MSc in Applied Informatics

Student Name: Tserga
Georgia

Project 1 (Supervised learning – Classification)

Introduction

The purpose of the report is to develop and evaluate different classification models with the aim of identifying companies that will declare bankruptcy, using supervised learning methods.

Classification is a subset of supervised learning where the output variable is categorical. Examples include identifying whether an email is spam or not, classifying images of animals, or diagnosing diseases based on medical records. The model learns to predict discrete class labels based on the input features.

Supervised learning is a type of machine learning where the model is trained on a labeled dataset. This means that each training example is paired with an output label. The goal of the supervised learning algorithm is to learn a mapping from inputs to outputs that can be used to predict the labels for new, unseen data.

The input data we are using is provided by an organization. This data includes company performance indicators, binary activity indicators, the company's status (healthy or bankrupt), and the year to which the indicators relate.

Data Visualization and Analysis

We are visualizing the data to understand the distribution and characteristics of healthy and bankrupt businesses over the years. This step is crucial as it helps in identifying patterns and anomalies in the data, which can guide the feature selection and model training processes.

Visualization 1: Number of Healthy and Bankrupt Businesses Per Year

a. Mapping Business Status:

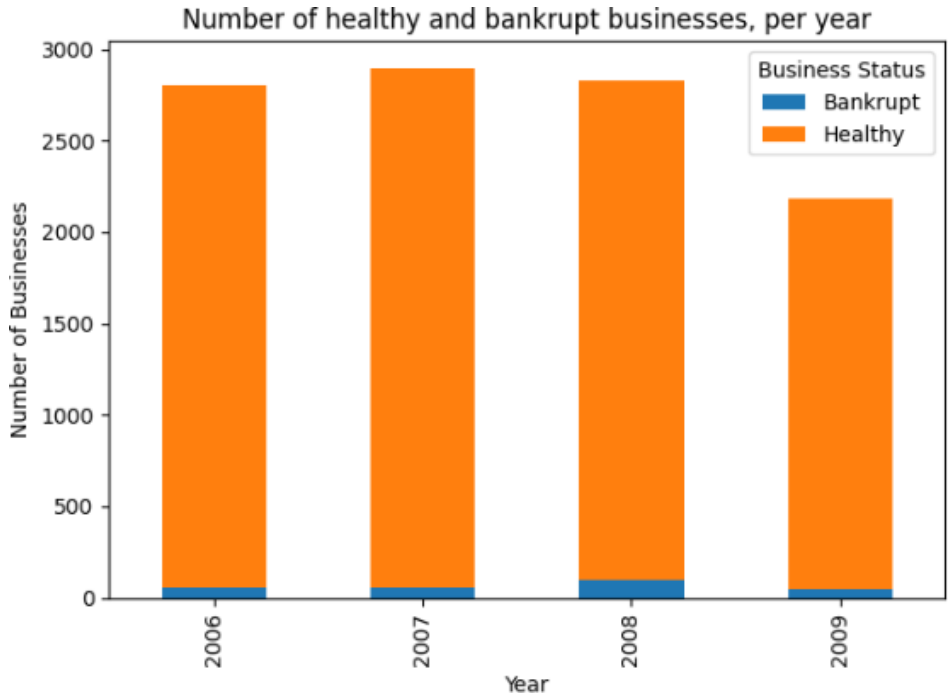
The Business_Status column is created by mapping the values of the ΕΝΔΕΙΞΗ ΑΣΥΝΕΠΕΙΑΣ (=2) (v+1) column to 'Healthy' and 'Bankrupt'. This makes the data more readable and interpretable.

b. Plotting the Data:

The businesses are grouped by year (ΕΤΟΣ) and business status (Business_Status), and the size of each group is calculated.

A stacked bar plot is created, as shown below, to visualize the number of healthy and bankrupt businesses each year.

This visualization provides a clear overview of the distribution of healthy and bankrupt businesses over the years, which is essential for understanding trends and potential seasonal effects in the data. From the following chart, we see that over these years, the number of healthy businesses generally remains high, close to 3000, except in 2009 when there is a noticeable drop. The number of bankrupt businesses is consistently low across all years. The chart highlights that while the majority of businesses remain healthy, there is a slight increase in bankruptcies over time, especially noticeable in the context of the decrease in healthy businesses in 2009.



Visualization 2: Indicator Values for Healthy and Bankrupt Businesses

a. Creating Subplots:

Two subplots are created side by side, with a specified figure size to ensure the plots are large enough for detailed analysis.

b. Box Plots for Healthy Businesses:

A box plot is created for the first 8 indicators of healthy businesses. Box plots are useful for visualizing the distribution, central tendency, and variability of the data.

