PROJECT REPORT

Bench Marking different Classification Methods

Prepared by Group 7:

Ashwini Awathe

Anurag Awathe

Brunda Somashekar

Dinesh Arasavalli

Harsha Vardhan Aitha

Rani Sravanthi Devi Lankalapalli

Abstract

This report presents a detailed description of comprehensive evaluation of seven classification methods applied to 20 diverse datasets using R. As different models perform in unique and distinct ways on the same dataset, we all know different models have different methods to categorize input data based on data structure, categorical predictors, and type of variables so we can expect distinct differences in output variables and performance parameters or metrics For Model evaluation parameters/metrics, we are considering two parts training and testing, As different models have different input training metrics, but for testing parameters, we are considering ROC, AUC, run time, confusion matrix,f1 value, accuracy and precision. An averaging is done for the value of 20 iterations for accuracy, precisions and moreover using same set of metrics for all classification models will greatly help during comparison. The datasets that are gathered, are from different online sources: Kaggle, UCI machine learning repository, and GitHub, For the project, we considered a few health or disease-related datasets through which we can predict the possibilities of that disease, a few datasets are related to credit loans, loan data and some datasets related automotive and wine quality.

Background and significance

Classification algorithms in machine learning are a subset of supervised learning techniques used to identify the category or class to which a new observation belongs, based on a training set of data containing observations whose category membership is known. These algorithms analyze the input data and use learned relationships to categorize new observations into predefined classes. They are used for their ability to simplify complex decision-making processes by categorizing data into distinct classes. This is crucial in many fields, such as medical diagnosis, where algorithms can help identify disease categories based on symptoms and tests, or in finance, where they can categorize transactions as fraudulent or legitimate. There are a wide range of classification models, each with its strengths and weaknesses. Common models include logistic regression, which is used for binary classification; decision trees, which are easy to interpret; random forests, an ensemble method that improves on the simplicity of decision trees; support vector machines, known for their effectiveness in high-dimensional spaces; and neural networks, which are particularly useful for complex, non-linear relationships.

In our study, we focus on benchmarking seven distinct classification methods across twenty diverse datasets. These methods are selected for their popularity and diverse algorithmic approaches. The datasets, sourced from public domains, encompass a wide range of sectors and complexities, ensuring a comprehensive evaluation platform. The primary objective of our benchmarking is to evaluate whether different classification methods perform differently on the same datasets. By doing so, we aim to shed light on the suitability of each method for various types of data. This involves assessing the performance of each algorithm in terms of metrics such as accuracy, precision, recall, F1 score, and area under the ROC curve (AUC).

Datasets

The table below shows all the 20 datasets we have taken for our benchmarking purposes from different public domains, mostly from Kaggle and University of California-Irvine (UCI) Machine Learning Repository and one from ISLR. The datasets cover diverse topics like Finance, Healthcare, Education, Social sciences etc. which vary in size, nature and complexity. The table below contains information of the names of datasets, N value which denotes the number of instances, P value which denotes the number of columns also predictor variables along with sources and links for the datasets.

Table:

Data set Names	Classification	No. Of instances(N)	No. Of col (P)	Source & Link
Loan data	Loan eligbility	614	11	Kaggle https://www.kaggle.com/datasets/burak3ergun/loan-data-set
Liver_patient	Prediction for liver disease	583	10	UCI https://archive.ics.uci.edu/dataset/225/ilpd+indian+liver+patient+dataset
stroke	Prediction for Brain disease	4932	10	Kaggle https://www.kaggle.com/datasets/shashwatwork/cerebral-stroke-predictionimbalaced-dataset
diabetes	Diabetes prediction	768	8	Kaggle https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database
Heart_attack	Prediction of heart stroke	302	13	Kaggle https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset
Breast_cancer	Prediction for Breast cancer wiscosin	569	32	UCI https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic
Wine_quality	Wine quality predictions	1600	12	UCI https://archive.ics.uci.edu/dataset/186/wine+quality
default	Credit card Yes or no/student	10000	3	ISLR https://rdrr.io/cran/ISLR/man/Default.html
diabetes2	Prediction for diabetes	520	16	Kaggle https://www.kaggle.com/datasets/andrewmvd/early-diabetes-classification

