Machine Learning project: predicting financial rating of companies

Fabio Morea

2022-03-02

Abstract: This notebook describes the final project for the course *introduction to machine learning* and, at the same time, a case study for Area Science Park in the frame of Innovation Intelligence FVG project. The user needs to predict, with a machine learning model, if a company has a high financial rating. The available data are the financial statements of the companies and, for a limited number of them, the financial rating that will be used as a label for the training of the model. The problem is to identify the model and the data preparation strategy that ensures the best performance on two different datasets, a balanced one (y7, minority class 45%) and the unbalanced one (y8, minority class 6%). In the case of the *balanced dataset*, the best performances are obtained with a Random Forrest model, but also a regression tree leads to good performances associated with an excellent interpretability and explainability of the results.In the case of *inbalanced dataset* it is necessary to apply an oversampling technique on the learning dataset which, associated with a random forrest, gives results in line with those of the previous case.

1. Introduction and problem statement

Background information

Research, innovation and highly skilled people are considered to be important factors in economic and social development. Economic support policies often include funds to support research (for example with the creation of public research infrastructures), companies (for example with tenders to co-finance innovative projects) and the training of people with the necessary skills.

Area Science Park [1] is a national research institution that manages a science and technology park located in Trieste (Italy) and is engaged in several projects aiming to support innovation at the regional level. Innovation Intelligence FVG [2] is a project, managed by Area Science Park and supported by Regione Friuli Venezia Giulia, which aims to monitor the performance of companies in terms of economic, employment and innovation results. .

In the frame of Innovation Intelligence project, Area Science Park needs to analyse the performance of groups of companies (such as the *tenants of Area Science Park*, about 60 companies that have their premises or research laboratories in the science and technology park, or the *regional cluster of metal and plastic manufacturing*, identified by a list of NACE activity sectors [3]) comparing them against similar groups in other regions. Information on companies is available in several datasets that either open source or available under a license for the project partners. A relevant source of information, **financial rating** is available from a rating agency [4], only for the companies that have their premises in Friuli Venezia Giulia.

Financial rating is a numerical variable ranging from 1 to 10, where low values denote an insufficient capability to meet financial obligations, and high values denote very good or excellent reliability. The value is generated by a proprietary algorithm summarizing the overall performance of a company in all its economic and financial areas: profitability, liquidity, solvency, efficiency, production. Only a part of the data used to generate financial ratings is available for this project, thus we may expect that the actual value will be hard to predict. According to some similar cases described in literature [5],[6] reasonably good predictions of rating can be based on *balance sheet data*. In our case we can access additional information on employees (number of employees, turnover and net balance in a given year), that may be a good proxy for company health and performance.

Objective and constraints

The Innovation Intelligence team at Area Science Park wants to explore the potential use of Machine Learning techniques to improve to predict the financial rating of companies, based on other relevant datasets (general company information from the *Italian Business Register* [7].

> The **objective** of the project to identify the model that ensures the best performance on two different cases, a balanced datasets, and an imbalanced datasets (which is notoriously a harder challenge for machine learning algorithms).

The user - or **target audience** - of the project study is the Innovation Intelligence team at Area Science Park, a small group of economic analysts, that have a robust domain knowledge but limited experience in data science. The **datasets** are given: companies working in the sector of metal and plastic manufacturing in Friuli Venezia Giulia, identified by NACE sectors 23 to 29.

Since the results of the project will be used by the Innovation Intelligence team to predict financial ratings of other groups of companies, some **constraints** apply. The analysis will be restricted to a binary classification, in order to obtain a relatively simple but still significant preliminary result. Explainability and interpretability of the model are appreciated, but not essential. Training and classification will be generally performed on a laptop, once a month, on datasets composed of 200 to 2000 companies, hence there are no specific constraints on time or computation effort.

Formal statement of the problem

Let a company be represented by a vector X in a multidimensional space, and associated with a binary label y indicates whether the company belongs to a group of *top performers* or not. The objective is to assess the performance of a binary classification model that predicts y under the following conditions: the dataset is composed of at least 1000 observations, homogeneous by sector and and company type, and the computation time musl be less than 1 hour on a state-of-the art personal computer.

The performance shall be assessed in two cases: a balanced dataset in which the minority class is between 40% and 50%, and an inbalanced dataset in which the minority class is less than 10%.

2. Assessment and performance indexes

Three indexes will be used to measure performance of each model, and effectiveness on unseen data: accuracy, sensitivity (or True Positive Rate, TPR, denotes the ability to correctly classify companies that are in the "top" group) and specificity (or True Negative Rate, TRN, denotes the ability to correctly classify companies that are in the "other" group).

$$Accuracy = (TruePositive + TrueNegative)/AllCases$$

$$Sensitivity = TruePositive/(TruePositive + FalseNegative) = 1 - FNR$$

$$Specificity = TrueNegative/(TrueNegative + FalsePositive) = 1 - FPR$$

The performance will be assessed experimentally using a k-fold cross validation procedure.

