



Stockholms  
universitet

# A Comparative Study of Random Forest and Gradient Boosting for Imbalanced Multiclass Classification

Liam Cuclair

Kandidatuppsats 2026:11  
Matematisk statistik  
Maj 2026

[www.math.su.se](http://www.math.su.se)

Matematisk statistik  
Matematiska institutionen  
Stockholms universitet  
106 91 Stockholm

# A Comparative Study of Random Forest and Gradient Boosting for Imbalanced Multiclass Classification

Liam Cuclair\*

May 2026

## Abstract

In this thesis, multiclass classification under varying levels of class imbalance is investigated using two machine learning methods: random forest and gradient boosting. The study is conducted on two datasets with different characteristics, while varying class sizes, hyperparameter settings, and methods for handling class imbalance. Model performance is evaluated primarily using balanced accuracy, supplemented by the F1-score. The imbalance methods considered are oversampling, undersampling, and class weights. The results show that oversampling is generally the most effective method for handling class imbalance, especially when the imbalance is large. When the imbalance is more moderate, class weights also performs well. Undersampling generally performs worse than the other methods, although it can be competitive in datasets that are more prone to overfitting. For datasets that are more prone to underfitting, gradient boosting generally outperforms random forest across all levels of class imbalance, whereas the two methods perform similarly for datasets that are more prone to overfitting. The results also indicate that hyperparameter tuning is important for both methods. For random forest, larger maximum depths generally improve performance or have little negative effect, suggesting a relatively low risk of overfitting. For gradient boosting, however, performance is more sensitive to model complexity, and the best results are obtained at intermediate maximum depths, where for datasets prone to overfitting higher maximum depths offer very similar results to intermediate maximum depths.

---

\*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden.  
E-mail: [liam.cuclair@math.su.se](mailto:liam.cuclair@math.su.se). Supervisor: Martina Favero & Jan-Olov Persson.

# Acknowledgements

I would like to thank my supervisors Martina Favero and Jan Olov Persson for their guidance and valuable feedback on different approaches. Artificial intelligence was not used in the process of writing this thesis.

# Contents

|          |                                                        |           |
|----------|--------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                                    | <b>5</b>  |
| 1.1      | Aims . . . . .                                         | 5         |
| 1.2      | Disposition . . . . .                                  | 5         |
| <b>2</b> | <b>Theoretical background</b>                          | <b>5</b>  |
| 2.1      | Statistical learning . . . . .                         | 6         |
| 2.2      | Supervised learning . . . . .                          | 6         |
| 2.3      | Training and evaluating models . . . . .               | 7         |
| 2.3.1    | Measurements . . . . .                                 | 7         |
| 2.4      | The bias variance trade off . . . . .                  | 8         |
| 2.5      | Decision trees . . . . .                               | 9         |
| 2.6      | Random forest . . . . .                                | 11        |
| 2.7      | Gradient boosting . . . . .                            | 11        |
| 2.8      | Imbalanced data . . . . .                              | 14        |
| <b>3</b> | <b>Simulation and modeling</b>                         | <b>15</b> |
| 3.1      | Data . . . . .                                         | 15        |
| 3.2      | Data simulation . . . . .                              | 16        |
| 3.2.1    | Hyperparameters . . . . .                              | 17        |
| 3.2.2    | Class imbalance . . . . .                              | 17        |
| 3.2.3    | Dealing with class imbalance . . . . .                 | 18        |
| 3.2.4    | Data simulation . . . . .                              | 18        |
| 3.3      | Procedure for handling simulation results . . . . .    | 19        |
| 3.3.1    | Procedure for the forest covertype results . . . . .   | 19        |
| 3.3.2    | Procedure for the handwritten digits results . . . . . | 20        |
| <b>4</b> | <b>Results</b>                                         | <b>20</b> |
| 4.1      | Random forest . . . . .                                | 20        |
| 4.1.1    | Minimum node size, first round . . . . .               | 21        |
| 4.1.2    | Maximum depth . . . . .                                | 22        |
| 4.1.3    | Minimum node size, second round . . . . .              | 24        |
| 4.1.4    | Class imbalance . . . . .                              | 26        |
| 4.1.5    | Imbalance methods . . . . .                            | 28        |
| 4.2      | Gradient boosting . . . . .                            | 30        |
| 4.2.1    | Maximum depth . . . . .                                | 30        |
| 4.2.2    | Class imbalance . . . . .                              | 32        |
| 4.2.3    | Imbalance method . . . . .                             | 34        |
| 4.3      | Comparison between models . . . . .                    | 36        |
| 4.4      | Class differences . . . . .                            | 38        |
| <b>5</b> | <b>Discussion</b>                                      | <b>39</b> |
| 5.1      | Limitations . . . . .                                  | 39        |
| 5.2      | Result discussion . . . . .                            | 40        |

|                                 |           |
|---------------------------------|-----------|
| <b>6 Conclusion</b>             | <b>43</b> |
| <b>7 References</b>             | <b>44</b> |
| <b>A Random forest</b>          | <b>46</b> |
| A.1 Maximum depth . . . . .     | 46        |
| A.2 Minimum node size . . . . . | 47        |
| A.3 Class imbalance . . . . .   | 49        |
| A.4 Imbalance method . . . . .  | 52        |
| <b>B Gradient boosting</b>      | <b>53</b> |
| B.1 Maximum depth . . . . .     | 53        |
| B.2 Class imbalance . . . . .   | 56        |
| B.3 Imbalance method . . . . .  | 59        |
| <b>C Method comparison</b>      | <b>61</b> |

# 1 Introduction

The class imbalance problem is a common issue in machine learning classification tasks. It arises because many machine learning algorithms assume a balanced dataset, that is, a dataset in which all classes contain approximately the same number of observations. If an imbalanced dataset is not handled appropriately, the performance of the resulting model may be affected. Several methods for dealing with class imbalance exist, and the most commonly used ones are investigated in this thesis.

Random forest and gradient boosting are two of the most successful machine learning methods in many areas of application, but neither method is immune to the class imbalance problem. For this reason, this thesis examines how these two methods perform under different levels of class imbalance, and how their performance can be improved for different imbalance settings.

## 1.1 Aims

The aim of this thesis is to investigate how different levels of class imbalance affect the performance of random forest and gradient boosting in multiclass classification with five classes. The thesis also aims to investigate how varying certain hyperparameters for the two methods affects the results, and how different methods for handling class imbalance, hereafter referred to as imbalance methods, influence model performance. This is done partly to determine which machine learning method should be chosen for different levels of class imbalance, which imbalance method should be preferred, and how the hyperparameters should be set for optimal performance. Another aim is to identify whether any systematic patterns arise from combinations of machine learning method, hyperparameter choice, imbalance method, and imbalance level.

## 1.2 Disposition

In Section 2, the theoretical background needed to understand the thesis is presented. The material is introduced in a way that requires only limited prior mathematical knowledge while still covering the essential mathematical concepts. Section 3 presents the methods used to obtain the results. Section 4 presents the results, and Section 5 discusses the results together with the limitations of the methods used. Section 6 presents the conclusions drawn from the discussion and answers the aims. Section 7 contains the references, and an appendix at the end of the thesis includes tables and plots that were not included in the main results section.

# 2 Theoretical background

Here the theoretical background necessary to understand the rest of the thesis is presented

## 2.1 Statistical learning

According to Hastie et al. (2009, p. 1) statistical learning refers to a set of methods for learning from data in order to predict an outcome variable based on observed features of the data (also called input variables or predictor variables). One example mentioned by Hastie et al. (2009, p. 2) is in the context of email classification, the outcome is whether an email is spam or not, while input variables might be the word count for different words in the emails. In order to train a model that can, for example, separate spam email, a training set of data is used, where the model has full access to the data in order to find out how the features and the outcome measurement correlate (Hastie et al., 2009, p. 1-2). The model can then be used on new unseen data where the outcome measurement isn't present. Statistical learning is divided into supervised and unsupervised learning (Hastie et al., 2009, p. 2). In supervised learning, the outcome variable is part of the learning process, while for unsupervised learning there are no measurements of the outcome (Hastie et al., 2009, p. 2). This thesis focuses on supervised learning.

## 2.2 Supervised learning

There are different types of output variables which influence what the supervised learning process looks like. According to Hastie et al. (2009, p. 9) the output can be quantitative, for example if the outcome is house prices, where some values are larger than other and if two outcomes are close in value then they are also close in reality. If the outcome is quantitative the preferred method is regression (Hastie et al., 2009, p. 10). Hastie et al. (2009, p. 9-10) also mention that the output can be qualitative where the output is a finite set of different classes, for example if the output variable is whether emails are spam or not (2 classes). For qualitative outputs there isn't any kind of order in the classes. If the outcome is qualitative the preferred method is classification. In this thesis only qualitative outcome variables will be studied and therefore classification will be the method used.

Hastie et al. (2009, p. 29) introduces the view of supervised learning as a function fitting paradigm where the model  $Y = f(X) + \epsilon$  represents the relationship between the output  $Y$  and the features  $X$ . The error between  $f(X)$  and  $Y$  is then represented as an additive term  $\epsilon$ . In this function fitting paradigm the goal of supervised learning is to approximate  $f$  as close as possible based on a training set of observations  $(x_i, y_i)$  for  $i = 1, \dots, N$ . A learning algorithm is used which produces outputs  $\hat{f}(x_i)$  where  $\hat{f}$  represents the approximation of  $f$ . The algorithm can use the training set to control how close  $\hat{f}(x_i)$  is to  $y_i$  and change  $\hat{f}$  in order to minimize this difference. In the end the hope is that  $\hat{f}$  is close enough to  $f$  so that  $\hat{f}$  can be successfully used on unseen data.

## 2.3 Training and evaluating models

According to Lindholm et al. (2022, p. 13) it is necessary to have a training set of observations  $(x_i, y_i)$  which the learning model can train on, where the output  $Y$  is known. It is also necessary to have a test set of previously unseen data where only the features  $X$  are known to the model, in order to evaluate how well the model performs on unseen data.