Heart_failure	Post-surgery heart failure prediction	299	13	UCI https://archive.ics.uci.edu/dataset/519/heart+failure+clinical+records
Student_data	Academic success and dropout prediction	4424	37	UCI https://archive.ics.uci.edu/dataset/697/predict+students+dropout+and+academic+success
Cell_samples	Prediction of classification of cell samples into 2 types	700	11	Github https://github.com/kvinlazy/Dataset/blob/master/cell_samples.csv?plain=1
New_model	Prediction for kidney disease	400	14	Kaggle https://www.kaggle.com/datasets/abhia1999/chronic-kidney-disease
Metabolic syndrome	Presence or absence of metabolic syndrome	2401	15	Kaggle https://www.kaggle.com/datasets/antimoni/metabolic-syndrome
gbsg	Breast cancer prediction yes/no	686	12	Kaggle https://www.kaggle.com/datasets/utkarshx27/breast-cancer-dataset-used-royston-and-altman
drug200	Classification of drugs into certain types	200	6	Github https://github.com/kvinlazy/Dataset/blob/master/drug200.csv
babies	Classification whether the baby's mom does smoke/not smoke	1237	8	Kaggle https://www.kaggle.com/datasets/debjeetdas/babies-birth-weight
Disease_syptom_and_patient	Prediction of diagnosis/assessment of specific disease	349	10	Kaggle https://www.kaggle.com/datasets/uom190346a/disease-symptoms-and-patient-profile-dataset

Rice_classification	Rice grain type prediction	18186	11	Kaggle https://www.kaggle.com/datasets/mssmartypants/rice-type-classification
heart	Prediction of heart disease	919	12	Kaggle https://www.kaggle.com/datasets/j ohnsmith88/heart-disease-dataset

Methodology

In this study, we benchmark seven classification models using R: Generative models such as LDA, QDA, Naïve bayes were selected for their ability to estimate class-specific data distributions and discriminative models like Logistic regression, support vector machines and decision trees, which are adept at creating decision boundaries to distinguish between different classes. The primary selection criteria were the model's effectiveness in handling linearly separable data. A brief description of the model’s functionality is listed below.

Generalized Linear Model (GLM) - Stats Package: glm()

GLM extends the traditional linear regression model by allowing for response variables that have error distribution models other than a normal distribution. It is particularly useful for modeling binary outcomes (as in logistic regression) or count data, among other types. By linking a function of the mean of the response variable to the predictors, GLM provides a flexible framework for analyzing diverse types of data.

Linear Discriminant Analysis (LDA) - MASS Package: lda()

LDA is a well-established method for dimensionality reduction and classification, which works by maximizing the separation between multiple classes through linear decision boundaries. This technique is particularly effective in scenarios where simplicity and computational efficiency are as important as predictive accuracy.

Quadratic Discriminant Analysis (QDA) - MASS Package: qda()

QDA extends the capabilities of Linear Discriminant Analysis by allowing for quadratic decision boundaries, making it more suitable for datasets where the class distribution is non-linear. This method enhances the model's adaptability to varying covariance structures among different classes, thus offering a more flexible approach in complex classification scenarios.

K-Nearest Neighbours (KNN) - Class Package: knn()

KNN is an instance-based learning method where the classification of a new observation is determined based on the majority class among its 'k' nearest neighbors in the feature space. This method is particularly valued for its simplicity and effectiveness, especially in cases where the data exhibits non-linear patterns.

Naive Bayes - e1071 Package:

Naive Bayes is a probabilistic classifier operates on the principle of Bayes' theorem, with the assumption that the predictors are independent of each other given the class. Despite its simplicity, Naive Bayes is known for its efficiency and effectiveness, especially in text classification and scenarios with high-dimensional data.

Support Vector Machine (SVM) - e1071 Package: svm()

VM is renowned for its effectiveness in both linear and non-linear classification, achieved through the use of kernel functions to transform the data into a higher-dimensional space. This method excels in handling complex and high-dimensional datasets, making it a robust tool for diverse classification challenges.

Random Forest - RandomForest Package: randomForest()

It is implemented for decision tree-based ensemble learning. It constructs multiple decision trees during training and outputs the class that is the mode of the classes of the individual trees. This method is well-regarded for its ability to handle large datasets with high dimensionality and its robustness against noise and overfitting

Model Parameters

Model	Training	Testing

Logistic Regression glm()	Time, Error rate	Accuracy, precision, ROC, AUC, Confusion matrix, Run time, F1 score
Linear discriminant analysis lda()	Time, Error rate	Accuracy, precision, ROC, AUC, Confusion matrix, Run time, F1 score
Quadrant discriminant analysis qda()	Time, Error rate	Accuracy, precision, ROC, AUC, Confusion matrix, Run time, F1 score
randomForest	Time, Error rate	Accuracy, precision, ROC, AUC, Confusion matrix, Run time, F1 score
Support vector machine svm()	Time, Error rate	Accuracy, precision, ROC, AUC, Confusion matrix, Run time, F1 score

Model	Testing
K nearest Neighbour	Accuracy, precision, ROC, AUC, Confusion matrix, Run time, F1 score
Naive Bayes	Accuracy, precision, ROC, AUC, Confusion matrix, Run time, F1 score

