Applied Mathematics and Computational Intelligence
Volume 14, No. 3, 2025 [57 – 78]

Credit Scoring: A Comparison of Machine Learning Models and Their Modifications

Jia Chong Ong[1*] and Lai Soon Lee[2]

[1,2]Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, 43400 Serdang, Selangor, MALAYSIA
[2]Laboratory of Computational Statistics and Operations Research, Institute for Mathematical Research, Universiti Putra Malaysia, 43400 Serdang, Selangor, MALAYSIA

* Corresponding author: ongjiachong@gmail.com

Received: 23 August 2024
Revised: 1 September 2024
Accepted: 18 October 2024

ABSTRACT

This study compares the performance of various machine learning models and their modifications across four benchmark credit scoring datasets to address the absence of comprehensive comparative analyses on multiple combinations of modifications in the credit scoring domain. Models studied include Logistic Regression (LR), Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), Random Forest (RF), and Multilayer Perceptron (MLP). Starting from these base models, a series of modifications encompassing feature scaling, resampling, feature selection, and hyperparameter tuning are added phase by phase to the previous models, where the optimal method from each modification is determined in each phase based on the accuracy, F1 score, precision, recall, area under the Receiver Operating Characteristic curve, fitting time and prediction time. Findings reveal LR's suitability for small datasets, while RF and MLP excel in larger ones. Standardization and Min-Max Scaling are generally effective, with Max-Abs Scaling enhancing RF. Synthetic Minority Oversampling Technique proves optimal for imbalanced datasets but no resampling is necessary for small balanced datasets. Analysis of Variance and Mutual Information perform similarly without tuning, while Grid Search slightly outperforms Random Search disregarding runtimes. The study concludes by presenting optimal models and alternatives.

Keywords: classification, comparative analysis, credit scoring, machine learning, modification techniques

1 INTRODUCTION

Credit scoring is a process used in the financial industry when evaluating the creditworthiness of an individual seeking credit. This process is crucial for lenders during the assessment and prediction of credit risk as it enables them to make informed decisions regarding loan approvals and minimize their potential financial losses.

Traditional credit-scoring models used statistical techniques such as Logistic Regression (LR) and Linear Discriminant Analysis (LDA) to determine a consumer's creditworthiness and assign a numerical score based on five major categories, including payment history, debt burden, length

of credit history, types of credit used, and new credit requests. This vastly simplifies the process of credit evaluation and risk assessment. However, lenders now face an information asymmetry as a numerical representation of a consumer's creditworthiness does not always provide the full picture. Other than accuracy, credit scoring models also face the problem of credit invisibility when classifying the creditworthiness of consumers with limited credit histories.

Therefore, credit scoring models such as FICO and VantageScore (a more recent competitor to FICO since 2006), evolved to meet the needs of the credit industry. Now, in response to the abundance of Big Data, these credit scoring models can harness the utility of machine learning and artificial intelligence (AI) techniques to enhance the performance of their scoring models. Promising machine learning models that have been developed over the years include Support Vector Machine (SVM), Random Forest (RF), and Multilayer Perceptron (MLP) [1–5]. However, AI techniques, including machine learning algorithms, do not always guarantee consistent results across different scenarios.

Due to the reliance on data-driven learning, existing credit-scoring models may not consistently deliver optimal performance across different datasets and scenarios. Therefore, it would be valuable to comprehensively analyze the impact of various modifications, such as resampling techniques, feature scaling methods, feature selection approaches, and hyperparameter tuning strategies, on the performance of credit scoring models. The effectiveness of the different modifications across different datasets can provide insights into their ability to improve credit risk assessment, and possibly lead to an optimal combination of modifications to enhance credit risk assessment.

Apart from that, there seems to be a research gap regarding comprehensive comparative analyses of modifications on multiple credit scoring models across diverse datasets. While some studies have explored specific modifications to credit scoring models [4, 6–8], there is a need for a comprehensive evaluation that considers the impact of various modifications, such as resampling, feature scaling, feature selection, and hyperparameter tuning techniques, on the performance of these models. Such an evaluation would provide valuable insights into the effectiveness of different strategies for credit risk assessment. By addressing this research gap, this study aims to contribute to the advancement of methodologies in credit risk assessment by searching for optimal combinations of modifications for different datasets.

This study also serves as an invitation for further exploration in the field of credit scoring models holistically. Many studies have studied specific modifications and fixed other parts of the credit-scoring algorithm which provides valuable insights under controlled conditions [9–11]. However, the performance of credit scoring models may be affected by the interactions between different methods used throughout the whole credit scoring process such as resampling, feature scaling, feature selection, and hyperparameter tuning techniques.

2 RELATED WORKS

The existing literature on credit scoring has focused on the exploration of different models [12–16] and their variations [6, 7, 17–20]. Many have also explored specific modifications to credit scoring models, such as resampling techniques [21–23], feature selection approaches [4, 8, 24], and hyperparameter tuning settings [25]. However, there is a lack of comprehensive comparative analyses

that consider the effectiveness of these modifications across multiple credit-scoring models and diverse datasets.

[1] studied the performance of several classification algorithms across eight credit scoring datasets. It was found that SVM and MLP performed well in terms of accuracy (ACC) and area under the Receiver Operating Characteristic curve (AUC), alongside the statistical models LR and LDA. [2] aimed to update the research by [1] with new alternative classification algorithms. The benchmarking study found that RF and MLP were very versatile classifiers, with RF being recommended as a benchmark against new classification algorithms. These two studies, although only looking at the base models, provided a solid foundation for researchers to understand the potential of certain classifiers, especially SVM, RF, and MLP.

Recently, [3] compared the performance between different ensemble models, which are RF, AdaBoost, XGBoost, LightGBM, and Stacking in terms of ACC, AUC, Kolmogorov-Smirnob statistic, Brier score, and operating time. These models are also compared to baseline classifiers, which are Neural Networks, Decision Trees, LR, Naive Bayes, and SVM. From the experiments, they discovered that ensemble models generally perform better than baseline classifiers, with RF being the best in all five performance criteria. This result is similar to the study conducted by [2]. However, this study only used one data source, that is the Lending Club Loan Data in 2018 Q4, and one hyperparameter tuning method which is grid search.

Another recent study by [4] compared the performance of five different machine learning models (Bayesian, Naive Bayesian, SVM, C5.0 Decision Tree, and RF) and three feature selection techniques (Chi-square, Gain rato, and Information gain), on the German Credit Data (GC) dataset. The objective of the research was to identify the best feature selection and machine learning model. The evaluation metrics used were ACC, F-value, False Positives (FP), False Negatives (FN), and training time. In this study, RF paired with the Chi-square feature selection method was found to be the best combination among the others. Again, this result was consistent with other studies [2, 3, 5]. However, this study also used only one data source, that is the German Credit Data, and only looked at the effects of feature selection and the selected models.

In 2021, [5] applied different machine learning and deep learning credit scoring models in a micro-finance environment and found that tree-based algorithms and ensemble classifiers performed better than others. RF had the best accuracy compared with other models such as decision tree, extra tree, XGBoost, AdaBoost, K-Nearest Neighbors, and MLP.

While existing studies have investigated the impact of individual modifications in credit-scoring models, there remains a research gap in conducting a comprehensive evaluation that encompasses a combination of resampling, feature scaling, feature selection, and hyperparameter tuning techniques. Hence, this study aims to address this gap by exploring and identifying optimal combinations of these modifications for different datasets, taking into account various evaluation metrics.

3 METHODOLOGY

Figure 1 outlines the general methodology that will be undertaken in this study. Four benchmark credit scoring datasets, namely Australian Credit Approval (AC), German Credit Data (GC), Lending Club Loan Data (LC) from 2007 to the third quarter of 2020 (2020Q3), and Give Me Some Credit Competition Data (GMSC) in 2011, are involved in this study.

