UNIVERSITY OF MACEDONIA

AIDA105 Machine Learning and Computer Vision

Homework 1- Supervised Machine Learning (Classification)

Dimitrios Mitras aid24005, Konstantinos Bougioukas aid24011

1. Introduction

The two basic categories of machine learning are supervised and unsupervised machine learning. In supervised learning, the model is trained on a labelled dataset, meaning that the input data is paired with corresponding output values/labels. Supervised learning problems are applicable to a variety of situations and data types. Common tasks in supervised learning include regression modelling for continuous output and classification when the output is discrete values or classes.
In this homework, our focus is on the case of binary classification modelling, where the output is a binary variable. Specifically, our aim is to compare and evaluate the performance of different classification methods in both unbalanced and balanced training datasets.

2. Problem description

We want to train a model that best predicts whether a company is healthy or will go bankrupt based on performance and activity indicators of the company. This is a binary classification problem.

2. Methods

*2.1. **Dataset description***

The performance of the classification algorithms will be studied in a dataset named "Dataset2Use_Assignment1" that contains *10716* observations and 13 columns. We are interested in the performance indicators of the companies (columns from A to H) and three binary indicators of activities (columns from I to K) which are referred to as features, input variables, or predictors of the dataset. The target variable for the models is the status of the company (healthy is coded with 1, and bankrupt is coded with 2). From an exploratory analysis of the data we observe in **Fig. 1** that there is severe class imbalance meaning that there is a significant difference in the frequency of companies between the two classes (healthy vs bankrupt).

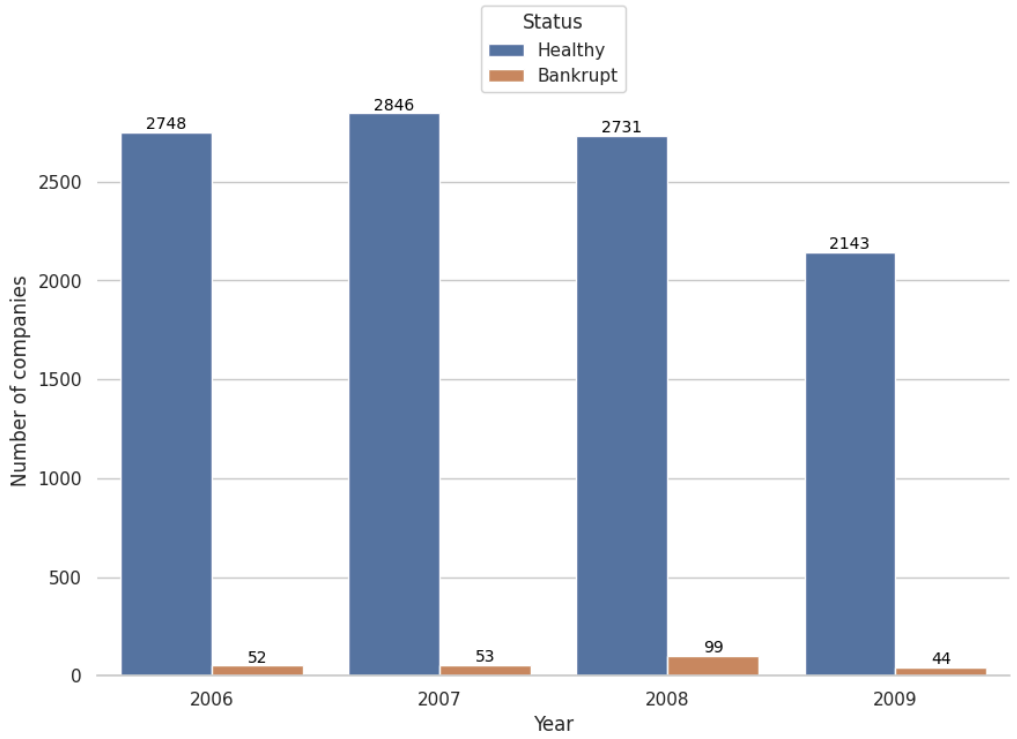

Figure 1. The barplot shows the class imbalance of the data between the years 2006-2009. The number of bankrupt companies is much less than the healthy companies.

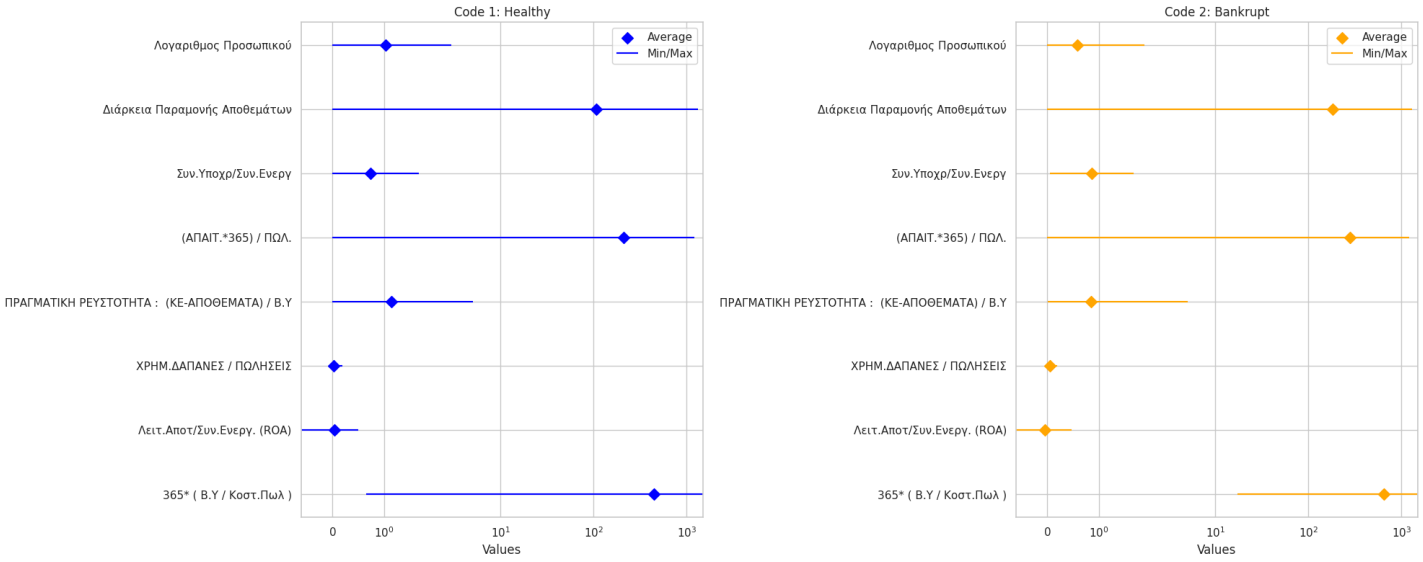

Figure 2. Plots of the performance indicators of the healthy (left-hand side plot) and bankrupt (right-hand side plot) companies. The dots represent the average value while the horizontal lines above or below the average represent the range of values (Min/Max) (Note that the x-axis is on a symmetrical log scale).

2.2. **Methodology for *training and testing the models***

After scaling all measured features using the normalisation technique (values [0-1]), we used a stratified split of the data set into 4-folds, which is particularly useful when dealing with imbalanced datasets. The goal of stratified sampling is to ensure that each fold (subset) of the data maintains the same distribution of classes as the entire dataset. Then, we trained and tested the models in this original dataset with the class imbalance. Additionally, in order to achieve a more balanced dataset, we examined the training set. If the distribution showed more than 3 healthy companies for each bankrupt one, we randomly selected as many healthy companies as needed to achieve a 3:1 ratio of healthy to bankrupt companies in the training set. The companies excluded from the training set were transferred to the test set. Then, we trained and tested the models again in this new more balanced data set.

We calculated the average value of the four folds for each performance metric and we presented the comparisons of the models in bar plots and line charts for both unbalanced and balanced data sets. To further explore the relative performance of the different models in the balanced test set, an analysis of variance (ANOVA) was conducted, considering that all the models utilised the data from the same four test folds (dependency of the measurements). In addition to the basic ANOVA results, we conducted pairwise comparisons of F1-Scores, Recall, and Specificity. The Hommel method (Hommel 1988) was employed to adjust the p-values of the comparisons with paired t-tests, ensuring a low familywise type I error. For all the statistical tests the level of significance for a two-tailed test was set α =0.05 .
All parts of the training and testing of the models were performed in Python 3.12. The average performance scores and graphical representation was conducted mainly in Excel. The statistical tests were conducted using R 4.3.0.

2.2. ***Classification methods used in this work***

a. Linear Discriminant Analysis (LDA): The LDA provides a probabilistic framework for classification and is particularly useful in scenarios where we have a dataset with multiple classes and we want to find a linear combination of features that best separates (discriminates) these classes (Rao 1973). The main goal of LDA is to maximise the distance between the means of different classes while minimising the spread or variance within each class. LDA assumes that the features are continuous and normally distributed, and that classes have the same covariance matrix. Note that we have a mix of continuous and binary variables as input. However, LDA is quite robust to the violation of the assumptions.

b. Logistic Regression: Logistic Regression is specifically designed for binary classification. It uses the logistic function (sigmoid function) to map the linear

combination of input features to a range between 0 and 1 (ref). Logistic regression is a direct probability model and doesn't require Bayes' rule, unlike discriminant analysis, for the conversion of outcomes into probabilities.

c. Decision Trees: A decision tree is a flowchart-like tree structure that operates by recursively partitioning the input data into subsets based on the values of different features. At each internal node of the tree, a decision is made regarding which feature to split on, and these decisions lead to the creation of branches that ultimately lead to the assignment of a class label at the leaf nodes.

d. Random Forests: A Random Forest combines the output of multiple decision trees (ensemble of trees) to reach a single result, usually trained with the bagging method. The general idea of the bagging method is that a combination of learning models increases the overall result.

e. K-Nearest Neighbours: The K-Nearest Neighbor (KNN) algorithm is a non-parametric supervised machine learning model. This is a very simple concept of identifying K-nearest neighbours for a given data point. The assumption is that similar items are closer together. Here the idea of close is a distance measure between two points. By finding the largest class of items that are close to the test data, we can conclude that test data belongs to that class.

f. Naïve Bayes: It is a probabilistic machine learning algorithm based on Bayes' theorem. It is considered "naïve" because it makes the assumptions that the features used to describe an observation are normally distributed and conditionally independent, given the class label. This method can be extremely fast relative to other classification algorithms.

g. Support Vector Machines: In machine learning, SVM is used to classify data by finding the optimal decision boundary, a hyperplane, that maximally separates different classes. This method offers valuable theoretical understanding of the two-class classification process, especially when assuming the linear separability of the data. If the class boundaries are not linearly separable, SVM uses a technique where the dimensional representation can be converted to higher dimension data. In this higher dimensional representation, the class boundaries become linearly separable, and SVM classifiers can provide a boundary.

h. Gradient Boosting Classifier: The idea behind Gradient Boosting is to build trees sequentially, with each tree correcting the errors of the previous ones. The process involves minimising a loss function (often a negative log-likelihood), and the learning is done by optimising the parameters of each weak learner (tree).