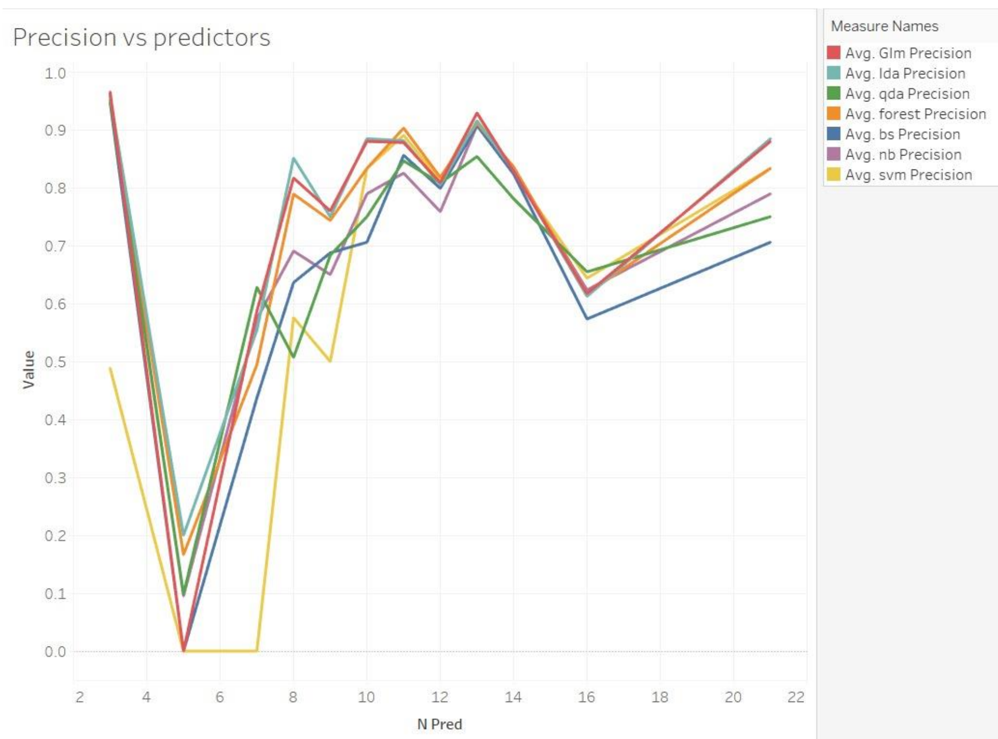
Data Cleaning

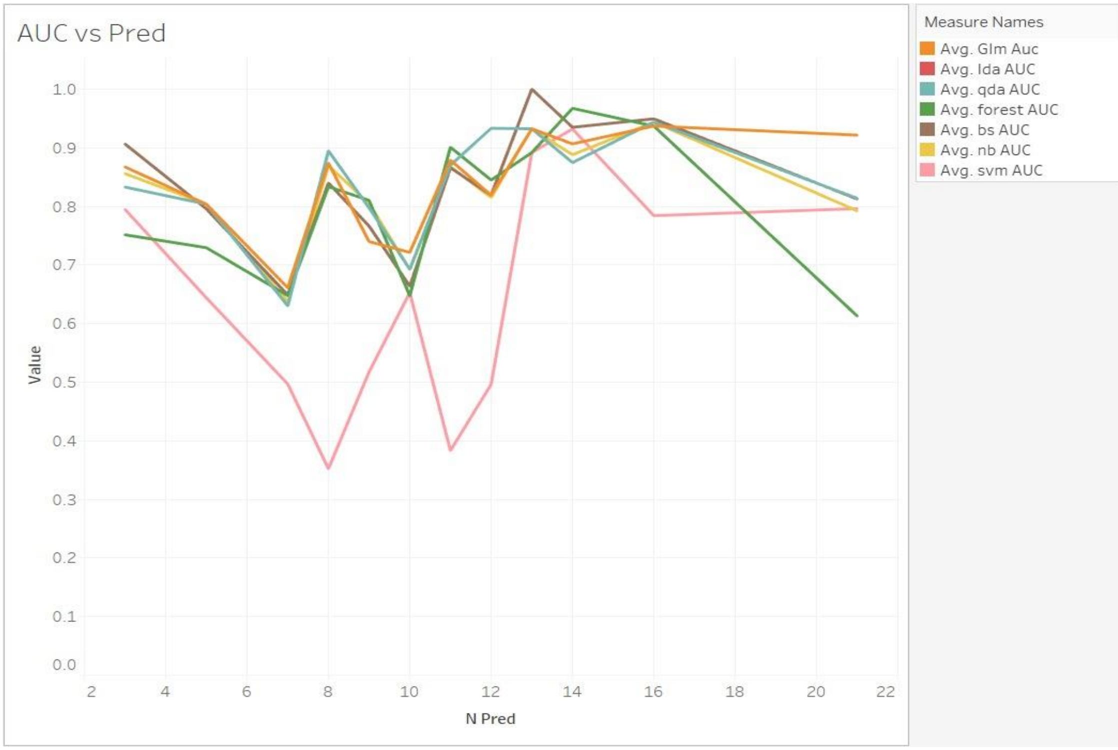
Our main focus is on converting target or label columns into a binary format (0 or 1), which simplifies the datasets for binary classification tasks. Additionally, we did identify and remove specific columns considered non-predictive or irrelevant to the analyses, ensuring a more focused dataset. The script also cleans the datasets by eliminating rows(non-predictive/irrelevant) with missing values in key columns, thereby enhancing data quality and consistency. Finally, each processed dataset is saved in a designated directory named Processed data, facilitating easy access and use for subsequent analysis or modelling tasks.