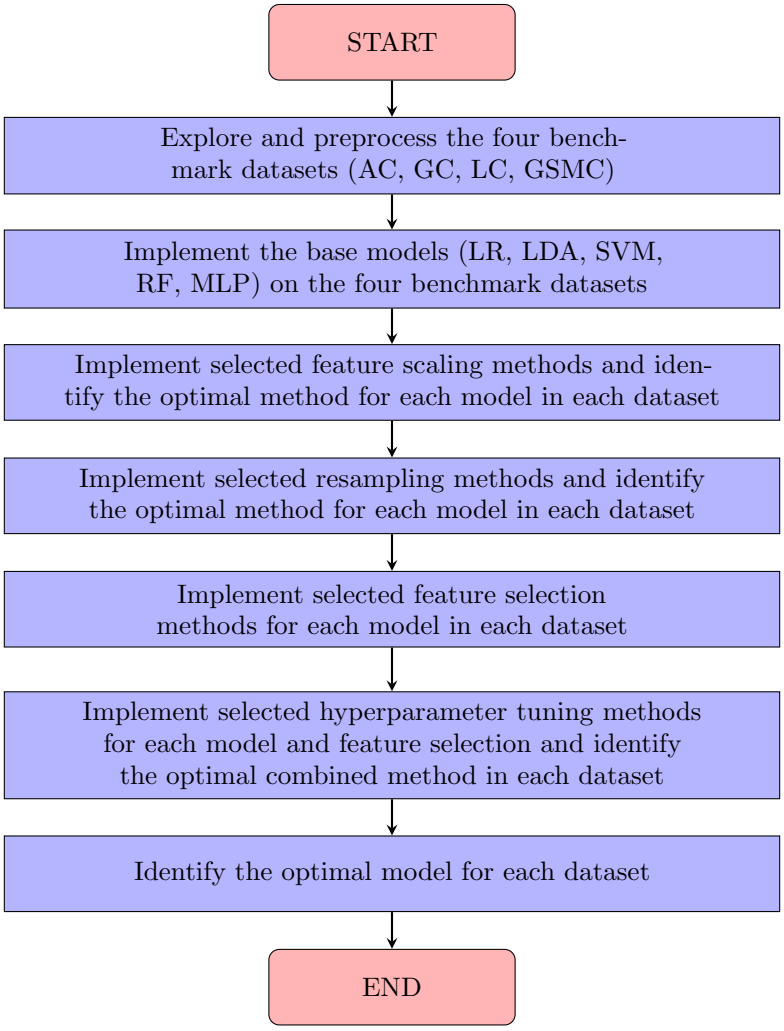

Figure 1 : Flowchart of Research Activities

In the first phase, the datasets are explored using Python to identify the features, number of samples, class distributions, and missing values. Table 1 shows the number of samples, features, and class distributions of the datasets. The datasets and their metadata information can be obtained from their relevant websites [26–29].

Based on the nature of the features and missing values, the dataset may be preprocessed through feature elimination, binning, encoding, or imputation. Features that are deemed unimportant for the analysis, such as identifiers and descriptions, are excluded from further consideration. This step aims to reduce noise and the number of irrelevant features. Categorical features with a large number

Table 1 : Overview of datasets including number of samples, features, and class distributions

Dataset	Samples	Features	Class Distribution
AC	690	14	383 non-defaulters, 307 defaulters
GC	1000	20	700 non-defaulters, 300 defaulters
LC	1,770,000	72	1,420,000 non-defaulters, 350,000 defaulters
GMSC	150,000	11	140,000 non-defaulters, 10,000 defaulters

of categories may be binned or grouped accordingly to reduce the number of distinct categories. This can simplify the analysis and prevent issues caused by high cardinality. Appropriate encoding techniques, such as one-hot encoding, label encoding, or ordinal encoding are applied to convert categorical features into numerical representations, depending on the nature of the categorical features. Instances or features with a high percentage of missing values (more than 90%) are removed from the dataset, as they may introduce bias. The remaining missing values are imputed with mean or median values for numerical features and mode for categorical features.

After a clean dataset is produced, the aforementioned base models are implemented on each of the four benchmark datasets. An 80/20 train-test split is first applied to the dataset, where 80% of the dataset is defined as the training set and 20% of the dataset as the testing set, all of which is done randomly. Then, the classifier is fitted with the training set at which the fitting time is also recorded. After that, the trained classifier are used to make class predictions based on the features from the testing set, at which the prediction is recorded.

After obtaining the prediction set, seven evaluation metrics, namely Accuracy (ACC), F1 Score (F1), Precision (P), Recall (R), Area Under the ROC Curve (AUC), fitting time, and prediction time are calculated based on the prediction set and the classes from the testing set. Here's why each metric is important:

- **Accuracy (ACC)**: This measures the overall correctness of the model's predictions, indicating the proportion of total correct predictions (both defaulters and non-defaulters). However, in imbalanced datasets, accuracy can be misleading because the model might be biased towards the majority class.

- **F1 Score (F1)**: This metric is the harmonic mean of precision and recall, providing a single measure that balances the two. It is especially useful when the costs of false positives and false negatives are different, which is often the case in credit scoring.

- **Precision (P)**: This indicates the proportion of true positive predictions among all positive predictions. High precision means that when the model predicts a borrower as risky, it is often correct, reducing the number of good borrowers incorrectly classified as risky.

- **Recall (R)**: This measures the model's ability to identify all actual risky borrowers. High recall ensures that most risky borrowers are correctly identified, minimizing the number of risky borrowers that are missed.

- **Area Under the ROC Curve (AUC)**: This metric provides an overall performance measure by illustrating the trade-off between true positive rate and false positive rate across different

thresholds. A higher AUC indicates better model performance in distinguishing between defaulters and non-defaulters.

- **Fitting time**: This measures how long it takes to train the model. It is important to understand the computational efficiency and feasibility of the model, especially with large datasets.

- **Prediction time**: This measures how long it takes for the model to make predictions on new data. It is crucial for assessing the model's practicality in real-time credit scoring applications.

The results are then tabulated as shown in the next section.

In the next phase, four feature scaling methods, namely Min-Max Scaling, Standardization, Max-Abs Scaling, and Robust Scaling are separately implemented on each of the base models for each dataset. After the train-test split, feature scaling is applied to the features of the training set and the testing test. The models proceed normally and the results are tabulated. Comparing the evaluation metrics between each base model and their respective modified models after feature scaling methods are applied, the optimal feature scaling method, which may include the case where it is optimal when no feature scaling is applied, is identified and recorded for each model in each dataset. The recorded method is chosen as the method applied for all further modifications in subsequent phases.

After feature scaling methods are chosen, two resampling methods, namely Random Undersampling (RUS) and Synthetic Minority Oversampling Technique (SMOTE) are separately implemented on each of the five chosen modified models from each dataset. Resampling is applied right before the train-test split is done. The model then proceeds similarly to the feature scaling phase, and the optimal resampling method for each model in each dataset is identified, recorded, and then chosen for subsequent phases.

After resampling methods are chosen, two feature selection methods, namely Analysis of Variance (ANOVA) and Mutual Information (MI) are again separately implemented on each of the five modified models from each dataset. Feature selection is applied right after the train-test split before feature scaling is done. First, each feature selection method is applied to the training set to obtain the relevant score assigned to each feature. The scores are then remapped using Min-Max Scaling, scaling them to a range of [0, 1]. Since score values are always non-negative, score signs are preserved. Features that scored above 10% of the highest score are selected. The training set and the testing set would be transformed to only include the selected features. The model then proceeds normally with feature scaling and so on until the calculation of evaluation metrics. During the study, it was observed that both feature selection methods without hyperparameter tuning provided similar results. Hence, the study moves onto the hyperparameter tuning phase without choosing an optimal feature selection method.

After the implementation of feature selection, two hyperparameter tuning methods, namely Grid Search (GS) and Random Search (RS) are further implemented separately on top of each feature selection method on each of the five modified models from each dataset. Hyperparameter tuning is implemented during which the classifiers are fitted with the training set, incorporating a 5-fold cross-validation approach. Hence, the whole model proceeds normally as the feature selection phase except during the fitting time. During fitting, the classifier is iteratively fitted with the training

data using different hyperparameters from a defined hyperparameter space. Table 2 shows a list of the hyperparameters and their values used to form the hyperparameter space.

