Assessing the Effectiveness of Various Text Classification Algorithms in Customer Complaint Classification: An Informative Resource for Data Scientists and Data Analysts

Yehia Helmy
Professor of Information Systems,
Business Information Systems
Department, Helwan University
Cairo, Egypt

Merna Ashraf
Business Information Systems
Department, Faculty of Commerce
and Business Administration,
Helwan University
Cairo, Egypt

Laila Abdelhamid
Information Systems Department
Faculty of Computers and AI,
Helwan University
Cairo, Egypt

ABSTRACT
Due to the numerous issues or challenges that aren't always within the company's control. Customers became unhappy. Customer complaint is the method by which they convey their dissatisfaction. Due to the rapid advancement of technology and the various convenient channels available for customers to voice their complaints, including email, web, and chatbots, online complaints have experienced exponential growth. As a result, classifying these complaints under the pertinent issue in time became a difficult task. Selecting the appropriate classification model and Fitting it with the proper training and testing ratios is a crucial topic that always faces researchers. This paper implements and compares the performance of six text classification machine learning algorithms used in multi-classification (SVM, KNN, NB, DT, RF, and GB) under two types of sampling (random and stratified) with the use of various data splitting ratios 50:50,80:20, 60:40, 70:30, and 90:10 on a Complaint Dataset. This paper aims to provide a roadmap for researchers working in the text classification field that helps them select the optimum classification model and splitting ratio. The results demonstrate that DT with an accuracy of 99%, F1-measure of 99%, and runtime of 1 second outperformed all other algorithms. And that the most suitable splitting ratio that fits most algorithms and acts as a secure base to work with is 80:20. It also indicates that using stratified sampling in multi-class text classification produces better results than random sampling.

Keywords
Text classification, Data splitting, Supervised machine learning, Multi-Classification, Random sampling and Stratified sampling, Complaint handling.

1. INTRODUCTION
Classification of online customers' complaints to their underlying issues and directing them to the appropriate department for resolution became a critical task. Manual analysis of complaints becoming time-consuming and ineffective because of the overwhelming volume of complaints that the business must process daily. As a result, automating the classification of complaints is essential for minimizing the workload, shortening the waiting time for the customer to receive a solution, and eliminating manual work (BOZYİĞİT, Doğan et al. 2022).

Therefore machine learning (ML) algorithms have become essential for the classification task. It has a great effect in classifying textual data to the appropriate issue with less time and cost (BOZYİĞİT, Doğan et al. 2022).

Text classification is a method that receives texts as input and assigns a label for it from an identified set of classes. It is the process of categorizing text into one or more categories(Miner 2012). As shown in Figure 1, there are many different classification types in machine learning (Sen, Hajra et al. 2020).

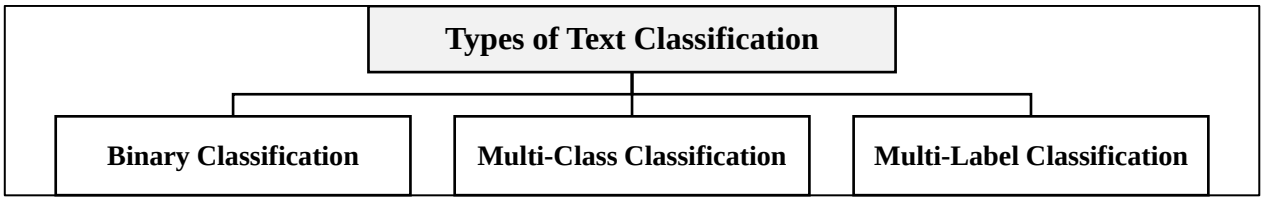

Fig 1: Types of Text Classification.

Binary classification refers to a classification that has two classes. Such as "yes" or "no", "0" or "1". The ML Algorithms that are popular and suitable for binary classification include Logistic Regression (LR), K-Nearest Neighbors (KNN), Decision Trees (DT), Support Vector Machine (SVM), and Naive Bayes (Tufail, Ma et al. 2020).

Multi-class classification refers to a classification that has more than two classes. Such as "high", "medium", and "low". ML Algorithms that are popular for Multi-class classification include K-Nearest Neighbors (KNN), Decision Trees (DT), Random Forest (RF), Adapted SVM (SVC and Linear), Gradient Boosting (GB) ,and Multinomial Naive Bayes (MNB) (Kadhim 2019).

Multi-label classification refers to a classification that has two or more classes, where one or more categories may be predicted for each instance. ML Algorithms that are popular for Multi-label classification include Multi-label Decision Trees, Multi-label Random Forests, and Multi-label Gradient Boosting (Endut, Hamzah et al. 2022).

Selecting the right classifier is one of the important steps in text classification. Without having an adequate understanding of every algorithm, finding the most effective model for text classification becomes difficult (Naseem, Razzak et al. 2021).

On the other hand, choosing the data splitting ratio that positively influences a classifier's performance is also a challenging task. Data splitting, also known as a train-test split, is the division of data into subsets for model training and model testing (Muraina 2022). Therefore, it is important to determine the appropriate classifier and splitting ratio that will result in accurate output that helps with the right decision-making.