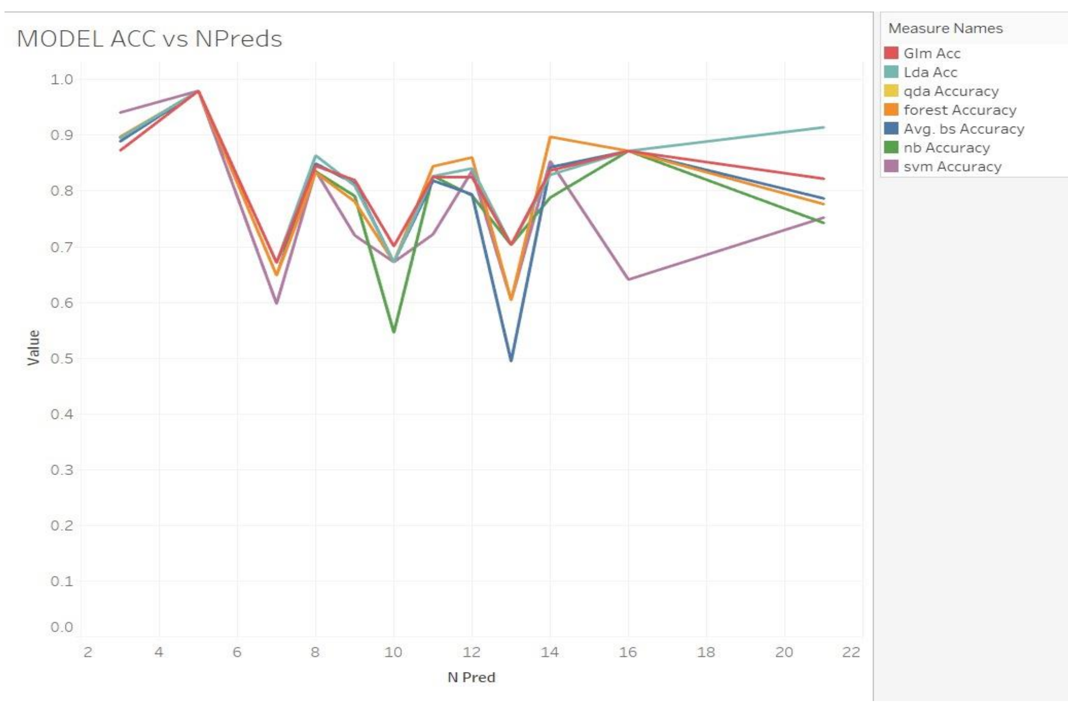
Here is the code snippet for the cleaning:

```
#new_model
new_model <- read.csv('new_model.csv')
new_model$Target <- ifelse(new_model$Target == 'enrolled' | new_model$Target == 'graduate', 1, 0)
write.csv(new_model, 'processeddata/new_model', row.names = FALSE)
rm('new_model')
```

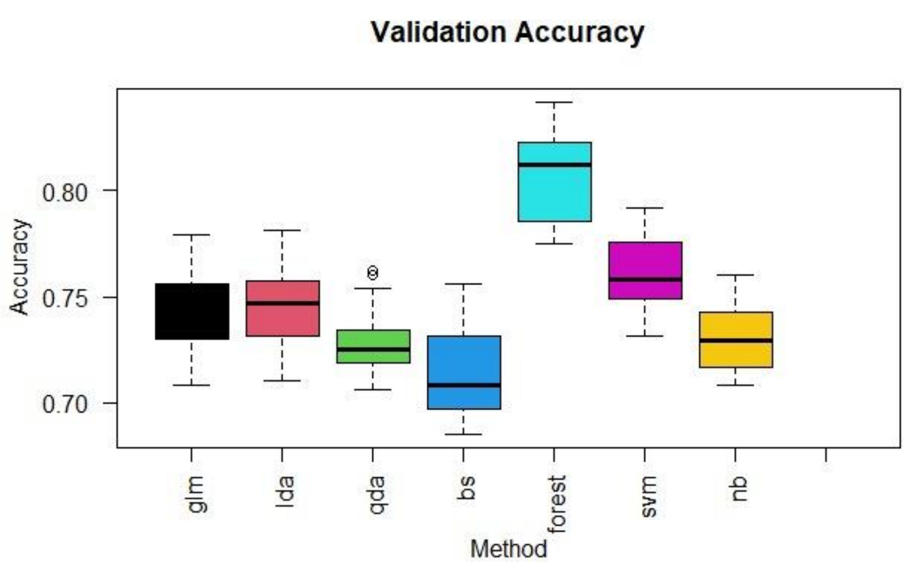
Data Analysis and Visualization:

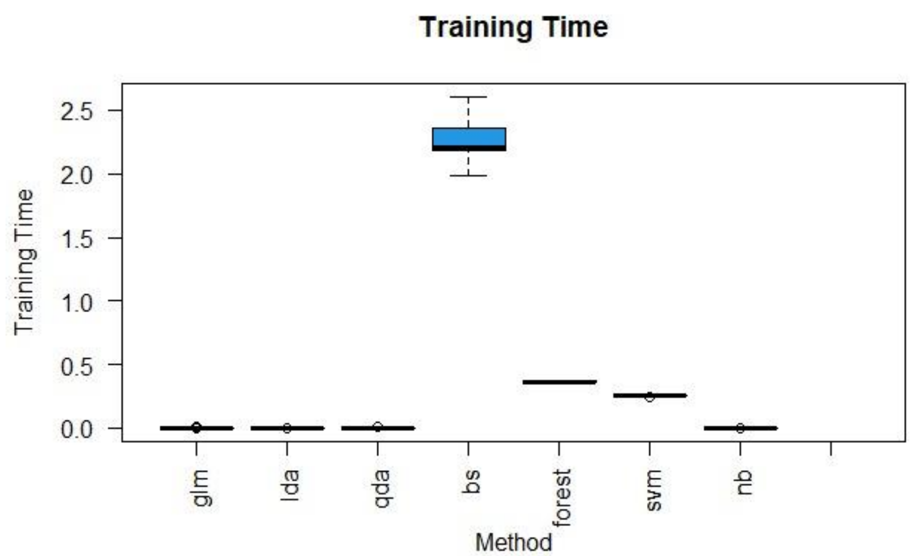


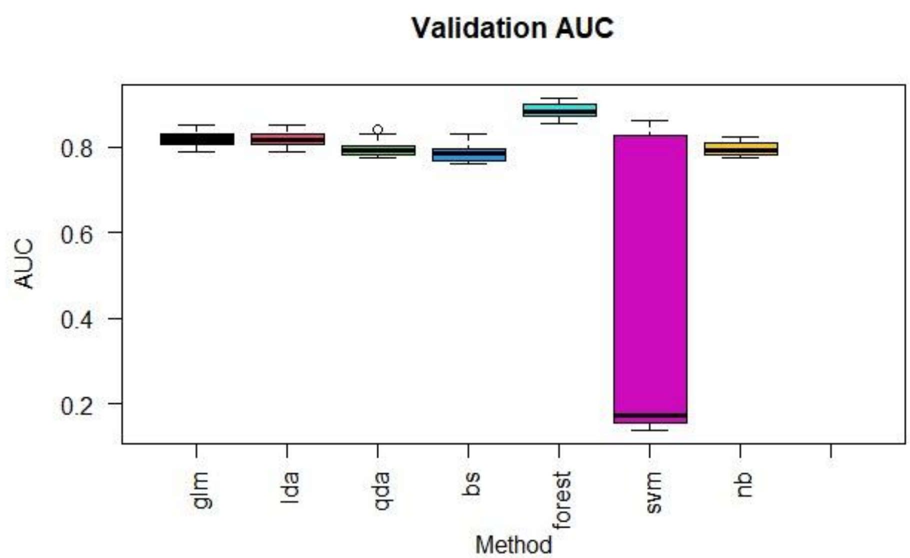

AUC vs Pred
Measure Names
Avg. Glm Auc
Avg. lda AUC
Avg. qda AUC
Avg. forest AUC
Avg. bs AUC
Avg. nb AUC
Avg. svm AUC
Value
N Pred
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
2
4
6
8
10
12
14
16
18
20
22


MODEL ACC vs NPreds
Measure Names
Glm Acc
Lda Acc
qda Accuracy
forest Accuracy
Avg. bs Accuracy
nb Accuracy
svm Accuracy
Value
N Pred
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
2
4
6
8
10
12
14
16
18
20
22