Table 2 : Hyperparameters and their values used to form the hyperparameter space

Model	Hyperparameter	Values
LR	C	[0.01, 0.1, 1, 10, 100]
	penalty	['none', 'l1', 'l2', 'elasticnet']
	solver	['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga']
LDA	solver	['svd', 'lsqr', 'eigen']
	shrinkage	[None, 'auto']
SVM	C	[0.1, 1, 10, 100, 1000]
	gamma	[1, 0.1, 0.01, 0.001, 0.0001]
RF	n_estimators	[10, 50, 100, 200]
	max_depth	[None, 3, 5, 7]
	min_samples_split	[2, 5, 10]
	min_samples_leaf	[1, 2, 4]
	max_features	[None, 'sqrt', 'log2']
	criterion	['gini', 'entropy', 'log_loss']
MLP	hidden_layer_sizes	[(50,), (100,), (150,)]
	activation	['identity', 'logistic', 'tanh', 'relu']
	solver	['adam', 'sgd', 'lbfgs']
	learning_rate	['constant', 'invscaling', 'adaptive']

Throughout each iteration of the 5-fold cross-validation, the accuracy score of the classifier is recorded. Upon completion of all iterations, the set of hyperparameter values yielding the highest average accuracy score across the folds is selected and applied to the model for final fitting. This whole process is included in the fitting time because the process depends on the type of classifier used. Then, the model continues as usual with prediction until the evaluation metrics are calculated. Both results from the feature selection and hyperparameter tuning phase are tabulated.

Lastly, based on all tabulated results, the optimal combination of modifications for each model in each dataset is chosen. By comparing between the models, the overall optimal model for each dataset can be identified. In the next section, all the tabulations of results mentioned above will be presented and discussed, alongside the optimal modification chosen for each phase for each model, the optimal combination of modifications for each model, and the optimal model overall.

4 RESULTS AND DISCUSSION

This section presents the tabulation of results obtained from this study, which aimed to evaluate and compare the performance of the base and modified models across four benchmark credit-scoring datasets. Subsequently based on the tabulation of data, some interesting findings and limitations that can be observed are discussed. Lastly, the optimal combination of modifications for each model in each dataset and the overall optimal model for each dataset is presented along with a few suggested alternatives.

As mentioned in previous sections, the models would involve LR, LDA, SVM, RF, and MLP along

with various modifications in feature scaling, resampling, feature selection, and hyperparameter tuning techniques. Each model's performance is evaluated based on key metrics including ACC, P, R, F1, and AUC. Credit scoring datasets involved include AC, GC, LC, and GMSC datasets.

It should be noted that in all subsequent tables within this section, values are highlighted with bold font when they are the highest among others within the same metric category. The values in the tables are indicative of the model's performance metrics, and higher values are desirable across ACC, F1, P, R, and AUC. For example, a model with an ACC of 90% would be better than a model with an ACC of 85%. Similarly, this is true for F1, P, R, and AUC. As described in the previous section, these metrics are derived from the analysis of the testing set, which is 20% of the whole dataset. Besides, it is also noteworthy that the emphasis in model selection will be placed on ACC and AUC, while the examination of additional metrics is still undertaken to provide supplementary insights. Recognizing that the best-performing model may not universally excel across all metrics, this evaluation aims to guide the selection of models based on accuracy and discrimination capabilities.

4.1 Base Models

This subsection analyzes the performance of five base credit scoring models using five key metrics across four benchmark credit scoring datasets. The models evaluated include Logistic Regression (LR), Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), Random Forest (RF), and Multi-Layer Perceptron (MLP).

Table 3 presents a detailed comparison of these models, showing their performance in terms of accuracy (ACC), F1 score (F1), precision (P), recall (R), and the area under the ROC curve (AUC) across the datasets.

Based on the analysis, LDA consistently performed well across most metrics, while SVM had the best precision on the AC dataset. Conversely, MLP performed the worst across all metrics on the AC dataset.

On the GC dataset, both LDA and RF exhibited strong performance. LDA achieved the highest accuracy and AUC, showcasing its ability to handle imbalanced data effectively. RF excelled in recall, highlighting its strength in capturing positive instances within an imbalanced dataset. This indicates that both LDA and RF are capable of managing the challenges posed by imbalanced datasets, each excelling in different aspects.

The consistency in results from the AC and GC datasets can be attributed to the size and balance of these datasets. The AC dataset (690 instances) is almost balanced (383 negative vs. 307 positive instances), while the GC dataset (1000 instances) is imbalanced (700 negative vs. 300 positive instances). This imbalance in the GC dataset contributes to the lower AUC scores observed but also emphasizes the strength of RF in recall and LDA's overall balanced performance.

For the LC dataset, RF outperformed other models across most metrics, while SVM had the highest precision. Both SVM and LR performed poorly overall, particularly in F1 and recall, due to the significant imbalance in the dataset (1,422,314 negative vs. 347,632 positive instances). This suggests that LDA, MLP, and especially RF are more resilient to imbalanced data when sufficient

Table 3 : Comparison of Base Models on Various Datasets

Dataset	Model	ACC (%)	F1 (%)	P (%)	R (%)	AUC (%)
AC	LR	82.6087	79.6610	87.0370	73.4375	81.9890
	LDA	**92.0290**	**91.7293**	88.4058	**95.3125**	**92.2508**
	SVM	89.1304	87.8049	**91.5254**	84.3750	88.8091
	RF	86.9565	85.2459	89.6552	81.2500	86.5709
	MLP	72.4638	68.8525	72.4138	65.6250	72.0017
GC	LR	72.0000	81.5789	76.0736	87.9433	60.9208
	LDA	**74.5000**	82.9431	**78.4810**	87.9433	**65.1581**
	SVM	73.0000	82.2368	76.6871	88.6525	62.1229
	RF	74.0000	**83.4395**	75.7225	**92.9078**	60.8607
	MLP	50.5000	54.3779	77.6316	41.8440	56.5152
LC	LR	80.3986	0.4676	51.2579	0.2349	50.0902
	LDA	89.9486	74.0412	74.9837	73.1220	83.5868
	SVM	82.7295	24.4544	**85.8122**	14.2590	56.8421
	RF	**90.1853**	**75.0830**	74.7373	**75.4319**	**84.6073**
	MLP	89.6313	73.2743	74.0584	72.5067	83.1568
GMSC	LR	93.2800	3.9085	48.2353	2.0368	50.9398
	LDA	**93.6000**	16.7224	**52.7704**	9.9354	54.6479
	SVM	92.7233	3.2787	15.1639	1.8381	50.5492
	RF	93.4100	26.8049	52.6163	17.9831	58.4091
	MLP	92.7167	**38.0493**	44.3197	**33.3333**	**65.1606**

data is available (1,769,946 instances).

On the GMSC dataset, LDA achieved the best accuracy and precision, while MLP excelled in F1, recall, and AUC. Despite this, all models showed high accuracy but performed poorly in F1, precision, recall, and AUC, reflecting the high imbalance in the dataset (139,974 negative vs. 10,026 positive instances). This indicates that while LDA, RF, and MLP show some resilience to imbalanced data, their performance still suffers when the dataset is highly imbalanced or not large enough (150,000 instances).

The performance of these models is influenced by the characteristics of the datasets. The AC dataset, with its balance between positive and negative instances, allowed LDA to excel. Conversely, the GC dataset's imbalance highlighted the strength of ensemble methods like RF. The large LC dataset underscored RF's ability to handle high variability in large volumes of data, while the high imbalance of the GMSC dataset revealed the challenges even robust models face under such conditions.

In summary, LDA and RF generally provided the best performance across different metrics and datasets. The specific strengths and weaknesses of each model were influenced by dataset characteristics such as size and balance. MLP, despite its challenges, demonstrated significant potential on the GMSC dataset, suggesting it may be particularly effective in certain scenarios.