As the objective is to identify which model achieves the best performance, there is no predetermined threshold for the accuracy, specificity and sensitivity. However, according to literature and previous experience, accuracy, sensitivity and specificity may be expected to be of the order of 80%. Such level may seem relatively high compared if compared to safety-critical applications, but is sufficient for the business case and in line with similar cases of binary classification of company performance found in scientific literature.

3. Proposed solution

The proposed solution is to evaluate the performance of two binary classification models:

- a *binary decision tree*, that offers the advantage of explainability and interpretability
- a *random Forrest*, that offers potentially a higher performance, according to a comprehensive literature eview [8].

on two different datasets, *Top7* (balanced) and *Top8*(unbalanced). The key steps to achieve the proposed solution are:

1) **feature engineering**: select and rescale features to be processed (matrix X) and generate the labels to be predicted for the two datasets (y7 for Top7 and y8 for Top8).
2) **assessment on the balanced dataset**, using the rpart() library, explore the effect of a flexibility parameter (minsplit) on the performance indexes (accuracy, sensitivity and selectivity). Identify an appropriate range for the flexibility parameter and optimize the decision tree, using a nested k-fold cross validation procedure. Finally, fit a Random Forrest model using caret library and compare the results.
3) **assessment on the inbalanced dataset**: using the rpart() library, explore the effect of a flexibility parameter (minsplit) on the performance indexes (accuracy, sensitivity and selectivity), as seen above. to produce a training dataset that Identify an appropriate range for the flexibility parameter and optimize the decision tree, using a nested k-fold cross validation procedure. Apply oversampling techniques (Up-sampling and down-sampling) to be used for training a regression tree and a random Forrest.

4. Experimental evaluation

4.1 Data

The data available from Innovation Intelligence needs to be pre-processed in order to obtain a *tidy* dataset suitable for ML. An extended explanation of all the features available in the original dataset is available in the Annexes.

The original dataset X consists of n = 1558 observations and p = 16 features (namely: idCompany, totAssets, totEquity, noi, personnel, prod, debts, deprec, valCost, totIntang, revenues, valAdded, yearsInBusiness, staffTurnover, staffBalance, StockAll). The sample is selected according to NACE codes and company type. The first selection on **company type** : we select all types that have a duty of disclosure of financial information, and therefore are suitable for the analysis, namely SU (società a responsabilità limitata con unico socio), SR (società a responsabilità limitata), SP (società per azioni), SD (società europea), RS (società a responsabilità limitata semplificata), RR (società a responsabilità limitata a capitale ridotto), AU (società per azioni con socio unico), AA (società in accomandita per azioni.

A further selection is based on **NACE codes** (further information in the Annexes).The selected sample is composed of companies that have at least one NACE code in one of the following Divisions: 22 (Manufacture of rubber and plastic products), 23 (Manufacture of other non-metallic mineral products), 24 (Manufacture of basic metals), 25 (Manufacture of fabricated metal products, except machinery and equipment), 26 (Manufacture of computer, electronic and optical products), 27 (Manufacture of electrical equipment) and 28 (Manufacture of machinery and equipment).

Some filters are applied: time-dependent data (such as balance sheet data, rating and employees flows) are filtered to year 2019, and company age (years in business) is filtered to values greater than 1.

Labels y7 and y8 are generated on the basis of financial ratings. Data is available for years 2018, 2019 and 2020; for the purpose of this study a single year (2019) will be selected. The choice of classes highlights a relevant issue in classification: imbalance in the distribution of labels.The balanced dataset isTop7 groups basically all above-average companies, while isTop8 groups about one quarter of the companies. In the following, the underrepresented class will be referred to as *minority class*, and the over represented class is referred to as *majority class*.

name	minority.class	majority.class
isTop7	0.456	0.544
isTop8	0.066	0.934

The objective of feature engineering is to build a dataset that contains a set of relevant variables for learning and prediction, appropriately scaled, in the form of a matrix. Specifically we will focus on calculating new features based on domain knowledge, checking variable correlation and normalizing the selected features by centering and scaling. The original features are highly correlated, as highlighted in the following correlation matrix.

Moreover, feature values range over different orders of magnitude (company age ranges from 1 to 150, while total assets ranges from 0 to 10^9 €).