### 2.3.1 Measurements

In order to evaluate how well the model performs on unseen test data it is necessary to use some kind of measurement. Assuming the model was tested on test data where the output was unknown to the model but the output is known to the user of the model, then there are some different measurements that can be used for classification. Assuming there is a class in focus, which will be considered the positive class, then these different values are used for the measurements and are defined in James et al. (2021, p. 152):

1. TP = The number of true positives, i.e. the number of input instances  $x_i$  that are correctly classified as belonging to the positive class.
2. FP = The amount of false positives, i.e. the number of input instances  $x_i$  that are incorrectly classified as belonging to the positive class.
3. TN = The amount of true negatives, i.e. the number of input instances  $x_i$  that are correctly classified as not belonging to the positive class.
4. FN = The amount of false negatives, i.e. the number of input instances  $x_i$  that are incorrectly classified as not belonging to the positive class.

According to James et al. (2021, p.153) sensitivity is defined as

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (1)$$

and measures the portion of all positives the model finds. James et al. (2021, p. 153) defines specificity as

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (2)$$

and measures the portion of all negatives the model finds. James et al. (2021, p. 153) also defines precision as

$$\text{precision} = \frac{TP}{TP + FP} \quad (3)$$

and measures the portion of all the instances predicted as positive that actually are positive. In Brodersen et al. (2010) balanced accuracy is defined as

$$\text{Balanced accuracy} = \frac{\text{Sensitivity} + \text{Specificity}}{2} \quad (4)$$

and measures how well the model balances sensitivity and specificity. According to Lindholm et al. (2022, p. 89) the F1-score is defined as

$$F1 = 2 * \frac{\text{precision} * \text{sensitivity}}{\text{precision} + \text{sensitivity}} \quad (5)$$

and measures how well the model balances sensitivity and precision.

## 2.4 The bias variance trade off

In section 2.2, the true relationship between the input  $X$  and the output  $Y$  is described as  $Y = f(X) + \varepsilon$ , where the aim is to find a good estimate  $\hat{f}$  of  $f$ . If it is assumed that  $E[\varepsilon] = 0$  and that  $\varepsilon$  is independent of  $X$ , for a new observation  $x_t$  with output  $y_t$  the expected test mean squared error (MSE) at  $x_t$  is defined as  $(E[y_t - \hat{f}(x_t)])^2$  (James et al., 2021, p. 33-34). According to James et al. (2021, p. 33-34) it can be shown that

$$(E[y_t - \hat{f}(x_t)])^2 = \text{Var}(\hat{f}(x_t)) + (\text{Bias}(\hat{f}(x_t)))^2 + \text{Var}(\varepsilon). \quad (6)$$

According to James et al. (2021, p. 34-35) the variance of  $\hat{f}$  measures how much  $\hat{f}$  changes for different training sets, while the bias of  $\hat{f}$  measures the systematic error of  $\hat{f}$  that exists no matter which training set is used. The variance of  $\varepsilon$  is the irreducible error, which means that the expected test  $MSE$  never can be less than  $\varepsilon$ . In general if a more complex method is used to create  $\hat{f}$  the bias is low and the variance is higher, while for less complex methods the bias is high while the variance is lower (James et al., 2021, p. 35). As can be seen in (6) the expected test MSE increases if either the variance or the bias increases. This is what the bias variance trade off means, that when choosing method complexity a too complex method will have too high variance, leads to overfitting and therefore be a bad choice, while not complex enough methods will have too high bias, leads to underfitting and therefore also be a bad choice. In order to minimize the expected test MSE a balance between bias and variance has to be found. This is illustrated in figure 1 below.

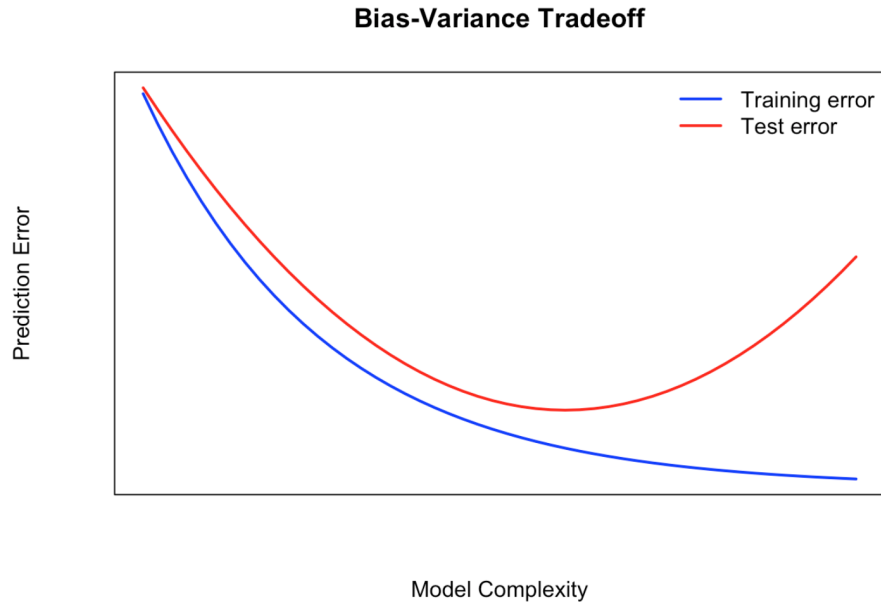


Figure 1: A visualization of how the training error decreases with lower bias and higher variance for higher model complexity, while the test sample starts performing worse when the variance on the training sample is too high for higher model complexity. The plot is inspired by Hasting et al. (2009, p. 38).

## 2.5 Decision trees

Decision trees are methods that can be used for both classification and regression. To illustrate the concept an example of a regression tree from James et al. (2021, page 328-329) is presented. In this example the output  $Y$  is the log salary in dollars for baseball players, and the two predictor variables  $X_1$  and  $X_2$  are the number of years played in major leagues and the number of hits made in the previous year, respectively. A decision tree divides the data into groups based on some splitting criterion and new observations that fall into one of the groups will be predicted to have the output which is the mean of the rest of the observations in that group, at least for regression (James et al., 2021, p. 328). The splits are called binary splits since they divide a group into two new groups. These groups are called nodes. An example of a decision tree can be seen in figure 2 below. However, it should be noted that the plot on the right in Figure 2 should have as many axes as there are predictor variables, which makes it more difficult to visualise for more predictor variables, although the underlying concept is the same.

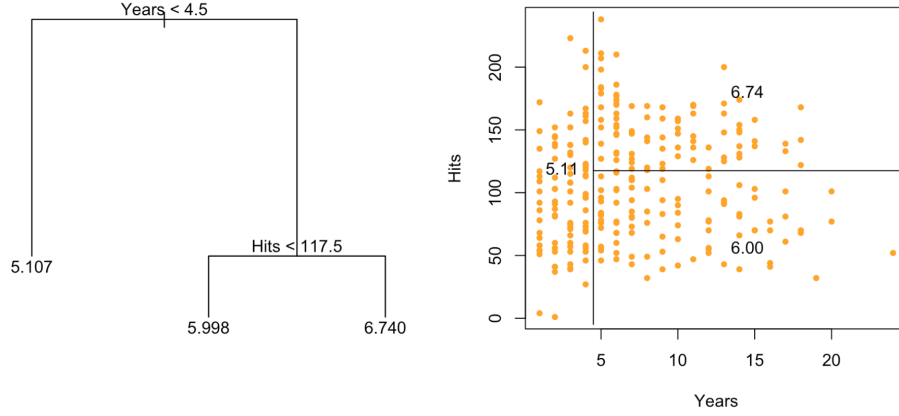


Figure 2: On the left the decision tree can be viewed and on the right it is visible how the data is split up based on the decision tree. The regression tree was constructed using the `tree()` function in the R package `tree` and the Hitters data, and then pruned using `prune.tree()` in order to recreate the plots from James et al. (2021, page 328-329).

When instead viewing classification new observations are instead assigned to the class that is most frequent in the node to which the observation belongs (James et al., 2021, p. 335). Since regression is not the focus for this thesis the criteria used for making the binary splits for regression will not be written about here. However, for classification one criterion that can be used for making the binary splits is the Gini index. According to James et al. (2021, p. 336) the Gini index is defined as

$$G = \sum_{k=1}^K p_{mk}(1 - p_{mk}) \quad (7)$$

where  $p_{mk}$  denotes the proportion of training observations in the  $m$ th node that belong to class  $k$ , and  $K$  is the number of classes. According to James et al. (2021, p. 336) this means that the Gini index measures the total variance across the  $K$  classes in the  $m$ th node. However, in order to compare the Gini index across different splits, the Gini index when node  $m$  is divided into the child nodes  $m_L$  and  $m_R$  with  $L$  and  $R$  observations, respectively, has to be weighted depending on  $L$  and  $R$  (Hastie et al., 2009, p. 309). In other words the Gini index of  $m_L$  and  $m_R$  should be multiplied with  $\frac{L}{L+R}$  and  $\frac{R}{L+R}$ , respectively. The Gini index has a small value for a node  $m$  if all  $p_{mk}$  are close to zero or one, which would mean that most observations in that node belong to the same class (James et al., 2021, p. 336). If the total weighted Gini index across all child nodes is small, the split is considered good. The objective is therefore to minimise the weighted Gini index. This is achieved using a greedy algorithm known as recursive binary splitting, which considers all predictor variables  $X_1, \dots, X_p$  and all possible split points for these variables, and selects the split that minimises

the total weighted Gini index (James et al., 2021, p. 331). Each split divides a node into two child nodes. This is repeated until no further improvement is possible, or often until some other stopping criterion is met (further details are given in section 2.6).

## 2.6 Random forest

A problem with decision trees according to James et al. (2021, p. 340) is that they generally suffer from high variance, meaning that the resulting model can vary substantially depending on the training set used. James et al. (2021, p.340) propose bagging as a way to reduce the variance. In the context of decision trees for classification, bagging involves training multiple decision trees on the training data.