4.2 Adding Feature Scaling

This subsection examines the performance of various feature scaling methods, including Min-Max Scaling, Standardization, Max-Abs Scaling, and Robust Scaling, applied to five base models across four credit scoring datasets. The analysis focuses on five key metrics: Accuracy (ACC), F1 Score (F1), Precision (P), Recall (R), and Area Under the Curve (AUC). Tables 4 - 7 display the comparative results for each model under different scaling methods on each dataset.

Table 4 : Comparison of Models on AC Dataset by Feature Scaling Methods

Model	Scaling	ACC (%)	F1 (%)	P (%)	R (%)	AUC (%)
LR	Min-Max	91.3043	90.7692	89.3939	92.1875	91.3640
	Standard	**92.7536**	**92.1875**	**92.1875**	**92.1875**	**92.7154**
	Max-Abs	91.3043	90.7692	89.3939	92.1875	91.3640
	Robust	91.3043	90.4762	91.9355	89.0625	91.1529
LDA	Min-Max	**92.0290**	**91.7293**	**88.4058**	**95.3125**	**92.2508**
	Standard	**92.0290**	**91.7293**	**88.4058**	**95.3125**	**92.2508**
	Max-Abs	**92.0290**	**91.7293**	**88.4058**	**95.3125**	**92.2508**
	Robust	**92.0290**	**91.7293**	**88.4058**	**95.3125**	**92.2508**
SVM	Min-Max	**89.8551**	**89.7059**	**84.7222**	**95.3125**	**90.2238**
	Standard	**89.8551**	**89.7059**	**84.7222**	**95.3125**	**90.2238**
	Max-Abs	**89.8551**	**89.7059**	**84.7222**	**95.3125**	**90.2238**
	Robust	**89.8551**	**89.7059**	**84.7222**	**95.3125**	**90.2238**
RF	Min-Max	87.6812	86.1789	89.8305	82.8125	87.3522
	Standard	86.9565	85.2459	89.6552	81.2500	86.5709
	Max-Abs	**89.1304**	**87.8049**	**91.5254**	**84.3750**	**88.8091**
	Robust	88.4058	86.8852	91.3793	82.8125	88.0279
MLP	Min-Max	**92.0290**	**91.3386**	92.0635	**90.6250**	**91.9341**
	Standard	**92.0290**	91.2000	**93.4426**	89.0625	91.8285
	Max-Abs	89.8551	88.8889	90.3226	87.5000	89.6959
	Robust	91.3043	90.3226	93.3333	87.5000	91.0473

In Table 4, which covers the AC dataset, it is evident that Standardization consistently outperformed other scaling methods for LR across all metrics. Max-Abs Scaling yielded the best results for the RF on this dataset. For MLP, Min-Max Scaling achieved the highest scores in most metrics, except for precision, where Standardization was superior. LDA showed no variation in performance regardless of the scaling method, indicating its robustness to different feature scaling techniques. SVM also displayed consistent performance across all scaling methods but showed noticeable improvement compared to not using scaling.

Moving to the GC dataset, as shown in Table 5, Min-Max Scaling emerged as the best performer for the LR model across all metrics. The RF model continued to perform optimally with Max-Abs Scaling, similar to its performance on the AC dataset. The SVM model, when scaled with Min-Max and Max-Abs methods, achieved the best results for most metrics, except for AUC, where Standardization was superior. The MLP model performed best with Max-Abs Scaling in terms of accuracy, F1 score, and recall, while Min-Max Scaling was best for precision and AUC. LDA once again remained unaffected by the choice of scaling method, similar to the observations in the AC dataset.

Table 5 : Comparison of Models on GC Dataset by Feature Scaling Methods

Model	Scaling	ACC (%)	F1 (%)	P (%)	R (%)	AUC (%)
LR	Min-Max	**77.0000**	**84.9673**	**78.7879**	**92.1986**	**66.4383**
	Standard	75.0000	83.3333	78.6164	88.6525	65.5127
	Max-Abs	76.5000	84.5902	78.6585	91.4894	66.0837
	Robust	75.5000	83.8284	78.3951	90.0709	65.3744
LDA	Min-Max	**74.5000**	**82.9431**	**78.4810**	**87.9433**	**65.1581**
	Standard	**74.5000**	**82.9431**	**78.4810**	**87.9433**	**65.1581**
	Max-Abs	**74.5000**	**82.9431**	**78.4810**	**87.9433**	**65.1581**
	Robust	**74.5000**	**82.9431**	**78.4810**	**87.9433**	**65.1581**
SVM	Min-Max	**76.0000**	**84.2105**	**78.5276**	**90.7801**	65.7291
	Standard	75.5000	83.7209	78.7500	89.3617	**65.8673**
	Max-Abs	**76.0000**	**84.2105**	**78.5276**	**90.7801**	65.7291
	Robust	75.0000	83.5526	77.9141	90.0709	64.5270
RF	Min-Max	75.0000	83.6601	77.5758	90.7801	64.0341
	Standard	75.0000	84.0764	76.3006	93.6170	62.0627
	Max-Abs	**76.0000**	**84.6154**	**77.1930**	**93.6170**	**63.7577**
	Robust	74.5000	83.3876	77.1084	90.7801	63.1867
MLP	Min-Max	75.0000	82.9932	**79.7386**	86.5248	**66.9912**
	Standard	73.0000	81.2500	79.5918	82.9787	66.0656
	Max-Abs	**75.5000**	**83.6120**	79.1139	**88.6525**	66.3601
	Robust	72.5000	80.9689	79.0541	82.9787	65.2182

Table 6 : Comparison of Models on LC Dataset by Feature Scaling Methods

Model	Scaling	ACC (%)	F1 (%)	P (%)	R (%)	AUC (%)
LR	Min-Max	89.9282	73.3358	76.2314	70.6521	82.6403
	Standard	**89.9463**	**73.3749**	**76.2992**	**70.6665**	**82.6570**
	Max-Abs	89.9285	73.3291	76.2473	70.6261	82.6307
	Robust	86.1680	59.3693	69.9853	51.5498	73.0795
LDA	Min-Max	**89.9486**	**74.0412**	**74.9837**	**73.1220**	**83.5868**
	Standard	**89.9486**	**74.0412**	**74.9837**	**73.1220**	**83.5868**
	Max-Abs	**89.9486**	**74.0412**	**74.9837**	**73.1220**	**83.5868**
	Robust	**89.9486**	**74.0412**	**74.9837**	**73.1220**	**83.5868**
SVM	Min-Max	89.9373	73.4890	**75.9928**	71.1449	82.8322
	Standard	**89.9718**	**73.6711**	75.9009	**71.5686**	**83.0139**
	Max-Abs	89.9370	73.4884	75.9916	71.1449	82.8321
	Robust	88.8322	70.2945	73.4440	67.4040	80.7306
RF	Min-Max	**90.2099**	**75.1220**	**74.8459**	**75.4002**	**84.6106**
	Standard	90.1726	75.0502	74.7062	75.3974	84.5864
	Max-Abs	90.1867	75.0693	74.7752	75.3657	84.5832
	Robust	90.1966	75.0921	74.8048	75.3815	84.5953
MLP	Min-Max	**90.4091**	75.7553	75.0878	76.4349	85.1257
	Standard	90.3983	**75.7431**	75.0304	76.4695	85.1321
	Max-Abs	90.3071	75.6089	74.6096	**76.6352**	**85.1380**
	Robust	90.2353	74.5995	**76.1130**	73.1450	83.7738