*2.3. **Performance Metrics***

In order to objectively evaluate our results, five different metrics are considered based on the confusion matrix (**Table 1**): Accuracy, Precision, Recall, the F1 score, and the AUC ROC.

Accuracy (ACC) is defined as: ACC = (TP + TN) / (TP + FP + TN + FN). Percentage of correct classification for all classes.

Precision (Pr) is defined as: Pr = TP / (TP + FP). It calculates how many correct positive predictions we have to the total predicted positive observations (also known as Positive Predictive Value - PPV).

Recall (Re) is defined as: Re = TP / (TP + FN). It represents the ratio of correctly predicted positive observations to the total actual positive observations. (also known as Sensitivity and True Positive Rate)

F1 score is defined as: F1 = (Pr × Re)/(Pr + Re). It calculates the weighted harmonic mean of precision and recall.

Specificity (True Negative Rate) is defined as: TNR = TN / (TN + FP). It measures the proportion of actual negative instances that are correctly identified as negative by the classification model.

Table 1. Confusion matrix.

		Predicted	
		healthy	**bankrupt**
Actual	**healthy**	healthy companies classified as healthy (True Negative-TN)	healthy companies classified as bankrupt (False Positive-FP)
	bankrupt	bankrupt companies classified as healthy (False Negative-FN)	bankrupt companies classified as bankrupt (True Positive-TP)

The **A**rea **U**nder the **R**eceiver **O**perating **C**haracteristic **C**urve (**AUC-ROC**) is a performance metric used to evaluate the discriminatory ability of a binary classification model. The AUC value ranges from 0 to 1, where:

- AUC = 0.5 indicates that the model's performance is equivalent to random chance.
- AUC > 0.5 suggests better-than-random performance.
- AUC = 1 signifies perfect classification.

3. Results

3.1. Difference between average performance scores (F1 , Recall) in Unbalanced vs Balanced datasets.

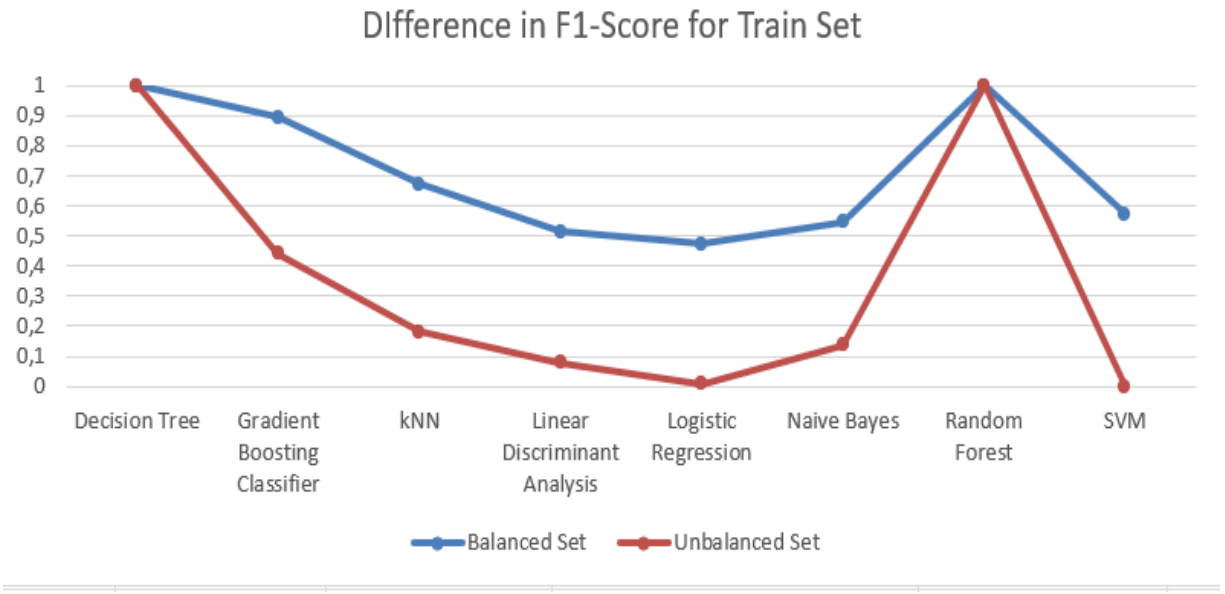

Figure 3. The Line graph shows the difference in average F1 - score for train sets. The blues line represents the balanced set and the red line represents the unbalanced set.

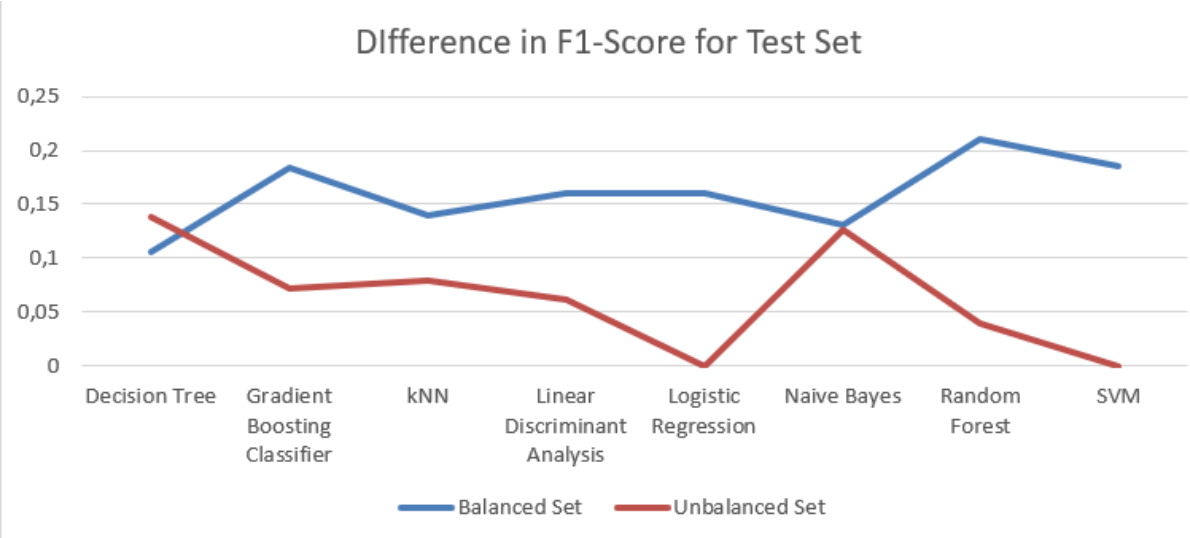

Figure 4. The Line graph depicts the difference in average F1 - score for test sets. The blues line represents the balanced set and the red line represents the unbalanced set.

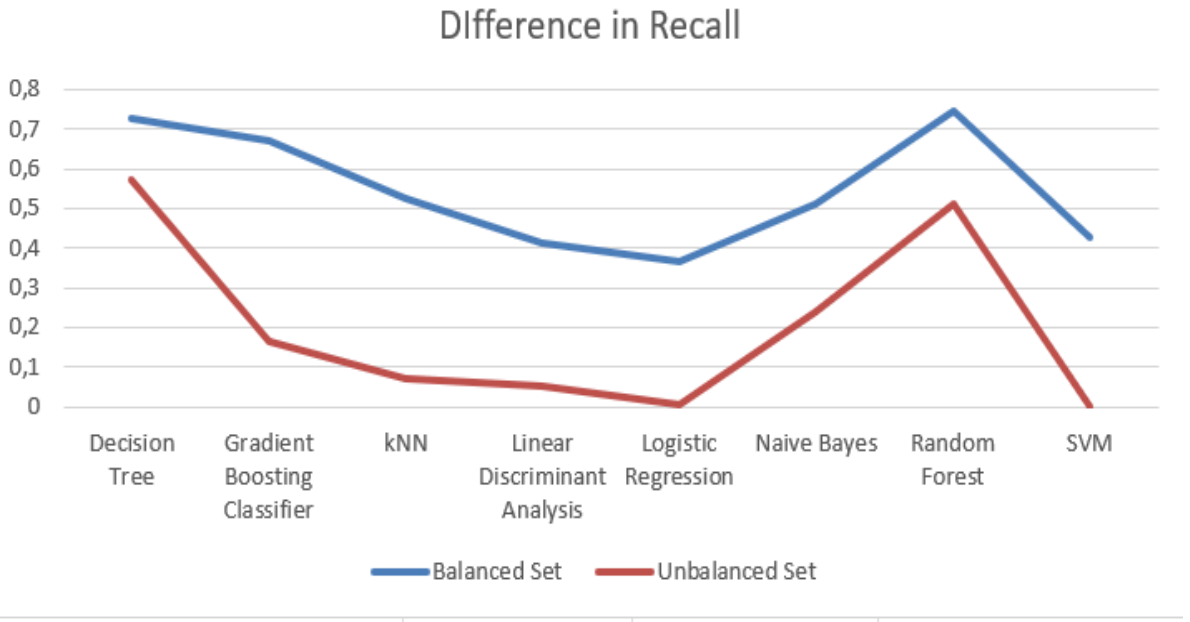

Figure 5. The Line graph depicts the difference in average Recall without filtering for train or test data set. The blues line represents the balanced set and the red line represents the unbalanced set.