Random forest is an implementation of bagging. A problem with bagging mentioned by James et al. (2021, p. 344) is that each tree is trained on exactly the same data, which means that if there is a very strong predictor variable probably all trees will use this predictor variable as the top split and therefore look quite similar. This is a problem since the purpose of bagging is to reduce the variance, but this reduction will be quite small if the trees are highly correlated. To address this issue a random forest will only allow a certain amount of all predictor variables to be part of each tree (James et al. 2021, p. 343-344). According to the inventors of random forest, only allowing  $\sqrt{p}$  of all  $p$  predictor variables is recommended for classification, even though the amount of predictor variables used should be viewed as a tuning parameter (Hastie et al., 2009, p. 592) also referred to as a hyperparameter (Lindholm et al., 2022, p. 22). Using only a subset of all predictor variables decorrelates the trees and therefore decreases the variance more effectively compared to using all  $p$  predictor variables for each tree (James et al., 2021, p. 344).

There are multiple hyperparameters used for random forest. One hyperparameter is the number of trees in the random forest. More trees generally reduce the prediction error on test data, however this effect stabilizes at around 200 trees (Hastie et al., 2009, p. 591). Another hyperparameter is the minimum node size, which means the minimum number of observations each node needs to contain (Hastie et al., 2009, p. 308). The minimum node size is a stopping criterion which prevents nodes from becoming too small. Another way of controlling the size of the trees is by imposing a maximum depth (Lyu et al., 2024). The maximum depth determines how deep each tree in the random forest is allowed to grow.

## 2.7 Gradient boosting

The concept of boosting is similar to bagging in that boosting also combines many decision trees. However, unlike bagging, the trees in boosting are grown sequentially, meaning that each new tree is constructed based on information

from the previously fitted trees (James et al., 2021, p. 345). Each new tree is fitted to a modified version of the data that depends on the errors made by the earlier trees.

There are several implementations of boosting. One example is Adaptive Boosting (AdaBoost) (Hastie et al., 2009, p. 337). However, the focus of this thesis is gradient boosting, specifically gradient boosting for multiclass classification.

Gradient boosting uses regression trees, even when doing classification (Chen & Guestrin, 2016, p. 786).  $K$  functions  $F_k(x)$  (one for each class  $1, \dots, K$ ) are used to map an observation  $x_i$  to a real number determined by the leaf nodes in the decision trees that the observation falls into (Chen & Guestrin, 2016, p. 786). This real number is converted to a probability through the function  $p_k(x)$  which by Friedman (2001, p. 1201) is defined as

$$p_k(x) = \frac{e^{F_k(x)}}{\sum_{l=1}^K e^{F_l(x)}} \quad (8)$$

which always returns a number between zero and one. In the beginning of the gradient boosting process  $F_k(x)$  will be put to zero for all classes and all observations, which means that  $p_k(x) = \frac{e^0}{K \cdot e^0} = \frac{1}{K}$  for all classes and observations (Friedman, 2001, p. 1201). This means that all observations will have an equal chance to be classified as each class. Based on the values of  $F_k(x)$  and  $p_k(x)$  the pseudo residuals will be calculated which are the actual values that the decision tree will be trained on (Hastie et al., 2009, p. 360-361). The pseudo residuals are defined for each observation in the dataset and will be the negative gradient of the loss function with respect to  $F_k(x)$  (Hastie et al., 2009, p. 360). The loss function used for multiple classification according to Friedman (2001, p. 1201) is defined as

$$L(\{y_k, F_k(x)\}_1^K) = - \sum_{k=1}^K y_k \log p_k(x) \quad (9)$$

for a single observation  $x$  where  $y_k$  is 1 if  $x$  belongs to class  $k$  and otherwise is 0. In other words for a single observation  $x$  the loss function can be written as  $-\log(p_k(x))$  for the class  $k$  that  $x$  belongs to. The goal is to minimise the loss function for all observations. Friedman (2001, p. 1201) defines the negative gradient of the loss function with respect to  $F_k(x)$  for an observation  $x_i$  as

$$r_{ik} = - \left[ \frac{\partial L(\{y_{il}, F_l(x_i)\}_{l=1}^K)}{\partial F_k(x_i)} \right] \quad (10)$$

which is equal to  $y_{ik} - p_k(x_i)$  according to Friedman (2001, p. 1201). For each class a regression decision tree will be fitted to the pseudo residuals  $r_{ik}$

for that class, which means that the goal of the tree is to predict the pseudo residuals well (Hastie et al., 2009, p. 361).

It is important to note that after the tree is fitted to the pseudo residuals, the value for each node is updated. So instead of the value of a node being the mean of all the observations in the node a different value is assigned to each leaf node (Friedman, 2001, p. 1201). The value  $v_{kn}$  for node  $n$  in the tree that belongs to class  $k$  is calculated through

$$v_{kn} = -\frac{K-1}{K} \frac{\sum_{i \in n} r_{ik}}{\sum_{i \in n} h_{ik}}. \quad (11)$$

according to Friedman (2001, p. 1201), however it is important to note that modern implementations of gradient boosting often remove the factor  $\frac{K-1}{K}$  for example in Chen & Guestrin (2016, p.787). This will be done in this thesis as well.

Since each observation is assigned to a node in each class specific tree this means that each observation has some real value  $v_{kn}$  for all  $k$ . This real value  $v_{kn}$  can now be added to  $F_k(x)$ . To reduce the risk of overfitting, a learning rate  $\eta < 1$  is introduced (James et al., 2021, p.346). Each update  $v_{kn}$  is multiplied by  $\eta$  before being added to  $F_k(x)$ , ensuring that each individual tree has a limited influence on the overall model (James et al., 2021, p.347). One iteration of the gradient boosting algorithm is thus finished by receiving a new updated set of functions  $F_k(x)$ , that can be used to calculate new probabilities  $p_k(x)$  and new pseudo residuals  $r_{ik}$ . The final functions  $F_{kM}(x)$  after  $M$  iterations can be calculated through

$$F_{kM}(x) = F_{k0}(x) + \eta \sum_{m=1}^M v_{k,m}(x) \quad (12)$$

where  $\eta$  is the learning rate, and  $v_{k,m}(x)$  is the update for observation  $x$ , for class  $k$  and iteration  $m$ . The final functions  $F_{kM}(x)$  can then be used to calculate the final probabilities  $p_k(x)$ . An observation is classified as the class  $k$  that has the highest probability  $p_k(x)$ . It is important to note that for each iteration  $p_k(x)$  will (probably) get closer and closer to correctly classify unseen data correctly, which means that the pseudo residuals will get smaller.

The regression decision trees in gradient boosting do binary splits based on the gain criterion (Chen & Guestrin, 2016, p. 787). Chen & Guestrin (2016, p. 787) define gain as

$$\text{Gain} = \frac{1}{2} \left( \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right) - \gamma \quad (13)$$

Here,  $G_L$  and  $G_R$  are the sums of gradients for the left and right child nodes, respectively, while  $H_L$  and  $H_R$  are the corresponding sums of second order gradients. There are two parameters  $\gamma$  and  $\lambda$  as well, which are two non negative constants. If there are any binary split with a positive gain, the binary split with the largest gain will be done (Chen & Guestrin, 2016, p. 787). This is where  $\gamma$  comes in and makes it more difficult to make a split. A larger  $\gamma$  will only allow binary splits that increase the gain significantly. The parameter  $\lambda$  controls so that the weight of a specific node doesn't become too large.

One hyperparameter that can be used for boosting to control the size of the trees is the maximum depth, which controls how deep the trees are allowed to become. The maximum depth is generally smaller for boosting compared to random forest (James et al., 2021, p. 346). Two other hyperparameters are the number of iterations of the boosting algorithm and the learning rate (James et al., 2021, p. 347). If the learning rate is small it is generally necessary to have a large number of iterations (James et al., 2021, p. 347). Boosting generally keeps improving when the number of iterations become larger, even though the improvement is smaller the more iterations there are (Hastie et al., 2009, p. 591). However there is a risk of overfitting if too many iterations are used (James et al., 2021, p. 347). Generally choosing a smaller learning rate improves the performance if there is enough iterations (James et al., 2021, p. 346). The parameters  $\lambda$  and  $\gamma$  mentioned earlier are also two hyperparameters.

## 2.8 Imbalanced data

Data is imbalanced when the classes have a different number of instances (Megahed et al., 2021, p. 1270). This means that some class might be very small while some other class might be very large. This can create problems. An example inspired by Megahed et al. (2021, p. 1270) is if 90 percent of the dataset belongs to one class, a classifier that predicts all observations as this class would have an accuracy of 90 percent, however it wouldn't be a good classifier, since it never classifies any observation as the other classes. There generally also is a small class bias against the classes which makes the model underestimate how often a small class appears (Megahed et al., 2021, p. 1270). In other words even if 10 percent of the dataset belongs to a smaller class, the model will underestimate how often this class appears and might only guess that 1 or 2 percent of observations belongs to this class.

In order to deal with class imbalance there are different methods. One method is undersampling. When undersampling, observations are randomly removed until all classes are equally large (Megahed et al. 2021, p.1270). A problem of undersampling is of course that valuable data is removed and if the class imbalance is large, this might mean removing a majority of the data. Another method is oversampling which means that data is added until all classes are equally large (Megahed et al, 2021, p.1270-1271). This can be done using SMOTE (synthetic majority oversampling technique). When using SMOTE

the  $k$  nearest neighbors for some random observation that belongs to one of the smaller classes is viewed and for one of these neighbors a random point on the line segment between the neighbor and the observation is chosen and added to the dataset (Megahed et al., 2021, p. 1271). This is repeated until all classes are equally large. The constant  $k$  is chosen based on the magnitude of the class imbalance (Megahed et al., 2021). A final way of dealing with class imbalance is through weights (Cui et al., 2019, p. 9268). This means that each observation gets a certain weight based on which class the observation belongs to. Using the inverse class frequency as weights is one option (Cui et al., 2019, p.9268). The inverse class frequency is the kind of weights used in this thesis. The weight for an observation belonging to class  $k$  can then be calculated through

$$w_k = \frac{N}{n_k * K} \quad (14)$$

where  $N$  is the total amount of observations in the dataset,  $n_k$  is the amount of observations in class  $k$  and  $K$  is the amount of classes. The random forest or boosting model will then take these weights into account when training on the data.