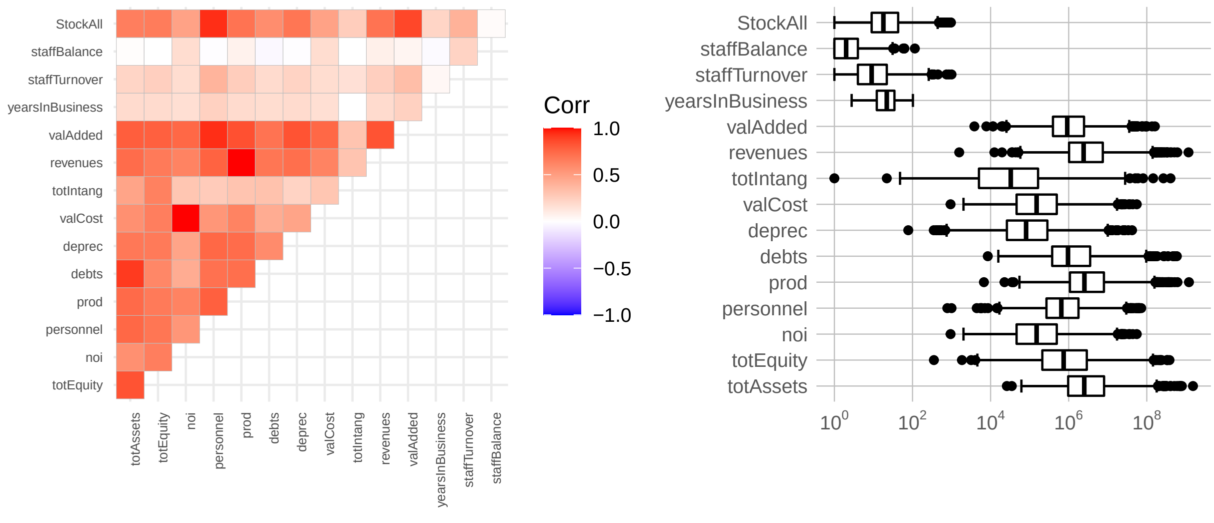


Figure 1: Correlation and order of magnitude of original features

We can tackle both issues by calculating new features that scale economic values to the company size, a common practice in economic analysts, that allows direct comparison of company performance regardless of company size. The new features are named 'rel*' as they are scaled to the total assets (totAssets) or the total number of employees (StockAll) of each company.

Correlation between the new features has significantly improved, as shown in the correlation matrix below.

The next step is to normalize (center and rescale) numeric features to the a similar range in order to improve the performance of the learning algorithm.

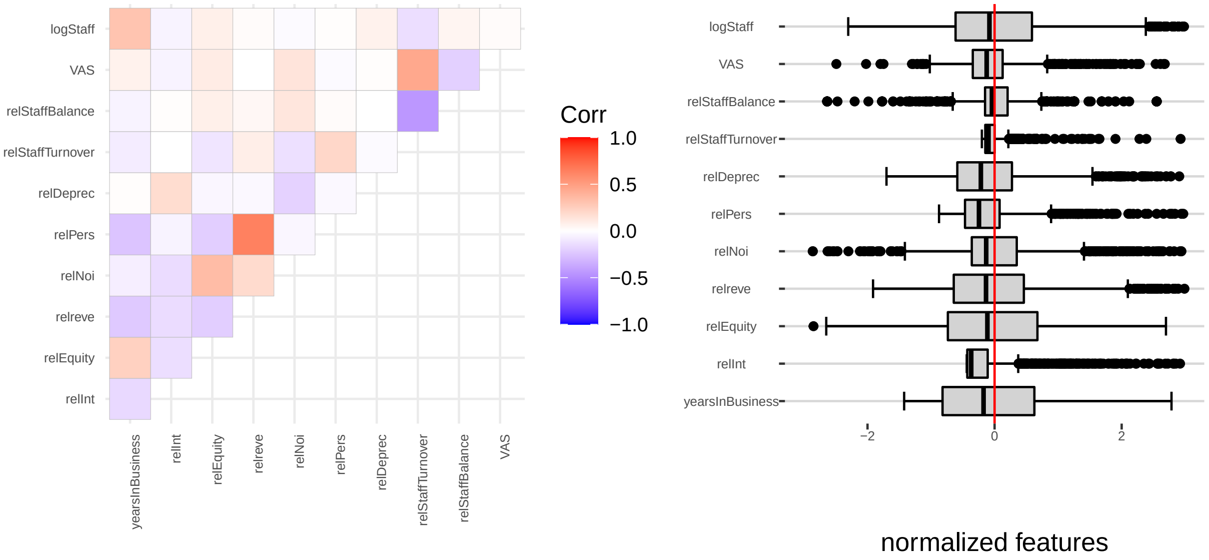


Figure 2: Correlation and distribution of after features engineering

All features are of the same order of magnitude. The dataset is composed of n = observations and p = features (namely:). There meaning of variables is self explanatory in some cases, but economic features may require an explanation:

4.2 Performance of decision trees on balanced dataset

Using the rpart() library, we can explore the effect of a flexibility parameter (minsplit) on the performance indexes (accuracy, sensitivity and selectivity). The decision tree will be tuned using the parameter minsplit, and setting the parameter cp to a fixed value (cp = .001) in order to allow a complexity budget, and keep the model reasonably simple to enhance explainability and interpretability.

The first step is to identify an appropriate range for the flexibility parameter and optimize the decision tree, using a nested k-fold cross validation procedure. Assessment of binary decision trees will be performed on a customized function *assessAccuracySensitivitySpecificity()*, while the random Forrest will be assessed using *caret* package.

The quality of prediction of a single decision tree can be assessed computing its **confusion matrix** (a 2x2 table in the case of binary classifiers) that shows the number of values classified correctly (on the main diagonal) or misclassified (off-diagonal). Error rate, accuracy, specificity and sensitivity are computed from the confusion matrix, comparing model predictions with known labels on *unseen data* from the test dataset.