This samples scores implies that it is crucial to consider the balance of the dataset when choosing a machine learning algorithm. An unbalanced dataset, where the number of

samples in each category differs significantly, can impact the performance of machine learning models. The main effects include:

1. **Overfitting to the Dominant Class**: When one type of sample is much more prevalent than others, the model may tend to focus on predicting that dominant class during training, resulting in a model that struggles to effectively handle the less frequent classes.
2. **Imbalance in Evaluation Metrics**: Classic metrics like accuracy,specificity may provide misleadingly high results. For example, a model that consistently predicts the dominant class might have high accuracy but fail to recognize the less frequent categories.

If the dataset is unbalanced, you may need to select an algorithm that is more robust to the uneven distribution of labels.

Hence, later in the comparison of the data, the balanced dataset will be utilised.

3.2. **Difference between average performance scores in Test vs Train datasets**

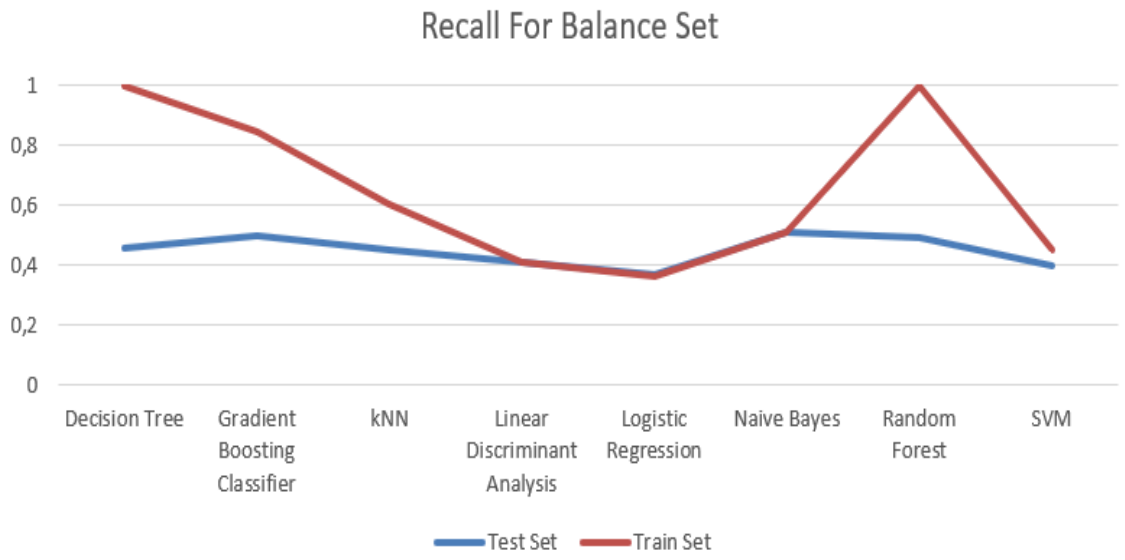

Figure 6. The Line graph shows the difference in average Recall for Balanced dataset. The blues line represents the Test set and the red line represents the Train set.

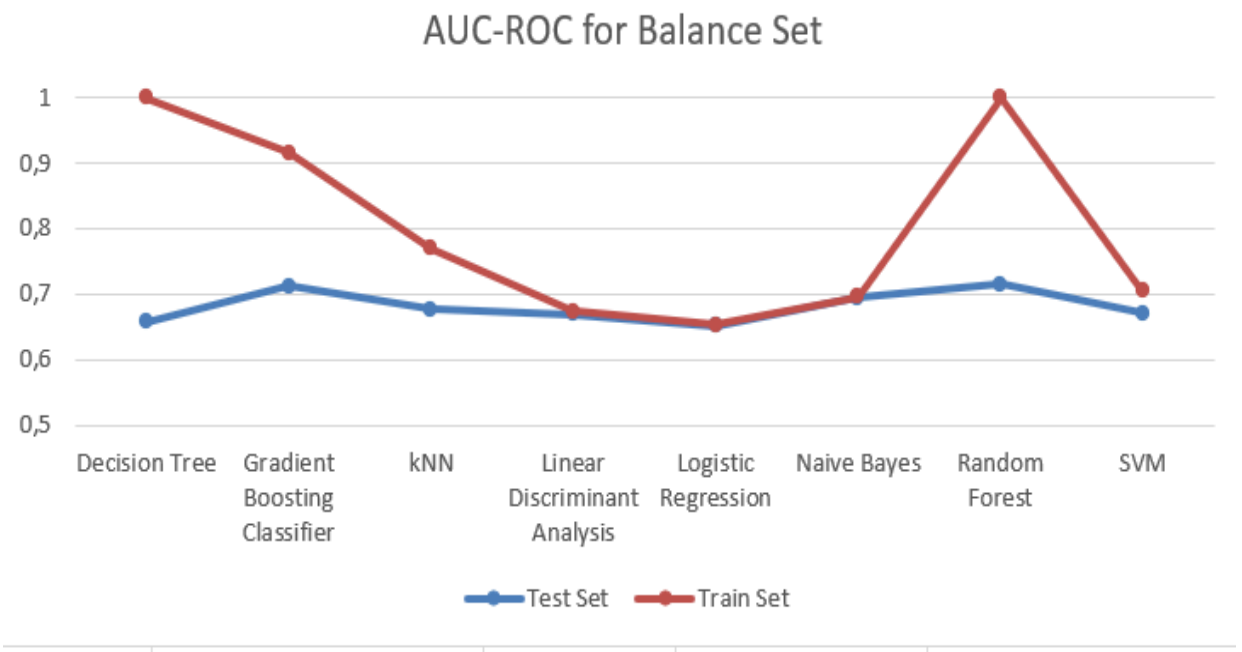

Figure 7. The Line graph shows the difference in average AUC-ROC for Balanced dataset. The blues line represents the Test set and the red line represents the Train set.

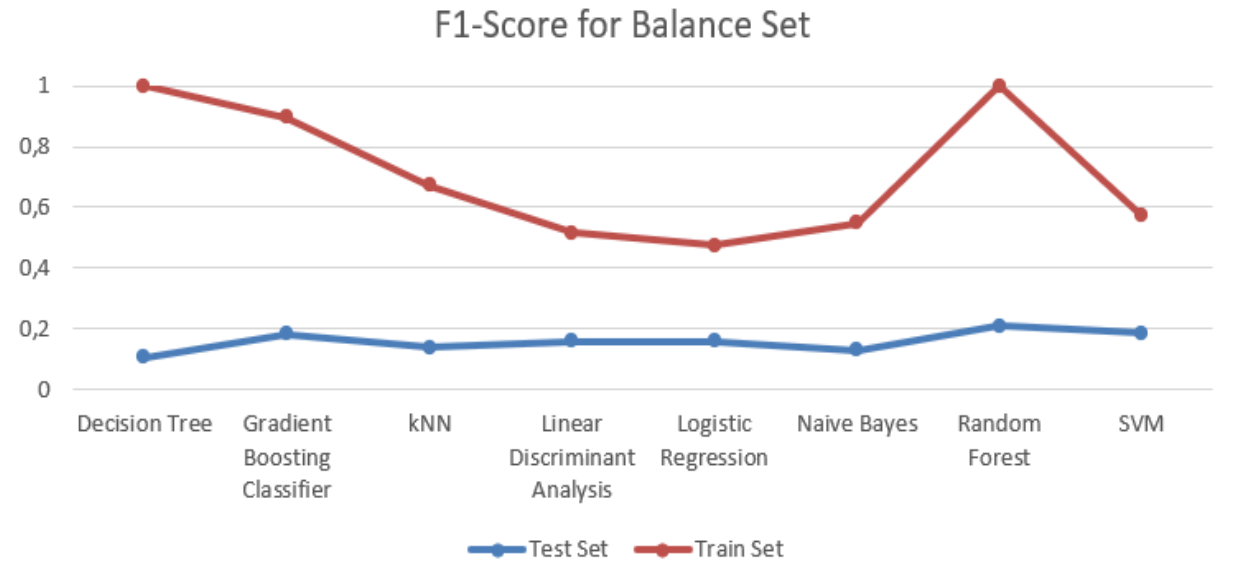

Figure 8. The Line graph shows the difference in average F1-Score for Balanced dataset. The blues line represents the Test set and the red line represents the Train set.

The difference in performance between the training and testing datasets by mentioning that the model exhibits superior metrics on the training set compared to the testing set. Model, generally, performs exceptionally well on the data it was trained on but struggles to generalize to new, unseen data. This observation prompts the need for

careful model evaluation and potential adjustments to enhance generalization performance.