Title
Model error rate VS N pred
Columns
N Pred
Rows
Measure Values
Measure Names
Avg. Glm Errorrate
Avg. lda Error rate
Avg. qda Error rate
Avg. forest Error rate
Avg. bs Error rate
Avg. nb Error rate
Avg. svm Error rate
Value
N Pred


Validation Accuracy
Accuracy
Method
glm
lda
qda
bs
forest
svm
nb





Model Metrics

A	B	C	D	E	F	G	H	I
datasets	**Metrics**	**glm**	**lda**	**qda**	**bs**	**forest**	**svm**	**Naive bais**
babies	**accuracy**	0.671388102	0.6713881	0.671388102	0.648725213	0.648725213	0.597733711	0.671388102
	AUC	0.661037314	0.66103731	0.6301315	0.648855217	0.647720446	0.496795942	0.637540885
	Precision	0.586667	0.554054	0.628205	0.436975	0.494382	0	0.573171
	Error rate(%)	16.90018	16.71743	16.42625	13.90018	15.89425	15.38884	14.04648
	F-score(%)	42.08027	41.84243	40.84441	39.58027	40.3583	40.12015	40.04962
breast cancer	**accuracy**	0.942857143	0.92857143	0.928571429	0.942857143	0.648725213	0.952380952	0.947619048
	AUC	0.98571429	0.98571429	0.987755102	0.974897959	0.647720446	0.023367347	0.985408163
	Precision	0.755869	0.803279	0.476	0.378378	0.494382	0	0.733333
	Error rate(%)	31.52647	30.37101	29.71988	28.52647	15.89425	0	0.733333
	F-score(%)	38.4707	38.41185	38.10762	28.52647	40.3583	36.67196	36.43271
cell samples	**accuracy**	0.96933333	0.96866667	0.968666667	0.965666667	0.648725213	0.969	0.968
	AUC	0.94173925	0.94173925	0.944255782	0.89831888	0.647720446	0.867821092	0.941739247
	Precision	0.95384615	0.98333333	0.90277778	0.94117647	0.494382	0	0.92647059
	Error rate(%)	30.82869	30.36576	30.0441	27.82869	15.89425	28.91713	28.26472
	F-score(%)	39.40755	39.31787	39.25944	36.90755	40.3583	37.75106	37.23568
default	**accuracy**	0.770562771	0.766233766	0.766233766	0.735930736	0.648725213	0.714285714	0.744588745
	AUC	0.800246914	0.800246914	0.792510288	0.752427984	0.647720446	0.220987654	0.796460905
	Precision	0.694444	0.75	0.045224	0.530612	0.494382	0.727273	0.33871
	Error rate(%)	25.35128	25.1517	24.93027	22.35128	15.89425	24.50399	23.89006
	F-score(%)	31.0246	30.90593	30.81006	28.5246	40.3583	30.15417	29.14506
diabetes	**accuracy**	0.871794872	0.766233766	0.871794872	0.871794872	0.648725213	0.641025641	0.871794872
	AUC	0.93702814	0.80024691	0.943719973	0.949382292	0.647720446	0.784145505	0.943719973
	Precision	0.617283951	0.75	0.654761905	0.573333333	0.494382	0.643835616	0.623529412
	Error rate(%)	28.21788	25.1517	28.11774	25.21788	15.89425	26.72409	25.51673
	F-score(%)	28.64963	30.90593	27.73851	26.14963	40.3583	27.02682	26.39038

