Journal of Statistics & Management Systems
ISSN 0972-0510 (Print), ISSN 2169-0014 (Online)
Vol. 29 (2026), No. 5, pp. 553–572
DOI : 10.47974/JSMS-1607

Comparison of machine learning classification algorithms under different data distributions

Ceyda Murat *
Esra Gökpınar †
Department of Statistics
Gazi University
Ankara 06500
Turkey

Abstract

In this study, we compare the performance of various machine learning algorithms through simulations conducted on datasets with different distributional characteristics. Widely used models such as Logistic Regression, Naive Bayes, k-Nearest Neighbors, Decision Trees, Support Vector Machines, Random Forest, AdaBoost, Gradient Boosting, and XGBoost are examined. The results indicate that the performance of classification algorithms is strongly influenced by the underlying data distribution and class balance. Specifically, Logistic Regression, Naive Bayes, and Support Vector Machines achieved high accuracy on normally distributed datasets, whereas tree-based methods such as Random Forest, Gradient Boosting, and XGBoost demonstrated greater robustness and superior outcomes on non-normal distributions and imbalanced class structures.

Subject Classification: *62H30, 68T01.*

Keywords: *Machine learning, Classification algorithms, Simulation study, Data distribution.*

1. Introduction

Machine learning, as an essential field of artificial intelligence, empowers computers to improve their decision-making capabilities by drawing patterns and insights from data. This field uses concepts such as data analysis, pattern recognition, and predictive modelling to create

* *ORCID-ID:* **0009-0006-1600-0285**
† *ORCID-ID:* **0000-0003-2148-4940**

* *E-mail:* **ceyda.murat.44@gmail.com (Corresponding Author)**
† *E-mail:* **eyigit@gazi.edu.tr**

data-driven models. Machine learning is widely used in data-driven decision-making and automation processes in many industries, including healthcare, finance, automotive, and telecommunications.

Among the most widely studied tasks in machine learning is classification, where the aim is to assign observations into predefined categories. In this problem, the goal is to predict a categorical target variable; models analyze patterns in the training data to determine which category each observation belongs to [1]. Among the algorithms commonly used in classification problems are Logistic Regression (LR), Decision Trees (DT), Support Vector Machines (SVM), k-Nearest Neighbour (kNN), Naive Bayes (NB), Artificial Neural Networks (ANN), and ensemble methods. Each of these algorithms has different assumptions, computational structures, and parameter sensitivities. Therefore, depending on the characteristics of the data set, one model may be more advantageous or disadvantageous than others.

Many comparative studies in the literature have focused on real-world datasets and mainly reported performance based on accuracy or similar measures. However, studies that systematically compare the overall performance of classification algorithms are quite limited. For example, Fernández-Delgado et al. [2] compared 179 algorithms across 121 datasets in the UCI machine learning database and found that Random Forest (RF), SVM, and artificial neural networks stood out. Similarly, Charoenpong et al. [3] also evaluated the advantages and disadvantages of LR, DT, SVM, NB, artificial neural networks, and random forest algorithms. However, a common problem in the literature is that such comparisons are often made with a limited number of datasets, and the results are significantly influenced by the characteristics of the dataset. Macià and Bernadó-Mansilla [4] emphasized that comparisons made with a small number of datasets can be misleading in terms of metrics such as average accuracy; while some algorithms may perform well in a narrow scope, their success may decrease as the diversity of the dataset increases.

In this context, simulation-based approaches offer a powerful alternative for evaluating the general performance of algorithms. Through simulations, data structural characteristics such as sample size, class distribution, and inter-variable correlation can be controlled, enabling a more systematic examination of the strengths and weaknesses of models within a cause-and-effect framework. However, there are very few studies in the literature on simulation-based classification analyses. For example, Ağlarcı and Bal [5] compared various machine learning algorithms through simulation for a response variable measured at the ranking level

and showed that the success levels of k-nearest neighbour and artificial neural networks, in particular, vary depending on the correlation and number of categories. Li and Guo [6] examined their proposed Random Logistic Machine model within the scope of a two-category classification problem by comparing it with some models through simulation. However, to the best of our knowledge, we could not find a comprehensive simulation study focusing on two-class problems and covering models developed for this problem. For this purpose, simulations based on scenarios representing different data structures were designed to evaluate the classification performance of machine learning algorithms, and nine different models (LR, NB, kNN, DT, SVM, RF, Ada, GB, and XGB) were systematically compared. The simulation datasets included both structures with multivariate normal distributions and those derived from non-normal distributions. Thus, the performance of the models under different conditions was evaluated, and it was revealed which data structures each algorithm was more advantageous for. In this respect, this study aims to fill an important gap in the literature; at the same time, it provides an applied guide for researchers on which algorithms to try.