Therefore, the goal of this paper is to test and compare the performance of six text classification algorithms used in multi-classification under two types of sampling with the use of various data splitting ratios on a complaint dataset in order to identify which algorithm and splitting ratio will produce the best results.

The rest of this paper is divided as follows: in Section 2 the theoretical basis of Different Multi-class Text Classification Machine learning algorithms are presented. In Section 3 Research methodology and related work are reviewed. In Section 4 the experimental analysis is presented. Section 5 shows the Results and Discussion. Finally, Section 6 closes the paper with a conclusion.

2. MACHINE LEARNING ALGORITHMS FOR MULTI-CLASS TEXT CLASSIFICATION.

The paper's goal is to classify customers' complaints to the relevant issue in order to forward them to the department in charge. And because customers' complaints are diverse and belong to different issues. Therefore, Out of many ML algorithms used in text classification, the focus will be on the well-known machine learning algorithms appropriate for multi-class text classification.

- **Support vector machine (SVM):** It is one of the known and often used methods for text classification. The training data are plotted in multi-dimensional space by the non-probabilistic binary linear classification technique known as SVM. After that, SVM uses a hyper-plane to classify the classes. If the classes in a multi-dimensional space cannot be split linearly, the method will introduce a new dimension. This procedure will go on until training data can be divided into two groups. The Supporting Vector Classifier (SVC) and the Linear Support Vector Machine are the two SVM subtypes that support multi-classification (Naseem, Razzak et al. 2021).

- **K-Nearest Neighbors (KNN):** It is one of the classifiers that made use of the knowledge of the identification that utilized the class of the query with regard to more than just the text that is nearby in the text area. As well as the K texts' classes that can be closed to it. While the kNN classifier examines the closest neighbors among the learning texts and uses the classes of the k neighbors to give weights to the class candidates, specified test texts are used to identify the class. It divides the texts into one or more predetermined classes based on their subject using the Euclidean distance formula. In order to classify texts, the classifier uses keywords from texts that have been matched to keywords from new texts. The classifier tries for potential classes for the text by identifying learning texts that have keywords nearest to them. The KNN classifier searches the corpus for the k keywords that may be near to element y (Kadhim 2019).

- **Multinomial Naïve Bayes (MNB):** It is one of the NB classifier families used for multinomial distributed data. It is frequently used as a starting point in text classification since it is quick and simple to use. This model is generative. It is assumed that a corpus of texts is created by choosing a class for each document and then individually creating each word of that document using a distribution appropriate to that class (Xu, Li et al. 2017).

.

- **Decision Tree (DT):** It is a classifier that employs a hierarchical mechanism to categorize the data, with predicate-based portioning at each node. It is a top-down strategy that begins from the root. It employed Different splitting techniques to divide the node, such as the single attribute split, which divides the node based on a single value or word phrase. Also, the document similarity split, which divides the node depending on how similar the two documents are. The node discriminant function is used to divide multivalued characteristics in the third case. A specific threshold can be selected, such as the tree's maximum depth, for convergence (Kalra and Aggarwal 2017).

- **Random Forest (RF):** It is a classifier that also known as an ensemble learning methodology, focuses on techniques to compare the outcomes of numerous trained models. A bootstrapped subset of the training text is used to train each tree in the DT classifier that makes up the RF classifier. At each decision node, a random subset of the characteristics is chosen, and the model only looks at a portion of these attributes. The main problem with using a single tree is that it has a lot of variety, which means the way the training data and features are organized might affect the outcome. For textual data, this classifier can be trained quickly, but it takes a while before it can make predictions (Naseem, Razzak et al. 2021).

- **Gradient Boosting (GB):** It is a tree-based classifier that boosts the accuracy of boots' prediction by repeatedly creating a better tree from earlier versions. The inaccuracy of the previous tree will be reduced with each iteration (Anwar, Pratiwi et al. 2021).

3. RESEARCH METHODOLOGY AND RELATED WORK

3.1 Research Methodology

The purpose of this paper is to find the classification algorithm and optimal splitting ratio that will speed up the process of assigning customers' complaints to the appropriate issue. By scanning the most widely used databases (Science Direct, Springer, and IEEE) to find articles that are relevant to automatic text classification in complaint handling. Only articles that were written in English and reference to at least one of the algorithms were examined to demonstrate how automated text classification is used in those articles. Additionally, a keyword search was conducted using the algorithms' names and text classification to find a large number of publications related to the goal of the paper. The target is to review articles published between “2017 to 2023".

3.2 Related Work

(Arusada, Putri et al. 2017) Applied Naïve Bayes and Support Vector Machine to classify customers' complaints from Twitter to determine which Algorithm performed better than another on a dataset with a size of 1.440 records and 4 different classes, two data splitting ratios of 70:30 and 60:40 were set. The findings indicated that SVM with a splitting ratio of 60:40 and an accuracy of 95% outperformed NB.