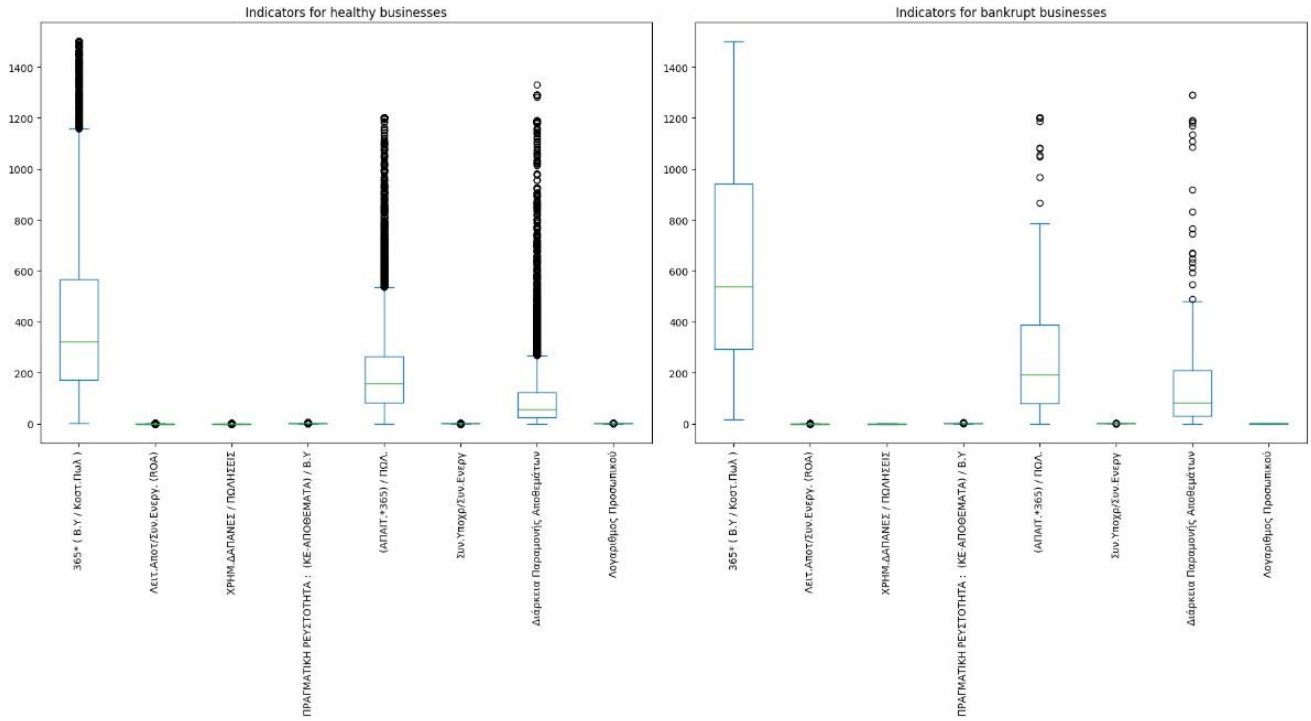
c. Box Plots for Bankrupt Businesses:

Similarly, a box plot is created for the first 8 indicators of bankrupt businesses.

The below box plots allow us to compare the distribution of various indicators between healthy and bankrupt businesses. Differences in these distributions can highlight which indicators are most influential in determining the business status, thus guiding feature selection for the classification model.

The left chart shows box plots for various financial indicators of healthy businesses. Each box plot represents a different financial metric. The first indicator on the left shows a wide range of values, with the interquartile range (IQR) spanning from around 200 to 600 and numerous outliers extending up to around 1400. This indicates significant variability in this metric among healthy businesses. The other indicators display much smaller ranges, with median values close to zero and very tight IQRs, suggesting that these metrics are relatively stable and consistent across healthy businesses. Some indicators, despite having low medians and IQRs, show many outliers, indicating that while most businesses have similar values, a few have significantly higher or lower values.

The right chart shows the same set of financial indicators but for bankrupt businesses. Similar to the healthy businesses, the first indicator on the left shows a large range of values, with the IQR from around 100 to 600 and outliers extending up to approximately 1400. This suggests that this metric varies widely among bankrupt businesses, similar to healthy ones. The subsequent indicators also display tight IQRs and median values close to zero, similar to the healthy businesses. However, the presence of outliers is more pronounced in the bankrupt businesses, suggesting greater variability in these metrics compared to healthy businesses. This increased variability and the presence of numerous outliers in these indicators might reflect the financial instability and diverse conditions leading to bankruptcy.



Data Preprocessing

1. Handling Missing Values

Before proceeding with model training, it's crucial to check for missing values in the dataset. Missing values can significantly impact the performance of machine learning models, so identifying and handling them appropriately is a key step in data preprocessing.

The code checks if there are any missing values in the DataFrame. If there are none, it prints a message indicating that there are no missing values. If there are missing values, it prompts the user to consider handling them appropriately.

```
Combined NaN Report:
                                          NaN Count  NaN Percentage
365* ( Β.Υ / Κοστ.Πωλ )                           0             0.0
Λειτ.Αποτ/Συν.Ενεργ. (ROA)                        0             0.0
ΧΡΗΜ.ΔΑΠΑΝΕΣ / ΠΩΛΗΣΕΙΣ                           0             0.0
 ΠΡΑΓΜΑΤΙΚΗ ΡΕΥΣΤΟΤΗΤΑ :  (ΚΕ-ΑΠΟΘΕΜΑΤΑ) / Β.Υ    0             0.0
(ΑΠΑΙΤ.*365) / ΠΩΛ.                               0             0.0
Συν.Υποχρ/Συν.Ενεργ                               0             0.0
Διάρκεια Παραμονής Αποθεμάτων                     0             0.0
Λογαριθμος Προσωπικού                             0             0.0
ΕΝΔΕΙΞΗ ΕΞΑΓΩΓΩΝ                                  0             0.0
ΕΝΔΕΙΞΗ ΕΙΣΑΓΩΓΩΝ                                 0             0.0
ΕΝΔΕΙΞΗ ΑΝΤΙΠΡΟΣΩΠΕΙΩΝ                            0             0.0
ΕΝΔΕΙΞΗ ΑΣΥΝΕΠΕΙΑΣ (=2) (v+1)                     0             0.0
ΕΤΟΣ                                              0             0.0
Business_Status                                   0             0.0

Good news! There are no missing values in the DataFrame.
```

2. Data Normalization

Normalization is an important step in data preprocessing, especially for machine learning algorithms that rely on distance metrics, such as k-nearest neighbors or neural networks. By scaling the features to a common range, we ensure that each feature contributes equally to the distance calculations.