Learning and prediction are based a stochastic process (splitting data into learn and test sets), hence the results vary for each sample. In order to evaluate the average performance of the method, we need to assess it over a number of samples using a structured experimental procedure such as k-fold cross validation, using the custom function compute.Kfold.AccuracySensitivitySpecificity()

Using the rpart() library, explore the effect of a flexibility parameter (minsplit) on the performance indexes (accuracy, sensitivity and selectivity). We scan the assessment indexes over a wide range of minsplit, from 1 to 300 (step 5).

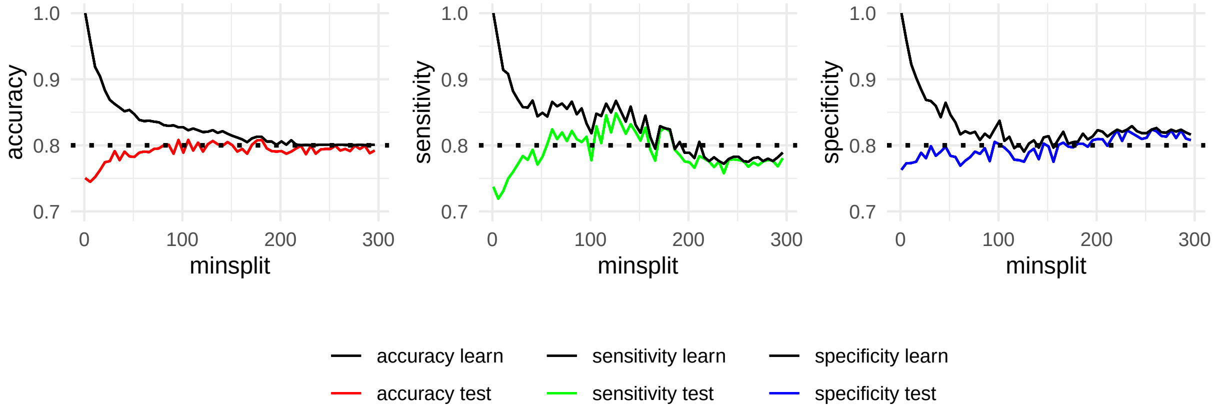


Figure 3: Experimental evaluation of accuracy, sensitivity and selectivity for a single decision tree on the balanced dataset y7

The figure below shows the results, comparing results on training data (black lines) against results on test data (colored lines). The horizontal axis is minsplit, therefore low values refer to complex trees, that have very high values of accuracy, sensitivity and specificity. Accuracy has several maxima for minsplit in range 80-200, and drops for smaller value (a typical sign of overfitting). Sensitivity and specificity drop for low values of minsplit, but have opposite behavior for intermediate values. Sensitivity is the percentage of true positives out of all observations that are in the Top group, i.e. it is the ability of the model to correctly classify a company that is a top performer. Symmetrically, specificity is the percentage of true negatives out of all observations that are not in the Top group, i.e. it is the ability of the model to correctly classify a company that is not top performer. In this example, a model with minsplit around 150 has a low sensitivity and a high specificity, hence they misclassify companies that are in the top group, and correctly classify companies that are not.

Sensitivity and specificity are inversely related should be considered together, and the choice for optimization depends on the needs of the project. In our example, if we need to optimize the model to achieve a high specificity, we should choose minsplit in the range 80 to 120. conversely, higher sensitivity can be achieved using larger values of minsplit, in the range 200-300.

Train for max accuracy, in a range of minsplit with high specificity

Train another tree for max accuracy, in a range of minsplit with high sensitivity

4.3 Performance of random forrest on balanced datast

A Random Forrest model can be fit to the train data and assessed on the test data, using caret library and the embedded repeated cross validation method. Performance is assessed using the custom function kFoldassesAccuracySensitivityScpecificityRF().

```
#generate data do plotrequire(caret)
data7r <- data7 %>%
  mutate(Class = case_when(isTop== TRUE   ~ "top" , isTop== FALSE   ~ "other")) %>%
  select(-isTop)
train.7 <- data7r %>% slice( indexes.learning)
test.7  <- data7r %>% slice(-indexes.learning)

# general control function for training
```