The structure of this study is organized as follows. In Section 2, widely used classification models are briefly introduced. Section 3 presents a simulation study in which the performance of these algorithms is compared on artificially generated datasets derived from both normal and non-normal distributions. The simulation results are evaluated using various performance metrics. Finally, Section 4 provides general conclusions and discussions.

2. Material And Methods

This section briefly introduces the LR, NB Classifier, kNN, DT, SVM, RF, Ada, GB, and XGB algorithms.

2.1 *Logistic Regression (LR)*

LR is a version of linear regression adapted to classification problems. When the dependent variable is binary, the sigmoid function is used to estimate the probability of the data. The model makes a linear distinction between classes.

2.2 *Naive Bayes (NB)*

Based on Bayes' theorem, NB predicts the class of an observation by computing conditional probabilities from training data. This table is based on feature values that must be considered when estimating a new observation [7]. In other words, using information learned from previously known data, an attempt is made to predict which category a new, unknown data point belongs to.

2.3 *K-Nearest Neighbors Algorithm (kNN)*

kNN operates by examining the surrounding points in the training data when a new instance is introduced and assigns its label according to the majority class of the nearest neighbors. It builds on the principle that data points positioned close to each other are likely to belong to the same category.

Then, the k nearest neighbours to the new data point are determined based on these distance measurements. Various calculation methods are used to determine the distance between examples. Euclidean distance, Manhattan distance, Chebyshev distance, and Minkowski distance are among the calculation methods used [8].

2.4 *Decision Trees (DT)*

Decision Trees (DT) are tree-based learning methods that partition a dataset into progressively smaller subsets by applying a sequence of decision rules. This recursive process continues until each subset contains instances with the same class label. Splits occur at internal nodes, while decisions are made at the leaf nodes. Among entropy-based approaches, ID3 and C4.5 are prominent. In classification and regression trees (CART), criteria such as the twoing method, the Gini index, and regression trees are commonly used.

2.5 *Support Vector Machines (SVM)*

Support Vector Machines identify the hyperplane that maximizes class separation, and kernel functions enable them to model complex nonlinear relationships. It handles high-dimensional data well and generalizes effectively. That said, training is slow, and performance can fluctuate based on the chosen hyperparameters [9].

2.6 *Random Forests (RF)*

Random Forests aggregate predictions from multiple decision trees trained on random subsets of the data, producing stable and less overfit models [10]. In this approach, every decision tree is built using a different randomly selected portion of the training set, and the final prediction is obtained by taking the class that receives the majority of votes across all trees.

2.7 *AdaBoost (Ada)*

The Ada algorithm is one of the powerful and widely used ensemble methods among boosting ensemble learning methods. In the Ada method, instead of training subsets of the training data set, different weights of the training data set are used [11]. The basis of this method is to improve itself at each step, focusing more on the instances where the model predicts incorrectly.

2.8 *Gradient Boosting Machine (GBM)*

Gradient Boosting Machine (GBM), introduced by Friedman [12], is an ensemble method designed to optimize the loss function by iteratively training weak decision tree classifiers. Its main principle is to combine many simple models, where each new one corrects the errors of the previous, producing a more accurate and reliable predictor. To prevent overfitting issues, the GBM uses a learning rate to scale the contribution from the new tree [13].

2.9 *XGBoost (XGB)*

XGB is an enhanced and optimized version of the GBM algorithm. It is specifically designed to improve performance. As a tree-based method, XGB can be effectively applied to both regression and classification problems. Its main objective is to build a strong predictive model by combining a series of weak learners.

3. Simulation Study

In this study, various classification algorithms were compared on artificial data sets obtained through simulation from normal and non-normal distributions using the Python programming language. The outcomes of the simulations were examined in a comparative manner,

with evaluation based on several performance indicators, including accuracy, precision, sensitivity, and the F1-score.