Below is a visual inspection of the transformed data.

```
   365* ( Β.Υ / Κοστ.Πωλ )  Λειτ.Αποτ/Συν.Ενεργ. (ROA)  \
0                 0.675945                    0.556137
1                 0.374377                    0.636057
2                 0.551084                    0.554176
3                 0.328492                    0.723895
4                 0.066540                    0.708216

   ΧΡΗΜ.ΔΑΠΑΝΕΣ / ΠΩΛΗΣΕΙΣ   ΠΡΑΓΜΑΤΙΚΗ ΡΕΥΣΤΟΤΗΤΑ :  (ΚΕ-ΑΠΟΘΕΜΑΤΑ) / Β.Υ  \
0                 0.192526                                        0.192989
1                 0.078877                                        0.282999
2                 0.467810                                        0.314916
3                 0.002904                                        0.269259
4                 0.000866                                        0.659892

   (ΑΠΑΙΤ.*365) / ΠΩΛ.  Συν.Υποχρ/Συν.Ενεργ  Διάρκεια Παραμονής Αποθεμάτων  \
0            0.473222            0.318549                      0.409274
1            0.279931            0.395393                      0.005969
2            0.292182            0.329722                      0.021471
3            0.345424            0.409265                      0.000914
4            0.098790            0.152183                      0.039296

   Λογαριθμος Προσωπικού  ΕΝΔΕΙΞΗ ΕΞΑΓΩΓΩΝ  ΕΝΔΕΙΞΗ ΕΙΣΑΓΩΓΩΝ  \
0               0.102743               0.0               1.0
1               0.401406               1.0               1.0
2               0.410972               0.0               1.0
3               0.488532               0.0               0.0
4               0.518277               0.0               1.0

   ΕΝΔΕΙΞΗ ΑΝΤΙΠΡΟΣΩΠΕΙΩΝ
0                      0.0
1                      1.0
2                      1.0
3                      0.0
4                      1.0
```

Data Splitting using Stratified K-Fold Cross Validation

In this section, the dataset is split into training and testing sets using Stratified K-Fold Cross Validation. This method is particularly useful for ensuring that each fold of the dataset is representative of the overall distribution of the target variable.

The results for each fold are printed, showing the distribution of classes in both training and testing sets.

```
Fold 1:
Train Index: [    0     1     3 ... 10712 10713 10714]
Test Index: [    2     7    13 ... 10707 10709 10715]
Train set - Healthy: 7851, Bankrupt: 186
Test set - Healthy: 2617, Bankrupt: 62
------------------------------
Fold 2:
Train Index: [    0     1     2 ... 10712 10713 10715]
Test Index: [    4     9    14 ... 10705 10710 10714]
Train set - Healthy: 7851, Bankrupt: 186
Test set - Healthy: 2617, Bankrupt: 62
------------------------------
Fold 3:
Train Index: [    2     4     5 ... 10712 10714 10715]
Test Index: [    0     1     3 ... 10706 10708 10713]
Train set - Healthy: 7851, Bankrupt: 186
Test set - Healthy: 2617, Bankrupt: 62
------------------------------
Fold 4:
Train Index: [    0     1     2 ... 10713 10714 10715]
Test Index: [    5     8    12 ... 10699 10711 10712]
Train set - Healthy: 7851, Bankrupt: 186
Test set - Healthy: 2617, Bankrupt: 62
------------------------------
```

Model Training, Evaluation, and Results Analysis

In this section, various classifiers are trained and evaluated using Stratified K-Fold Cross Validation. The performance metrics such as F1 Score, ROC AUC Score, and Confusion Matrix are calculated and visualized for each fold of the cross-validation. Finally, the results are stored in a DataFrame and saved to a CSV file for further analysis.

1. Definition of Machine Learning Models:

    a. Linear Discriminant Analysis (LDA)
    Linear Discriminant Analysis as its name suggests is a linear model for classification and dimensionality reduction. Most commonly used for feature extraction in pattern classification problems. [4]

    b. Logistic Regression (LR)
    The supervised Machine Learning classification algorithm Logistic Regression is used to predict the likelihood of a target variable. The logistic function, also called the sigmoid function is an S shaped curve that can take any real-valued number and map it into a value between 0 and 1. In general, Logistic Regression refers to binary logistic regression with binary target variables, but it can also predict two additional types of target variables [3].

    c. Decision Tree (DT)
    Decision Tree is a rule-based supervised machine learning algorithm used in both regressions, as well as classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. While the node at the top of the decision tree is the root node [2].