Table 7 : Comparison of Models on GMSC Dataset by Feature Scaling Methods

Model	Scaling	ACC (%)	F1 (%)	P (%)	R (%)	AUC (%)
LR	Min-Max	93.2933	2.8958	50.8475	1.4903	50.6933
	Standard	**93.4267**	**8.8725**	**63.5762**	**4.7690**	**52.2862**
	Max-Abs	93.2933	2.8958	50.8475	1.4903	50.6933
	Robust	93.2733	2.8874	46.1538	1.4903	50.6826
LDA	Min-Max	**93.3600**	**16.7224**	**52.7704**	**9.9354**	**54.6479**
	Standard	**93.3600**	**16.7224**	**52.7704**	**9.9354**	**54.6479**
	Max-Abs	**93.3600**	**16.7224**	**52.7704**	**9.9354**	**54.6479**
	Robust	**93.3600**	**16.7224**	**52.7704**	**9.9354**	**54.6479**
SVM	Min-Max	93.2900	3.0814	50.0000	1.5897	**50.7377**
	Standard	**93.3033**	0.9857	**62.5000**	0.4968	50.2377
	Max-Abs	93.2900	3.0814	50.0000	1.5897	**50.7377**
	Robust	85.3500	**8.0352**	6.9414	**9.5380**	50.1704
RF	Min-Max	93.4167	27.0949	52.7299	18.2315	58.5280
	Standard	93.4100	26.9671	52.5937	18.1321	58.4783
	Max-Abs	**93.4900**	**28.1192**	**54.2614**	**18.9767**	**58.9131**
	Robust	93.4467	27.4003	53.3813	18.4302	58.6363
MLP	Min-Max	**93.6633**	25.5386	60.3704	16.1947	57.7150
	Standard	93.5467	**26.4996**	56.1997	**17.3373**	**58.1827**
	Max-Abs	93.6200	25.0587	59.1497	15.8967	57.5535
	Robust	93.5967	19.9250	**61.9171**	11.8728	55.6738

Table 6 presents the results for the LC dataset. Here, LR again showed the best performance with Standardization, consistent with its performance on the AC dataset. However, the RF model now performed best with Min-Max Scaling across all metrics, differing from its optimal scaling method in the other datasets. The SVM model showed the highest overall performance with Standardization but had the best precision with Min-Max Scaling. For the MLP model, the results were mixed: Min-Max Scaling was best for accuracy and F1 score, Robust Scaling for precision, and Max-Abs Scaling for recall and AUC. Although Standardization did not top any single metric for MLP, it remained a strong overall performer alongside Min-Max Scaling.

Lastly, the GMSC dataset, presented in Table 7, reveals that Standardization was the most effective scaling method for the LR model across most metrics. For the LDA model, all scaling methods yielded identical results, consistent with its performance on other datasets. The SVM model performed best with Min-Max Scaling for most metrics, except for AUC, where Max-Abs Scaling was superior. The RF model showed the best performance with Max-Abs Scaling, similar to its performance on the AC and GC datasets. For MLP, Min-Max Scaling and Standardization were both strong performers, with the former excelling in precision and the latter in recall.

In summary, the analysis indicates that the optimal feature scaling method varies depending on the model and dataset. Standardization and Max-Abs Scaling often emerged as top performers across different models and datasets, while LDA remained largely insensitive to the choice of scaling method. This variability underscores the importance of carefully selecting and testing feature scaling methods to achieve the best model performance for specific applications. Table 8 shows the feature scaling methods that are determined to be the best and are chosen to be carried on to subsequent subsections.

Table 8 : Chosen Feature Scaling Methods for Each Model in Each Dataset

Model	AC	GC	LC	GMSC
LR	Standard	Min-Max	Standard	Standard
LDA	None	None	None	None
SVM	Min-Max	Max-Abs	Standard	Standard
RF	Max-Abs	Max-Abs	Min-Max	Max-Abs
MLP	Min-Max	Max-Abs	Min-Max	Standard

4.3 Adding Resampling

This subsection presents the analysis of the performance of resampling methods, including Random Undersampling (RUS) and Synthetic Minority Oversampling Technique (SMOTE), applied to the five models scaled based on methods as shown in Table 8. Their performances will be compared based on the aforementioned five metrics across the four datasets. Tables 9 - 12 present the comparison of these models between resampling methods with each table depicting the results on each dataset.

Table 9 : Comparison of Models on AC Dataset by Resampling Methods

Model	**Sampling**	**ACC (%)**	**F1 (%)**	**P (%)**	**R (%)**	**AUC (%)**
LR	RUS	82.1138	82.2581	78.4615	86.4407	82.2828
	SMOTE	**85.7143**	**85.7143**	**82.5000**	**89.1892**	**85.8446**
LDA	RUS	83.7398	84.1270	79.1045	89.8305	83.9778
	SMOTE	**84.4156**	**84.8101**	**79.7619**	**90.5405**	**84.6453**
SVM	RUS	83.7398	84.3750	78.2609	91.5254	84.0440
	SMOTE	**85.0649**	**85.5346**	**80.0000**	**91.8919**	**85.3209**
RF	RUS	83.7398	82.7586	84.2105	81.3559	83.6467
	SMOTE	**89.6104**	**89.3333**	**88.1579**	**90.5405**	**89.6453**
MLP	RUS	**87.8049**	**87.3950**	**86.6667**	88.1356	**87.8178**
	SMOTE	86.3636	86.6242	81.9277	**91.8919**	86.5709

Table 10 : Comparison of Models on GC Dataset by Resampling Methods

Model	**Sampling**	**ACC (%)**	**F1 (%)**	**P (%)**	**R (%)**	**AUC (%)**
LR	RUS	69.1667	68.9076	69.4915	68.3333	69.1667
	SMOTE	**85.7143**	**85.9155**	**82.9932**	**89.0511**	**85.7843**
LDA	RUS	67.5000	67.2269	67.7966	66.6667	67.5000
	SMOTE	**85.0000**	**85.3147**	**81.8792**	**89.0511**	**85.0850**
SVM	RUS	67.5000	66.6667	68.4211	65.0000	67.5000
	SMOTE	**84.6429**	**85.0174**	**81.3333**	**89.0511**	**84.7353**
RF	RUS	66.6667	65.5172	67.8571	63.3333	66.6667
	SMOTE	**85.0000**	**85.1064**	**82.7586**	**87.5912**	**85.0544**
MLP	RUS	67.5000	67.7686	67.2131	68.3333	67.5000
	SMOTE	**81.7857**	**82.1053**	**79.0541**	**85.4015**	**81.8616**

For this subsection, there is a clear pattern that can be observed throughout all datasets with minor

Table 11 : Comparison of Models on LC Dataset by Resampling Methods

Model	Sampling	ACC (%)	F1 (%)	P (%)	R (%)	AUC (%)
LR	RUS	88.5547	88.5930	88.5783	88.6076	88.5546
	SMOTE	**91.6842**	**91.7276**	**91.4209**	**92.0364**	**91.6835**
LDA	RUS	88.5562	88.6783	88.0164	89.3502	88.5536
	SMOTE	**91.7708**	**91.8718**	**90.9252**	**92.8382**	**91.7688**
SVM	RUS	88.4303	88.4893	88.3187	88.6606	88.4296
	SMOTE	**91.7393**	**91.7949**	**91.3511**	**92.2431**	**91.7384**
RF	RUS	88.6655	88.9310	87.1601	90.7754	88.6587
	SMOTE	**93.6122**	**93.7320**	**92.1725**	**95.3452**	**93.6089**
MLP	RUS	88.8848	89.1722	87.1883	91.2485	88.8773
	SMOTE	**92.9546**	**92.9063**	**93.7265**	**92.1003**	**92.9562**

Table 12 : Comparison of Models on GMSC Dataset by Resampling Methods

Model	Sampling	ACC (%)	F1 (%)	P (%)	R (%)	AUC (%)
LR	RUS	**72.8746**	**69.5920**	**77.2333**	63.3266	**72.6902**
	SMOTE	65.7707	65.3279	65.9760	**64.6924**	65.7674
LDA	RUS	63.5752	63.5206	62.3835	**64.6999**	63.5969
	SMOTE	**65.0616**	**64.7735**	**65.1088**	64.4416	**65.0597**
SVM	RUS	63.6500	63.2004	62.7255	63.6826	63.6506
	SMOTE	**65.1652**	**64.8310**	**65.2546**	**64.4129**	**65.1629**
RF	RUS	78.1102	77.3244	78.5414	76.1445	78.0722
	SMOTE	**92.1111**	**92.0827**	**92.1306**	**92.0348**	**92.1109**
MLP	RUS	88.9129	88.7570	90.3187	87.2484	88.9182
	SMOTE	**93.1167**	**93.0492**	**93.7908**	**92.3192**	**93.1152**

exceptions, hence the results will be discussed together here. Based on Table 9 - 12, it can be observed that SMOTE works best in most cases when the data is imbalanced (GC, LC, and GMSC datasets), especially on the GC and LC datasets where it scored the highest across all metrics. The only exception is where LR worked better with RUS on the GMSC dataset.