In the simulation study, the sample size was set to 1000. To create the classification model, the dependent variable y was taken as a two-class categorical variable, and the independent variable X was taken as any data type. Accordingly, in scenarios 1-3, the independent variable X was treated as continuous, while in scenarios 4-6, the independent variable X was treated as both continuous and categorical. When creating the dependent variable, data was generated from a Bernoulli distribution with probability values of p = 0.5, 0.3, and 0.2. When p = 0.5, the positive and negative class states were kept balanced, while with p = 0.2 and p = 0.3, the positive class ratios were kept at approximately 20% and 30%, respectively, to create an unbalanced data set. When creating the independent variable, data was generated under a specific distribution. In addition to the generated independent variables, noise variables were created for each variable to include measurement errors in the model and were added to the main variables. To obtain the categorical independent variables in scenarios 4-6, after initially obtaining continuous independent variables, these variables were converted to categorical independent variables based on their quartile values.

In the first stage, a simulation study was conducted for variables generated from a multivariate normal distribution, followed by a simulation study on variables generated from non-normal distributions. Each scenario was repeated 1000 times, and the average accuracy, precision, sensitivity, and F1-Score values of the classification algorithms were calculated.

3.1 *Simulation Study for Variables Generated from a Multivariate Normal Distribution*

In this part of the study, five independent variables were generated based on a multivariate normal distribution. The corresponding variance-covariance matrix was then constructed as shown below.

$$\sum = \begin{bmatrix} 1.0 & 0.2 & 0.6 & 0.6 & 0.4 \\ 0.2 & 1.0 & 0.6 & 0.6 & 0.6 \\ 0.6 & 0.6 & 1.0 & 0.6 & 0.6 \\ 0.6 & 0.6 & 0.6 & 1.0 & 0.2 \\ 0.4 & 0.6 & 0.6 & 0.2 & 1.0 \end{bmatrix}$$

Additional noise variables were produced from a multivariate normal distribution with zero mean and a unit covariance matrix, and these were incorporated into the main variables. For scenarios 4-6 created for 5 independent variables, the third quartile value ($\mathbb{Q}_3$) was calculated for the X_1 variable, and observations greater than this value were assigned 1, while others were assigned 0. For the X_2 variable, the median ($\mathbb{Q}_2$) was calculated, and observations greater than the median were assigned a value of 1, while the others were assigned a value of 0. These cases are presented in Table 1.

Table 1

Variable settings and simulation configurations

Cases	X_1, X_2	X_3, X_4, X_5	p
Case 1	Real-valued	Real-valued	0.5
Case 2	Real-valued	Real-valued	0.3
Case 3	Real-valued	Real-valued	0.2
Case 4	Binary	Real-valued	0.5
Case 5	Binary	Real-valued	0.3
Case 6	Binary	Real-valued	0.2

Table 2

Performances of classification algorithms for a multivariate normal distribution under Case1-Case3

Cases	Models	F1- Score	Precision	Recall	Accuracy
Case 1	LR	0.8663	0.8678	0.8661	0.8663
	NB	0.8672	0.8683	0.8673	0.8672
	kNN	0.8471	0.8483	0.8474	0.8471
	DT	0.8266	0.8323	0.8236	0.8275
	SVM	0.8646	0.8667	0.8640	0.8647
	RF	0.8544	0.8564	0.8539	0.8545
	Ada	0.8500	0.8520	0.8496	0.8502
	GB	0.8564	0.8580	0.8562	0.8565
	XGB	0.8505	0.8499	0.8527	0.8502

Contd...

Case 2	LR	0.7985	0.8299	0.7726	0.8846
	NB	0.8033	0.7867	0.8238	0.8805
	kNN	0.7667	0.7978	0.7417	0.8664
	DT	0.7350	0.7674	0.7111	0.8485
	SVM	0.7890	0.8392	0.7482	0.8816
	RF	0.7773	0.8119	0.7491	0.8730
	Ada	0.7705	0.7927	0.7529	0.8673
	GB	0.7778	0.8113	0.7502	0.8731
	XGB	0.7715	0.7955	0.7528	0.8681
Case 3	LR	0.7426	0.8081	0.6924	0.9055
	NB	0.7519	0.7222	0.7896	0.8971
	kNN	0.7041	0.7698	0.6551	0.8916
	DT	0.6625	0.7210	0.6226	0.8757
	SVM	0.7265	0.8267	0.6543	0.9032
	RF	0.7156	0.7865	0.6624	0.8964
	Ada	0.7093	0.7564	0.6745	0.8912
	GB	0.7164	0.7837	0.6654	0.8964
	XGB	0.7088	0.7604	0.6701	0.8917

Table 3

Simulation results for variables generated from a multivariate normal distribution for Case4-Case6

Cases	Models	F1- Score	Precision	Recall	Accuracy
Case 4	LR	0.8576	0.8612	0.8555	0.8585
	NB	0.8463	0.8591	0.8353	0.8490
	kNN	0.8322	0.8398	0.8264	0.8340
	DT	0.8182	0.8265	0.8133	0.8202
	SVM	0.8548	0.8583	0.8530	0.8557
	RF	0.8414	0.8435	0.8411	0.8421
	Ada	0.8436	0.8463	0.8427	0.8444
	GB	0.8466	0.8480	0.8469	0.8472
	XGB	0.8378	0.8395	0.8379	0.8385

Contd...