### 3 Simulation and modeling

In order to investigate how variations in hyperparameters, class imbalance levels, and methods for handling class imbalance (hereafter referred to as imbalance methods) affect the performance of random forest and gradient boosting (hereafter referred to as boosting), a simulation study is conducted. This section describes the methods used for data simulation and aggregation.

#### 3.1 Data

In order to analyse how random forest and boosting perform, data are required. Using real world data increases the likelihood that the results are relevant for practical applications, which motivates the use of the forest cover type dataset. However, since this dataset has certain characteristics that may influence the results, a second dataset (the handwritten digits dataset) is also included. This allows the methods to be evaluated under different data conditions.

The forest cover type dataset (retrieved from Blackard (1998)) contains observations of seven forest cover types, where each of the 581,012 observations represents a 30 by 30 meter area. The data are collected from four wilderness areas, which are represented by four binary predictor variables. In addition, 40 binary variables indicate the presence of different soil types. The remaining predictor variables are quantitative and are described in Table 1.

| Variable                           | Unit / Values  | Description                                             |
|------------------------------------|----------------|---------------------------------------------------------|
| Elevation                          | meters         | Elevation in meters                                     |
| Aspect                             | azimuth        | Aspect in degrees azimuth                               |
| Slope                              | degrees        | Slope in degrees                                        |
| Horizontal Distance To Hydrology   | meters         | Horizontal distance to nearest surface water features   |
| Vertical Distance To Hydrology     | meters         | Vertical distance to nearest surface water features     |
| Horizontal Distance To Roadways    | meters         | Horizontal distance to nearest roadway                  |
| Hillshade 9am                      | 0 to 255 index | Hillshade index at 9am, summer solstice                 |
| Hillshade Noon                     | 0 to 255 index | Hillshade index at noon, summer solstice                |
| Hillshade 3pm                      | 0 to 255 index | Hillshade index at 3pm, summer solstice                 |
| Horizontal Distance To Fire Points | meters         | Horizontal distance to nearest wildfire ignition points |

Table 1: Attribute description for the forest covertype dataset.

Since the dataset is very large, a subset is used in this study. The two smallest classes are removed so that five classes remain. From these, 4000 observations per class are randomly sampled, resulting in a dataset of 20,000 observations.

The handwritten digits dataset (retrieved from MNIST Database of Handwritten Digits (2021)) contains 70,000 images of handwritten digits. Each image is represented as a 28 by 28 pixel grid, resulting in 784 predictor variables. Each pixel takes a value between 0 and 255, corresponding to the grayscale intensity, where 0 represents black and 255 represents white. The response variable is the digit the picture portrays, with options 0 to 9.

To ensure that the handwritten digits dataset differs structurally from the forest cover type dataset, it is modified to have fewer predictor variables and a weaker relationship between predictors and response. Specifically, only every 60th pixel is retained, reducing the number of predictors to 13. In addition, only digits 0 to 4 are included, with 4000 observations per class, resulting in a dataset comparable in size and class structure to the forest cover type dataset.

### 3.2 Data simulation

In order to investigate how hyperparameters, class imbalance, and imbalance methods affect model performance, a large number of random forest and boost-

ing models are trained using varying hyperparameter settings, imbalance methods, and datasets with different levels of class imbalance.

### 3.2.1 Hyperparameters

For random forest, the number of trees is fixed at 300. Since increasing the number of trees generally improves performance up to a certain point, 300 is chosen as a compromise between computational cost and predictive performance.

The number of predictor variables considered at each split is set to the square root of the total number of predictors, which is the standard recommended value for classification problems.

The splitting criterion is the Gini index, as it is commonly used and well suited for classification tasks.

The minimum node size is one of the hyperparameters that varies between models. The values used are 2, 3, 5, 7, 10, 14, 18, 24, and 30. These values are chosen on an approximately exponential scale, since the marginal effect of increasing minimum node size decreases for larger values.

The maximum depth of the trees also varies between models, with values 5, 8, 11, 14, 17, 20, 23, and 26.

For Boosting, the learning rate and number of boosting rounds are fixed at 0.01 and 3000, respectively. A small learning rate is generally preferred when combined with a sufficiently large number of boosting rounds, as this allows the model to learn more gradually while maintaining stable performance. The chosen values therefore reflect a trade off between model performance and computational cost.

For Boosting, only the maximum depth is varied between models, taking the values 2, 4, 6, 8, 10, 12, and 14.

The hyperparameters  $\gamma$  and  $\lambda$  are not the focus of this study and are therefore fixed at their default values in the implementation of gradient boosting used. The default values are 0 and 1 respectively.

### 3.2.2 Class imbalance

The class sizes for classes 0, 1, 2, 3, and 4 are defined by the expressions  $2000 - 2i$ ,  $2000 - i$ ,  $2000$ ,  $2000 + i$ , and  $2000 + 2i$ , respectively, for different values of  $i$ . This construction ensures that class 0 is always the smallest class and class 4 is always the largest class. The values of  $i$  considered are 0, 500, 800, 900, 950, and 980. These values are chosen to increase in a non-linear manner, since the effect of class imbalance is expected to become more pronounced as the imbalance

becomes more extreme. Importantly, the total number of observations is kept constant at 10,000 across all settings, ensuring that all models are trained on datasets with the same amount of information.

### 3.2.3 Dealing with class imbalance

The different imbalance methods that will be used will be oversampling, under-sampling, weights and also a default option where no way of dealing with class imbalance is used.

### 3.2.4 Data simulation

First, the level of class imbalance and the imbalance method are selected from the 24 possible combinations ( $6 \times 4$ ). A subset of 10,000 observations is then sampled from the full set of 20,000 observations, while preserving the specified class imbalance structure.

The dataset is subsequently split into a training set (80%) and a test set (20%). If undersampling or oversampling is used, the resampling is applied only to the training set.

For each combination of hyperparameters (7 values of maximum depth for boosting, and 8 values of maximum depth combined with 9 values of minimum node size for random forest, resulting in 72 combinations), a model is trained on the training data. If class weights are used, they are incorporated directly into the training process. Each trained model is then evaluated on the test set, and the results are stored in a confusion matrix.

In total, this results in  $6 \times 4 \times 8 \times 9 = 1728$  confusion matrices for Random Forest and  $6 \times 4 \times 7 = 168$  confusion matrices for Boosting.

Since a single run of the simulation may be sensitive to random variation and therefore increase variance in the results, the procedure is repeated multiple times. However, due to computational constraints, the full simulation is not repeated from the beginning for every configuration. Instead, an initial run is used to identify a suitable value for minimum node size based on average performance.

After fixing the minimum node size at this selected value, the simulation is repeated eight additional times. These results are used for the main analysis (see Section 3.3) to determine the optimal maximum depth. Once this is established, a final set of eight simulations is performed where maximum depth is fixed and minimum node size is varied again, allowing for a more detailed analysis of its effect.

This entire simulation procedure is applied to both the forest cover type dataset and the handwritten digits dataset.

### 3.3 Procedure for handling simulation results

Here the procedures that will be used to aggregate and summarize the results obtained from the simulation are presented.

#### 3.3.1 Procedure for the forest coartype results

First, a preliminary analysis of minimum node size is performed. For each value of minimum node size, the mean and median balanced accuracy are calculated across all model results obtained with that value, and the results are summarised in a table and visualised in a plot.

After the eight simulations with fixed minimum node size have been completed, a separate analysis is performed for each of the three remaining variables: maximum depth, class imbalance, and imbalance method. When one variable is analysed, the results are grouped by all combinations of the other two variables. In other words, the data are divided into multiple subdatasets, one for each combination of the remaining variables. For each subdataset, the mean balanced accuracy and mean F1-score are calculated for each value of the variable of interest. The median is also calculated, but only for comparison with the mean.

For each variable, a table is created for each class, including the average across classes, and for each of the two performance measures, balanced accuracy and F1-score. Each row in a table corresponds to one subdataset. For the lowest value of the variable of interest, the mean value is stored directly in the corresponding column. For all other values, the difference between that mean and the mean for the lowest value is stored instead. This makes it easier to compare how balanced accuracy and F1-score change across values of the variable. Each value of the variable is therefore represented by one column.

In addition, the tables include a column indicating whether the difference between the mean and median exceeds 0.005 in balanced accuracy or F1-score, depending on which measure is under consideration. If such a difference occurs, another column indicates whether the mean or the median is larger. The tables also include columns specifying the class and the performance measure under consideration.

The same procedure is repeated for minimum node size after the third simulation, where maximum depth is fixed.

The tables are primarily used to provide an overview of the results and to identify patterns that may be worth examining further. When a pattern is observed, it is illustrated using plots, which are included in the Results section,

whereas the corresponding tables are placed in the appendix. These plots are shown only for balanced accuracy, for the sake of simplicity. However, if the F1-score differs substantially from balanced accuracy for a particular result, this is noted in the Results section.

After these comparisons, the top 10 models for random forest and boosting, ranked by mean balanced accuracy and later also by mean F1-score, are collected in tables. A separate table is created for each imbalance level and each performance measure. The median for each model is also included in these tables.

Finally, the mean and median balanced accuracy for each class are calculated across the results from the initial analysis of the balanced datasets in order to determine whether some classes are consistently easier or harder to classify, even when the dataset is balanced.

It is important to note that for some models, F1-score is undefined. In cases where an F1 value is NA, a row is added to the table to make it clear which scenarios contain missing values.