diabetes2	**accuracy**	0.87179487	0.87179487	0.72815534	0.689320388	0.648725213	0.747572816	0.762135922
	AUC	0.93702814	0.93702814	0.816765873	0.800992064	0.647720446	0.8125	0.79781746
	Precision	0.61728395	0.6125	0.916667	0.967742	0.494382	0.980392	0.925234
	Error rate(%)	28.21788	28.16429	33.36506	30.4037	15.89425	31.80823	30.44432
	F-score(%)	28.64963	28.18725	27.95463	26.89748	40.3583	27.79784	27.77191
diseaase symptom and patient profile	**accuracy**	0.72330097	0.72815534	0.851449275	0.862318841	0.648725213	0.829710145	0.81884058
	AUC	0.79325397	0.79325397	0.878486268	0.886576538	0.647720446	0.114913775	0.881147541
	Precision	0.960396	0.957447	0.75531915	0.71764706	0.494382	0.78481013	0.62857143
	Error rate(%)	33.4037	33.36573	24.74007	21.9757	15.89425	23.16419	22.71356
	F-score(%)	29.39748	28.83186	35.16969	33.85906	40.3583	34.48309	33.95685
gbsg	**accuracy**	0.703296703	0.703296703	0.703296703	0.494505495	0.648725213	0.604395604	0.703296703
	AUC	0.93236715	0.93236715	0.93236715	1	0.647720446	0.89178744	0.93236715
	Precision	0.929032	0.915584	0.853933	0.907407	0.494382	0.928571	0.907407
	Error rate(%)	25.11034	25.05382	24.18534	22.11034	15.89425	22.47507	22.11352
	F-score(%)	22.6928	22.48571	22.44394	22	40.3583	22.38204	22.01053
heart	**accuracy**	0.833333333	0.844444444	0.844444444	0.755555556	0.648725213	0.9	0.822222222
	AUC	0.745989305	0.745989305	0.901069519	0.865641711	0.647720446	0.15040107	0.818181818
	Precision	0.84313726	0.82	0.86046512	0.88	0.494382	0.84	0.88888889
	Error rate(%)	37.45304	37.36505	35.60775	34.45304	15.89425	35.02439	34.60299
	F-score(%)	31.58473	31.02206	31.00353	29.08473	40.3583	29.97248	29.21467
heart attack	**accuracy**	0.701149425	0.672413793	0.672413793	0.672413793	0.648725213	0.672413793	0.545977012
	AUC	0.721397511	0.721397511	0.692757535	0.663817664	0.647720446	0.652871495	0.692307692
	Precision	0.88	0.884615	0.75	0.705882	0.494382	0.833333	0.789474
	Error rate(%)	32.16712	0.884615	30.95071	29.16712	15.89425	30.00017	29.22308
	F-score(%)	37.03253	36.70704	36.6802	34.53253	40.3583	35.35671	34.66247

heart failure	accuracy	0.826388889	0.826388889	0.826388889	0.763888889	0.648725213	0.708333333	0.819444444
	AUC	0.746732026	0.746732026	0.778244631	0.720588235	0.647720446	0.595938375	0.787348273
	Precision	0.52	0.5	0.366667	0.375	0.494382	0	0.383333
	Error rate(%)	33.44708	33.02564	31.85121	30.44708	15.89425	31.15327	30.84438
	F-score(%)	32.31941	32.05399	31.58486	29.81941	40.3583	30.175	30.12477
liver patient	accuracy	0.836611195	0.829046899	0.829046899	0.842662632	0.648725213	0.853252648	0.788199697
	AUC	0.906602737	0.906602737	0.874687843	0.934891619	0.647720446	0.932104685	0.888083109
	Precision	0.828571	0.829787	0.780822	0.823529	0.494382	0.823944	0.82963
	Error rate(%)	36.96793	36.96078	36.19716	33.96793	15.89425	35.15558	34.91803
	F-score(%)	33.65621	33.17562	32.86879	31.15621	40.3583	31.29071	31.25909
loan_data	accuracy	1	0.999083578	0.999083578	0.986803519	0.648725213	0.569648094	0.999633431
	AUC	1	1	1	0.998689737	0.647720446	0.420114273	1
	Precision	1	1	1	1	0.494382	1	0.986872
	Error rate(%)	26.94409	26.67916	26.62018	23.94409	15.89425	24.53163	24.20974
	F-score(%)	22.0721	22.06825	22.06218	22	40.3583	22.03312	22.02728
metabolic syndrome	accuracy	0.979706489	0.979706489	0.979706489	0.979706489	0.648725213	0.979706489	0.979706489
	AUC	0.803658928	0.803658928	0.803658928	0.795572129	0.647720446	0.643837709	0.803658928
	Precision	0	0.2	0.098326	0	0.494382	0	0.095238
	Error rate(%)	19.29598	18.89829	17.47413	16.29598	15.89425	16.86464	16.53975
	F-score(%)	33.08324	33.05868	32.81459	30.58324	40.3583	32.05749	31.04949
new model	accuracy	0.720833333	0.727083333	0.727083333	0.735416667	0.648725213	0.741666667	0.727083333
	AUC	0.812905024	0.812905024	0.785520173	0.777001603	0.647720446	0.185300676	0.791025016
	Precision	0.777778	0.776423	0.713376	0.738516	0.494382	0.796748	0.76
	Error rate(%)	31.06872	30.9841	29.90313	28.06872	15.89425	28.49535	28.4437
	F-score(%)	31.72806	30.96912	30.37432	29.22806	40.3583	30.29405	29.74368