(Anwar, Pratiwi et al. 2021) Applied Random Forest and Gradient Boosting to classify public-sector customers' complaints data. To determine which algorithm outperforms another on a dataset with a size of 44961 records and 10 different classes, a data splitting ratio of 80:20 was set. The findings indicated that RF and GB both have an accuracy of 73%.

(Goncarovs 2019) Applied active learning support vector machine and decision tree to classify customer complaints in online banking. Only 20% of the training data was used to train the model in order to compare the performance of the algorithms on a dataset of 1000 records and 8 different classes. The findings indicated that active learning SVM with an accuracy of 86.4% outperformed DT which has an accuracy of 56.6%.

(Li and Li 2019) Applied Naive Bayes to classify customers' railway complaints on a dataset of 14651 records and 7 different classes, a data splitting ratio of 80:20 was set. The findings indicated that it achieved an accuracy of 78%.

(BOZYİĞİT, Doğan et al. 2022) Applied Naïve Bayes, K-Nearest Neighbors, Support Vector Machine, Random Forest, and Gradient Boosting to classify customers' complaints in the Food industry on a dataset of 2217 records and 5 different classes, a data splitting was performed by dividing the data set into 10 pieces by cross-validation. The findings indicated that GB outperformed the other algorithms with an accuracy of 88%.

(Bazzan, Echeveste et al. 2023) Applied Naïve Bayes, Support Vector Machine, Random Forest, and Gradient Boosting to classify customers' complaints in residential projects on a dataset of 2765 records and 6 different classes. Three data splitting ratios of 70: 30, 75:25, and 80:20 were set. According to the results, GB had the greatest accuracy rate (82.89%) under a splitting ratio of 70:30. The runtime was 4.59 h, which was longer than that of other models. The second one, RF, ran for almost 4 hours and had an accuracy of 81.68%. Finally, despite quick processing times, Naive Bayes and SVM had the lowest accuracy, with 77.71% and 77.95%, respectively.

(HaCohen-Kerner, Dilmon et al. 2019) Applied Support Vector Machine, and Random Forest to classify customers' complaints in insurance on a dataset of 2073 records and 7 different classes, a data splitting ratio of 63: 33 was set. The findings indicated that SVM with an accuracy of 82.78% outperformed RF with an accuracy of 76.55%.

(Hasan, Matin et al. 2020) Applied Support Vector Machine to classify online customers' complaints for a restaurant on a dataset of 7280 records and 3 different classes, data splitting was performed by dividing the data set into 5 pieces by cross-validation. The findings indicated that SVM achieved an accuracy of 91.53%.

(Choi 2018) Applied K-Nearest Neighbors, Support Vector Machine, and Decision Tree to classify online customers' complaints in the mobile telecom sector on a dataset of 10000 records and 7 different classes. A data splitting ratio of 60: 40 was set. The findings indicated that KNN with an accuracy of 79.40% outperformed SVM with an accuracy of 71.77%, and DT with an accuracy of 62.90%.

(Ali, Guru et al. 2019) Applied K-Nearest Neighbors and Support Vector Machine to classify customers' complaints in the farming industry on a dataset of 3700 records and 5 different classes, a data splitting ratio of 50: 50 was set. The findings indicated that KNN with an accuracy of 93.53% outperformed SVM with an accuracy of 93.38%.

4. EXPERIMENTAL ANALYSIS

4.1 Experiment Settings

To effectively evaluate the efficacy of the aforementioned algorithms at various splitting ratios the following procedures were used.

- First, Different text preprocessing techniques, including cleaning and normalization, tokenization, stop word removal, and stemming & lemmatization were applied to a complaint data set that was obtained from " https://www.kaggle.com/" to prepare it for text classification. The dataset consists of 4782 records and 20 different classes.
- Second, two sampling techniques random sampling and stratified sampling with ratios of "50:50," "80:20," "60:40," "70:30," and "90:10" were utilized.
- Third, the Scikit-learn package of the Python programming language was employed to perform the experiment.

4.2 Evaluation Metrics

For the evaluation of the classification algorithms and comparison, four performance measures were used accuracy, Precision, Recall, and F1-Score.

- **Accuracy**

Accuracy is one of the most widely used metrics for classification performance which is calculated as the ratio of samples that are correctly classified to all samples (Tharwat and Informatics 2020).

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

Where

True Positive (TP): Is the number of predictions when the classifier properly identifies the positive class as positive
True Negative (TN): Is the number of predictions in which the classifier properly identified the negative class as negative
False Positives (FP): Is the proportion of predictions in which a classifier predicts a negative class as a positive one.
False Negative (FN): Is the proportion of predictions in which the classifier misinterprets the positive class as the negative class.
(Tharwat and Informatics 2020)

- **Precision**

Precision is the ratio of accurately classified positive samples to the total number of positive predicted samples (Tharwat and Informatics 2020).

$$\textbf{Precision} = \frac{TP}{TP+FP}$$

- **Recall**

Recall shows the proportion of positively identified positive samples to all positive samples (Tharwat and Informatics 2020).