3.3. **Average** *performance scores of methods in the train and test sets for unbalanced and balanced data sets*

The previous introductory sections (3.1 , 3.2) provide some general information about the differences of the Train-Test and Balanced- Unbalanced datasets.
The following graphs contain metrics comparing the classification methods. Initially, the metrics of all calculated scores are presented. The results of each metric concern the average value of the 4 folds

Recall

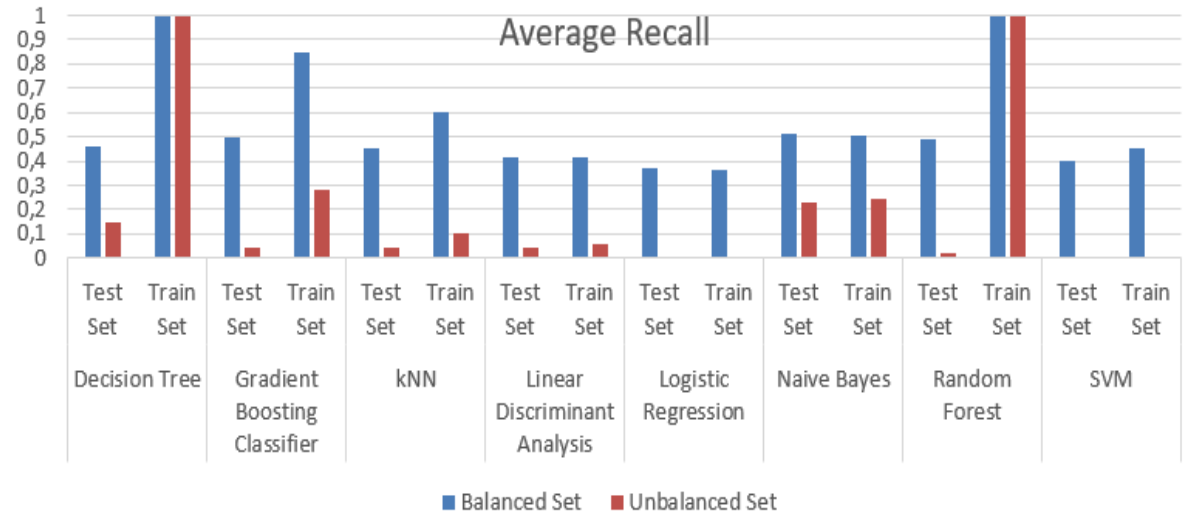

Figure 9. The Bar plot shows the average Recall for every classification method. The blues bar represents the Balanced set and the red line represents the Unbalanced set.

Precision

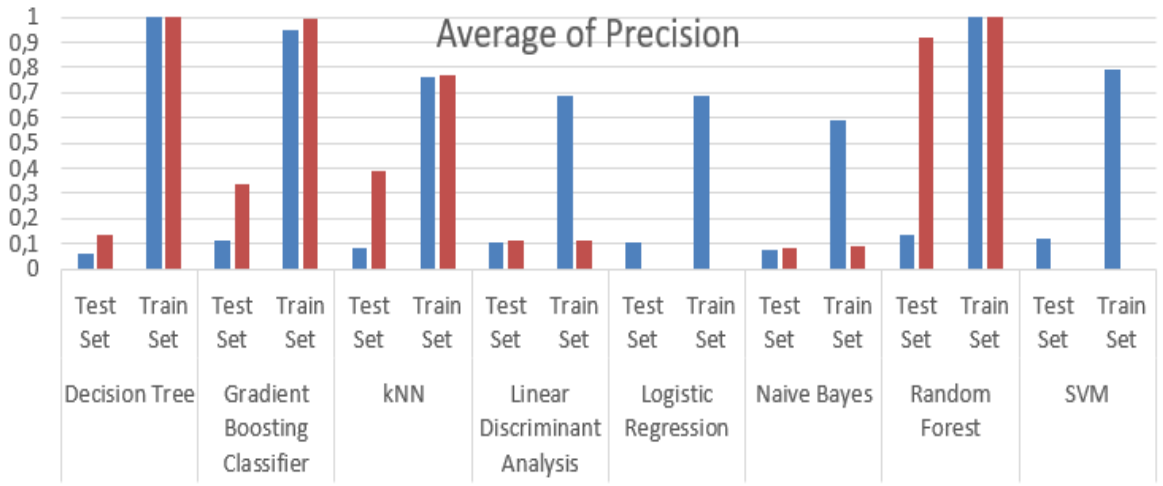

Figure 10. The Bar plot shows the average Precision for every classification method. The blues bar represents the Balanced set and the red line represents the Unbalanced set.

F1-Score

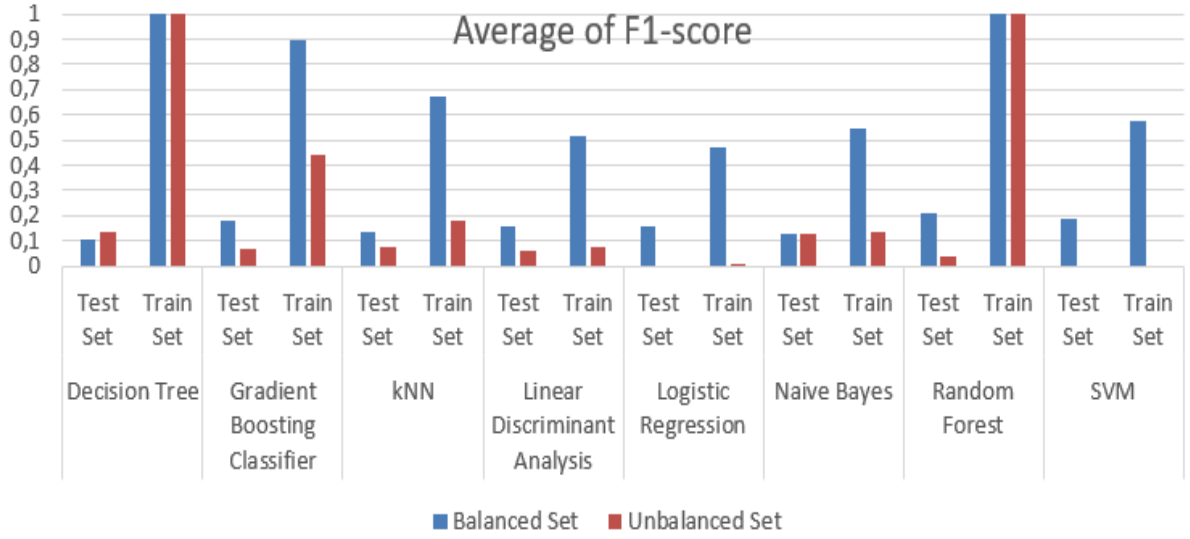

Figure 11. The Bar plot shows the average F1-Score for every classification method. The blues bar represents the Balanced set and the red line represents the Unbalanced set.

ROC-AUC

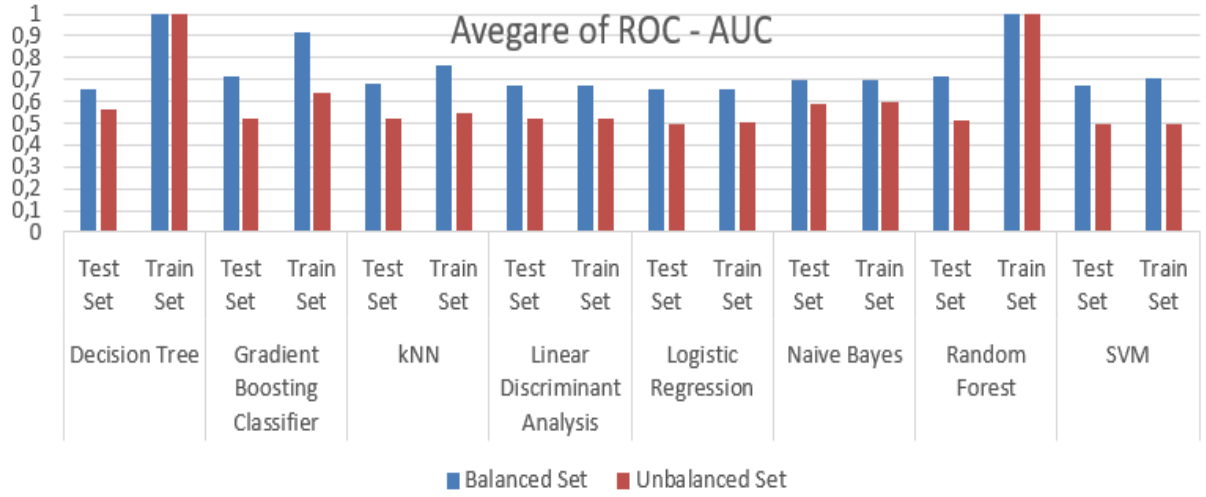

Figure 12. The Bar plot shows the average ROC-AUC for every classification method. The blues bar represents the Balanced set and the red line represents the Unbalanced set.

Specificity

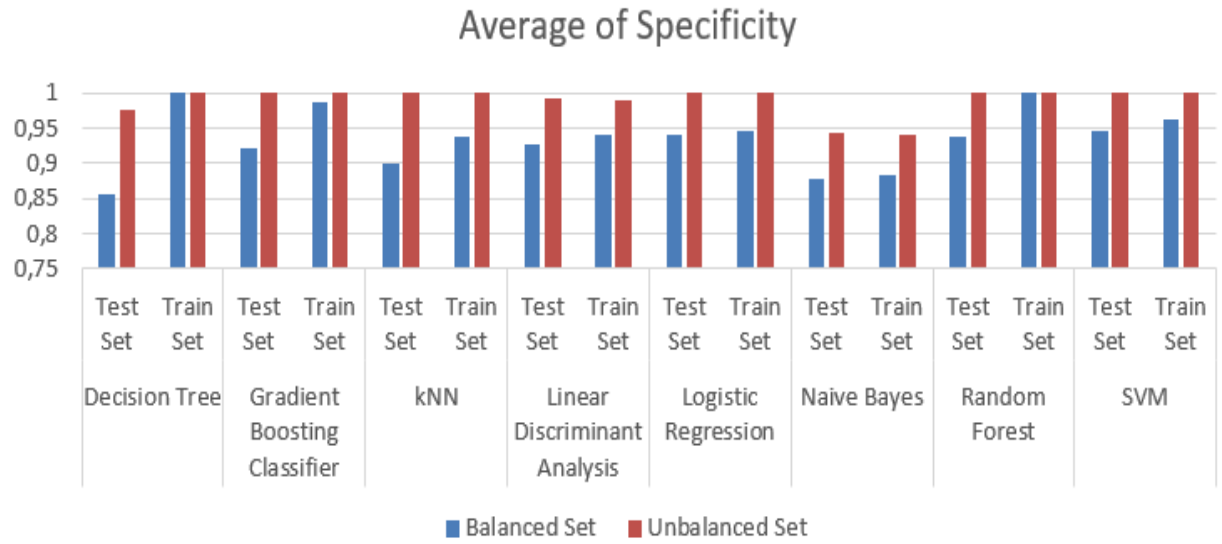

Figure 13. This Bar plot shows the average Specificity for every classification method. The blues bar represents the Balanced set and the red like represents the Unbalanced set.