### 3.3.2 Procedure for the handwritten digits results

For the handwritten digits dataset, only the tables for the top 10 models and the table showing differences in class wise performance on the balanced dataset will be retained. Otherwise, only the same plots created for the forest cover type dataset will be produced for the handwritten digits dataset, since these were the most interesting results for comparison. The first minimum node size plot and the maximum depth plots will be used to select suitable values for the second and third simulations, respectively, for the handwritten digits dataset.

## 4 Results

The results from the simulation and modeling are presented in this section. Important to notice is that in most of the plots presented in this section the unit of the y-axis is not the balanced accuracy, but rather the effect on the balanced accuracy compared to a base level. The balanced accuracy of the base level is unique for every line in each single plot. The results for the F1-score will only be mentioned in case they differ systematically from the results for the balanced accuracy.

### 4.1 Random forest

Here the results for the random forest models are presented.

#### 4.1.1 Minimum node size, first round

Across all models, differences in balanced accuracy between different minimum node sizes are small, both in terms of mean and median values (Figure 3). The mean is generally slightly lower than the median, indicating some outliers and reduced robustness. Overall, there is a weak tendency for smaller minimum node sizes to yield slightly higher balanced accuracy, but the effect is minor.

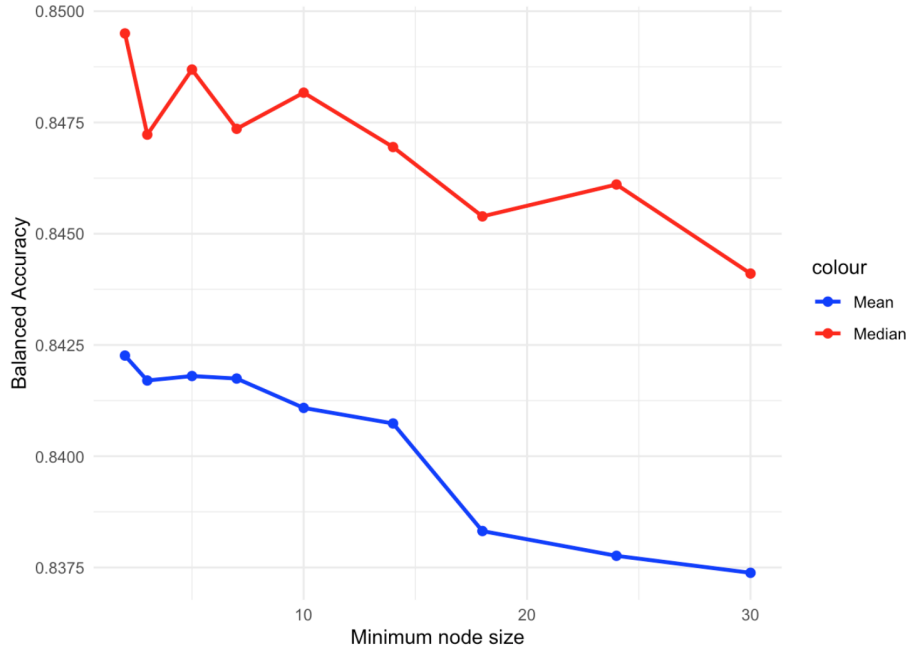


Figure 3: Median and mean balanced accuracy across all models for each value of the minimum node size in the forest covertype dataset.

A similar pattern is observed for the handwritten digits dataset (Figure 4), where no clear relationship between minimum node size and balanced accuracy is present. Some variability between mean and median indicates outliers, but no consistent trend is visible.

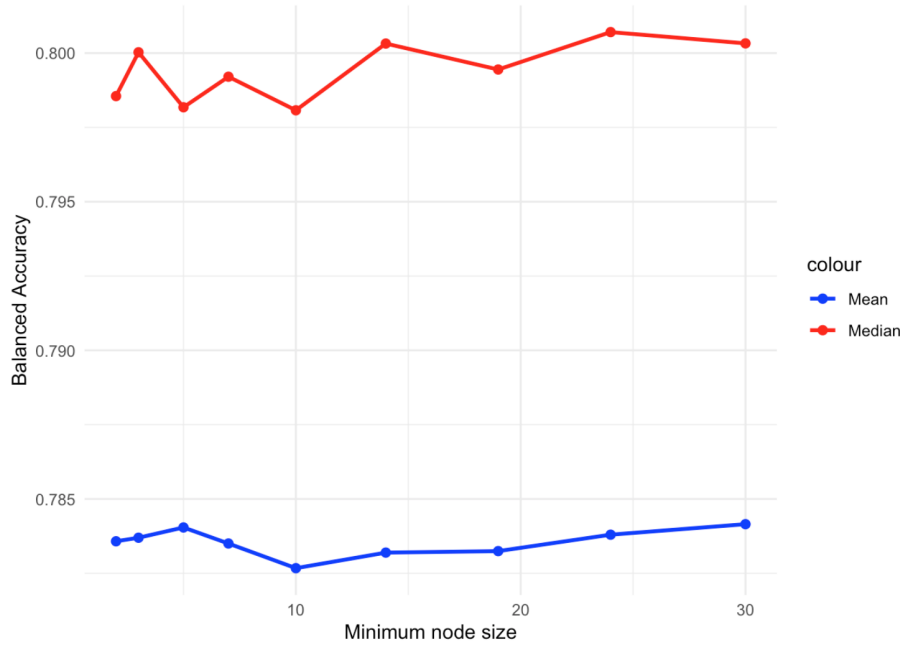


Figure 4: Median and mean balanced accuracy across all models for each value of the minimum node size in the handwritten digits dataset.

Since minimum node size has little effect in both datasets, a value of 2 is used in the second simulation for both datasets to ensure that the maximum depth constraint becomes the primary factor controlling tree complexity, making it easier to isolate and analyse the effect of maximum depth on balanced accuracy.

#### 4.1.2 Maximum depth

These results are obtained using a minimum node size of 2. Overall, increasing maximum depth generally leads to higher balanced accuracy (Figure 5.1), although the gains diminish at higher depths. In several cases, a maximum depth of 26 does not yield the best performance (Figure 5.4). However, under oversampling, maximum depth 26 consistently achieves the highest balanced accuracy, although the improvement over depth 23 is negligible (Figure 5.3). The relationship between maximum depth and balanced accuracy appears approximately logarithmic, both overall (Figure 5.1) and under oversampling (Figure 5.3). A similar pattern is observed for more balanced datasets (imbalance levels 1000 and 2000), where depth 26 performs best (Figure 5.2). In contrast, for highly imbalanced datasets with weighting or undersampling, slightly smaller depths than 26 occasionally yield better performance (Figure 5.4). Figure 5.1 further shows that the effect of maximum depth is strongest when no imbalance method is used, followed by oversampling. Weights and undersampling show weaker sensitivity to depth.

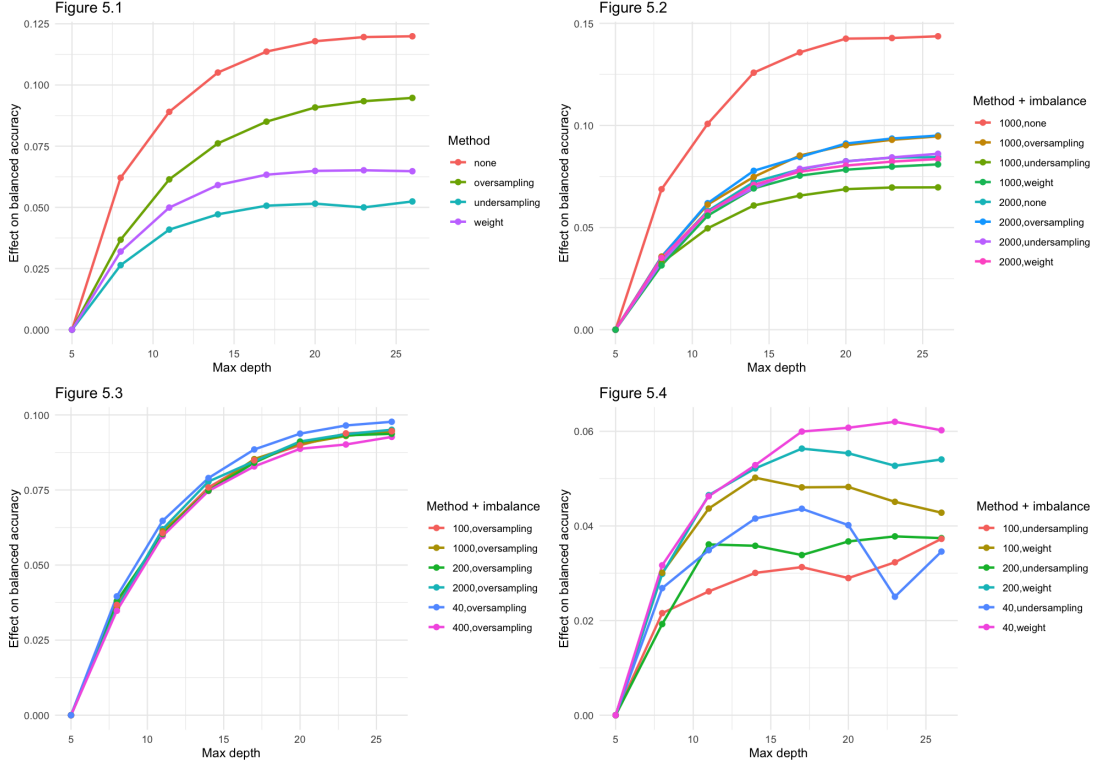


Figure 5: Effect of maximum depth on balanced accuracy compared to baseline level 5, plotted against maximum depth for different combinations of imbalance level and imbalance methods, for the forest cover type dataset. In Figure 5.1, the mean effect on balanced accuracy across the six imbalance levels is shown for each imbalance method. In figure 5.2 the results with an imbalance level of 1000 or 2000 have been plotted. In figure 5.3 all results with oversampling are shown. In figure 5.4 all results with undersampling or weights and an imbalance of 40, 100 or 200 are plotted.