For the case where data is small and balanced, even though SMOTE mostly worked better than RUS, observe from Table 4 that LR, LDA, SVM, and MLP all performed better without any resampling done on the AC dataset. RF still performed better after SMOTE was applied. Also, notice that the effect of SMOTE is highly effective for RF and MLP on the GMSC dataset based on Table 12.

The overall results based on Tables 9 - 12 show that SMOTE sampling is suitable for most cases especially when the dataset is imbalanced, while no resampling is required when the dataset is small and balanced. Table 13 shows the resampling methods that are determined to be the best and are chosen to be carried on to subsequent subsections.

Table 13 : Chosen Resampling Methods for Each Model in Each Dataset

Model	AC	GC	LC	GMSC
LR	None	SMOTE	SMOTE	RUS
LDA	None	SMOTE	SMOTE	SMOTE
SVM	None	SMOTE	SMOTE	SMOTE
RF	SMOTE	SMOTE	SMOTE	SMOTE
MLP	None	SMOTE	SMOTE	SMOTE

4.4 Adding Feature Selection and Hyperparameter Tuning

This subsection provides an analysis of the performance of feature selection and hyperparameter tuning methods applied to five models that have been scaled and resampled using the methods outlined in Table 8 and Table 13. The feature selection methods include the Analysis of Variance (ANOVA) F-test and Mutual Information (MI), while the hyperparameter tuning methods are Grid Search (GS) and Random Search (RS). Their performances are compared across five metrics for four datasets, with the results summarized in Tables 14 - 17.

In the AC dataset, Table 14 reveals that SVM was insensitive to any combinations of feature selection or hyperparameter tuning, as indicated by unchanged metric values before and after these methods were applied (see Table 4). A similar pattern was observed for LDA, with consistently lower scores across all metrics for both ANOVA and MI without hyperparameter tuning. Most models showed indifference to the type of feature selection method when hyperparameter tuning was not involved, except for MLP, which performed better with MI, as seen by its higher scores across metrics in Table 14. Generally, MI performed better or equally well compared to ANOVA on the AC dataset.

The best performance on the AC dataset was achieved by the MLP model with either ANOVA-RS or MI-GS combinations. However, it is noteworthy that LR with Standardization performed equally well without any resampling, feature selection, or hyperparameter tuning. These three models emerged as the best overall for the AC dataset. When comparing their fitting and prediction times, LR with Standardization was the fastest, making it the best model for the AC dataset.

For the GC dataset, Table 15 shows that ANOVA and MI performed equally well across all metrics, regardless of the hyperparameter tuning method used. GS outperformed other methods for LR across all five metrics. For LDA and SVM, both ANOVA and MI performed better without hyperparameter tuning, possibly due to overfitting. RF performed better with GS than with RS for both ANOVA and MI. For MLP, ANOVA showed overfitting issues with both search methods, although GS still performed better. Interestingly, Random Search performed better than Grid Search with MI, indicating that Random Search can sometimes provide similar or better results due to its inherent variability.

LDA with ANOVA and MI without hyperparameter tuning performed the best on the GC dataset, but the overall best model was LR with Min-Max Scaling and SMOTE, demonstrating that LR is effective for small datasets, while SMOTE helps with imbalanced datasets.

Table 16 presents the performance comparison on the LC dataset, noting that data on GS for

Table 14 : Comparison of Models on AC Dataset by Feature Selection and Hyperparameter Tuning Methods

Model	Selection	Search	ACC (%)	F1 (%)	P (%)	R (%)	AUC (%)
LR	ANOVA	None	91.3043	90.6250	90.6250	90.6250	91.2584
		Grid	90.5797	89.6000	91.8033	87.5000	90.3716
		Random	87.6812	85.9504	91.2281	81.2500	87.2466
	MI	None	91.3043	90.6250	90.6250	90.6250	91.2584
		Grid	90.5797	89.6000	91.8033	87.5000	90.3716
		Random	**92.0290**	**91.4729**	90.7692	**92.1875**	**92.0397**
LDA	ANOVA	None	90.5797	90.2256	86.9565	93.7500	90.7939
		Grid	**92.0290**	**91.7293**	**88.4058**	**95.3125**	**92.2508**
		Random	**92.0290**	**91.7293**	**88.4058**	**95.3125**	**92.2508**
	MI	None	90.5797	90.2256	86.9565	93.7500	90.7939
		Grid	**92.0290**	**91.7293**	**88.4058**	**95.3125**	**92.2508**
		Random	**92.0290**	**91.7293**	**88.4058**	**95.3125**	**92.2508**
SVM	ANOVA	None	**89.8551**	**89.7059**	**84.7222**	**95.3125**	**90.2238**
		Grid	**89.8551**	**89.7059**	**84.7222**	**95.3125**	**90.2238**
		Random	**89.8551**	**89.7059**	**84.7222**	**95.3125**	**90.2238**
	MI	None	**89.8551**	**89.7059**	**84.7222**	**95.3125**	**90.2238**
		Grid	**89.8551**	**89.7059**	**84.7222**	**95.3125**	**90.2238**
		Random	**89.8551**	**89.7059**	**84.7222**	**95.3125**	**90.2238**
RF	ANOVA	None	**88.9610**	**88.7417**	**87.0130**	**90.5405**	**89.0203**
		Grid	87.0130	86.8421	84.6154	89.1892	87.0946
		Random	85.7143	85.8974	81.7073	**90.5405**	85.8953
	MI	None	**88.9610**	**88.7417**	**87.0130**	**90.5405**	**89.0203**
		Grid	87.0130	86.8421	84.6154	89.1892	87.0946
		Random	88.3117	88.1579	85.8974	**90.5405**	88.3953
MLP	ANOVA	None	88.4058	87.3016	88.7097	85.9375	88.2390
		Grid	91.3043	90.4762	91.9355	89.0625	91.1529
		Random	**92.7536**	**92.1875**	**92.1875**	**92.1875**	**92.7154**
	MI	None	89.1304	88.0000	90.1639	85.9375	88.9147
		Grid	**92.7536**	**92.1875**	**92.1875**	**92.1875**	**92.7154**
		Random	92.0290	91.4729	90.7692	**92.1875**	92.0397

RF and MLP is absent due to computational limitations. Despite these limitations, ANOVA and MI performed similarly without hyperparameter tuning. GS and RS performed equally well for LR, LDA, and SVM. However, only LR showed improvement with hyperparameter tuning. RF performed better with RS than without hyperparameter tuning, while MLP showed signs of overfitting. RF with ANOVA and RS was the best-performing model overall, with RF with MI and RS being a good alternative due to its shorter fitting time of 1.30 hours, which is approximately one-third of its ANOVA counterpart of 3.68 hours.