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

- **F1-Score**

F1- score also known as the F-measure. It denotes the harmonic mean of recall and precision (Tharwat and Informatics 2020).

$$\textbf{F1-Score} = 2 * \frac{Precision*Recall}{Precision+Recall} = \frac{2TP}{2TP+FP+FN}$$

5. RESULTS AND DISCUSSION

This section presents the results of implementing classification algorithms under different splitting ratios with random and stratified sampling. The following sub-section presents the results.

5.1 Results of Applying Random Sampling with Different Splitting Ratios.

Table 1. Results of Classification Algorithms with (50:50) Splitting Ratio –under Random Sampling.

	Data Splitting (50:50)				
Algorithms	**Accuracy (%)**	**Precision**	**Recall**	**F1-Score**	**Time(s)**
SVM(SVC)	36%	0.039	0.100	0.056	152
SVM(Linear)	74%	0.497	0.441	0.443	1.6
KNN	99%	0.837	0.850	0.842	2
MNB	50%	0.119	0.138	0.102	0.6
DT	99%	0.889	0.900	0.894	0.9
RF	37%	0.231	0.092	0.078	1.7
GB	85%	0.412	0.454	0.420	170

Table 2. Results of Classification Algorithms with (80:20) Splitting Ratio –under Random Sampling.

	Data Splitting (80:20)				
Algorithms	**Accuracy (%)**	**Precision**	**Recall**	**F1-Score**	**Time(s)**
SVM(SVC)	34%	0.045	0.111	0.064	153
SVM(Linear)	75%	0.601	0.499	0.512	2
KNN	99%	0.829	0.842	0.835	1.8
MNB	50%	0.132	0.158	0.115	0.6
DT	99%	0.998	0.998	0.998	1
RF	33%	0.199	0.094	0.076	1.9
GB	79%	0.298	0.359	0.308	325

Table 3. Results of Classification Algorithms with (60:40) Splitting Ratio –under Random Sampling.

	Data Splitting (60:40)				
Algorithms	**Accuracy (%)**	**Precision**	**Recall**	**F1-Score**	**Time(s)**
SVM(SVC)	35%	0.046	0.111	0.065	151
SVM(Linear)	76%	0.495	0.419	0.424	2
KNN	99%	0.873	0.889	0.880	2
MNB	49%	0.132	0.154	0.113	0.6
DT	99%	0.842	0.842	0.842	1
RF	35%	0.185	0.097	0.081	1.7
GB	0.9%	0.001	0.050	0.001	226

Table 4. Results of Classification Algorithms with (70:30) Splitting Ratio –under Random Sampling.

	Data Splitting (70:30)				
Algorithms	**Accuracy (%)**	**Precision**	**Recall**	**F1-Score**	**Time(s)**
SVM(SVC)	35%	0.047	0.111	0.066	156
SVM(Linear)	76%	0.534	0.421	0.431	1.9
KNN	99%	0.872	0.889	0.879	2
MNB	50%	0.133	0.157	0.116	0.6
DT	99%	0.842	0.842	0.842	1
RF	36%	0.250	0.099	0.086	1.8
GB	93%	0.648	0.578	0.589	194

Table 5. Results of Classification Algorithms with (90:10) Splitting Ratio –under Random Sampling.

Algorithms	*Data Splitting (90:10)*				
	Accuracy (%)	**Precision**	**Recall**	**F1-Score**	**Time(s)**
SVM(SVC)	34%	0.043	0.111	0.062	146
SVM(Linear)	74%	0.551	0.482	0.490	2
KNN	99%	0.825	0.842	0.832	1
MNB	53%	0.134	0.171	0.127	0.6
DT	99%	0.907	0.944	0.917	1
RF	34%	0.206	0.094	0.078	2
GB	92%	0.669	0.670	0.659	300

5.2 Results of Applying Stratified Sampling with Different Splitting Ratios.

Table 6. Results of Classification Algorithms with (50:50) Splitting Ratio –under Stratified Sampling.

Algorithms	*Data Splitting (50:50)*				
	Accuracy (%)	**Precision**	**Recall**	**F1-Score**	**Time(s)**
SVM(SVC)	36%	0.040	0.100	0.057	119
SVM(Linear)	78%	0.576	0.513	0.526	1.7
KNN	99%	0.845	0.850	0.847	3
MNB	52%	0.122	0.147	0.107	0.6
DT	99%	0.925	0.950	0.933	0.9
RF	43%	0.189	0.108	0.101	1.5
GB	90%	0.640	0.690	0.649	158

Table 7. Results of Classification Algorithms with (80:20) Splitting Ratio –under Stratified Sampling.

Algorithms	*Data Splitting (80:20)*				
	Accuracy (%)	**Precision**	**Recall**	**F1-Score**	**Time(s)**
SVM(SVC)	36%	0.047	0.111	0.066	141
SVM(Linear)	79%	0.487	0.460	0.464	2
KNN	99%	0.933	0.944	0.938	2
MNB	56%	0.137	0.173	0.132	0.6
DT	99%	0.998	0.998	0.998	1
RF	45%	0.209	0.128	0.117	1.9
GB	97%	0.718	0.748	0.730	265

Table 8. Results of Classification Algorithms with (60:40) Splitting Ratio –under Stratified Sampling.