In Figure 6, several differences compared with Figure 5 can be observed. The effect of maximum depth on balanced accuracy is substantially smaller for the handwritten digits dataset (Figure 6.1), although a general positive trend remains. However, multiple exceptions occur, particularly at higher maximum depths (Figure 6.4). No clear performance differences are observed between maximum depths from 11 to 26 (Figures 6.1–6.3). For very imbalanced datasets combined with weighting or undersampling, depths around 8 or 11 sometimes outperform depth 26, although differences remain small (Figure 6.4). As in the forest cover type dataset, the strongest effect of maximum depth is observed when no imbalance method is used, followed by oversampling. Weights and undersampling show weaker effects overall. For the third simulation of the handwritten digits dataset, maximum depth 26 is selected, as it yields among the highest balanced accuracies while allowing minimum node size to act as the primary stopping criterion.

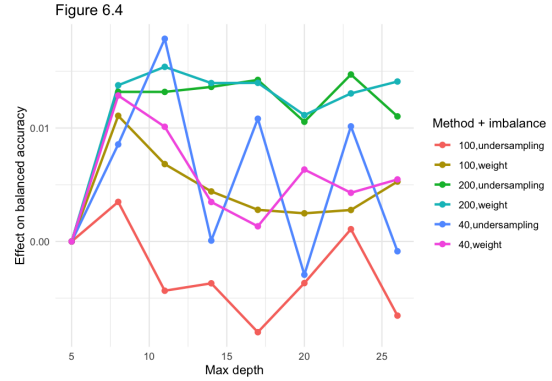
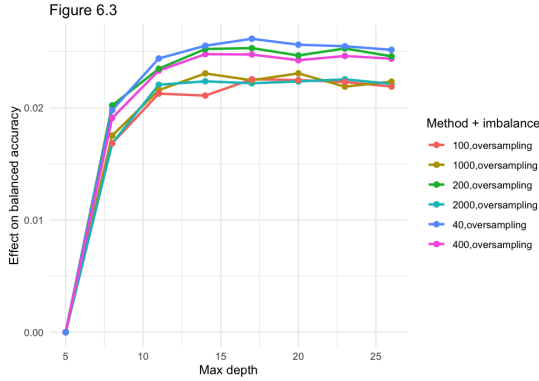
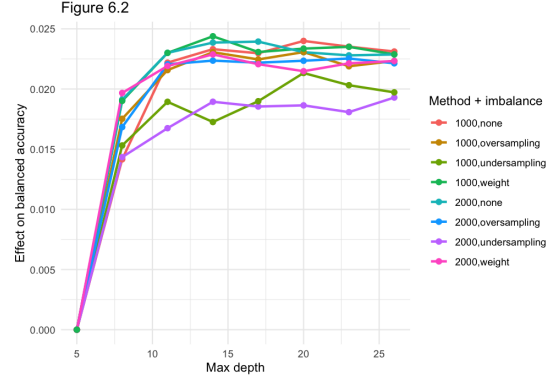
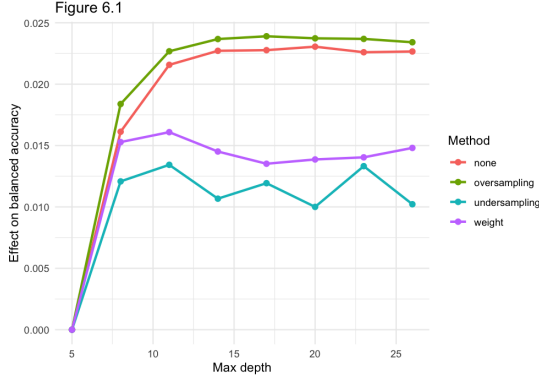


Figure 6: Effect of maximum depth on balanced accuracy compared to baseline level 5, plotted against maximum depth for different combinations of imbalance level and imbalance methods, for the handwritten digits dataset. In Figure 6.1, the mean effect on balanced accuracy across the six imbalance levels is shown for each imbalance method. In figure 6.2 the results with an imbalance level of 1000 or 2000 have been plotted. In figure 6.3 all results with oversampling are shown. In figure 6.4 all results with undersampling or weights and an imbalance of 40, 100 or 200 are plotted.

#### 4.1.3 Minimum node size, second round

These results are obtained using a maximum depth of 26. As shown in Figure 7, minimum node size has only a minor effect on balanced accuracy. Overall, smaller values tend to perform slightly better, with values of 2 or 3 giving the highest scores. However, the differences are very small and show no consistent pattern, suggesting that observed variations are mainly due to noise and outliers.

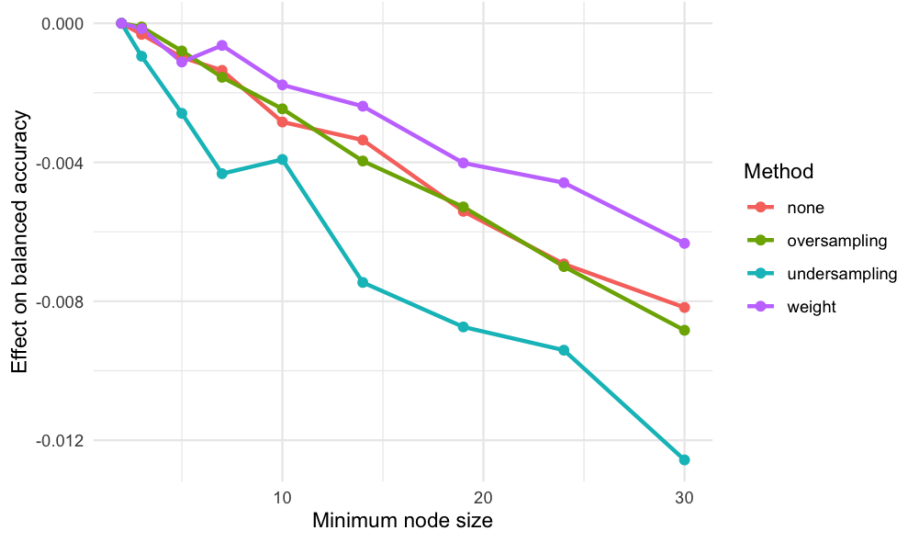


Figure 7: The average effect on balanced accuracy for different values of minimum node size compared to minimum node size 2, for the 6 models (for the 6 different imbalance level) with the same imbalance method, for the forest cover type dataset.

When comparing Figures 7 and 8, similar patterns are observed across datasets, but with even smaller effects in Figure 8. For oversampling and no imbalance method, minimum node size has no noticeable effect on balanced accuracy. For undersampling, a weak negative relationship is observed, where larger minimum node sizes slightly reduce performance, although the effect is smaller than in Figure 7. In contrast, weighting shows a small positive relationship, where larger minimum node sizes slightly improve balanced accuracy.

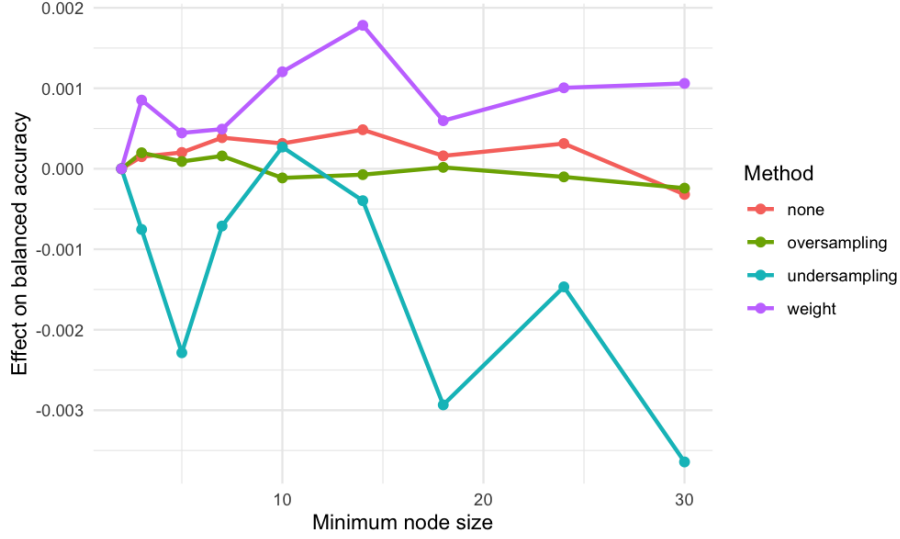


Figure 8: The average effect on balanced accuracy for different values of minimum node size compared to minimum node size 2, for the 6 models (for the 6 different imbalance level) with the same imbalance method, for the handwritten digits dataset.

#### 4.1.4 Class imbalance

These results are obtained using a minimum node size of 2. As shown in Figure 9.1, more balanced datasets generally yield higher balanced accuracy. For oversampling, however, balanced accuracy is largely insensitive to imbalance level, except at maximum depth 5, where more balanced datasets show a slight advantage (Figure 9.2). Undersampling exhibits a stronger dependence on class imbalance than the other methods (Figure 9.1). The same figure also shows that, at an imbalance level of 2000, the effect of class imbalance is slightly larger for no imbalance method than for weights, whereas the opposite holds for the remaining imbalance levels. For weighting, Figure 9.3 shows that an imbalance level of 1000 generally performs better than 2000. For smaller maximum depths, an imbalance level of 400 also yields higher balanced accuracy than both 1000 and 2000. Overall, the effect of class imbalance under weighting increases with maximum depth (Figure 9.3). However, this trend is not consistent for F1-score, where the effect is larger at smaller maximum depths. Therefore, results in Figure 9.3 should be interpreted with caution.