    d. Random Forest (RF)
    Random Forest is an ensemble learning method for classification. It builds decision trees on different samples of the given dataset and takes the average to improve the predictive accuracy of the dataset. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing categorical variables and usually performs better results [1].

    e. k-Nearest Neighbors
    The k-nearest neighbors (KNN) algorithm is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the

grouping of an individual data point. It is one of the popular and simplest classification and regression classifiers used in machine learning today. [2]

f. Naïve Bayes
Naïve Bayes is a simple learning algorithm that utilizes Bayes' rule together with a strong assumption that the attributes are conditionally independent given the class. While this independence assumption is often violated in practice, naïve Bayes nonetheless often delivers competitive classification accuracy. Coupled with its computational efficiency and many other desirable features, this leads to naïve Bayes being widely applied in practice [5]

g. Support Vector Machines (SVM)
Support Vector Machines (SVM) are supervised machine learning algorithms used to classify featured objects. The objective is to find a hyperplane in an n-dimensional feature space that clearly classifies the data points representing objects in the feature space.[6]

h. Multi-layer Perceptron Classifier (MLP) «personal choice»
MLP Classifier is a a modern feedforward artificial neural network, consisting of multiple connected layers -the input layer, output layer and hidden layer. Modern feedforward networks are trained using the backpropagation method. It is frequently used for different machine-learning tasks, including classification and regression.

2. Results Recording:

Results for each fold, including classifier name, fold number, set (train or test), number of samples, number of bankrupt instances, TP, TN, FP, FN, F1 Score, and AUC, are recorded in the results list.

3. Results Storage:

A DataFrame results_df is created from the results list.

Results are saved to a CSV file (/content/drive/MyDrive/Results/unbalancedDataOutcomes.csv) using to_csv.

4. Repeating the above experiment

The above experiment was repeated, but this time, the code checks the class distribution in the training set.

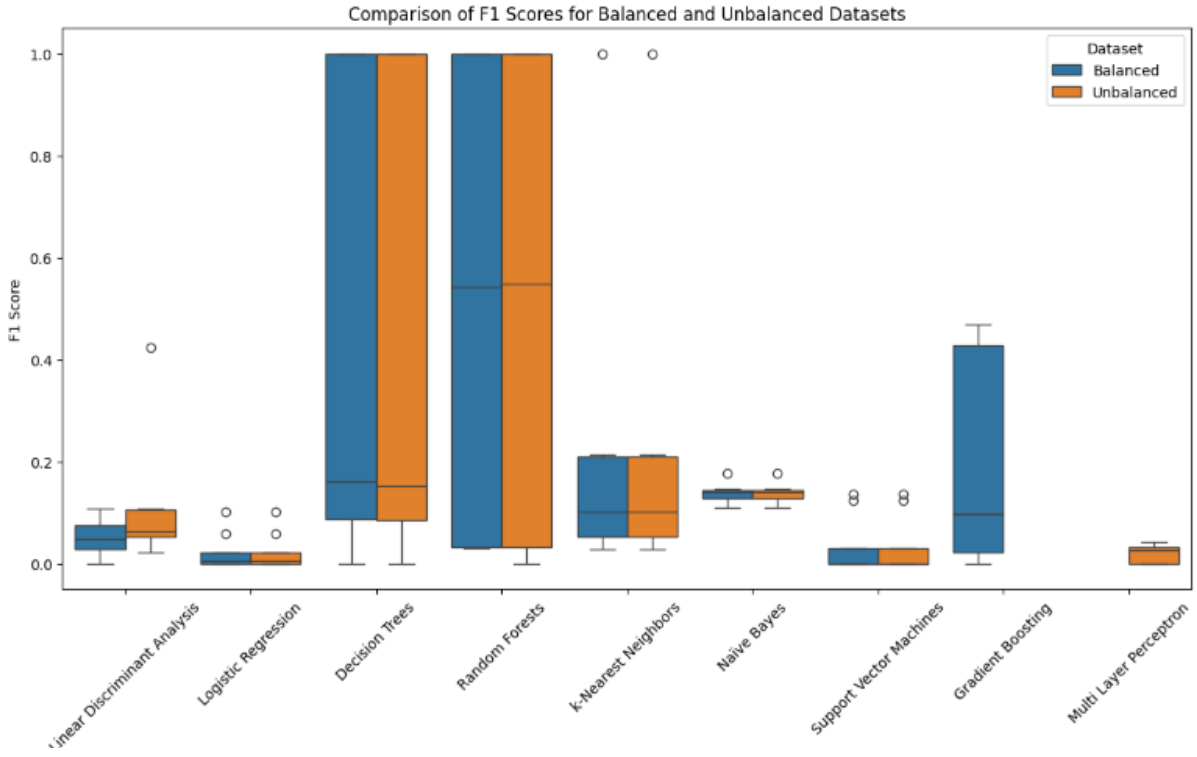
If the ratio of healthy to bankrupt companies is more than 3:1, it performs random undersampling of healthy companies to ensure a balanced class distribution.