On the GMSC dataset, Table 17 shows that ANOVA and MI yielded similar results for LR, except for MI-RS. LDA showed consistent patterns as in the GC and LC datasets, while SVM began to show inconsistencies with different feature selection and hyperparameter tuning combinations, suggesting these methods have varying effects as the dataset size and imbalance increase. RF outperformed other models significantly on the GMSC dataset, demonstrating its effectiveness for

Table 15 : Comparison of Models on GC Dataset by Feature Selection and Hyperparameter Tuning Methods

Model	Selection	Search	ACC (%)	F1 (%)	P (%)	R (%)	AUC (%)
LR	ANOVA	None	79.6429	78.6517	80.7692	76.6423	79.5799
		Grid	**80.7143**	**79.8507**	**81.6794**	**78.1022**	**80.6595**
		Random	80.3571	79.5539	81.0606	78.1022	80.3098
	MI	None	79.6429	78.6517	80.7692	76.6423	79.5799
		Grid	**80.7143**	**79.8507**	**81.6794**	**78.1022**	**80.6595**
		Random	80.3571	79.5539	81.0606	78.1022	80.3098
LDA	ANOVA	None	**85.3571**	**85.6140**	**82.4324**	**89.0511**	**85.4346**
		Grid	84.2857	84.7222	80.7947	**89.0511**	84.3857
		Random	84.2857	84.7222	80.7947	**89.0511**	84.3857
	MI	None	**85.3571**	**85.6140**	**82.4324**	**89.0511**	**85.4346**
		Grid	84.2857	84.7222	80.7947	**89.0511**	84.3857
		Random	84.2857	84.7222	80.7947	**89.0511**	84.3857
SVM	ANOVA	None	**82.5000**	**82.0513**	**82.3529**	**81.7518**	**82.4843**
		Grid	78.9286	77.7358	80.4688	75.1825	78.8500
		Random	78.9286	77.7358	80.4688	75.1825	78.8500
	MI	None	**82.5000**	**82.0513**	**82.3529**	**81.7518**	**82.4843**
		Grid	78.9286	77.7358	80.4688	75.1825	78.8500
		Random	78.9286	77.7358	80.4688	75.1825	78.8500
RF	ANOVA	None	**78.9286**	**78.0669**	**79.5455**	**76.6423**	**78.8806**
		Grid	**78.9286**	**78.0669**	**79.5455**	**76.6423**	**78.8806**
		Random	76.0714	75.6364	75.3623	75.9124	76.0681
	MI	None	78.2143	77.6557	77.9412	77.3723	78.1966
		Grid	**80.7143**	**80.1471**	**80.7407**	**79.5620**	**80.6901**
		Random	79.2857	78.6765	79.2593	78.1022	79.2609
MLP	ANOVA	None	**79.6429**	**78.8104**	**80.3030**	**77.3723**	**79.5952**
		Grid	78.2143	76.9811	79.6875	74.4526	78.1354
		Random	75.7143	74.4361	76.7442	72.2628	75.6419
	MI	None	78.2143	77.3234	78.7879	75.9124	78.1660
		Grid	77.5000	76.4045	78.4615	74.4526	77.4361
		Random	**78.5714**	**77.7778**	**78.9474**	**76.6423**	**78.5310**

large and highly imbalanced datasets, while MLP also performed well regardless of the combinations used.

Overall, ANOVA and MI performed similarly well for all models across datasets, especially for LR and LDA. GS generally outperformed RS, but RS reduced fitting time significantly for large datasets and provided competitive results due to its variability. Hence, the optimal hyperparameter tuning method depends on the specific goals and the classifiers or datasets used in the analysis.

Lastly, based on the overall results from Tables 3 - 17 and conclusions made for each dataset, Tables 18 and 19 present the optimal models for each dataset and their suggested close-performing alternatives.

Table 16 : Comparison of Models on LC Dataset by Feature Selection and Hyperparameter Tuning Methods[a,b]

Model	Selection	Search	ACC (%)	F1 (%)	P (%)	R (%)	AUC (%)
LR	ANOVA	None	90.2961	90.3938	89.6581	91.1417	90.2945
		Grid	90.3362	90.4312	**89.7158**	**91.1582**	90.3346
		Random	**90.3365**	**90.4314**	89.7175	91.1568	**90.3350**
	MI	None	90.2961	90.3938	89.6581	91.1417	90.2945
		Grid	**90.3362**	**90.4312**	**89.7158**	**91.1582**	**90.3346**
		Random	**90.3362**	90.4311	89.7172	91.1564	**90.3346**
LDA	ANOVA	None	**90.4826**	**90.6777**	**89.0171**	**92.4014**	**90.4790**
		Grid	90.4559	90.6529	88.9798	92.3901	90.4522
		Random	90.4559	90.6529	88.9798	92.3901	90.4522
	MI	None	**90.4826**	**90.6777**	**89.0171**	**92.4014**	**90.4790**
		Grid	90.4559	90.6529	88.9798	92.3901	90.4522
		Random	90.4559	90.6529	88.9798	92.3901	90.4522
SVM	ANOVA	None	**90.3437**	**90.4558**	89.5828	**91.3459**	**90.3418**
		Grid	90.3081	90.4173	89.5739	91.2768	90.3062
		Random	90.2903	90.3905	**89.6325**	91.1613	90.2887
	MI	None	**90.3427**	**90.4548**	**89.5813**	**91.3456**	**90.3408**
		Grid	90.3128	90.4225	89.5746	91.2866	90.3110
		Random	90.3017	90.4117	89.5625	91.2771	90.2999
RF	ANOVA	None	93.6292	93.4900	**95.3910**	91.6634	93.6255
		Grid	-	-	-	-	-
		Random	**93.7038**	**93.7536**	93.1893	**94.3249**	**93.7026**
	MI	None	93.6102	93.4725	**95.3449**	91.6722	93.6066
		Grid	-	-	-	-	-
		Random	**93.6289**	**93.6803**	93.1023	**94.2656**	**93.6277**
MLP	ANOVA	None	**92.9147**	**92.8288**	93.7878	**91.8891**	**92.9128**
		Grid	-	-	-	-	-
		Random	92.2055	92.0215	**94.4329**	89.7301	92.2102
	MI	None	**92.8947**	**92.8270**	**93.5407**	92.1241	**92.8932**
		Grid	-	-	-	-	-
		Random	91.9541	92.1132	90.4912	**93.7943**	91.9507

[a] "-" indicates unavailable data. Data is unavailable due to insufficient computational resources.
[b] Results are based on 2-fold cross-validation.

5 CONCLUSION

As empirical research was done for each model on four benchmark credit scoring datasets (Australian, German, Lending Club, and Give Me Some Credit 2011 Competition Datasets) and less optimal methods were eliminated from each of the four modification phases, it was found that LR was sufficient for small datasets while RF and MLP were better for larger datasets. As for the modifications, Min-Max Scaling worked well in general, Max-Abs Scaling mostly paired well with RF, Standardization paired well with LR, and Robust Scaling did not perform well with any model. SMOTE was preferred for imbalanced datasets while no sampling is required when datasets are small and balanced. Feature selection and hyperparameter tuning did not always improve the performance of models due to overfitting. ANOVA and MI performed similarly without hyperparameter tuning,

Table 17 : Comparison of Models on GMSC Dataset by Feature Selection and Hyperparameter Tuning Methods