Case 5	LR	0.7839	0.8213	0.7530	0.8761
	NB	0.7857	0.7652	0.8107	0.8679
	kNN	0.7516	0.7802	0.7288	0.8562
	DT	0.7261	0.7629	0.6987	0.8431
	SVM	0.7747	0.8306	0.7300	0.8735
	RF	0.7621	0.7967	0.7343	0.8632
	Ada	0.7633	0.7934	0.7395	0.8632
	GB	0.7672	0.8034	0.7379	0.8664
	XGB	0.7560	0.7841	0.7339	0.8587
Case 6	LR	0.7182	0.7899	0.6642	0.8974
	NB	0.7273	0.6869	0.7783	0.8849
	kNN	0.6843	0.7351	0.6462	0.8827
	DT	0.6436	0.7083	0.6003	0.8698
	SVM	0.6957	0.8125	0.6151	0.8944
	RF	0.6902	0.7609	0.6378	0.8874
	Ada	0.6947	0.7467	0.6560	0.8866
	GB	0.6959	0.7640	0.6451	0.8891
	XGB	0.6849	0.7401	0.6440	0.8834

According to the simulation data in Table 2-Table 3, when classes are balanced (p=0.5), considering the accuracy criterion in the performance comparison of models, it has been observed that the performance of NB, LR, and SVM models is higher than that of other models. In scenarios where class imbalance increases, i.e., when the p value decreases, criteria such as precision, sensitivity, and F1 score come to the fore instead of the accuracy criterion. In this case, when the performance of the models is compared according to these criteria, a significant decrease is observed compared to the accuracy criterion. When sensitivity and F1 score are considered, the NB model stands out, while when precision is considered, the Support Vector Machines model stands out.

3.2 *Simulation Study for Variables Generated from Log-Normal Distribution*

In this section, independent variables were generated from a multivariate normal distribution and then their natural logarithms were taken to obtain values from a log-normal distribution. Thus, random variables with a log-normal distribution and a specific dependency structure were generated.

The mean values for the independent variables were set to 3 for y=0 and 5 for y=1. The noise variables were generated from a log-normal distribution with a mean of 0 and a standard deviation of 1.

Table 4

Simulation results for variables generated from log-normal distribution for Case1-Case3

Cases	Models	F1- Score	Precision	Recall	Accuracy
Case 1	LR	0.8920	0.9205	0.8663	0.8960
	NB	0.8489	0.9421	0.7742	0.8635
	kNN	0.8876	0.8890	0.8873	0.8884
	DT	0.8781	0.8816	0.8763	0.8792
	SVM	0.8942	0.9173	0.8732	0.8974
	RF	0.8971	0.8988	0.8964	0.8978
	Ada	0.8928	0.8930	0.8938	0.8934
	GB	0.8984	0.8985	0.8992	0.8990
	XGB	0.8931	0.8923	0.8951	0.8937
Case 2	LR	0.8356	0.8961	0.7853	0.9080
	NB	0.8168	0.8865	0.7603	0.8985
	kNN	0.8303	0.8537	0.8110	0.9013
	DT	0.8154	0.8349	0.8011	0.8921
	SVM	0.8393	0.8887	0.7976	0.9090
	RF	0.8446	0.8654	0.8273	0.9093
	Ada	0.8371	0.8528	0.8252	0.9044
	GB	0.8467	0.8629	0.8336	0.9101
	XGB	0.8389	0.8535	0.8275	0.9054
Case 3	LR	0.7890	0.8726	0.7250	0.9241
	NB	0.7842	0.8228	0.7543	0.9186
	kNN	0.7827	0.8277	0.7476	0.9186
	DT	0.7614	0.7964	0.7368	0.9094
	SVM	0.7905	0.8716	0.7287	0.9244
	RF	0.8020	0.8442	0.7689	0.9256
	Ada	0.7917	0.8220	0.7693	0.9207
	GB	0.8010	0.8334	0.7761	0.9244
	XGB	0.7938	0.8249	0.7700	0.9215