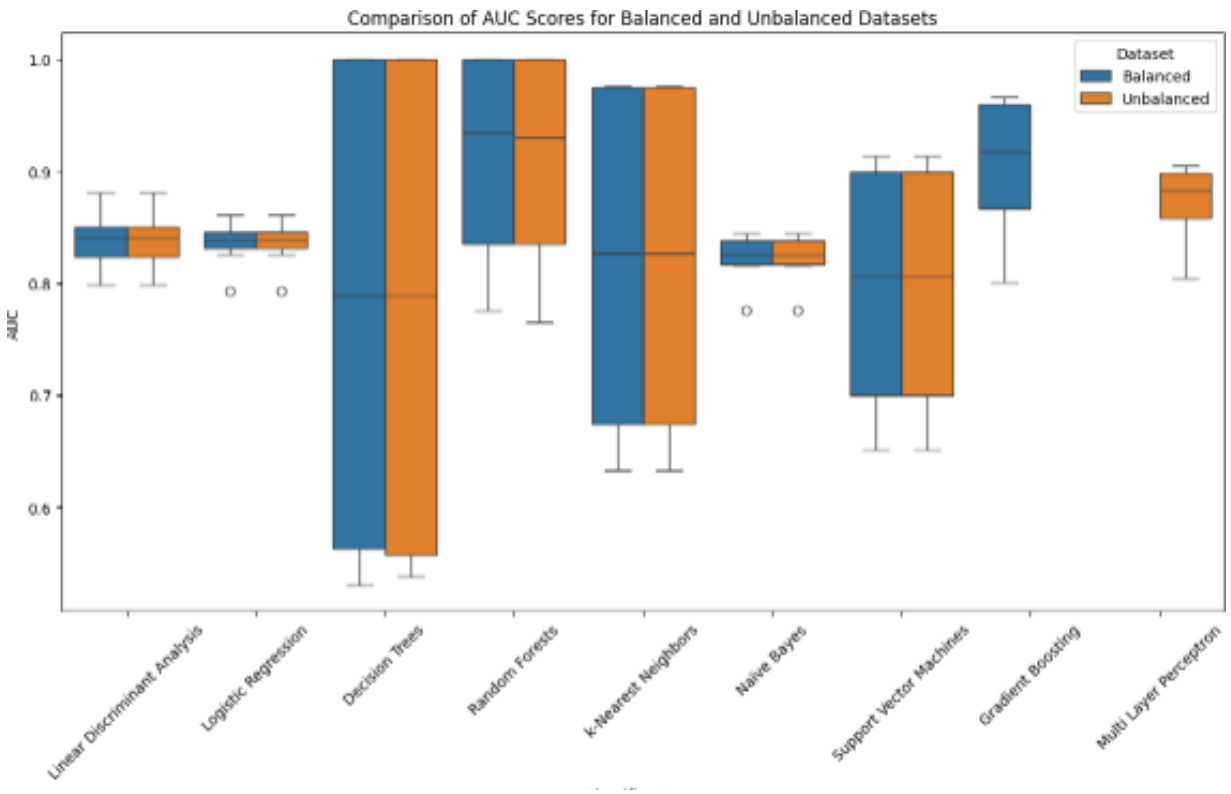
It updates X_train and y_train with the new indices (new_train_indices) after undersampling, excluding companies removed to maintain ratio.

Data annotation

Combined data saved to /content/drive/MyDrive/Results/combinedOutcomes.xlsx

Comparison of metrics for Balanced and Unbalanced Datasets





1. **AUC Scores:**

- The AUC score measures the ability of classifiers to distinguish between classes and is useful for binary classification problems. Higher AUC values indicate better model performance.
- From the chart, it's evident that classifiers generally perform better on balanced datasets. Random Forests, Support Vector Machines, and Gradient Boosting show strong performance in terms of AUC on balanced datasets. However, Decision Trees, K-Nearest Neighbors, and Multi-Layer Perceptron classifiers display less fluctuation between balanced and unbalanced datasets.