Algorithms	*Data Splitting (60:40)*				
	Accuracy (%)	**Precision**	**Recall**	**F1-Score**	**Time(s)**
SVM(SVC)	36%	0.047	0.111	0.066	139
SVM(Linear)	79%	0.487	0.460	0.464	2
KNN	99%	0.933	0.944	0.938	1.8
MNB	56%	0.137	0.173	0.132	0.7
DT	99%	0.998	0.998	0.988	1
RF	40%	0.191	0.115	0.104	1.9
GB	97%	0.716	0.744	0.728	264

Table 9. Results of Classification Algorithms with (70:30) Splitting Ratio –under Stratified Sampling.

Algorithms	*Data Splitting (70:30)*				
	Accuracy (%)	**Precision**	**Recall**	**F1-Score**	**Time(s)**
SVM(SVC)	36%	0.043	0.100	0.060	138
SVM(Linear)	79%	0.554	0.518	0.512	1.8
KNN	99%	0.843	0.850	0.846	3
MNB	55%	0.123	0.153	0.115	0.6
DT	99%	0.925	0.950	0.933	1
RF	44%	0.163	0.113	0.103	1.8
GB	22%	0.095	0.102	0.060	288

Table 10. Results of Classification Algorithms with (90:10) Splitting Ratio –under Stratified Sampling.

Algorithms	*Data Splitting (90:10)*				
	Accuracy (%)	**Precision**	**Recall**	**F1-Score**	**Time(s)**
SVM(SVC)	36%	0.052	0.118	0.072	140
SVM(Linear)	79%	0.527	0.507	0.507	2
KNN	99%	0.998	0.998	0.998	1
MNB	56%	0.146	0.187	0.143	0.6
DT	99%	0.998	0.998	0.998	1
RF	41%	0.205	0.125	0.110	2
GB	97%	0.741	0.762	0.751	239

5.3 Discussion

In the previous subsection, the impacts of various classification algorithms and data splitting ratios on the complaint dataset were presented. The findings show that the impact of data splitting ratios differs with respect to the various classification algorithms applied.

As shown in Table 1 to Table 5, which applied different splitting ratios including 50:50, 80:20,60:40, 70:30, and 90:10 respectively under random sampling with the six famous algorithms that are used in multi-class text classification of customers' complaints. The findings indicate that DT and KNN outperformed the other algorithms in all splitting ratios with an accuracy of 99%, Precision ranging from 83% to 99%, Recall ranging from 84% to 99%, F1-score ranging from 83% to 99%, and runtime 1-2 seconds. Also, when stratified sampling was applied as shown in Table 6 to Table 10, The findings indicate that KNN and DT outperformed the other algorithms in all splitting ratios with an accuracy of 99%, Precision ranging from 84% to 99%, Recall ranging from 85% to 99%, F1-score ranging from 85% to 99%, and runtime 1-2 seconds.

To clarify the findings, three comparisons were performed. The aim of the first one is to identify which sample technique to apply to multi-class text classification, the second one is to determine the optimal splitting ratio, and the third one is to determine the multi-class text classification algorithm that yields the best results. Table 11, summarized the results of the first comparison.

Table 11. Comparison between Random Sampling and Stratified Sampling on six text classification algorithms.

Algorithms	**Sampling Method**	**Evaluation Metrics**							
		Accuracy		**Precision**		**Recall**		**F1-score**	
		min	**max**	**min**	**max**	**min**	**max**	**min**	**max**
KNN	**Random**	99%	99%	83%	99%	84%	99%	83%	99%
	Stratified	**99%**	**99%**	**84%**	**99%**	**85%**	**99%**	**85%**	**99%**
DT	**Random**	99%	99%	83%	99%	84%	99%	83%	99%
	Stratified	**99%**	**99%**	**84%**	**99%**	**85%**	**99%**	**85%**	**99%**
SVM(linear)	**Random**	74%	76%	50%	60%	41%	48%	42%	51%
	Stratified	78%	79%	49%	58%	46%	52%	46%	53%
MNB	**Random**	49%	53%	12%	13%	13%	17%	10%	13%
	Stratified	52%	56%	12%	15%	15%	19%	11%	14%
GB	**Random**	0.9%	93%	0.1%	66%	5%	67%	0.1%	66%
	Stratified	22%	97%	9.5%	74%	10%	76%	6%	75%
SVM(SVC)	**Random**	33%	37%	4%	25%	6%	11%	5%	8%
	Stratified	36%	45%	4%	21%	10%	13%	6%	12%
RF	**Random**	33%	37%	4%	25%	6%	11%	5%	8%
	Stratified	36%	45%	4%	21%	10%	13%	6%	12%

As shown in Table 11, according to the four evaluation metrics that were employed and by examining the minimum and maximum values for each algorithm under stratified sampling and random sampling. It appears that stratified sampling has superior minimum and maximum values than random sampling, hence it is preferable to use stratified sampling for multi-classification.