Table 5

Simulation results for variables generated from log-normal distribution for Case 4-Case 6

Cases	Models	F1- Score	Precision	Recall	Accuracy
Case 4	LR	0.8871	0.9143	0.8626	0.8908
	NB	0.8519	0.9335	0.7850	0.8644
	kNN	0.8815	0.8814	0.8829	0.8819
	DT	0.8756	0.8809	0.8723	0.8766
	SVM	0.8851	0.9122	0.8607	0.8888
	RF	0.8889	0.8914	0.8878	0.8896
	Ada	0.8886	0.8898	0.8891	0.8890
	GB	0.8933	0.8943	0.8936	0.8938
	XGB	0.8864	0.8872	0.8868	0.8869
Case 5	LR	0.8300	0.8854	0.7839	0.9047
	NB	0.8240	0.8613	0.7925	0.8994
	kNN	0.8189	0.8359	0.8054	0.8941
	DT	0.8092	0.8333	0.7907	0.8892
	SVM	0.8248	0.8826	0.7770	0.9020
	RF	0.8333	0.8552	0.8154	0.9031
	Ada	0.8312	0.8505	0.8161	0.9015
	GB	0.8390	0.8571	0.8245	0.9060
	XGB	0.8293	0.8457	0.8166	0.9002
Case 6	LR	0.7843	0.8605	0.7252	0.9228
	NB	0.7868	0.7952	0.7840	0.9175
	kNN	0.7680	0.8061	0.7387	0.9135
	DT	0.7552	0.7923	0.7282	0.9087
	SVM	0.7710	0.8684	0.6986	0.9198
	RF	0.7869	0.8280	0.7548	0.9208
	Ada	0.7838	0.8153	0.7609	0.9188
	GB	0.7914	0.8245	0.7658	0.9217
	XGB	0.7819	0.8118	0.7594	0.9179

When examining the simulation results in Table 4 and Table 5, it is observed that in cases where classes are balanced (p=0.5), the most successful models are tree-based methods (RF, Ada, GB, XGB) and LR. GB and RF models have demonstrated high performance in all performance metrics. In scenarios with increased class imbalance (p=0.3 and p=0.2), tree-based methods continued to perform well in terms of both sensitivity and F1-score. However, in these scenarios, a significant decline in the performance of the LR model was observed, particularly in terms of sensitivity and F1-score values. In terms of accuracy, the LR and SVM models generally yielded successful results. In all scenarios, the kNN, NB, and DT models performed worse than other methods. This may be due to the independence assumption not being fully compatible with the multivariate log-normal distribution structure, particularly in the NB model.

3.3 *Simulation Study for Variables Generated from Multivariate t-Distribution*

In this section, the independent variables are generated from a multivariate t-distribution. In the simulation study, the degree of freedom (ν) is set to 3. The μ parameter, i.e., the mean value, is set to 3 for y=0 and 5 for y=1 for the independent variables. To better reflect real-world variability in the simulation, noise variables were also included in the model in addition to the independent variables. Data for the noise variables were generated from a multivariate t-distribution with a degree of freedom (ν) of 3. For scenarios 4-6, while obtaining categorical variables from the independent variables obtained from the multivariate t-distribution, the third quartile value ($\mathbb{Q}_3$) was calculated for the X_1 variable, and observations greater than this value were assigned 1, while others were assigned 0. For the X_2 variable, the median ($\mathbb{Q}_2$) was calculated, and observations greater than the median were assigned a value of 1, while the others were assigned a value of 0.

According to the simulation data in Table 6 and Table 7, since the t distribution has a distribution structure containing extreme values, a general decline in model performance has been observed. In cases where classes are balanced (p=0.5), it has been observed that LR, NB, and SVM achieve higher accuracy values compared to other models. In scenarios where class imbalance increased (p=0.3 and p=0.2), although accuracy values remained high, significant decreases were observed in sensitivity and F1-score values for many models, particularly LR. In these scenarios, the NB model stood out with higher sensitivity and F1-score values

Table 6

Simulation results for variables generated from a multivariate t-distribution for Case1-Case3