The reason the Specificity results on the unbalanced set are so good is due to the algorithm identifying healthy companies , which are significantly more numerous compared to the bankrupt ones. As mentioned earlier, specificity may provide

misleadingly high results. For example, a model that consistently predicts the dominant class might have high specificity but fail to recognize the less frequent categories.

In general, the positions presented in sections 3.1 and 3.2 regarding balanced and unbalanced datasets, as well as train-test sets, are reinforced with the support of the graphs.

*3.4. **Comparison of Average performance scores of methods in the balanced test sets***

The following plots summarized the 6 metrics for the average values for each classification method of the balanced test set.

Recall-Specificity

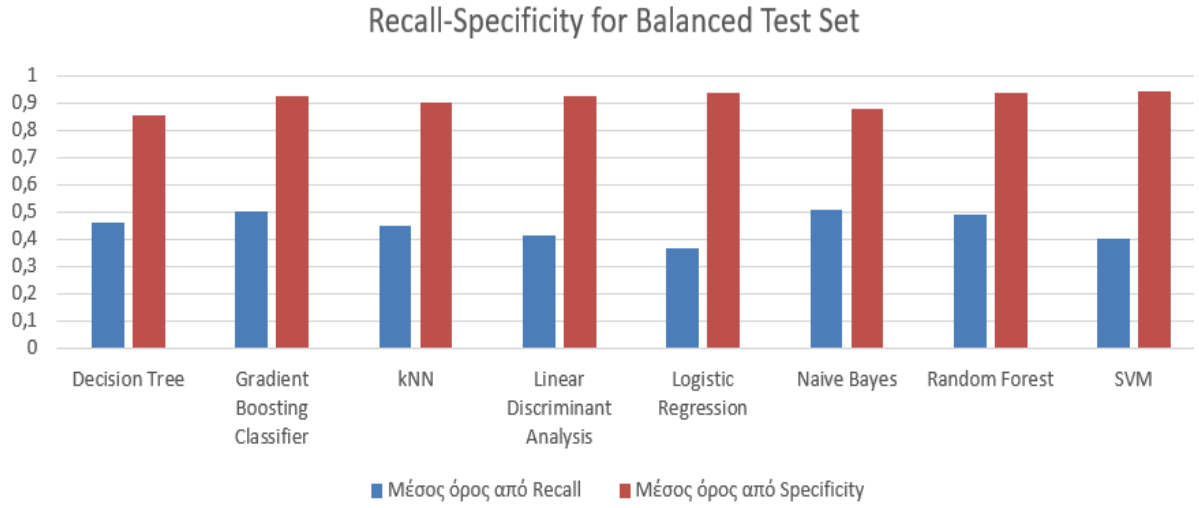

Figure 14. This Bar plot shows the average Specificity and average Recall for every classification method in the Balanced Test Set

ROC_AUC - Accuracy

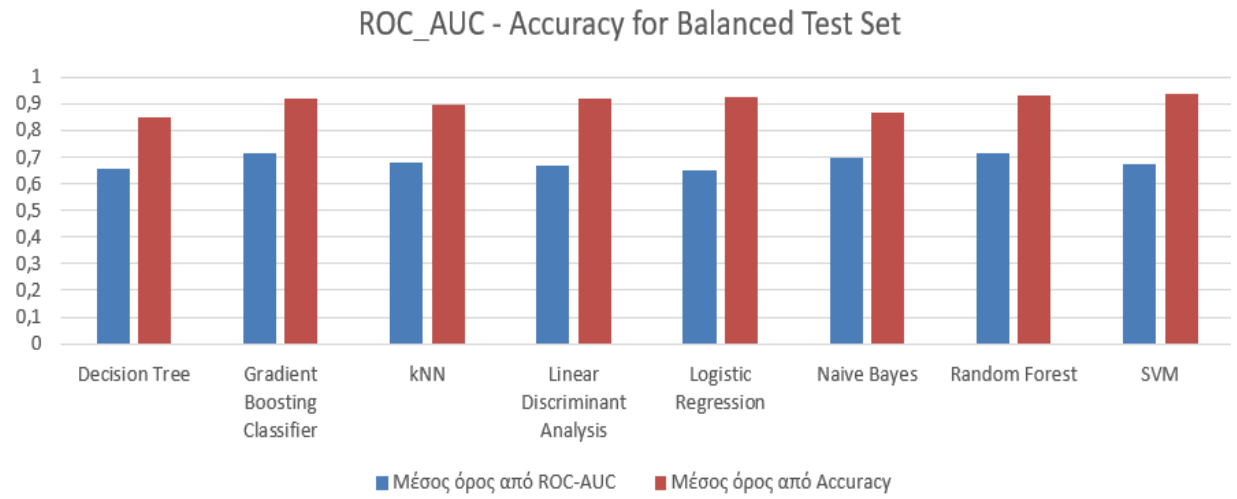

Figure 15. This Bar plot shows the average ROC_AUC and average Accuracy for every classification method in the Balanced Test Set

F1_Score - Precision

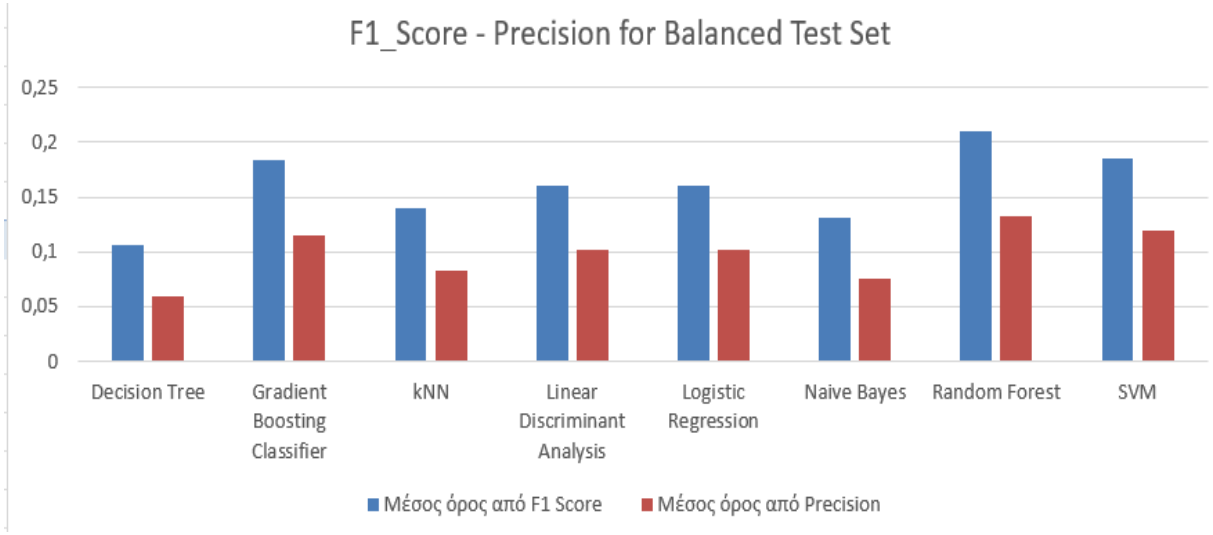

Figure 16. This Bar plot shows the average F1-Score and average Precision for every classification method in the Balanced Test Set

3.5. Exploring Models that Satisfy Dual Performance Constraints

The search is underway to identify a model that satisfies the following two conditions:

1. The model must successfully identify at least 60% of the companies that will go bankrupt.
2. The model must successfully identify at least 70% of the companies that will **not** go bankrupt.

The selected method should meet both constraints. The above conditions correspond to Recall and Specificity, respectively. As seen in the previous analysis of the average

performance metrics for Recall and Specificity (Figure 14), the constraints for recall are not met. Nevertheless, a closer examination of each fold reveals that, for a specific fold of the Naive Bayes, both constraints are achieved. In the following plots, these 2 metric techniques are presented for each classification method in each fold, focusing, always, on the balanced test set.

Recall

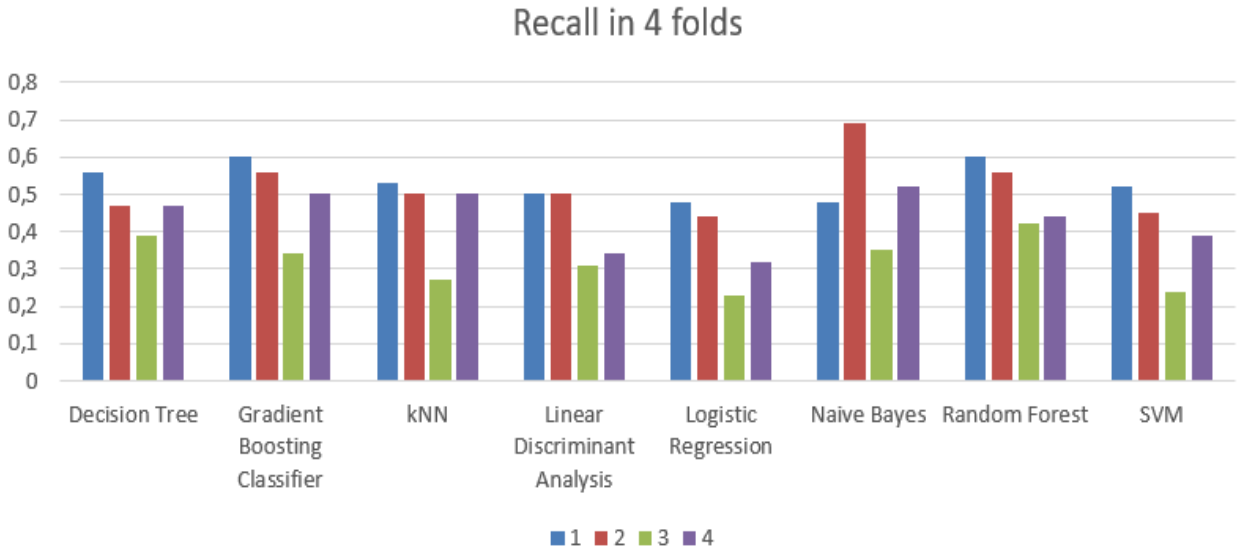

Figure 17. This Bar plot shows the Recall for every classification method for every coloured fold in the Balanced Test Set.