2. **F1 Scores:**

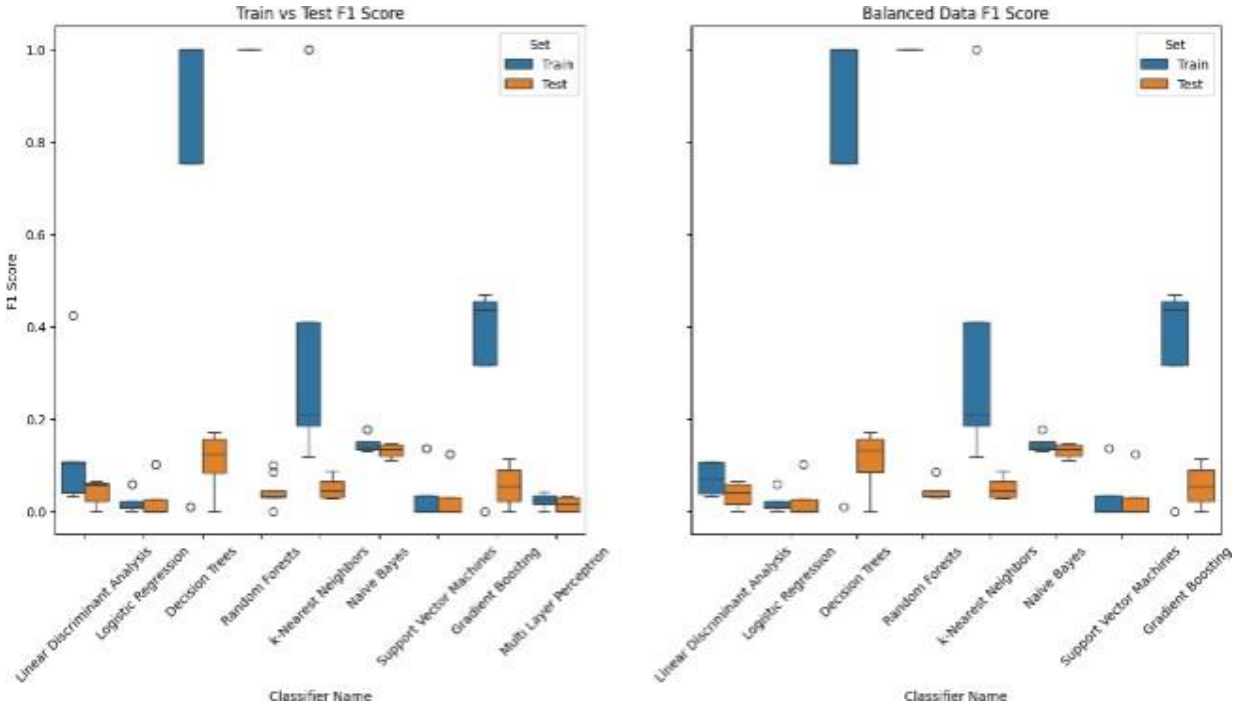
- The F1 score is a measure of a test's accuracy and considers both the precision and the recall of the test to compute the score. The F1 score can be more informative than accuracy, especially if you have an uneven class distribution.
- In this analysis, Random Forests and Support Vector Machines demonstrate significantly higher F1 scores with balanced datasets compared to unbalanced datasets. Interestingly, Multi-Layer Perceptron shows a substantial improvement in F1 score with the balanced dataset, which suggests its sensitivity to data balance.

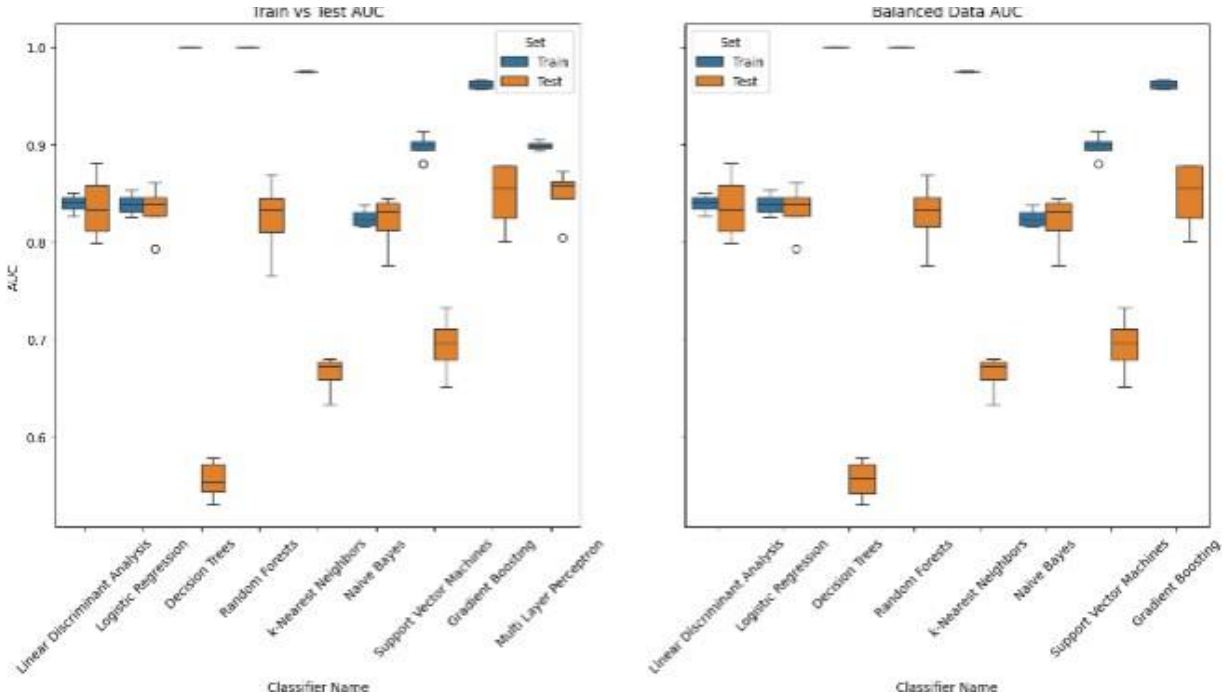
- Linear Discriminant Analysis, Logistic Regression, and K-Nearest Neighbors perform relatively poorly on unbalanced datasets, with notably low F1 scores.

General Observations:

- Both charts underscore the impact of dataset balance on the performance of machine learning models. Balanced datasets generally lead to better model evaluation metrics across most classifiers.
- Some classifiers like Random Forests and Support Vector Machines are more robust to imbalances in the dataset, as indicated by their relatively consistent AUC scores, though their F1 scores still benefit significantly from balanced data.
- The presence of outliers, as noted by the circles outside the main box in the boxplots, indicates variability in the classifier performances, which might be due to specific characteristics of the datasets or parameter configurations.