Cases	Models	F1- Score	Precision	Recall	Accuracy
Case 1	LR	0.8076	0.8069	0.8102	0.8079
	NB	0.8018	0.8015	0.8066	0.8022
	kNN	0.7865	0.7861	0.7890	0.7870
	DT	0.7615	0.7635	0.7628	0.7626
	SVM	0.8050	0.8039	0.8085	0.8053
	RF	0.7906	0.7907	0.7927	0.7913
	Ada	0.7850	0.7852	0.7872	0.7856
	GB	0.7937	0.7934	0.7961	0.7942
	XGB	0.7851	0.7826	0.7898	0.7850
Case 2	LR	0.6542	0.7553	0.5827	0.8175
	NB	0.6936	0.7159	0.6817	0.8209
	kNN	0.6729	0.6978	0.6540	0.8106
	DT	0.6367	0.6622	0.6202	0.7896
	SVM	0.6827	0.7560	0.6271	0.8265
	RF	0.6713	0.7223	0.6318	0.8156
	Ada	0.6624	0.7084	0.6266	0.8098
	GB	0.6734	0.7242	0.6337	0.8169
	XGB	0.6657	0.7013	0.6382	0.8091
Case 3	LR	0.4825	0.6760	0.3850	0.8404
	NB	0.5985	0.6355	0.5776	0.8486
	kNN	0.5725	0.6251	0.5357	0.8428
	DT	0.5334	0.5737	0.5092	0.8256
	SVM	0.5666	0.7060	0.4812	0.8560
	RF	0.5611	0.6583	0.4962	0.8477
	Ada	0.5645	0.6452	0.5095	0.8456
	GB	0.5651	0.6554	0.5040	0.8477
	XGB	0.5638	0.6308	0.5170	0.8428

Table 7

Simulation results for variables generated from a multivariate t-distribution for Case4-Case6

Cases	Models	F1- Score	Precision	Recall	Accuracy
Case 4	LR	0.7874	0.7905	0.7864	0.7899
	NB	0.7764	0.7870	0.7688	0.7813
	kNN	0.7617	0.7688	0.7571	0.7658
	DT	0.7519	0.7587	0.7489	0.7560
	SVM	0.7915	0.7943	0.7911	0.7939
	RF	0.7730	0.7736	0.7748	0.7750
	Ada	0.7761	0.7791	0.7755	0.7788
	GB	0.7806	0.7821	0.7813	0.7828
	XGB	0.7695	0.7696	0.7719	0.7714
Case 5	LR	0.6414	0.7372	0.5726	0.8118
	NB	0.6773	0.6827	0.6774	0.8099
	kNN	0.6402	0.6703	0.6173	0.7953
	DT	0.6221	0.6530	0.6016	0.7853
	SVM	0.6606	0.7380	0.6028	0.8175
	RF	0.6491	0.6954	0.6137	0.8046
	Ada	0.6514	0.7059	0.6097	0.8078
	GB	0.6578	0.7076	0.6195	0.8101
	XGB	0.6461	0.6810	0.6197	0.7999
Case 6	LR	0.5235	0.6836	0.4317	0.8467
	NB	0.5938	0.5943	0.6015	0.8385
	kNN	0.5368	0.5944	0.4966	0.8319
	DT	0.5155	0.5672	0.4845	0.8226
	SVM	0.5267	0.6957	0.4327	0.8488
	RF	0.5387	0.6290	0.4783	0.8397
	Ada	0.5487	0.6446	0.4860	0.8439
	GB	0.5473	0.6431	0.4836	0.8433
	XGB	0.5379	0.6101	0.4884	0.8355

compared to other models, while the SVM model demonstrated successful performance in terms of precision. In general, the model performance obtained on data with a t-distribution was weaker compared to the log-normal distribution. This can be explained by the fact that the multivariate t-distribution may contain more outliers, making it more challenging for models to adapt.

3.4 *Simulation Study for Variables with Gamma and t-Distributions*

In this study, a correlated random data generation method was used to simulate variables with different probability distributions but maintaining a specific correlation structure. The following steps were followed in this method: First, random samples were generated from a multivariate normal distribution. The covariance matrix given above was used in this generation. Each variable generated from the multivariate normal distribution was transformed using the standard normal cumulative distribution function, producing values that follow a uniform distribution between 0 and 1. For these transformed variables, the inverse CDF method was used to convert them to Gamma and t-distributions:

Thus, variables with Gamma and t distributions but retaining their correlation were obtained. This approach is a Gaussian-Copula-based method widely used for simulating data with correlated different distributions.