Specificity

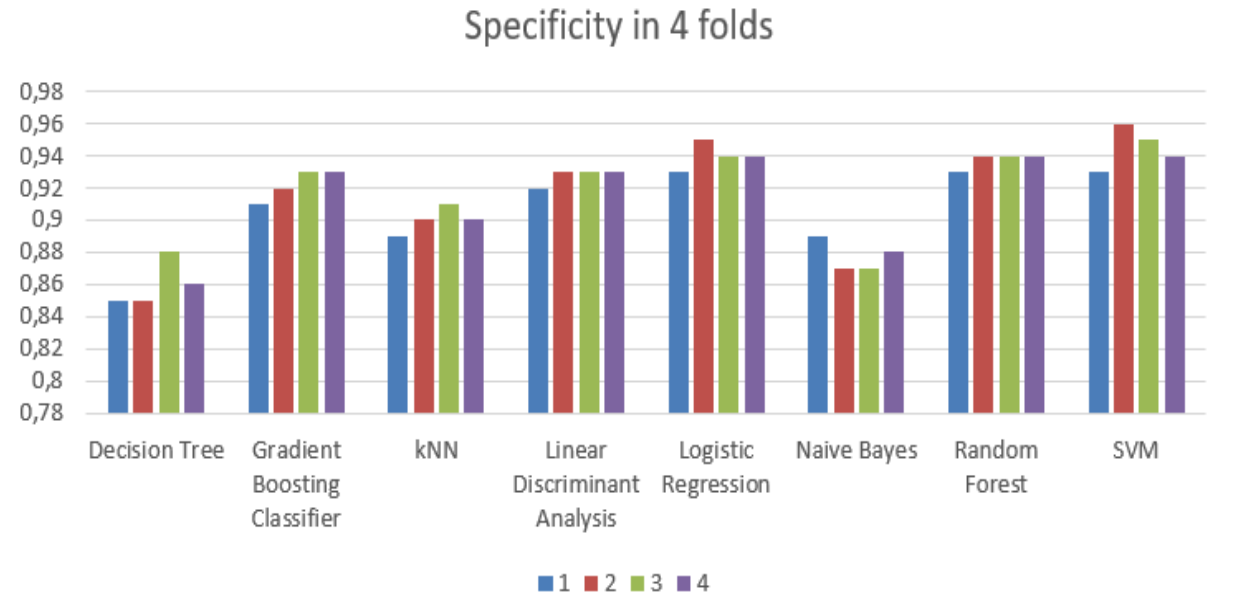

Figure 18. This Bar plot shows the Specificity for every classification method for every coloured fold in the Balanced Test Set.

*3.6. **Comparison of performance scores and selection of the best classification methods***

The ensuing chart elucidates the cumulative 4-fold metrics of Recall and Specificity across all classification techniques.

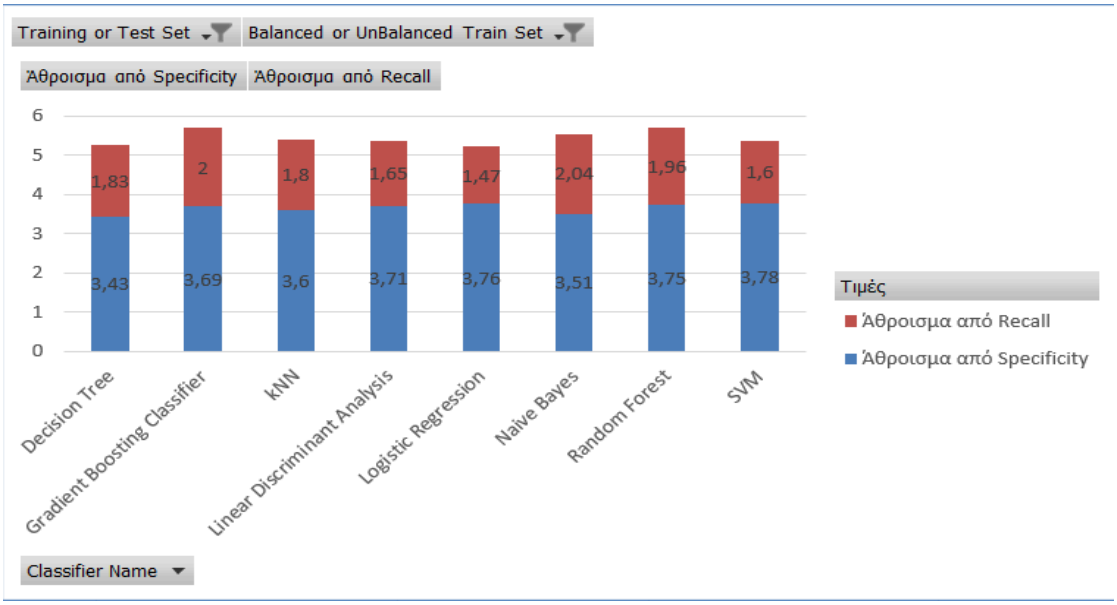

Figure 19. This Bar plot shows the sum of Specificity+Recall for all folds in every classification method.

Considering Figure19, Figure 17, Figure 18, Figure 14, the most robust classification methods emerge as the Gradient Boosting Classifier, Naive Bayes, and Random Forest. In instances where our sole focus is on recall and gauging the likelihood of a company going bankrupt, our preference leans towards Naive Bayes. It satisfies the previously established conditions within a fold.

Nevertheless, Naive Bayes exhibits relatively subpar outcomes in the Specificity metric, indicating its limited ability to predict the status of healthy companies. The other two methods showcase comparatively superior results in Specificity.

However, it is important to note that Naive Bayes has limitations and is less complex than the Gradient Boosting Classifier and Random Forest. As for the Random Forest and Gradient Boosting Classifier, these algorithms have the capability to handle more complexity in the data and adapt better to more intricate problems.

The fact that these two methods are the best for these two metrics(Figure 19), is confirmed. So, the final decision, between these two classification methods, will be made based on the F1-Score performance.. The best-performing model, considering the F1 Score is the **Random Forest Classifier** (Figure 16).

4. Statistical Performance

All the ANOVA results for F1-Score, Recall and Specificity metrics were significant ($p < 0.001$, $p = 0.004$, $p < 0.001$, respectively) (**Table 2**).

Table 2. ANOVA results for the F1-Score, Recall, and Specificity.

Metric	df1	df2	F	p
F1-Score	7	21	12.415	<0.001
Recall	7	21	4.31	0.004
Specificity	7	21	54.197	<0.001

df, degrees of freedom , F-statistic

In **Figure 20** we observe that the Random forest seems to perform better on F1-Score than the other algorithms. Additionally, in **Table 3** we present the results of paired t-test comparisons for all the methods. We included both the p-value and the adjusted p-value. We found 11 important comparisons ($p<0.05$); note that Gradient Boosting Classifier Vs Random Forest ($p=0.032$). However, significant differences in F1-Score after adjustment of the p-values were kept for the following two comparisons: the Gradient Boosting Classifier Vs Naive Bayes ($p.adj = 0.04$) and Random Forest Vs Linear Discriminant Analysis ($p.adj = 0.009$) .
Note that we have only four measurements (because we have four folds) so the power of these statistical tests is relatively low (underpower test). Additionally, the p-values become even smaller after the adjustment of the p-value for the number of comparisons (23 pairwise comparisons).

