01/01/2023

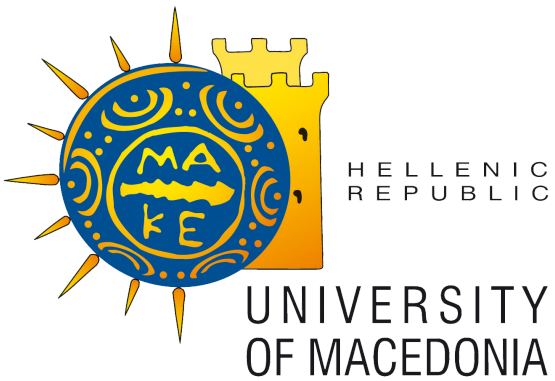
# MACHINE LEARNING
## 2nd Task

**Leisure**:
Applied Informatics

**Department**:
Introduction to Computer Science and Technology

**Theme:**"Classification problems»

**Club**:
1. Michael Zervas, ics20015
2. Nestoras Sotiriou, ics20012
3. Christos Drikos, ics20043




Academic year 2022-2023

# Table of Contents



# Charts Table of Contents

## Introduction

Given some financial indicators of Greek companies, as well as some additional data concerning them, 4 classification models are used in order to find the best one that predicts, based on these data, when a company is bankrupt or not. The 4 classification models used are: Linear Discriminant Analysis, Decision Trees, k-Nearest Neighbors and Naïve Bayes. The selection of the optimal model was made based on the metric F1 score.

# Methods applied

Language used: Python

We have at our disposal data of various companies over a period of 4 years from 2006 to 2009. The volume of data amounts to a total of 10716 (in rows), for all years and companies.

2The question)
This query is implemented with the appropriate use of groupby and aggregate. The number of healthy and bankrupt companies is calculated for each year and the min, max and mean are calculated for each category separately.

3The question)
The data is divided into 80% training data and 20% testing data.

5The question)
The following models were selected:

- Linear Discriminant Analysis (LDA):
  The purpose of LDA is to achieve a good separation between the categories (classes), while simultaneously reducing the dimensions (variables of which the data are composed). To achieve the best separability, the data during its transformation is projected into a space of smaller dimension than the initial, with the result that the number of their dimensions (that is, the variables they are composed of) is reduced, so that subsequently the classification of new data is more accurate.

- Decision Trees:
  In this particular method, the technique of "divide and conquer" is applied, conducting a greedy search to locate the optimal split points within a tree. The problem space is partitioned into regions of instances that have the same value for some attribute variable, and the process is repeated recursively until all or most of the records are classified into specific classes, thus representing the resulting model as a decision tree. . Whether or not all data points are classified as homogeneous sets depends largely on the complexity of the decision tree.

- k-Nearest Neighbors (kNN):
  kNN tries to predict the correct class for the testing data by calculating the distance between the testing data and all training points. Then, it selects the k number of points that are closest to the test data. The kNN algorithm calculates the probability that the test data belongs to the "k" training data classes and the class with the highest probability will be selected. k=3 was used to solve the exercise problem.

- Naïve Bayes:
  The Naïve Bayes algorithm relies on Bayes' probability theorem to calculate the probability that the input sample belongs to each possible class, based on the attribute values. The class with the highest probability is then selected as the predicted class.
  There are 3 types of Naïve Bayes algorithms: Gaussian Naïve Bayes, Multinomial Naïve Bayes and Bernoulli Naïve Bayes. Gaussian Naïve Bayes was used here, whereby when we have continuous feature values, we assume that the values associated with each class are distributed according to the Gaussian or normal distribution.

