TRUE	51%	4%	94%	6%	96%	90%
FCR	Upsampled	SVC	FALSE	51%	73%	25%	75%	27%	30%
AG	Downsampled	DecisionTree	TRUE	51%	23%	75%	25%	77%	73%
FCR	Downsampled	SVC	FALSE	51%	67%	31%	69%	33%	35%
AG	Upsampled	SVC	TRUE	50%	44%	56%	44%	56%	56%
AG	Downsampled	GaussianNB	TRUE	50%	56%	44%	56%	44%	44%
AG	Upsampled	DecisionTree	FALSE	50%	13%	88%	13%	87%	82%
FCR	Upsampled	XGB	TRUE	50%	0%	100%	0%	100%	92%
AG	Upsampled	SVC	FALSE	50%	44%	56%	44%	56%	55%
FCR	Upsampled	XGB	FALSE	50%	1%	100%	0%	99%	92%
FCR	Upsampled	GaussianNB	FALSE	49%	82%	19%	81%	18%	22%
FCR	Downsampled	GaussianNB	TRUE	49%	8%	94%	6%	92%	86%
AG	Upsampled	XGB	FALSE	49%	1%	100%	0%	99%	92%
FCRC	Upsampled	GradientBoosting	TRUE	49%	1%	100%	0%	99%	92%
AG	Upsampled	XGB	TRUE	49%	2%	100%	0%	98%	91%
FCRC	Upsampled	Sequential	FALSE	49%	15%	88%	13%	85%	80%
FCRC	Upsampled	XGB	FALSE	49%	2%	100%	0%	98%	91%

Dataset	Subset	Classifier Name	Feature Selection	AUC	Type 1	Type 2	Recall	Specificity	Accuracy
FCR	Upsampled	RandomForest	FALSE	49%	2%	100%	0%	98%	91%
FCR	Upsampled	RandomForest	TRUE	49%	2%	100%	0%	98%	91%
FCRC	Upsampled	DecisionTree	FALSE	49%	9%	94%	6%	91%	85%
FCRC	Upsampled	XGB	TRUE	49%	3%	100%	0%	97%	90%
FCR	Upsampled	KNeighbors	FALSE	48%	47%	56%	44%	53%	52%
FCRC	Upsampled	KNeighbors	FALSE	48%	41%	63%	38%	59%	58%
AG	Upsampled	RandomForest	FALSE	48%	3%	100%	0%	97%	90%
FCR	Upsampled	LogisticRegression	FALSE	48%	54%	50%	50%	46%	47%
FCRC	Upsampled	LogisticRegression	FALSE	48%	35%	69%	31%	65%	63%
FCRC	Upsampled	KNeighbors	TRUE	47%	36%	69%	31%	64%	61%
FCRC	Upsampled	GaussianNB	FALSE	47%	87%	19%	81%	13%	18%
FCRC	Upsampled	GaussianNB	TRUE	47%	87%	19%	81%	13%	18%
AG	Downsampled	SVC	FALSE	47%	43%	63%	38%	57%	56%
AG	Upsampled	GaussianNB	TRUE	46%	83%	25%	75%	17%	21%
AG	Upsampled	GaussianNB	FALSE	46%	78%	31%	69%	22%	26%
FCR	Downsampled	Sequential	FALSE	43%	70%	44%	56%	30%	32%
FCR	Upsampled	Sequential	FALSE	41%	30%	88%	13%	70%	66%
AG	Downsampled	GaussianNB	FALSE	40%	77%	44%	56%	23%	26%