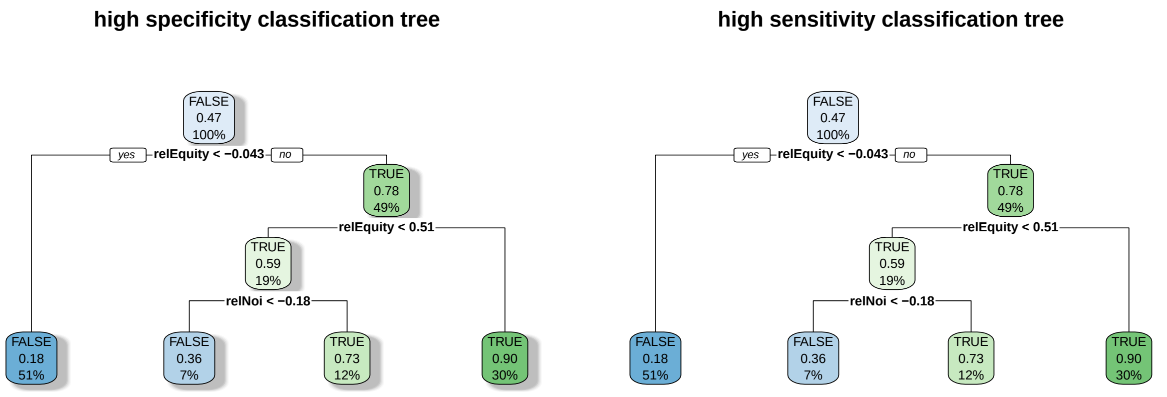
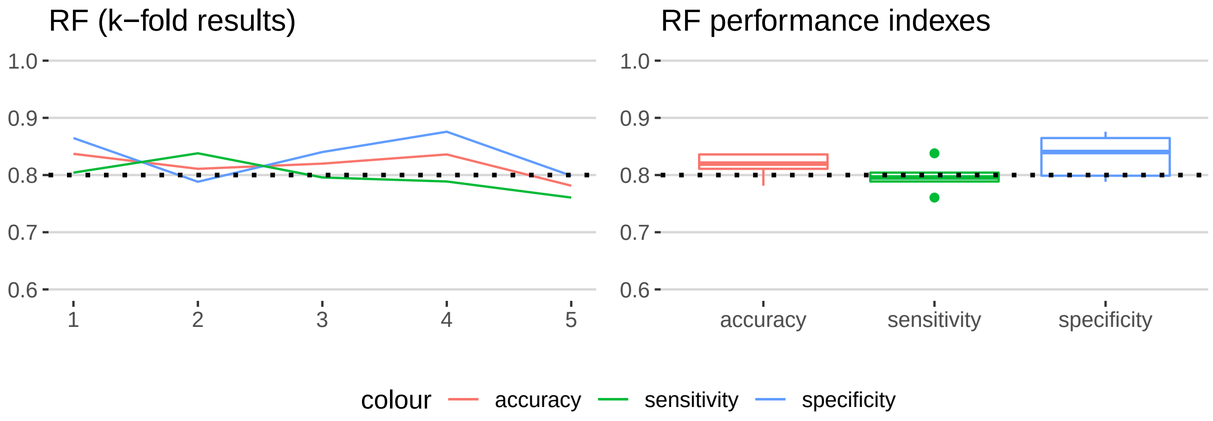


Figure 4: A visual representation of the binary decision tree optimized for the imbalanced dataset y7

```
ctrl <- trainControl(method = "repeatedcv",
                     number = 5,#10
                     repeats = 3,#5
                     summaryFunction = twoClassSummary,
                     savePredictions = "final",
                     classProbs = TRUE)


data.to.plot.rf.7 <- kFoldassesAccuracySensitivityScpecificityRF(data7r,
                                                            ctrl,
                                                            name='y7 RF',
                                                            k_folds=5)

data.to.plot.7  <- data.to.plot.7 %>%
  add_row( tibble_row(indic = 'accuracy',   method =  'y7 RF', value =  mean(data.to.plot.rf.7$accuracy)
  add_row( tibble_row(indic = 'sensitivity',method =  'y7 RF', value =  mean(data.to.plot.rf.7$sensitivi
  add_row( tibble_row(indic = 'specificity',method =  'y7 RF', value =  mean(data.to.plot.rf.7$specifici
```



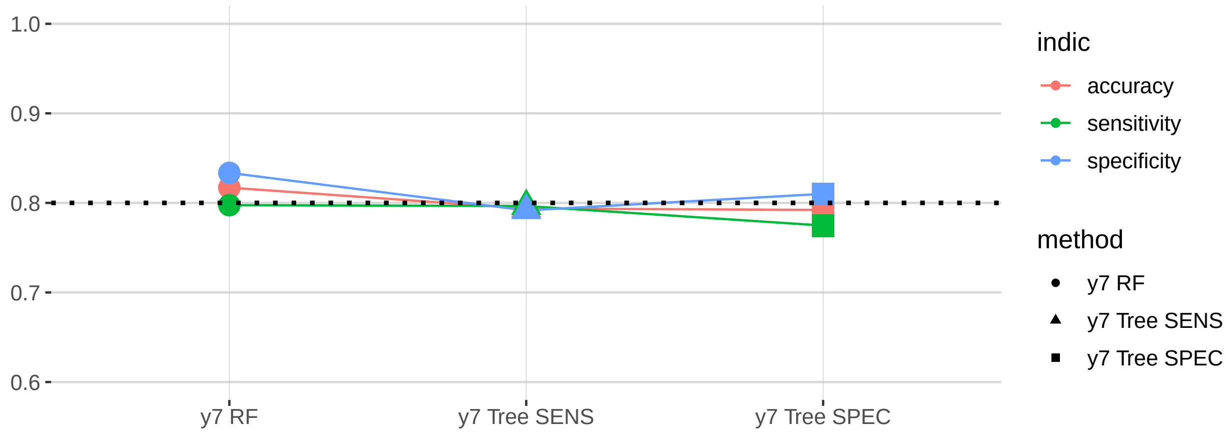


Figure 5: Experimantal assessment of accuracy, sensitivity and specificity for a random forrest model on a balanced dataset y7, using k-fold cross validation

4.4 Performance of decision trees on inbalanced dataset

In classification problems, a disparity in the frequencies of the observed classes can have a significant negative impact on model fitting. we will first of all assess the performance of decision trees, using the procedure seen above.