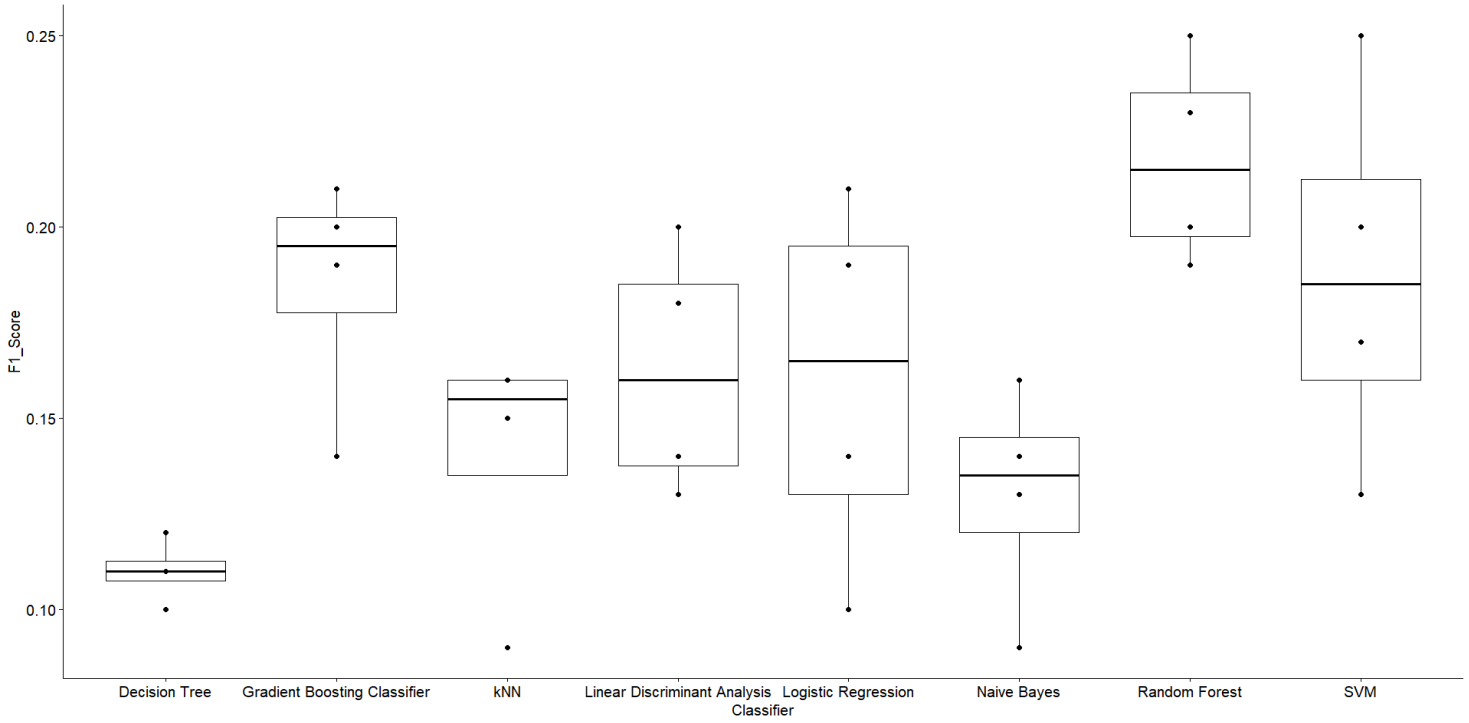

Figure 20. Boxplots of the F1-Score for the different algorithms in the balanced test data.

Table 3. Comparisons of F1-Score for all the methods in the balanced test data.

.y.	Method 1	Method 2	n1	n2	statistic	df	p	p.adj	p.adj.signif
F1_Score	Decision Tree	Gradient Boosting Classifier	4	4	-4.523	3.000	**0.020**	0.283	ns
F1_Score	Decision Tree	kNN	4	4	-1.686	3.000	0.190	0.640	ns
F1_Score	Decision Tree	Linear Discriminant Analysis	4	4	-2.922	3.000	0.061	0.430	ns
F1_Score	Decision Tree	Logistic Regression	4	4	-1.936	3.000	0.148	0.592	ns
F1_Score	Decision Tree	Naive Bayes	4	4	-1.188	3.000	0.320	0.861	ns
F1_Score	Decision Tree	Random Forest	4	4	-6.945	3.000	**0.006**	0.135	ns

F1_Score	Decision Tree	SVM	4	4	-2.850	3.000	0.065	0.448	ns
F1_Score	Gradient Boosting Classifier	kNN	4	4	9.000	3.000	**0.003**	0.072	ns
F1_Score	Gradient Boosting Classifier	Linear Discriminant Analysis	4	4	2.377	3.000	0.098	0.490	ns
F1_Score	Gradient Boosting Classifier	Logistic Regression	4	4	2.100	3.000	0.127	0.533	ns
F1_Score	Gradient Boosting Classifier	Naive Bayes	4	4	11.000	3.000	**0.002**	**0.040**	*
F1_Score	Gradient Boosting Classifier	Random Forest	4	4	-3.806	3.000	**0.032**	0.338	ns
F1_Score	Gradient Boosting Classifier	SVM	4	4	-0.190	3.000	0.861	0.861	ns
F1_Score	kNN	Linear Discriminant Analysis	4	4	-1.567	3.000	0.215	0.645	ns
F1_Score	kNN	Logistic Regression	4	4	-1.265	3.000	0.295	0.861	ns
F1_Score	kNN	Naive Bayes	4	4	1.732	3.000	0.182	0.640	ns
F1_Score	kNN	Random Forest	4	4	-5.894	3.000	**0.010**	0.186	ns
F1_Score	kNN	SVM	4	4	-2.875	3.000	0.064	0.447	ns
F1_Score	Linear Discriminant Analysis	Logistic Regression	4	4	0.264	3.000	0.809	0.861	ns
F1_Score	Linear Discriminant Analysis	Naive Bayes	4	4	2.931	3.000	0.061	0.427	ns
F1_Score	Linear Discriminant Analysis	Random Forest	4	4	-19.053	3.000	**0.000**	**0.009**	**

F1_Score	Linear Discriminant Analysis	SVM	4	4	-2.402	3.000	0.096	0.480	ns
F1_Score	Logistic Regression	Naive Bayes	4	4	2.038	3.000	0.134	0.536	ns
F1_Score	Logistic Regression	Random Forest	4	4	-4.867	3.000	**0.017**	0.261	ns
F1_Score	Logistic Regression	SVM	4	4	-4.371	3.000	**0.022**	0.296	ns
F1_Score	Naive Bayes	Random Forest	4	4	-9.245	3.000	**0.003**	0.067	ns
F1_Score	Naive Bayes	SVM	4	4	-4.176	3.000	**0.025**	0.309	ns
F1_Score	Random Forest	SVM	4	4	2.449	3.000	0.092	0.480	ns

Abbreviations: ns, non-significant; df, degrees of freedom; p, p-value of the statistical test. Bold numbers indicate statistical significant unadjusted/adjusted results.

In **Figure 21** we observe that the Random forest, Gradient Boosting Classifier, and Naive Bayes seem to perform better on Recall among the algorithms explored. Additionally, in **Table 4** we present the results of paired t-test comparisons for all the methods. We included both the p-value and the adjusted p-value. We found 6 important comparisons ($p < 0.05$). However, significant differences in Recall after adjustment of the p-values were kept only for one comparison: the Gradient Boosting Classifier Vs SVM (p.adj = 0.021).

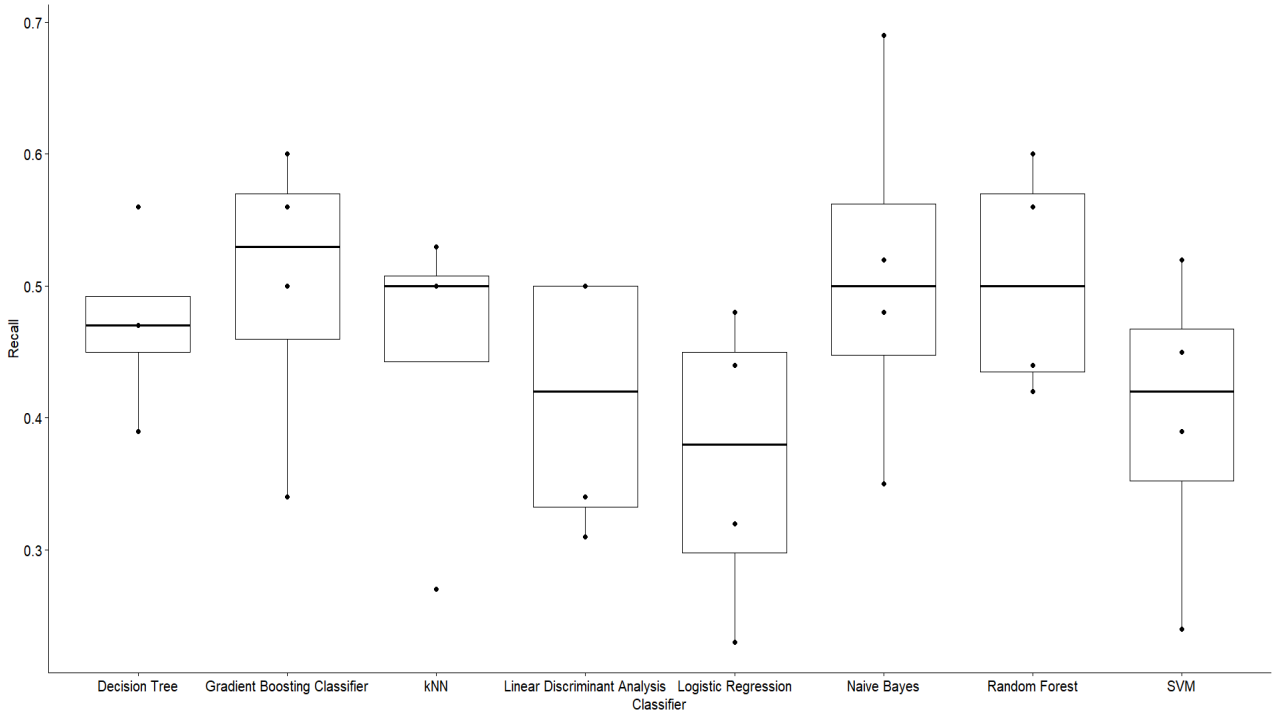

Figure 21. Boxplots of the Recall for the different algorithms in the balanced test data.

Table 4. Comparisons of the Recall for all the methods in the balanced test data.