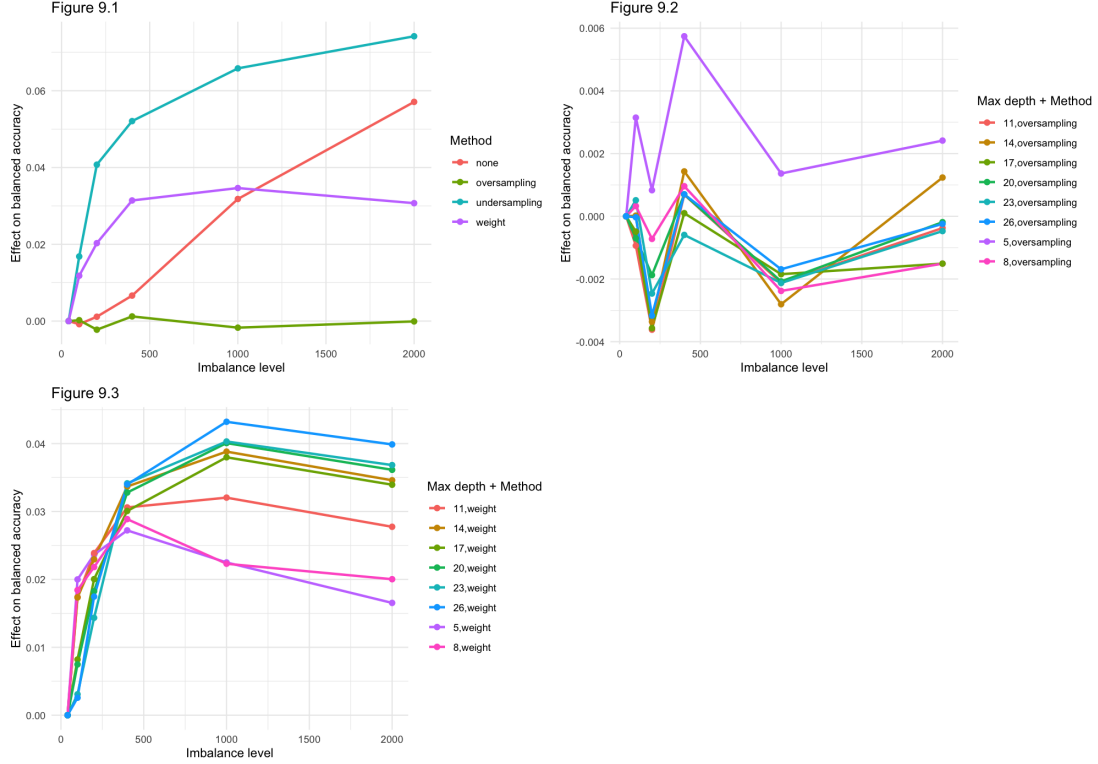


Figure 9: Effect of imbalance level on balanced accuracy compared to baseline level 40, plotted against imbalance level for different combinations of maximum depth and imbalance methods, for the forest cover type dataset. In figure 9.1 the mean effect on balanced accuracy across the eight different maximum depths is shown for each imbalance method. In figure 9.2 the results with oversampling are shown. In figure 9.3 all results with weights can be seen.

When comparing the forest cover type dataset and the handwritten digits dataset, several differences emerge. As shown in Figure 10.1 compared to Figure 9.1, the effect of class imbalance under undersampling is smaller for the handwritten digits dataset. In contrast, the effect of class imbalance under no imbalance method is larger for the handwritten digits dataset. A similarity between datasets is that oversampling shows very limited sensitivity to class imbalance (Figure 10.1). However, Figure 10.2 suggests a slight decrease in balanced accuracy as datasets become more balanced, except at maximum depth 5. This effect is mainly driven by the comparison with imbalance level 40, otherwise differences between imbalance levels are small. For weighting, no consistent pattern is observed between imbalance levels 400, 1000, and 2000 in the handwritten digits dataset (Figure 10.3), which contrasts with the forest cover type dataset (Figure 9.3), where 400 and 1000 generally outperform 2000.

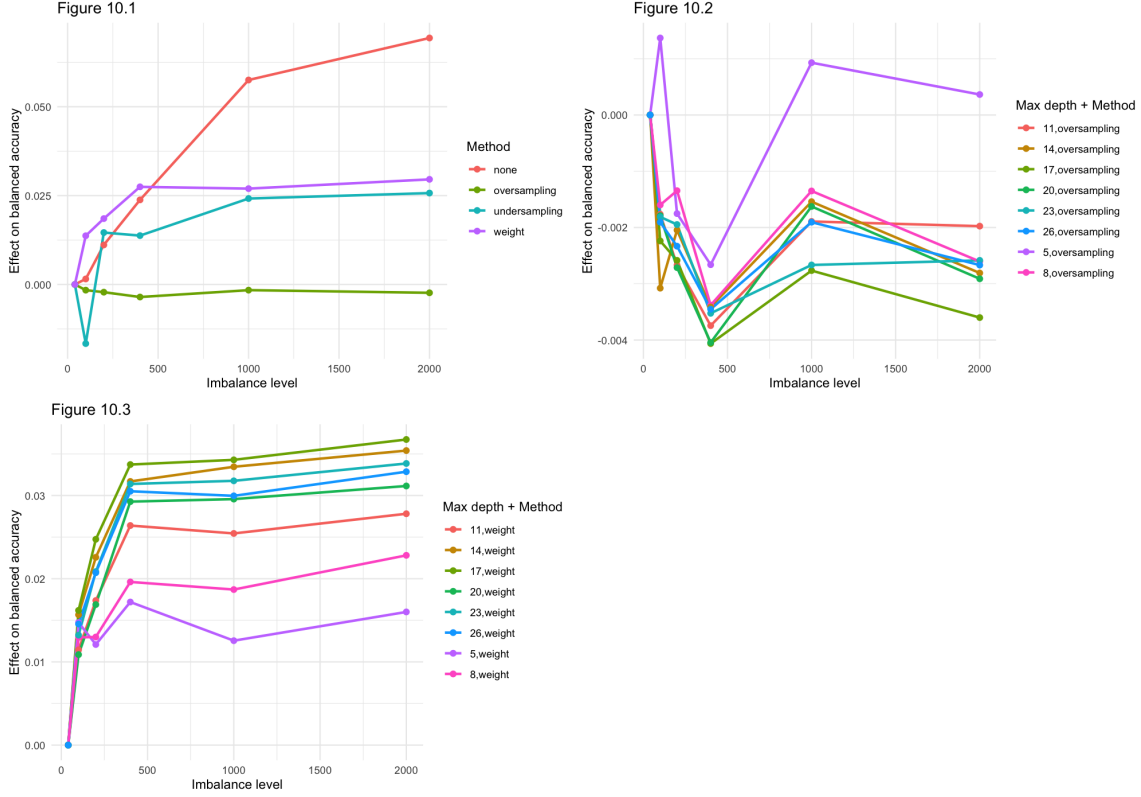


Figure 10: Effect of imbalance level on balanced accuracy compared to baseline level 40, plotted against imbalance level for different combinations of maximum depth and imbalance methods, for the handwritten digits dataset. In figure 10.1 the mean effect on balanced accuracy across the eight different maximum depths is shown for each imbalance method. In figure 10.2 the results with oversampling are shown. In figure 10.3 all results with weights can be seen.

#### 4.1.5 Imbalance methods

These results are obtained using a minimum node size of 2. Results for fully balanced datasets are not of primary interest, since observed differences mainly reflect variation in the test set rather than the effect of imbalance methods. Overall, oversampling tends to yield the highest balanced accuracy, as shown in Figures 11.1 and 11.2. For more balanced datasets and low maximum depth values, weights and oversampling perform similarly. Undersampling generally yields slightly higher balanced accuracy than using no imbalance method when maximum depth is small, and substantially higher when maximum depth is 5 (see Figure 11.1). For larger maximum depths, however, using no imbalance method tends to perform better. For more imbalanced datasets (smallest class sizes of 40 or 100), no imbalance method generally yields higher balanced accuracy (Figure 11.2), except when maximum depth is 5, and when the smallest

class size is 100 with maximum depth 8 (Figure 11.3). Overall, the influence of imbalance method is stronger for smaller maximum depths, as shown in Figure 11.1.

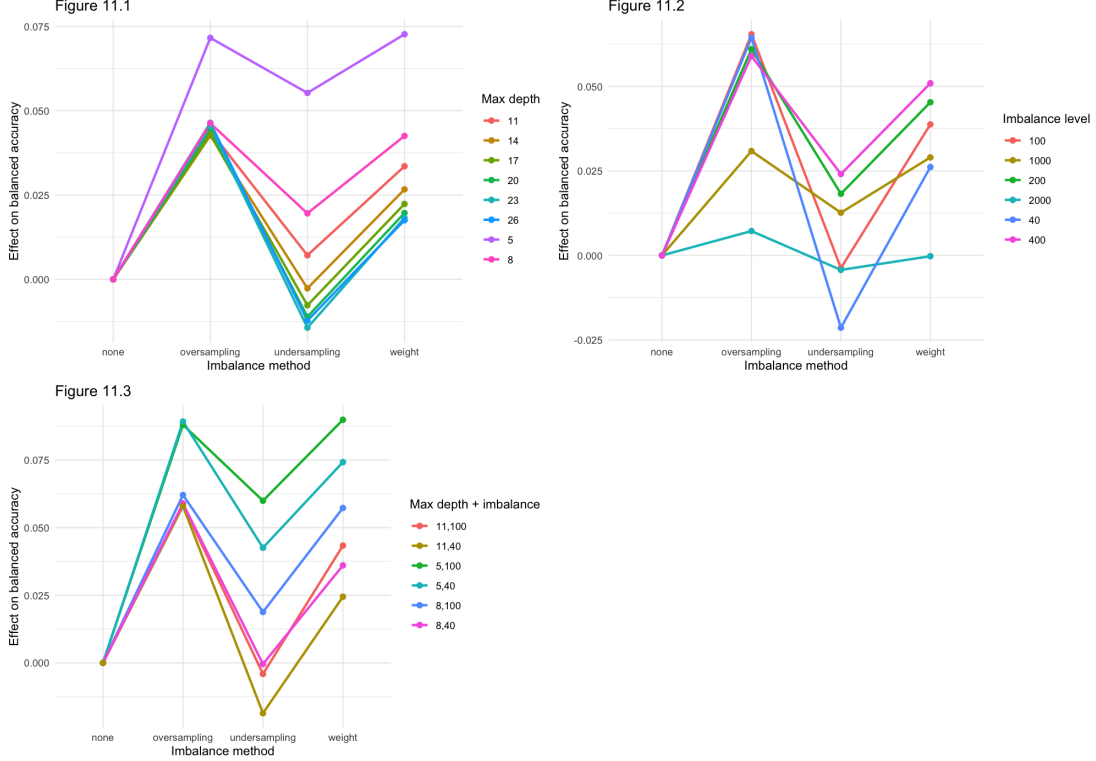


Figure 11: Effect of imbalance method on balanced accuracy compared to baseline level none, plotted against imbalance method for different combinations of imbalance level and maximum depths, for the forest cover type dataset. In figure 11.1 the mean effect on balanced accuracy across the 6 different imbalance levels is shown for each maximum depth. In figure 11.2 the mean effect on balanced accuracy across the 8 different maximum depths is shown for each imbalance level. In figure 11.3 all results with maximum depth of 5, 8 or 11 and an imbalance of 40 or 100 have been plotted.