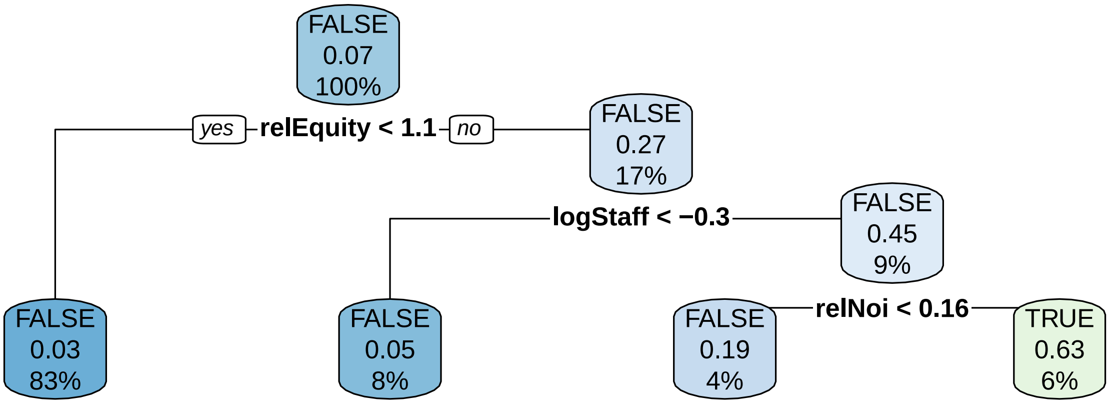
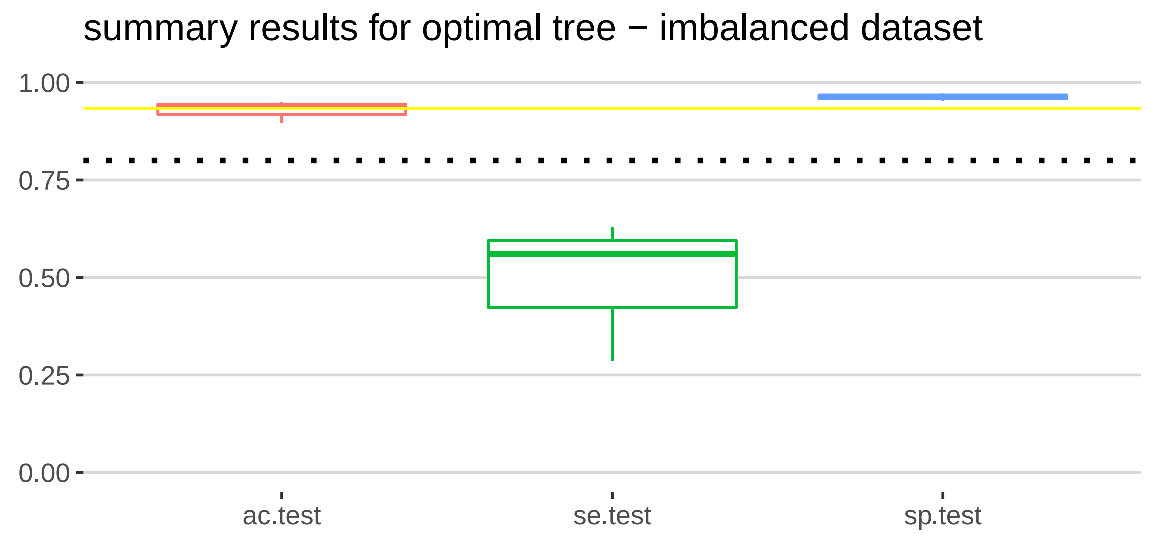


Figure 6: A visual representation of the binary decision tree optimized fot the imbalanced dataset y8

Even after optimization, a decision tree model has a very poor performance. Accuracy and specificity soar to the level of minority class, while sensitivity drops below 50%.



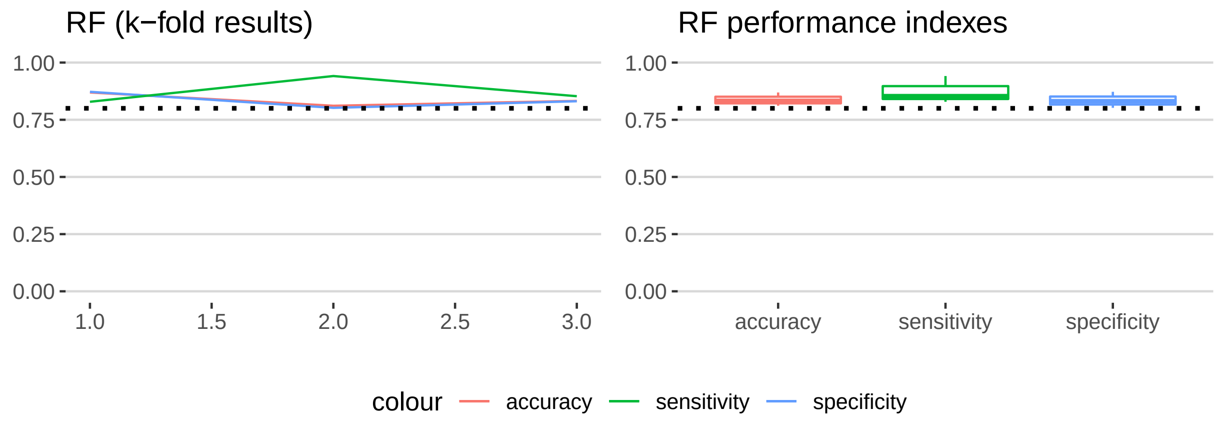
This is known as the "accuracy paradox" in unbalanced datasets, where simply predicting the majority class provides high accuracy, associated with a high bias: the model misclassifies the "top" performing companie