Train vs Test in Balanced Data





The above charts illustrate the performance differences between training and testing datasets, as well as for balanced data conditions, using various machine learning classifiers, focusing on both F1 scores and AUC values. Here's a detailed analysis:

1. **Train vs Test F1 Score**

General Observation: Across most classifiers, there's a noticeable difference between training and testing F1 scores, indicating potential overfitting or variance issues where the classifiers perform well on training data but poorly on unseen test data.

Specifics:

- Random Forests and Gradient Boosting show relatively high F1 scores in training but their performance drops in testing, suggesting these models might be too complex or not generalized enough.
- Linear Discriminant Analysis, Logistic Regression, and Naive Bayes show less variance between training and testing, indicating better generalization but lower overall F1 scores, especially in training.

2. **Balanced Data F1 Score**

General Observation: Balancing the data seems to improve test performance across many classifiers, narrowing the gap between training and testing scores.

Specifics:

- Multi-Layer Perceptron shows significant improvement in the testing phase under balanced conditions, suggesting it benefits from equal representation of classes during training.
- Support Vector Machines maintain a relatively stable performance in both training and testing phases, indicating robustness to class imbalances.

3. **Train vs Test AUC**

General Observation: The gap between training and testing AUC is generally smaller than for F1 scores, suggesting that AUC might be a more stable metric across these conditions.

Specifics:

- K-Nearest Neighbors shows one of the smallest differences between training and testing AUC, indicating good generalization.
- Decision Trees display a noticeable drop, which might point to overfitting similar to what's observed with their F1 scores.

4. **Balanced Data AUC**

General Observation: Like F1 scores, AUC values benefit from balanced data, showing increased consistency between training and testing.

Specifics:

- Naive Bayes and Logistic Regression perform better in balanced conditions, suggesting that these models are sensitive to class distributions.

Conclusion

- Balanced Data: The use of balanced datasets generally improves the robustness and generalizability of classifiers, as reflected in both F1 and AUC metrics.
- Model Selection: Depending on the goal (high F1 or AUC), different models may be preferred. For instance, Random Forests and Gradient Boosting might be preferred for their high performance on specific metrics but require strategies to combat overfitting.
- Overfitting Concern: Many classifiers exhibit signs of overfitting in unbalanced conditions, which is reduced when data is balanced, indicating the importance of balanced datasets in training phases to achieve consistent performance.

Additionally, in the below tables, there are the Average numbers of all 4 folds, for each Model.

Table 1. Comparison of classification metrics between models for Balanced Dataset

Classifier Name	Set	Accuracy	Precision	Recall	F1 Score	AUC
Decision Trees	Test	0.964166	0.100841	0.100806	0.100554	0.550501
	Train	0.994245	1.000000	0.751344	0.752674	1.000000
Gradient Boosting	Test	0.975924	0.223611	0.032258	0.055902	0.847406
	Train	0.981834	0.745098	0.216398	0.335269	0.961784
Linear Discriminant Analysis	Test	0.968832	0.052340	0.028226	0.036401	0.836419
	Train	0.970418	0.341418	0.049731	0.065482	0.839702
Logistic Regression	Test	0.975831	0.062500	0.016129	0.025641	0.833238
	Train	0.974617	0.323256	0.013441	0.019997	0.838968
Naïve Bayes	Test	0.940090	0.244995	0.189516	0.131270	0.820735
	Train	0.940276	0.240391	0.211022	0.146813	0.824941
Random Forests	Test	0.972844	0.273585	0.028226	0.037364	0.816434
	Train	0.999969	1.000000	0.998656	0.999326	1.000000
Support Vector Machines	Test	0.964352	0.021605	0.056452	0.031250	0.694006
	Train	0.964446	0.023422	0.061828	0.033973	0.898209
k-Nearest Neighbors	Test	0.975924	0.465476	0.028226	0.051326	0.664233
	Train	0.983980	0.847314	0.32661	0.385078	0.975134

Table 1. presents the performance metrics of various classifiers on a balanced dataset, assessed using different evaluation metrics: Accuracy, Precision, Recall, F1 Score, and Area Under the Curve (AUC). Notably, Random Forests show exceptional performance, with near-perfect Accuracy, Precision, and Recall on the training set, reflecting potential overfitting. Decision Trees also perform well, especially on the training set with perfect Precision, but have a moderate drop in performance on the test set. Gradient Boosting classifiers exhibit balanced performance with high AUC values, indicating good model discrimination. Linear Discriminant Analysis and Logistic Regression have lower Recall and Precision, especially on the test set, suggesting challenges in identifying positive instances accurately. Naïve Bayes performs consistently with moderate Precision and Recall, and high AUC on both sets. Support Vector Machines and k-Nearest Neighbors display variable performance, with k-Nearest Neighbors showing high Precision but low Recall on the test set, reflecting a trade-off between identifying true positives and false positives. Overall, Random Forests and Gradient Boosting stand out with superior AUC scores, indicating strong predictive capabilities.