rice classification	accuracy	0.812355448	0.792542122	0.792542122	0.862413366	0.648725213	0.732166696	0.762541485
	AUC	0.732894215	0.732894215	0.81658843	0.811965845	0.647720446	0.438964575	0.822545854
	Precision	1	1	1	1	0.494382	1	0.916667
	Error rate(%)	31.56263	31.42599	29.75751	28.56263	15.89425	29.12439	28.90383
	F-score(%)	35.79805	35.67945	35.55336	33.29805	40.3583	34.58777	33.34159
stroke	accuracy	0.775962696	0.821365874	0.821365874	0.812478997	0.648725213	0.912478585	0.824789654
	AUC	0.792888565	0.792888565	0.72157556	0.914475633	0.647720446	0.721588896	0.769952365
	Precision	0.976744	0.931818	1	0.953488	0.494382	0.976744	1
	Error rate(%)	18.91379	18.5508	18.35635	15.91379	15.89425	17.82342	16.31046
	F-score(%)	22.46158	22.37775	22.32575	22	40.3583	22.08003	22.01477
student	accuracy	0.821969824	0.896354124	0.896354124	0.868773955	0.648725213	0.842188966	0.812548576
	AUC	0.835458236	0.835458236	0.903666902	0.792228456	0.647720446	0.812395676	0.826669426
	Precision	1	1	1	1	0.494382	1	1
	Error rate(%)	33.21841	32.77922	32.17209	30.21841	15.89425	31.72565	30.5321
	F-score(%)	22.11104	22.08068	22.07276	22	40.3583	22.05692	22.05103
wine quality	accuracy	0.821964567	0.914258646	0.914258646	0.786589421	0.648725213	0.752369889	0.742589642
	AUC	0.921549954	0.921549954	0.813698774	0.812444963	0.647720446	0.796134985	0.792271966
	Precision	0.88	0.884615	0.75	0.705882	0.494382	0.83333	0.789474
	Error rate(%)	25.10096	24.98453	24.26514	22.10096	15.89425	23.27163	22.30986
	F-score(%)	31.27422	31.15327	30.41303	28.77422	40.3583	29.92438	28.96723
drug 200	accuracy	0.816579857	0.836511976	0.836511976	0.832191676	0.831498731	0.771546585	0.761478895
	AUC	0.892132533	0.892132533	0.965412885	0.77254137	0.783396524	0.842127396	0.8122249
	Precision	0.773809524	0.79069767	0.75531915	0.717647059	0.8125	0.784810127	0.628571429
	Error rate(%)	33.98386	33.11406	32.86652	30.98386	31.78689	31.70438	31.70349
	F-score(%)	32.62608	32.56694	32.54324	30.12608	30.91335	30.57962	30.46333

Benchmarked Metrics

MODEL	GLM	LDA	QDA	RF	BS	NB	SVM
AVG ACC	0.822	0.831	0.83	0.82	0.8	0.803	0.77

MODEL	GLM	LDA	QDA	RF	BS	NB	SVM
AVG AUC	0.84	0.81	0.83	0.81	0.83	0.83	0.57

MODEL	GLM	LDA	QDA	RF	BS	NB	SVM
AVG PRESICIO	0.787	0.799	0.715	0.785	0.71	0.73	0.64

MODEL	GLM	LDA	QDA	RF	BS	NB	SVM
AVG ER	28.82%	28.50%	27.87%	25.80%	25.82%	24.81	25.39

Evaluating Differences and Model Metrics: The evaluation process involves splitting each dataset into a training set (70%) and a test set (30%). Each classification method is applied to these datasets, and its performance is measured using the aforementioned metrics. Furthermore, we implement method-specific parameter tuning through cross-validation on the training set to optimize each model's performance. This dual evaluation—before and after optimization—allows us to comprehensively assess the out-of-the-box efficiency of the algorithms as well as their potential when fine-tuned.

Significance of Our Study: Our study is significant as it provides empirical evidence on the performance of various classification methods across multiple datasets, helping practitioners make informed decisions about algorithm selection based on specific data characteristics. This benchmarking exercise not only contributes to the theoretical understanding of these methods but also offers practical insights into their real-world applicability.

Conclusion

The line graph depicts a comparison of various predictive models' accuracies across a series of predictions. One model demonstrates the highest accuracy, peaking above the others, while another model shows the lowest, dipping below the rest at certain points. Overall trends suggest that some models' accuracies fluctuate significantly with the number of predictions, indicating variability in performance. The specific model names and their corresponding accuracy levels at different prediction counts are color-coded for easy identification.

References

1. James, Witten, Hastie and Tibshirani, An Introduction to Statistical Learning with Applications in R (ISLR).
2. Provost and Fawcett, Data Science for Business (DSFB) - What You Need to Know about Data Mining and Data-Analytic Thinking, O'REILLY, 2013.
3. https://www.rdocumentation.org/
4. The Comprehensive R Archeive network: https://cran.r-project.org/
5. https://www.projectpro.io/article/7-types-of-classification-algorithms-in-machine-learning/435
6. https://www.simplilearn.com/tutorials/machine-learning-tutorial/classification-in-machine-learning
7. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8306704/