Model	Selection	Search	ACC (%)	F1 (%)	P (%)	R (%)	AUC (%)
LR	ANOVA	None	69.9825	68.4155	70.6392	**66.3276**	69.9120
		Grid	**72.7250**	**70.0602**	**75.8294**	65.1068	**72.5779**
		Random	**72.7250**	**70.0602**	**75.8294**	65.1068	**72.5779**
	MI	None	69.9825	68.4155	70.6392	**66.3276**	69.9120
		Grid	**72.7250**	70.0602	**75.8294**	65.1068	**72.5779**
		Random	72.7001	**70.0738**	75.7236	65.2085	72.5554
LDA	ANOVA	None	**61.5967**	**62.5479**	**60.8582**	**64.3341**	**61.6051**
		Grid	61.4074	62.2989	60.7142	63.9686	61.4152
		Random	61.4074	62.2989	60.7142	63.9686	61.4152
	MI	None	**61.5967**	**62.5479**	**60.8582**	**64.3341**	**61.6051**
		Grid	61.4074	62.2989	60.7142	63.9686	61.4152
		Random	61.4074	62.2989	60.7142	63.9686	61.4152
SVM	ANOVA	None	62.1629	62.9743	61.4713	**64.5527**	62.1702
		Grid	63.4006	51.7176	**75.5178**	39.3242	63.3269
		Random	**66.7191**	**64.4633**	68.9077	60.5575	**66.7002**
	MI	None	62.1682	62.9634	61.4862	64.5132	62.1754
		Grid	**66.4601**	**69.7771**	63.3378	**77.6739**	**66.4944**
		Random	63.0809	58.7907	**66.2637**	52.8324	63.0495
RF	ANOVA	None	90.6287	90.4595	**91.8303**	89.1290	90.6241
		Grid	**91.7575**	**91.7664**	91.3865	**92.1495**	**91.7587**
		Random	90.2250	90.0013	91.8145	88.2583	90.2190
	MI	None	90.6555	90.4720	**91.9898**	89.0035	90.6504
		Grid	**91.7485**	**91.7559**	91.3938	**92.1208**	**91.7497**
		Random	90.7966	90.6229	**92.0722**	89.2185	90.7917
MLP	ANOVA	None	83.3631	83.2816	83.4328	83.1309	83.3624
		Grid	**83.7900**	**83.6345**	**84.1809**	83.0951	**83.7878**
		Random	83.3810	83.3813	83.1244	**83.6397**	83.3818
	MI	None	**83.6953**	**83.6119**	83.7818	**83.4426**	**83.6945**
		Grid	83.6078	83.1038	**85.4606**	80.8736	83.5994
		Random	83.4078	82.9115	**85.1899**	80.7517	83.3996

while GS performed slightly better than RS disregarding runtime. Lastly, this study also presented an optimal model and a few suggested alternative models for each dataset.

REFERENCES

[1] B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen, "Benchmarking state-of-the-art classification algorithms for credit scoring," *Journal of the Operational Research Society*, vol. 54, no. 6, pp. 627–635, 2003.

[2] S. Lessmann, B. Baesens, H.-V. Seow, and L. C. Thomas, "Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research," *European Journal of Operational Research*, vol. 247, no. 1, pp. 124–136, 2015.

Table 18 : Optimal Model for Each Dataset

Dataset	Classifier	Scaling	Resampling	Selection	Tuning
AC	LR	Standard	None	None	None
GC	LR	Min-Max	SMOTE	None	None
LC	RF	Min-Max	SMOTE	ANOVA	Random Search
GMSC	MLP	Standard	SMOTE	None	None

Table 19 : Suggested Alternative Models for Each Dataset

Dataset	Classifier	Scaling	Resampling	Selection	Tuning
AC	MLP	Min-Max	None	ANOVA	Random Search
AC	MLP	Min-Max	None	MI	Grid Search
GC	LDA	None	SMOTE	ANOVA	None
GC	LDA	None	SMOTE	MI	None
LC	RF	Min-Max	SMOTE	MI	Random Search
LC	MLP	Min-Max	SMOTE	MI	Random Search
GMSC	RF	Max-Abs	SMOTE	None	None

[3] Y. Li and W. Chen, "A comparative performance assessment of ensemble learning for credit scoring," *Mathematics*, vol. 8, no. 10, p. 1756, 2020.

[4] S. K. Trivedi, "A study on credit scoring modeling with different feature selection and machine learning approaches," *Technology in Society*, vol. 63, p. 101413, 2020.

[5] A. Ampountolas, T. N. Nde, P. Date, and C. Constantinescu, "A machine learning approach for micro-credit scoring," *Risks*, vol. 9, no. 4, p. 50, 2021.

[6] X. Liu, H. Fu, and W. Lin, "A modified support vector machine model for credit scoring," *International Journal of Computational Intelligence Systems*, vol. 3, no. 6, pp. 797–804, 2010.

[7] A. Ghodselahi, "A hybrid support vector machine ensemble model for credit scoring," *International Journal of Computer Applications*, vol. 17, no. 5, pp. 1–5, 2011.

[8] X. Zhang, Y. Yang, and Z. Zhou, "A novel credit scoring model based on optimized random forest," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, 2018, pp. 60–65.

[9] D. Tripathi, D. R. Edla, A. Bablani, A. K. Shukla, and B. R. Reddy, "Experimental analysis of machine learning methods for credit score classification," *Progress in Artificial Intelligence*, vol. 10, no. 3, pp. 217–243, 2021.

[10] H.-W. Teng, M.-H. Kang, and I.-H. Lee, "Improving credit scoring: A rescaled cluster-then-predict approach," *SSRN Electronic Journal*, 2023.

[11] N. Kozodoi, J. Jacob, and S. Lessmann, "Fairness in credit scoring: Assessment, implementation and profit implications," *European Journal of Operational Research*, vol. 297, no. 3, pp. 1083–1094, 2022.

[12] D. Durand, "Risk elements in consumer instatement financing," National Bureau of Economic Research, New York, Tech. Rep., 1941.

[13] E. I. Altman, "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy," *The Journal of Finance*, vol. 23, pp. 589–609, 1968.

[14] J. A. Ohlson, "Financial ratios and the probabilistic prediction of bankruptcy," *Journal of Accounting Research*, vol. 18, no. 1, p. 109, 1980.

[15] R. K. Chhikara, "The state of the art in credit evaluation," *American Journal of Agricultural Economics*, vol. 71, no. 5, pp. 1138–1144, 1989.

[16] W. E. Hardy and J. L. Adrian, "A linear programming alternative to discriminant analysis in credit scoring," *Agribusiness*, vol. 1, no. 4, pp. 285–292, 1985.

[17] B. Zhu, W. Yang, H. Wang, and Y. Yuan, "A hybrid deep learning model for consumer credit scoring," in *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, 2018, pp. 205–208.

[18] F. Shen, R. Wang, and Y. Shen, "A cost-sensitive logistic regression credit scoring model based on multi-objective optimization approach," *Technological and Economic Development of Economy*, vol. 26, no. 2, pp. 405–429, 2019.

[19] L. Munkhdalai, J. Y. Lee, and K. H. Ryu, "Hybrid credit scoring model using classification methods and association rules," in *Advances in Intelligent Information Hiding and Multimedia Signal Processing*, J.-S. Pan, J. Li, P.-W. Tsai, and L. C. Jain, Eds. Springer International Publishing, 2020, pp. 251–258.

[20] E. Dumitrescu, S. Hué, C. Hurlin, and S. Tokpavi, "Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects," *European Journal of Operational Research*, vol. 297, no. 3, pp. 1178–1192, 2022.

[21] S. F. Crone and S. Finlay, "Instance sampling in credit scoring: An empirical study of sample size and balancing," *International Journal of Forecasting*, vol. 28, no. 1, pp. 224–238, 2012.

[22] A. I. Marqués, V. García, and J. S. Sánchez, "On the suitability of resampling techniques for the class imbalance problem in credit scoring," *Journal of the Operational Research Society*, vol. 64, no. 7, pp. 1060–1070, 2013.

[23] S.-J. Yen and Y.-S. Lee, "Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset," *Intelligent Control and Automation, Lecture Notes in Control and Information Sciences*, vol. 344, pp. 731–740, 2006.

[24] Y. E. Orgler, "A credit scoring model for commercial loans," *Journal of Money, Credit and Banking*, vol. 2, no. 4, p. 435, 1970.

[25] R. Y. Goh, L. S. Lee, H.-V. Seow, and K. Gopal, "Hybrid harmony search–artificial intelligence models in credit scoring," *Entropy*, vol. 22, no. 9, pp. 321–357, 2020.

[26] R. Quinlan. (1987) Statlog (Australian Credit Approval). UCI Machine Learning Repository. [Online]. Available: https://doi.org/10.24432/C59012

[27] H. Hofmann. (1994) Statlog (German Credit Data). UCI Machine Learning Repository. [Online]. Available: https://doi.org/10.24432/C5NC77

[28] Yash. (2020) Lending Club 2007-2020Q3. Kaggle. [Online]. Available: https://www.kaggle.com/datasets/ethon0426/lending-club-20072020q1

[29] W. Cukierski. (2011) Give Me Some Credit. Kaggle. [Online]. Available: https://kaggle.com/competitions/GiveMeSomeCredit