4.5 Performance of Random Forrest on inbalanced dataset

Similarly, a Random Forrest model has poor performance. One technique for resolving such a class imbalance is to sub-sample the training data in a manner that mitigates the issues. Examples of sampling methods for this purpose are:

- down-sampling: randomly subset all the classes in the training set so that their class frequencies match the minority class. In the case of y8, about 75% of the observations are the majority class. Down-sampling would randomly sample the majority class to be the same size as the minority class (so that only a part of the training data is used to fit the model).
- up-sampling: randomly sample (with replacement) the minority class to be the same size as the majority class. caret contains a function (upSample) to do this.
- hybrid methods: down-sample the majority class and synthesize new data points in the minority class. The above mentioned methods are implemented in the caret package as "downSample" and "upSample".

The procedure is executed using caret library, specifying sub-sampling when using train so that it is conducted inside of resampling.



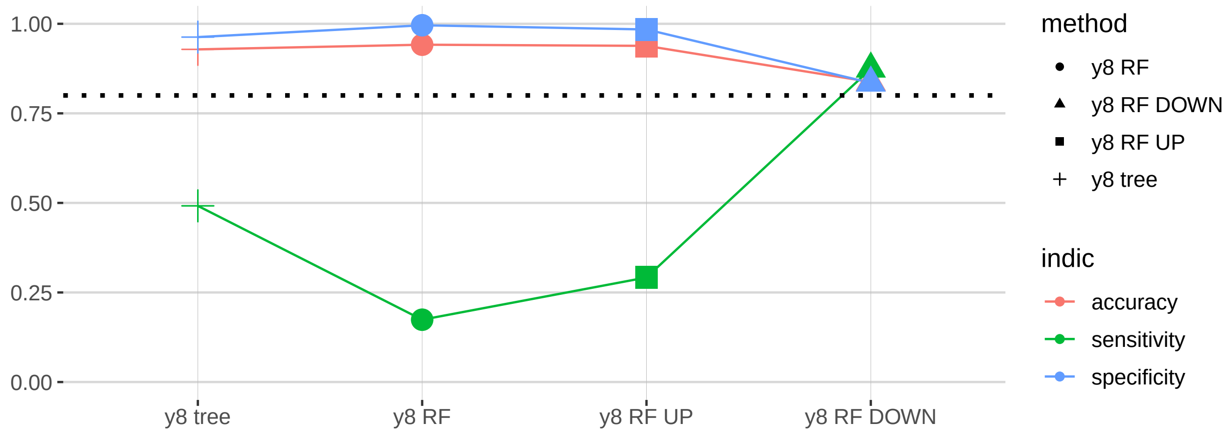


Figure 7: Experimantal assessment of accuracy, sensitivity and specificity for a random forrest model on a inbalanced dataset y8, using oversampling, undersampling