A total of 5 independent variables were generated within the scope of the simulation. The first two variables (were generated from the Gamma distribution, while the other three variables () were generated from the t distribution. In the case of variables obtained from the Gamma distribution, the parameters were set as shape $k = 2$ and scale $\theta = 1$. For the variables generated from the t distribution, the degrees of freedom $\nu=5$ were determined. In this context, the mean values of the independent variables are set to 3 for y=0 and 5 for y=1, thereby shifting the positions of these distributions relative to the respective means. The noise variables added to the independent variables are also obtained from the gamma and t-distributions in the same manner and shifted according to the specified values.

Table 8

Simulation results for variables generated from variables with gamma and t-distributions for Case1-Case3

Cases	Models	F1- Score	Precision	Recall	Accuracy
Case 1	LR	0.8126	0.8137	0.8135	0.8130
	NB	0.8121	0.8158	0.8104	0.8131
	kNN	0.8080	0.7874	0.8317	0.8029
	DT	0.8038	0.7848	0.8275	0.7989
	SVM	0.8272	0.8129	0.8441	0.8241
	RF	0.8223	0.8139	0.8329	0.8206
	Ada	0.8212	0.8049	0.8405	0.8176
	GB	0.8261	0.8126	0.8420	0.8234
	XGB	0.8191	0.8034	0.8374	0.8155
Case 2	LR	0.6855	0.7501	0.6359	0.8272
	NB	0.7126	0.7056	0.7241	0.8268
	kNN	0.6895	0.6982	0.6858	0.8168
	DT	0.6733	0.6898	0.6665	0.8089
	SVM	0.7054	0.7633	0.6609	0.8366
	RF	0.7036	0.7415	0.6741	0.8318
	Ada	0.7074	0.7270	0.6948	0.8298
	GB	0.7100	0.7422	0.6852	0.8341
	XGB	0.7013	0.7178	0.6905	0.8257
Case 3	LR	0.5746	0.7077	0.4909	0.8564
	NB	0.6388	0.6269	0.6577	0.8521
	kNN	0.5926	0.6416	0.5581	0.8479
	DT	0.5739	0.6185	0.5484	0.8397
	SVM	0.5873	0.7385	0.4958	0.8627
	RF	0.6047	0.6946	0.5429	0.8595
	Ada	0.6102	0.6663	0.5735	0.8554
	GB	0.6126	0.6956	0.5545	0.8611
	XGB	0.6055	0.6613	0.5657	0.8538

Table 9

Simulation results for variables generated from variables with gamma and t-distributions for Case4-Case6

Cases	Models	F1- Score	Precision	Recall	Accuracy
Case 4	LR	0.8075	0.8079	0.8088	0.8082
	NB	0.8039	0.8061	0.8035	0.8051
	kNN	0.7892	0.7789	0.8017	0.7870
	DT	0.7845	0.7743	0.7985	0.7822
	SVM	0.8133	0.8034	0.8255	0.8116
	RF	0.8009	0.7956	0.8084	0.8101
	Ada	0.8089	0.7988	0.8215	0.8071
	GB	0.8100	0.8018	0.8203	0.8187
	XGB	0.7985	0.7891	0.8102	0.7967
Case 5	LR	0.6947	0.7453	0.6550	0.8297
	NB	0.7121	0.6928	0.7364	0.8235
	kNN	0.6750	0.6844	0.6704	0.8088
	DT	0.6586	0.6835	0.6430	0.8033
	SVM	0.6941	0.7597	0.6444	0.8322
	RF	0.6868	0.7232	0.6586	0.8222
	Ada	0.6979	0.7225	0.6800	0.8259
	GB	0.6953	0.7350	0.6643	0.8277
	XGB	0.6826	0.7070	0.6643	0.8171
Case 6	LR	0.5894	0.7047	0.5139	0.8594
	NB	0.6410	0.6015	0.6931	0.8464
	kNN	0.5806	0.6172	0.5550	0.8417
	DT	0.5521	0.6084	0.5181	0.8357
	SVM	0.5575	0.7351	0.4583	0.8579
	RF	0.5852	0.6706	0.5265	0.8531
	Ada	0.5955	0.6665	0.5473	0.8539
	GB	0.5964	0.6860	0.5347	0.8573
	XGB	0.5877	0.6472	0.5454	0.8491

According to the simulation data in Table 8-Table 9, in scenarios where classes are balanced (p=0.5), the highest performance was observed in SVM, GB, and RF models. In this scenario, in addition to the models having high accuracy rates, metrics such as sensitivity and F1-score also showed similar values, demonstrating a balanced performance. In scenarios with increased class imbalance (p=0.3 and p=0.2), all models experienced a decrease in sensitivity and F1-score values, but NB produced successful results against this imbalance. In terms of accuracy, the SVM model demonstrated successful performance.