The second comparison was carried out to identify which is the best splitting ratio for all algorithms under random sampling and stratified sampling. In other words, which splitting ratio represents a secure base for researchers to work, the one that did not result in high deviations in results as illustrated in Table 12.

Table 12. Comparison between Different splitting ratios on six text classification algorithms.

Algorithm	Evaluation metrics	Random Sampling					Stratified Sampling				
		50:50	80:20	60:40	70:30	90:10	50:50	80:20	60:40	70:30	90:10
KNN	Accuracy	99%	99%	99%	99%	99%	99%	99%	99%	99%	99%
	Precision	84%	83%	87%	87%	83%	85%	93%	93%	84%	99%
	Recall	85%	84%	89%	89%	84%	85%	94%	94%	85%	99%
	F1-score	84%	84%	88%	88%	83%	85%	94%	94%	85%	99%
DT	Accuracy	99%	99%	99%	99%	99%	99%	99%	99%	99%	99%
	Precision	89%	99%	84%	84%	91%	93%	99%	99%	93%	99%
	Recall	90%	99%	84%	84%	94%	95%	99%	99%	95%	99%
	F1-score	89%	99%	84%	84%	92%	93%	99%	99%	93%	99%
SVM(Linear)	Accuracy	74%	75%	76%	76%	74%	78%	79%	79%	79%	79%
	Precision	50%	60%	50%	53%	55%	58%	49%	50%	55%	53%
	Recall	44%	45%	42%	42%	48%	51%	46%	46%	52%	51%
	F1-score	44%	51%	42%	43%	49%	53%	46%	46%	51%	51%
MNB	Accuracy	50%	50%	49%	50%	53%	52%	56%	56%	55%	56%
	Precision	12%	13%	13%	13%	13%	12%	14%	14%	12%	15%
	Recall	14%	16%	15%	16%	17%	15%	17%	17%	15%	19%
	F1-score	10%	12%	11%	12%	13%	11%	13%	13%	12%	14%
GB	Accuracy	85%	79%	0.9%	93%	92%	90%	97%	97%	22%	97%
	Precision	41%	30%	0.1%	65%	67%	64%	72%	72%	10%	74%
	Recall	45%	40%	5%	58%	67%	69%	75%	74%	10%	76%
	F1-score	42%	31%	0.1%	59%	66%	65%	73%	73%	6%	75%
SVM(SVC)	Accuracy	36%	34%	35%	35%	34%	36%	36%	36%	36%	36%
	Precision	4%	5%	5%	5%	4%	4%	5%	5%	4%	5%
	Recall	10%	11%	11%	11%	11%	10%	11%	11%	10%	12%
	F1-score	7%	6%	5%	7%	6%	6%	7%	7%	6%	7%
RF	Accuracy	37%	33%	35%	36%	34%	43%	45%	40%	44%	41%
	Precision	23%	20%	19%	25%	21%	20%	21%	20%	16%	21%
	Recall	9%	9%	10%	10%	9%	11%	13%	16%	11%	13%
	F1-score	9%	8%	8%	9%	8%	10%	12%	10%	10%	11%

As shown in Table 12, the performance of different algorithms changed with different splitting ratios. This indicates that while a particular splitting ratio might be the best for one algorithm, it might be the worst for another. As a result, The results shown in Table 12 were used to create a matrix that demonstrates which of them can be regarded as the most suitable splitting ratio for all algorithms.

Table 13. A matrix demonstrates the best splitting ratio.

	Random Sampling					Stratified Sampling				
Algorithm	50:50	80:20	60:40	70:30	90:10	50:50	80:20	60:40	70:30	90:10
KNN					✓				✓	
DT			✓	✓		✓			✓	
SVM(Linear)	✓							✓		
MNB			✓			✓				
GB			✓						✓	
SVM(SVC)	✓				✓	✓			✓	
RF			✓							✓

As shown in Table 13, all splitting ratios have a flaw in one or two algorithms except 80:20 .Therefore, for researchers who focus on multi-class text classification, this splitting ratio might be regarded as a safe base.

The third comparison was conducted to identify the best classification algorithms in terms of the four evaluation metrics and time. For the comparison, the highest value of each algorithm was taken from the random and stratified sampling regardless of the splitting ratio. Figure 2, Figure 3, Figure 4, Figure 5, and Figure 6 illustrate the results.