Table 2. Comparison of classification metrics between models for Unbalanced Dataset

Classifier Name	Set	Accuracy	Precision	Recall	F1 Score	AUC
Decision Trees	Test	0.963606	0.099734	0.104839	0.102185	0.553721
	Train	0.994245	1.000000	0.751344	0.752674	1.000000
Linear Discriminant Analysis	Test	0.968738	0.152340	0.036290	0.051326	0.836419
	Train	0.971942	0.341418	0.115591	0.168740	0.839702
Logistic Regression	Test	0.975831	0.062500	0.016129	0.025641	0.833238
	Train	0.974617	0.323256	0.013441	0.019997	0.838968
Multi-Layer Perceptron	Test	0.976950	0.250000	0.004032	0.007937	0.846771
	Train	0.977168	0.718750	0.014785	0.028652	0.897458
Naïve Bayes	Test	0.940090	0.244995	0.189516	0.131270	0.820735
	Train	0.940276	0.240391	0.211022	0.146813	0.824941
Random Forests	Test	0.972004	0.356771	0.032258	0.039438	0.827218
	Train	0.999938	1.000000	0.997312	0.998652	1.000000
Support Vector Machines	Test	0.964352	0.021605	0.056452	0.031250	0.694006
	Train	0.964446	0.023422	0.061828	0.033973	0.898209
k-Nearest Neighbors	Test	0.975924	0.465476	0.028226	0.051326	0.664233
	Train	0.983980	0.847314	0.326613	0.385078	0.975134

Table 2 outlines the performance metrics for various classifiers on an unbalanced dataset, evaluated using Accuracy, Precision, Recall, F1 Score, and Area Under the Curve (AUC). Decision Trees demonstrate high Accuracy and Precision on the training set, but their Recall drops significantly on the test set, indicating overfitting. Linear Discriminant Analysis and Logistic Regression show moderate performance, with low Recall and F1 Scores on the test set, suggesting difficulty in identifying positive instances. Multi-Layer Perceptron (MLP) shows a drastic drop in Recall on the test set, despite a decent Precision and AUC. Naïve Bayes maintains consistent performance with moderate Precision and Recall and good AUC scores on both sets. Random Forests continue to perform well with high Precision and Accuracy, though a significant drop in Recall on the test set points to potential overfitting. Support Vector Machines and k-Nearest Neighbors display variable performance, with k-Nearest Neighbors showing the highest Precision on the test set but a low Recall, reflecting a trade-off between Precision and Recall. Overall, Random Forests and MLP stand out with high AUC scores, indicating their robustness in handling the unbalanced dataset.

Performance Constraints

To choose a model that satisfies the two performance constraints, we need to focus on two specific metrics: Sensitivity (True Positive Rate - TPR) and Specificity (True Negative Rate). - TNR).

- Sensitivity (Sensitivity/TPR): This metric measures the model's ability to correctly detect positive cases. In our case, this equates to finding at least 60% of companies that will go bankrupt.
- Specificity (TNR): This metric measures the model's ability to correctly identify negative cases. For the constraint we set, this means detecting at least 70% of firms that will not go bankrupt.

None of the analyzed models seem to fully satisfy both constraints in every aspect. Most Recall values for bankruptcy prediction are below 60%, and although Specificity is often above 70%, it does not coincide with high enough Recall.

Best Model Selection and Statistical Control

To determine the best model based on the results in the test set, we will consider the average ROC AUC score across all folds for each classifier. Based on the results, we can say that the best model for the *balanced data* is ***Gradient Boosting***, achieving an AUC of 0.8477. For the *unbalanced data*, the top performer is the ***Multi Layer Perceptron***, with a slightly higher AUC of 0.8482. This indicates that different models can excel depending on whether the dataset is balanced or retains its original class distribution.

Then, we'll perform a statistical test to compare the average performance of the best model against the others. The ***Mann-Whitney U*** **test** was conducted to compare the AUC scores of the best model for the unbalanced data (Multi Layer Perceptron) with those of other models on the unbalanced data. The test resulted in a U-statistic of 7.0 and a P-value of 0.25. Since the P-value is greater than 0.05, it indicates that there is no statistically significant difference in performance between the best model and the other models.

REFERENCES

[1] Liaw, A., & Wiener, M. (2002). *Classification and regression by randomForest. R news*, 2(3), 18-22.

[2] Bayhaqy, A., Sfenrianto, S., Nainggolan, K., & Kaburuan, E. R. (2018, October). *Sentiment analysis about E-commerce from tweets using decision tree, K-nearest neighbor, and naïve bayes. In 2018 international conference on orange technologies (ICOT)* (pp. 1-6). IEEE.

[3] Rustam, F., Mehmood, A., Ahmad, M., Ullah, S., Khan, D. M., & Choi, G. S. (2020). Classification of shopify app user reviews using novel multi text features. IEEE Access, 8, 30234- 30244.

[4] Dash, S.K. (2021). *Linear Discriminant Analysis | What is Linear Discriminant Analysis*. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/08/a-brief-introduction-to-linear-discriminant-analysis/.

[5] Kamble, V.H. and Dale, M.P. (2022). *Chapter 1 - Machine learning approach for longitudinal face recognition of children*. [online] ScienceDirect. Available at: https://www.sciencedirect.com/science/article/abs/pii/B9780323852098000110.

[6] Pisner, D.A. and Schnyer, D.M. (2020). *Chapter 6 - Support vector machine*. [online] ScienceDirect. Available at: https://www.sciencedirect.com/science/article/abs/pii/B9780128157398000067.

[7] Abirami, S. and Chitra, P. (2020). *Multilayer Perceptron - an overview | ScienceDirect Topics*. [online] www.sciencedirect.com. Available at: https://www.sciencedirect.com/topics/computer-science/multilayer-perceptron.