4. Conclusion

In simulation studies, the effect of variables with different distributional characteristics on model performance was examined, and evaluations were made based on the findings. These findings highlight the critical role of the distributional structure of the data in model selection and serve as a guide for determining the most appropriate algorithms for different types of data. The effectiveness of linear models under a multivariate normal distribution supports the preference for linear methods when working with such data. In contrast, the superiority of tree-based methods in log-normal and gamma and t-distributions indicates that non-linear relationships can be better learned in these distributions. In data structures that may contain extreme values, such as the t distribution, a decline in the overall performance of the models was observed; however, SVM and NB produced more successful results despite this challenging structure.

Overall, class imbalance and the addition of categorical variables caused a decrease in F1-score values, particularly in skewed data sets, and some models were observed to struggle in identifying the positive class. In such cases, tree-based methods demonstrated resilience against both class imbalance and data complexity, yielding more consistent results. In conclusion, model selection based on the distributional characteristics and class distribution of the dataset is of great importance in improving classification success.

This research evaluated the performance of several classification methods using datasets that were synthetically produced from multiple probability distributions. Simulation results showed that model performance varied significantly depending on the data structure. Under a multivariate normal distribution, LR, NB, and SVM quite successful results. This indicates that linear models are effective for data suitable for

normal distribution. However, when the data had skewed or complex distributions (t-distribution, log-normal distribution, gamma and t-distribution), the performance of linear methods decreased; in contrast, SVM and tree-based methods were more robust and successful. Additionally, while the performance of many models decreased in structures containing outliers such as the t distribution, SVM and NB behaved more consistently under these challenging conditions. However, it has been observed that some models, such as DT, are inadequate in complex structures and prone to overfitting. Realistic data problems such as class imbalance have also had negative effects on model performance. In these cases, the most successful results have generally been obtained with SVM and tree-based methods.

The findings reveal that in machine learning-based classification studies, not only the choice of algorithms but also the distributional characteristics of the data set, class balance, and variable types must be considered. Future studies could enable a more comprehensive evaluation of model performance by conducting broader analyses on data sets with different distributions.

Conflict of interest

There is no conflict of interest in this study.

References

[1] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge, U.K.: Cambridge University Press (2014).

[2] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3133–3181 (2014).

[3] J. Charoenpong, B. Pimpunchat, S. Amornsamankul, W. Triampo, and N. Nuttavut, "A comparison of machine learning algorithms and their applications," *Int. J. Simulation: Systems, Science & Technology*, vol. 20, no. 4 (2019).

[4] N. Macià and E. Bernadó-Mansilla, "Towards UCI+: A mindful repository design," *Information Sciences*, vol. 261, pp. 237–262 (2014).

[5] A. V. Aglarci and C. Bal, "Classification performance of machine learning methods in different data structures," *Communications in*

Statistics – Simulation and Computation, vol. 53, no. 12, pp. 6471–6489 (2024).

[6] Y. S. Li and C. Y. Guo, "Random logistic machine (RLM): Transforming statistical models into machine learning approach," *Communications in Statistics – Theory and Methods*, vol. 53, no. 21, pp. 7517–7525 (2024).

[7] S. Ray, "A quick review of machine learning algorithms," in *Proc. Int. Conf. on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, Bhubaneswar, India, pp. 35–39 (Feb. 2019).

[8] S. Zhang, M. Zong, K. Sun, Y. Liu, and D. Cheng, "Efficient kNN algorithm based on graph sparse reconstruction," in *Proc. 10th Int. Conf. on Advanced Data Mining and Applications (ADMA)*, Guilin, China, pp. 356–369 (Dec. 2014).

[9] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proc. 22nd Int. Conf. on Machine Learning (ICML)*, Bonn, Germany, pp. 625–632 (Aug. 2005).

[10] M. Denil, D. Matheson, and N. de Freitas, "Narrowing the gap: Random forests in theory and in practice," in *Proc. Int. Conf. on Machine Learning (ICML)*, Atlanta, GA, USA, pp. 665–673 (Jan. 2014).

[11] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. 13th Int. Conf. on Machine Learning (ICML)*, pp. 148–156 (July 1996).

[12] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232 (2001).

[13] J. Yoon, "Forecasting of real GDP growth using machine learning models: Gradient boosting and random forest approach," *Computational Economics*, vol. 57, no. 1, pp. 247–265 (2021).

Received September, 2025