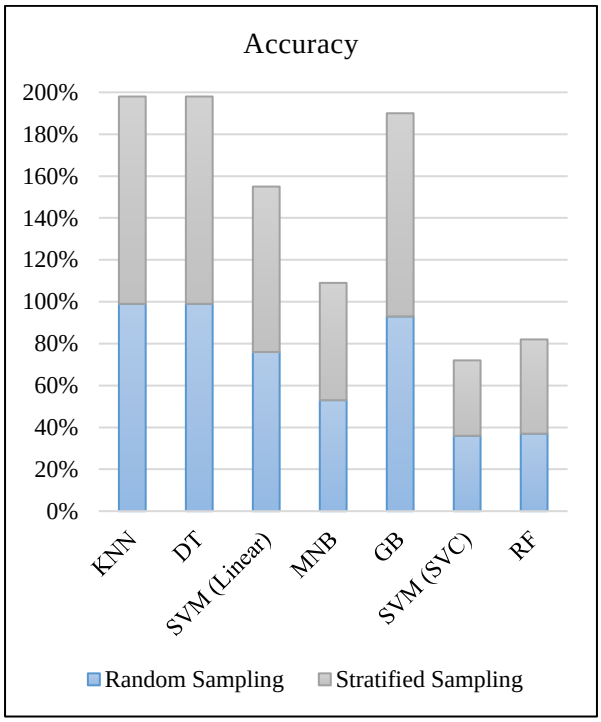

Fig 2: The text classification Algs. Based on Accuracy.

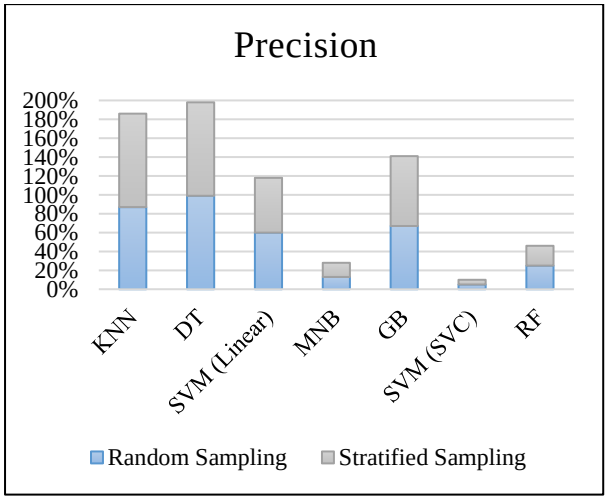

Fig 3: The text classification Algs. Based on Precision.

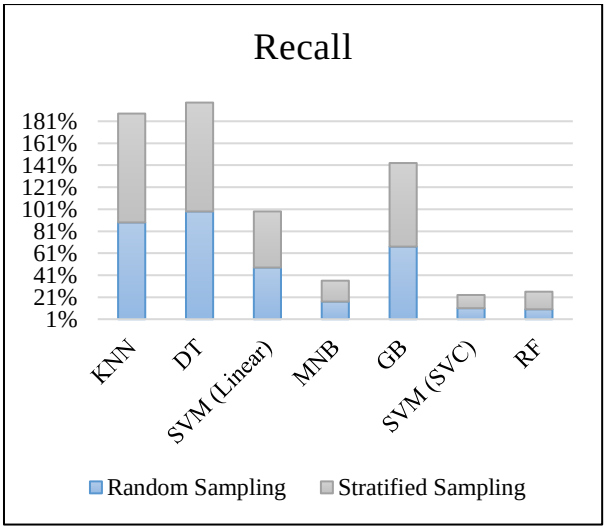

Fig 4: The text classification Algs. Based on Recall.

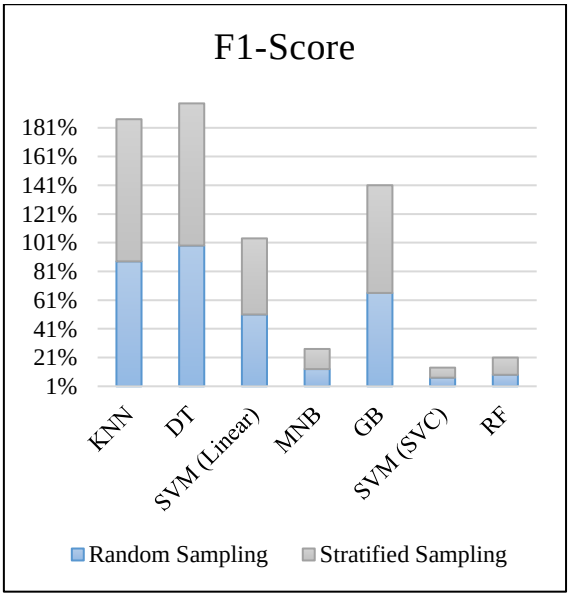

Fig 5: The text classification Algs. Based on F1-Score.

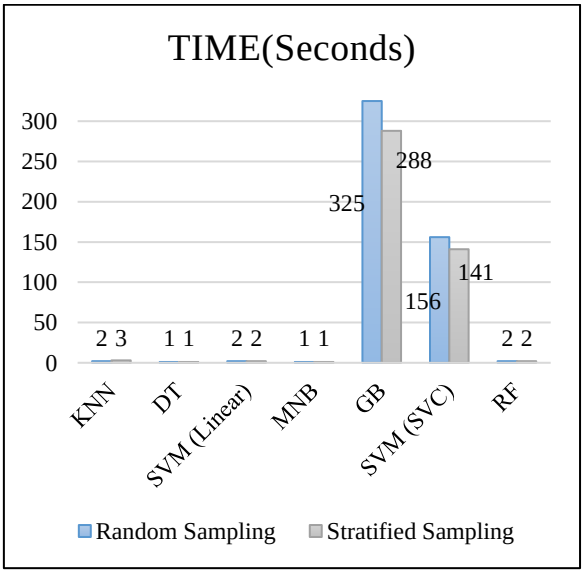

Fig 6: The text classification Algs. Based on Time.