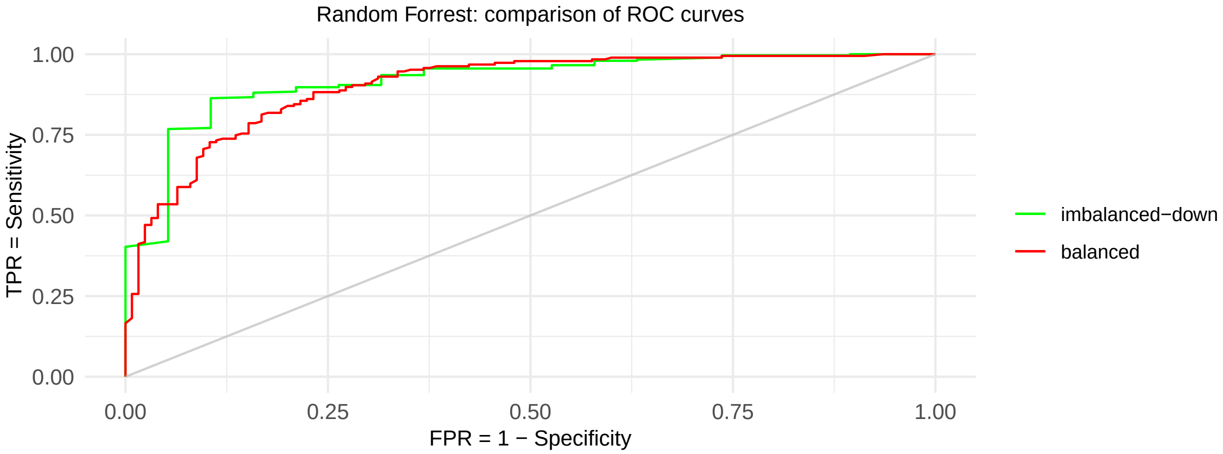
4.6 Comparison of results

The accuracy of best-performing models is summarized in the following table. Random Forrest models (RF) ensure consistently higher performance compared to decision tree models. On the balaced dataset the RF model can achieve an accuracy of 0.83, associate with values of sensitivity and specificity, that are higher than the corresponding decision tree (model Tree SENS). When dealing with the inbalanced data, Random Forrest can achieve results that are in line with the previous case, provided that the learning phase is performed on a suitably oversampled dataset. In this case the oversampling strategy that performed best is "downsampling".

method	dataset	accuracy	sensitivity	specificity
Tree SPEC	y7	0.792	0.775	0.810
Tree SENS	y7	0.794	0.797	0.792
RF	y7	0.817	0.797	0.834
RF	y8	0.942	0.174	0.996
RF DOWN	y8	0.838	0.874	0.835

The comparison of performance can be visualized using the ROC Curve (Receiver Operating Characteristics), measuring the AUC (Area Under The Curve).

dataset	method	auc
balanced	orig	0.902
inbalanced	down	0.918



The AUC index is similato for both curves, hence the predictions on an inbalanced dataset can be expected to achieve the same quality of predictions on a balanced dataset, when

5. Concluding remarks

This project demonstrates that a machine learning algorithm can predict whether a company belongs to the group of "top performers" with accuracy, sensitivity of the order of 80%, and that quality of predictions depends strongly on the definition on the degree of imbalance of the dataset.

The performance has been demonstrated for a binary classification algorithm (namely a single tree, generated by the recursive partitioning algorithm of r-part library) and for a random Forrest. The dataset used for learning and validation is composed of n = 1558 observations and p = 13 features, obtained from a larger dataset of regional companies, filtered on NACE sectors of activity and on year 2019.

Two case studies have been examined in detailed, using different definitions of the y label (isTop8, a narrow definition addressing about 15% of the sample) and isTop7 (a balanced dataset addressing about half of the companies).

The model has been tuned to maximize accuracy on test data using a nested k-fold cross validation loops for tuning and assessing experimentally the performance, a procedure that gives good results for balanced datasets. The imbalanced datasets has been treated with a *random Forrest* model and an oversampling of the minority class, to achieve a good accuracy, and improving on sensitivity and specificity.

In both cases computation time is of the order of minutes, depending on the number of folds in k-fold cross validation, which is in line with the constraints of the project. The accuracy of predictions has been measured experimentally, and results are in all cases around 80%. Specifically, key results are summarized in the following table.

The quality of predictions depends on the selection of companies included in the dataset: we may expect that smaller, homogeneous sets may result in higher accuracy (e.g. selecting only companies of a given size, or of a single NACE sector) and the exact definition of the label. In future applications the model *should be learned for specific subsets and specific definitions of label*, and accuracy (or quality of predictions) has to be evaluated for each case.

These results suggest further investigations. First, as the optimal decision trees are simple and do not use many of the available features, we should assess the predictive power of each feature and work on a smaller model with the advantage of improving interpretability and explainability. Second, explore other supervised ML methods such as multi-class decision trees, SVM, naive Bayes and KNN. Moreover, some unsupervised techniques may be applied, such as Principal Component Analysis, estimate of intrinsic dimension and k-means clustering.

Finally, as data is available also for 2018 and 2020, we may compare the performance of models on same-year data as well as using unseen data from other years. A relevant question is to train a model on data from 2018 and 2019, and compare predictions on 2020, when disruptive change in economy took place.

References

1. Area science park - institutional website. http://www.areasciencepark.it;

2. Innovation intelligence FVG - project website. http://www.innovationintelligence.it;

3. Eurostat - NACE codes. https://ec.europa.eu/eurostat/web/nace-rev2;

4. ModeFinance. The MORE Philosophy: The credit rating technology by modeFinance. https://www.modefinance.com/en/data-science;

5. Gandin I, Cozza C. Can we predict firms' innovativeness? The identification of innovation performers in an Italian region through a supervised learning approach. PLOS ONE. 2019;14: e0218175.

6. Wu J, Zhang Z, Zhou SX. Credit Rating Prediction Through Supply Chains: A Machine Learning Approach. Production and Operations Management. 2021.

7. Italian Business Register. https://italianbusinessregister.it/en/home;

8. Fernández-Delgado M, Cernadas E, Barro S, Amorim D. Do we need hundreds of classifiers to solve real world classification problems? The journal of machine learning research. 2014;15: 3133–3181.

9. James G, Witten D, Hastie T, Tibshirani R. An Introduction to Statistical Learning: With Applications in R. Springer Science & Business Media; 2013.