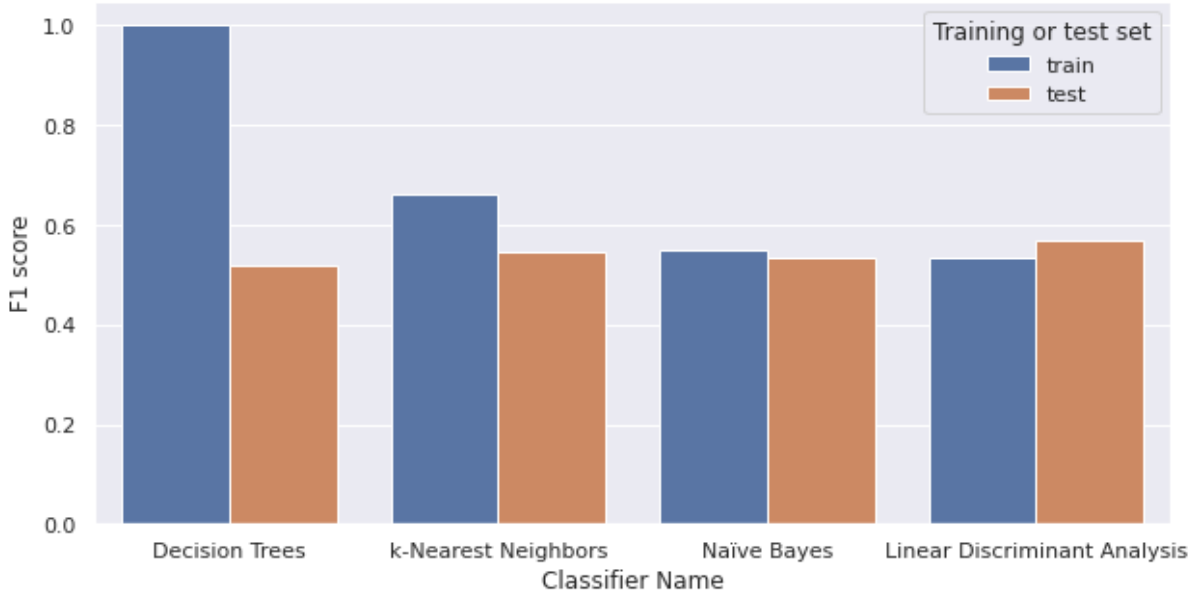
7Thequestion)
4 metrics were used to evaluate the performance of each trained model.
Analytically:
Accuracy: This is the number of correct predictions made by the model, divided by the total number of predictions made. It is a measure of the overall correct classification rate of the model.
Precision: This is the number of true positive predictions made by the model, divided by the total number of positive predictions made. It is a measure of the percentage of positive predictions that were actually correct.
Recall: This is the number of true positive predictions made by the model, divided by the total number of true positives in the data. It is a measure of the percentage of true positive cases that were correctly predicted.

F1 score: This is the harmonic mean of Precision and Recall. It is a measure of the balance between Precision and Recall, with higher values indicating a better balance.

These four evaluation metrics are used to evaluate the performance of a classifier and can provide information on various aspects of its behavior. For example, a classifier with high Accuracy can

not necessarily have high Precision or Recall and vice versa. The F1 score can provide a single metric that reflects the overall performance of the classifier.

Below is an overall plot of the performance of all classifiers in terms of the F1 score metric, both on the training set and the test set.




In the above diagrams we observe a big difference in the results between train and test sets in the Decision Tree Classifier. This is due to the fact that the Decision Tree was overfitting the training dataset. Thus, as expected, its performance on the test set was greatly reduced compared to that on the training set.

## Conclusions

Based on the F1-score metric, the best performing model (although the differences between them are small) seems to be Linear Discriminant Analysis (LDA), then Naïve Bayes with kNN and finally Decision Tree. To improve the Decision Tree so that it does not overfit, we can either prune parts of the complete tree that will be created, or create a tree from the beginning with fewer branches than were created. To improve all models, we could, if possible, use more data. Increasing the amount of training data can often help a model identify more general patterns, which can improve its performance on the test set.

## Sources

https://medisp.uniwa.gr/ https://www.ibm.com/topics/decision-trees https://nemertes.library.upatras.gr/items/4391cc25-e726-448a-ab83-85f3a8ffe46f https: //medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4

https://www.kaggle.com/code/prashant111/naive-bayes-classifier-in-python