Based on the previous Figures, DT performed better in terms of accuracy, precision, recall, f1-Score, and time than all the other algorithms. This indicates that using DT can produce very accurate results in the case of multi-class text classification.

Finally, This study investigated the varying data splitting ratios under two sampling techniques on six multi-class text classification algorithms in order to determine the best-split, sampling technique, and classification algorithm that best fit to improve the text classification. The findings can assist researchers in defining parameters that will enable them to achieve better results.

6. CONCLUSION

In this paper, different classification algorithms, different splitting ratios, and sampling techniques were discussed. It was emphatically put that the splitting ratio and sampling techniques that were utilized greatly affected the performance of the classification model. It is therefore suggested that: First, caution must be taken in selecting the splitting ratio and that an 80:20 split is the safest and produces the least amount of variation in the results. Second, care must be taken in selecting the right classifier that gives the best accuracy, precision, recall, and F1 score. Third, Do not overlook the runtime either, as it can get monotonous if the activities are repeated again for an extended period of time. The experiment's finding indicates that DT outperformed all the other classifiers and that stratified sampling is better in multi-classification.

7. ACKNOWLEDGMENTS

We would like to express our gratitude to the researchers who contributed to this study. Their invaluable contributions allowed us to complete this study successfully. We appreciate and thank them for their efforts and support, as well as their time and dedication.

8. REFERENCES

[1] Ali, M., et al. (2019). Classifying Arabic farmers' complaints based on crops and diseases using machine learning approaches. Recent Trends in Image Processing and Pattern Recognition: Second International Conference, RTIP2R 2018, Solapur, India, December 21–22, 2018, Revised Selected Papers, Part III 2, Springer.

[2] Anwar, M. T., et al. (2021). "Automatic Complaints Categorization Using Random Forest and Gradient Boosting." **3**(1): 210106.

[3] Arusada, M. D. N., et al. (2017). Training data optimization strategy for multiclass text classification. 2017 5th International Conference on Information and Communication Technology (ICoIC7), IEEE.

[4] Bazzan, J., et al. (2023). "An Information Management Model for Addressing Residents' Complaints through Artificial Intelligence Techniques." **13**(3): 737.

[5] BOZYİĞİT, F., et al. (2022). "Categorization of customer complaints in food industry using machine learning approaches." **5**(1): 85-91.

[6] Choi, C. (2018). Predicting customer complaints in mobile telecom industry using machine learning algorithms, Purdue University.

[7] Endut, N., et al. (2022). "A Systematic Literature Review on Multi-Label Classification based on Machine Learning Algorithms." **11**(2): 658.

[8] Goncarovs, P. (2019). Active learning svm classification algorithm for complaints management process automatization. 2019 60th International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS), IEEE.

[9] HaCohen-Kerner, Y., et al. (2019). "Automatic classification of complaint letters according to service provider categories." **56**(6): 102102.

[10] Hasan, T., et al. (2020). Machine learning based automatic classification of customer sentiment. 2020 23rd International Conference on Computer and Information Technology (ICCIT), IEEE.

[11] Kadhim, A. I. J. A. I. R. (2019). "Survey on supervised machine learning techniques for automatic text classification." **52**(1): 273-292.

[12] Kalra, V. and R. Aggarwal (2017). Importance of Text Data Preprocessing & Implementation in RapidMiner. ICITKM.

[13] Li, L. and W. J. T. v. Li (2019). "Naive Bayesian automatic classification of railway service complaint text based on eigenvalue extraction." **26**(3): 778-785.

[14] Miner, G. (2012). Practical text mining and statistical analysis for non-structured text data applications, Academic Press.

[15] Muraina, I. (2022). Ideal dataset splitting ratios in machine learning algorithms: general concerns for data scientists and data analysts. 7th International Mardin Artuklu Scientific Research Conference.

[16] Naseem, U., et al. (2021). "A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models." **20**(5): 1-35.

[17] Sen, P. C., et al. (2020). Supervised classification algorithms in machine learning: A survey and review. Emerging technology in modelling and graphics, Springer**:** 99-111.

[18] Tharwat, A. J. A. C. and Informatics (2020). "Classification assessment methods."

[19] Tufail, A. B., et al. (2020). "Binary classification of Alzheimer's disease using sMRI imaging modality and deep learning." **33**: 1073-1090.

[20] Xu, S., et al. (2017). Bayesian multinomial Naïve Bayes classifier to text classification. Advanced Multimedia and Ubiquitous Engineering: MUE/FutureTech 2017 11, Springer.