.y.	Method 1	Method 2	n1	n2	statistic	df	p	p.adj	p.adj.signif
Recall	Decision Tree	Gradient Boosting Classifier	4	4	-0.949	3.000	0.413	0.938	ns
Recall	Decision Tree	kNN	4	4	0.635	3.000	0.571	0.938	ns
Recall	Decision Tree	Linear Discriminant Analysis	4	4	1.796	3.000	0.170	0.938	ns
Recall	Decision Tree	Logistic Regression	4	4	3.422	3.000	**0.042**	0.627	ns
Recall	Decision Tree	Naive Bayes	4	4	-0.563	3.000	0.613	0.938	ns
Recall	Decision Tree	Random Forest	4	4	-1.320	3.000	0.279	0.938	ns

Recall	Decision Tree	SVM	4	4	2.527	3.000	0.086	0.938	ns
Recall	Gradient Boosting Classifier	kNN	4	4	2.970	3.000	0.059	0.767	ns
Recall	Gradient Boosting Classifier	Linear Discriminant Analysis	4	4	3.114	3.000	0.053	0.738	ns
Recall	Gradient Boosting Classifier	Logistic Regression	4	4	8.277	3.000	**0.004**	0.092	ns
Recall	Gradient Boosting Classifier	Naive Bayes	4	4	-0.195	3.000	0.858	0.938	ns
Recall	Gradient Boosting Classifier	Random Forest	4	4	-0.174	3.000	0.873	0.938	ns
Recall	Gradient Boosting Classifier	SVM	4	4	14.142	3.000	**0.001**	**0.021**	*
Recall	kNN	Linear Discriminant Analysis	4	4	0.867	3.000	0.450	0.938	ns
Recall	kNN	Logistic Regression	4	4	2.519	3.000	0.086	0.938	ns
Recall	kNN	Naive Bayes	4	4	-1.180	3.000	0.323	0.938	ns
Recall	kNN	Random Forest	4	4	-1.270	3.000	0.294	0.938	ns
Recall	kNN	SVM	4	4	2.315	3.000	0.104	0.938	ns
Recall	Linear Discriminant Analysis	Logistic Regression	4	4	3.000	3.000	0.058	0.754	ns
Recall	Linear Discriminant Analysis	Naive Bayes	4	4	-1.874	3.000	0.158	0.938	ns
Recall	Linear Discriminant Analysis	Random Forest	4	4	-8.343	3.000	**0.004**	0.090	ns

Recall	Linear Discriminant Analysis	SVM	4	4	0.440	3.000	0.690	0.938	ns
Recall	Logistic Regression	Naive Bayes	4	4	-2.614	3.000	0.079	0.900	ns
Recall	Logistic Regression	Random Forest	4	4	-7.857	3.000	**0.004**	0.107	ns
Recall	Logistic Regression	SVM	4	4	-2.263	3.000	0.109	0.938	ns
Recall	Naive Bayes	Random Forest	4	4	0.084	3.000	0.938	0.938	ns
Recall	Naive Bayes	SVM	4	4	1.910	3.000	0.152	0.938	ns
Recall	Random Forest	SVM	4	4	3.772	3.000	**0.033**	0.522	ns

Abbreviations: ns, non-significant; df, degrees of freedom; p, p-value of the statistical test
Bold numbers indicate statistical significant unadjusted/adjusted results.

In **Figure 22** we observe that the Random forest, SVM, and Logistic regression seem to perform better on Specificity. Additionally, in **Table 5** we present the results of paired t-test comparisons for all the methods. We included both the p-value and the adjusted p-value. We found 13 significant differences in Specificity after adjustment for multiple comparisons (p.adj <0.05), most of them due to the low values of the Decision Tree algorithm .

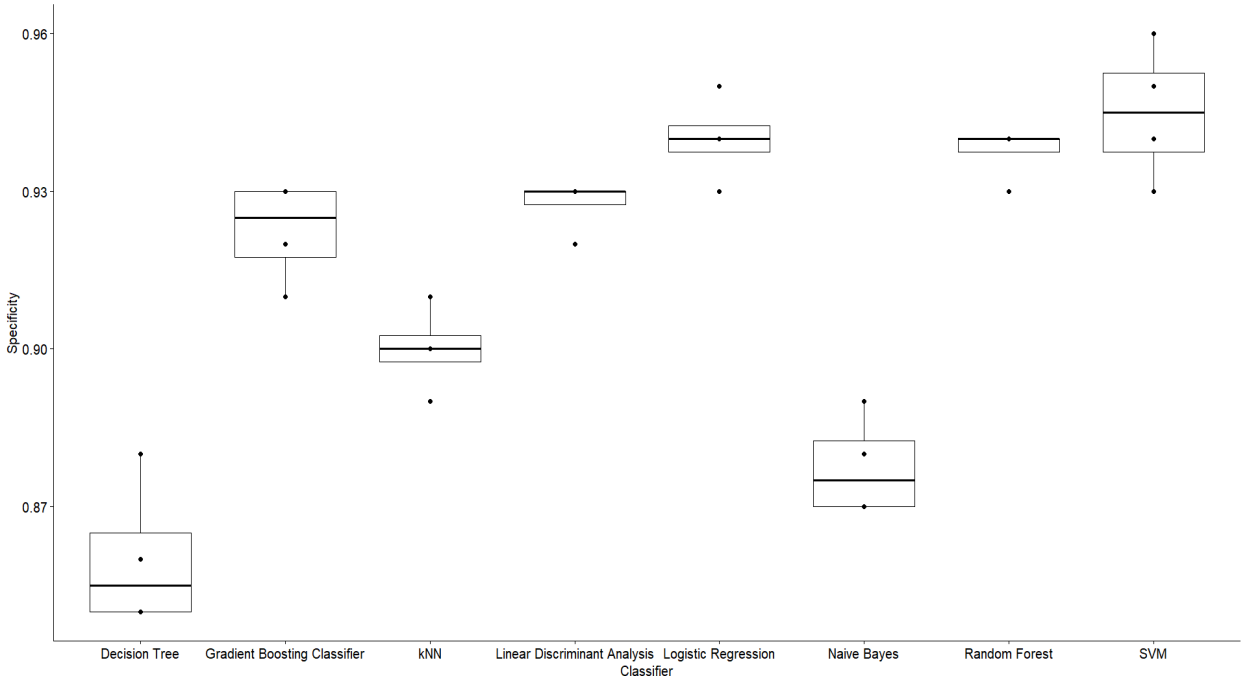

Figure 22. Boxplots of the Specificity for the different algorithms in the balanced test data.

Table 5. Comparisons of the Specificity for all the methods in the balanced test data.

.y.	Method 1	Method 2	n1	n2	statistic	df	p	p.adj	p.adj.signif
Specificity	Decision Tree	Gradient Boosting Classifier	4	4	-13.056	3.000	**0.001**	**0.017**	*
Specificity	Decision Tree	kNN	4	4	-9.798	3.000	**0.002**	**0.034**	*
Specificity	Decision Tree	Linear Discriminant Analysis	4	4	-10.729	3.000	**0.002**	**0.028**	*
Specificity	Decision Tree	Logistic Regression	4	4	-9.798	3.000	**0.002**	**0.034**	*
Specificity	Decision Tree	Naive Bayes	4	4	-1.698	3.000	0.188	0.376	ns
Specificity	Decision Tree	Random Forest	4	4	-12.318	3.000	**0.001**	**0.021**	*
Specificity	Decision Tree	SVM	4	4	-9.815	3.000	**0.002**	**0.034**	*

Specificity	Gradient Boosting Classifier	kNN	4	4	9.000	3.000	**0.003**	**0.041**	*
Specificity	Gradient Boosting Classifier	Linear Discriminant Analysis	4	4	-1.732	3.000	0.182	0.364	ns
Specificity	Gradient Boosting Classifier	Logistic Regression	4	4	-3.656	3.000	0.035	0.248	ns
Specificity	Gradient Boosting Classifier	Naive Bayes	4	4	5.196	3.000	0.014	0.112	ns
Specificity	Gradient Boosting Classifier	Random Forest	4	4	-5.196	3.000	0.014	0.112	ns
Specificity	Gradient Boosting Classifier	SVM	4	4	-3.576	3.000	0.037	0.251	ns
Specificity	kNN	Linear Discriminant Analysis	4	4	-11.000	3.000	**0.002**	**0.026**	*
Specificity	kNN	Logistic Regression	4	4	-9.798	3.000	**0.002**	**0.034**	*
Specificity	kNN	Naive Bayes	4	4	2.635	3.000	0.078	0.287	ns
Specificity	kNN	Random Forest	4	4	-15.000	3.000	**0.001**	**0.013**	*
Specificity	kNN	SVM	4	4	-9.000	3.000	**0.003**	**0.041**	*
Specificity	Linear Discriminant Analysis	Logistic Regression	4	4	-5.000	3.000	0.015	0.123	ns
Specificity	Linear Discriminant Analysis	Naive Bayes	4	4	7.071	3.000	0.006	0.064	ns
Specificity	Linear Discriminant Analysis	Random Forest	4	4	-360,287,970,189,637	3.000	**0.000**	**0.000**	****
Specificity	Linear Discriminant Analysis	SVM	4	4	-3.656	3.000	0.035	0.248	ns

Specificity	Logistic Regression	Naive Bayes	4	4	7.319	3.000	0.005	0.062	ns
Specificity	Logistic Regression	Random Forest	4	4	1.000	3.000	0.391	0.391	ns
Specificity	Logistic Regression	SVM	4	4	-1.732	3.000	0.182	0.364	ns
Specificity	Naive Bayes	Random Forest	4	4	-8.485	3.000	**0.003**	**0.045**	*
Specificity	Naive Bayes	SVM	4	4	-6.088	3.000	0.009	0.089	ns
Specificity	Random Forest	SVM	4	4	-1.567	3.000	0.215	0.391	ns

Abbreviations: ns, non-significant; df, degrees of freedom; p, p-value of the statistical test
Bold numbers indicate statistical significant unadjusted/adjusted results.

5. Conclusions

We have presented a comparative study of classifiers for unbalanced and balanced data sets using a stratified split of the data set into 4-folds. Model, generally, performs exceptionally well on the data it was trained on but struggles to generalise to test data. In the balanced data, no model fulfilled the recall constraint in terms of average values. However, a value exceeding 60% was observed in only one fold of the Naive Bayes model. In contrast, all classifiers satisfied the specificity criterion (>70%).

The optimal classifier for the balanced data seems to be the Random Forest method among the ones explored. Specifically, the comparison of the F1-Score between the Random Forest and Gradient Boosting Classifier is significant ($p<0.05$). However, in this analysis the statistical tests may have not adequate power under a small number of measurements (only four measurements for each method) and the p-values become even smaller after the adjustment for the number of comparisons (23 pairwise comparisons).

References

1. Rao, C. R. (1973). Linear statistical inference and its applications. Wiley Series in Probability and Statistics.
2. G. HOMMEL, A stagewise rejective multiple test procedure based on a modified Bonferroni test, Biometrika, Volume 75, Issue 2, June 1988, Pages 383–386, https://doi.org/10.1093/biomet/75.2.383