When comparing the forest cover type dataset and the handwritten digits dataset, several differences emerge. In Figures 12.1 and 12.3, undersampling generally yields higher balanced accuracy than using no imbalance method, regardless of maximum depth. This was not observed in Figures 11.1 and 11.3. Across both imbalance levels and maximum depths, oversampling consistently achieves the highest balanced accuracy (Figure 12). A further difference compared to Figure 11 is that oversampling clearly outperforms weights even at smaller maximum depths (Figures 12.1 and 12.3). For highly imbalanced datasets, undersampling also achieves higher balanced accuracy than using no imbalance method (Figure 12.2).

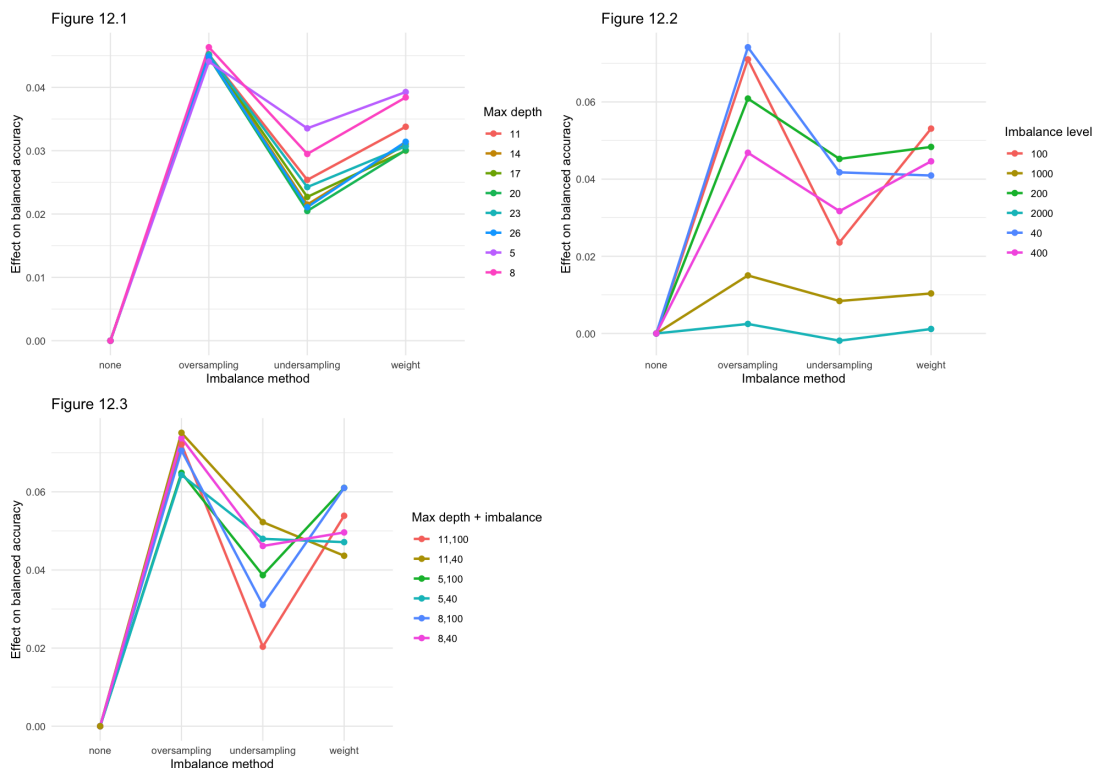


Figure 12: Effect of imbalance method on balanced accuracy compared to baseline level none, plotted against imbalance method for different combinations of imbalance level and maximum depths, for the handwritten digits dataset. In figure 12.1 the mean effect on balanced accuracy across the 6 different imbalance levels is shown for each maximum depth. In figure 12.2 the mean effect on balanced accuracy across the 8 different maximum depths is shown for each imbalance level. In figure 12.3 all results with maximum depth of 5, 8 or 11 and an imbalance of 40 or 100 have been plotted.

## 4.2 Gradient boosting

### 4.2.1 Maximum depth

There is generally a slightly higher balanced accuracy for larger maximum depths, although the differences between maximum depths 8, 10, 12, and 14 are very small in most cases, and occasionally also for 6, as shown in Figure 13.1. Overall, the increase in balanced accuracy appears to follow an approximately logarithmic pattern, as shown in Figure 13.1. The effect of maximum depth is also larger for more balanced datasets, as shown in Figure 13.3. However, for oversampling, there is no clear difference in the effect across imbalance levels, as shown in Figure 13.2. For undersampling, maximum depth has almost no effect on balanced accuracy when the class imbalance is large, as shown in Figure 13.4. Overall, the effect of maximum depth is largest when oversampling

is used, as shown in Figure 13.1.

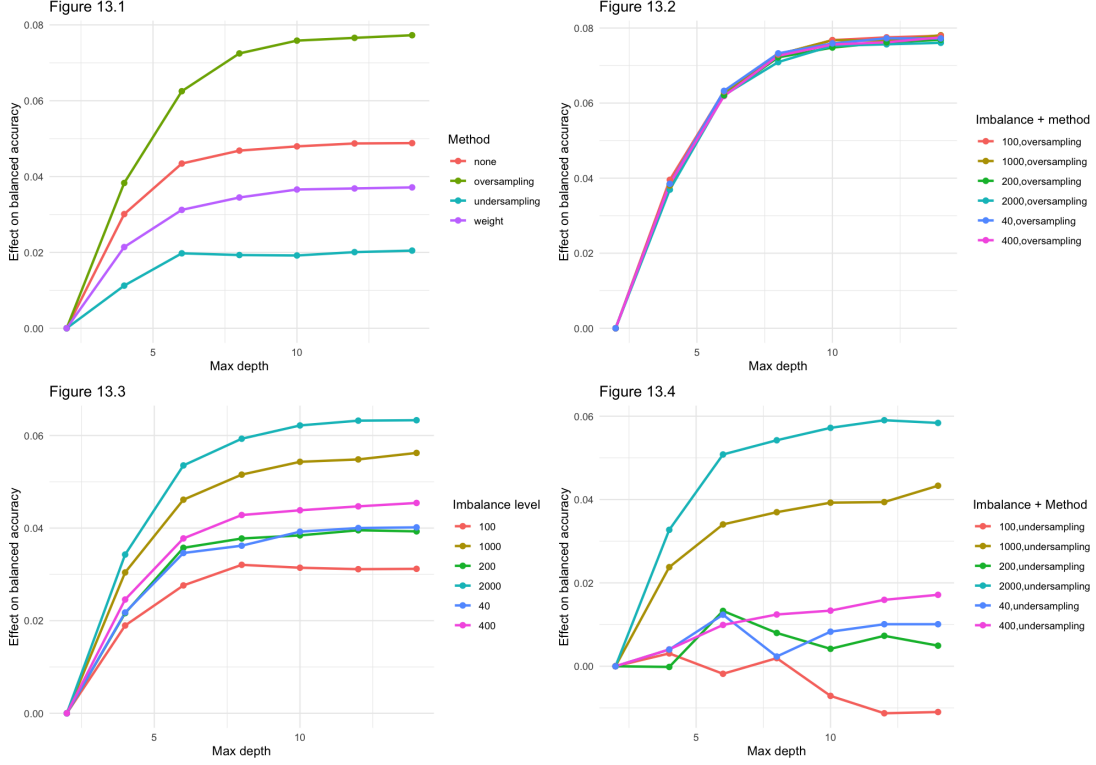


Figure 13: Effect of maximum depth on balanced accuracy compared to baseline level 2, plotted against maximum depth for different combinations of imbalance and imbalance methods, for the forest cover type dataset. In figure 13.1 the mean effect on balanced accuracy across the 6 different imbalance levels is shown for each imbalance method. In figure 13.2 all results with oversampling have been plotted. In figure 13.3 the mean effect on balanced accuracy across the 4 different imbalance methods is shown for each imbalance level. In figure 13.4 all results with undersampling have been plotted.

Some notable differences emerge when comparing Figures 13 and 14. In Figure 14.1, the maximum depth that generally yields the highest balanced accuracy is around 4, while larger maximum depths tend to result in lower balanced accuracy. In Figure 14.2, however, only a maximum depth of 14 results in clearly lower balanced accuracy than a maximum depth of 2, whereas maximum depths of 4 or 6 generally achieve the highest performance. The effect of maximum depth is also substantially smaller in Figure 14 than in Figure 13. In Figure 14.3, smaller maximum depths (approximately 2–4) tend to perform better for more imbalanced datasets, while slightly larger maximum depths (approximately 4–6) perform better for more balanced datasets. The same pattern is also visible in Figure 14.4 for undersampling. However, this pattern is not observed in Figure 14.2, suggesting that it does not hold for oversampling.

Finally, in Figure 14.1, the negative effect of large maximum depths appears stronger for weights and undersampling than for using no imbalance method or oversampling.

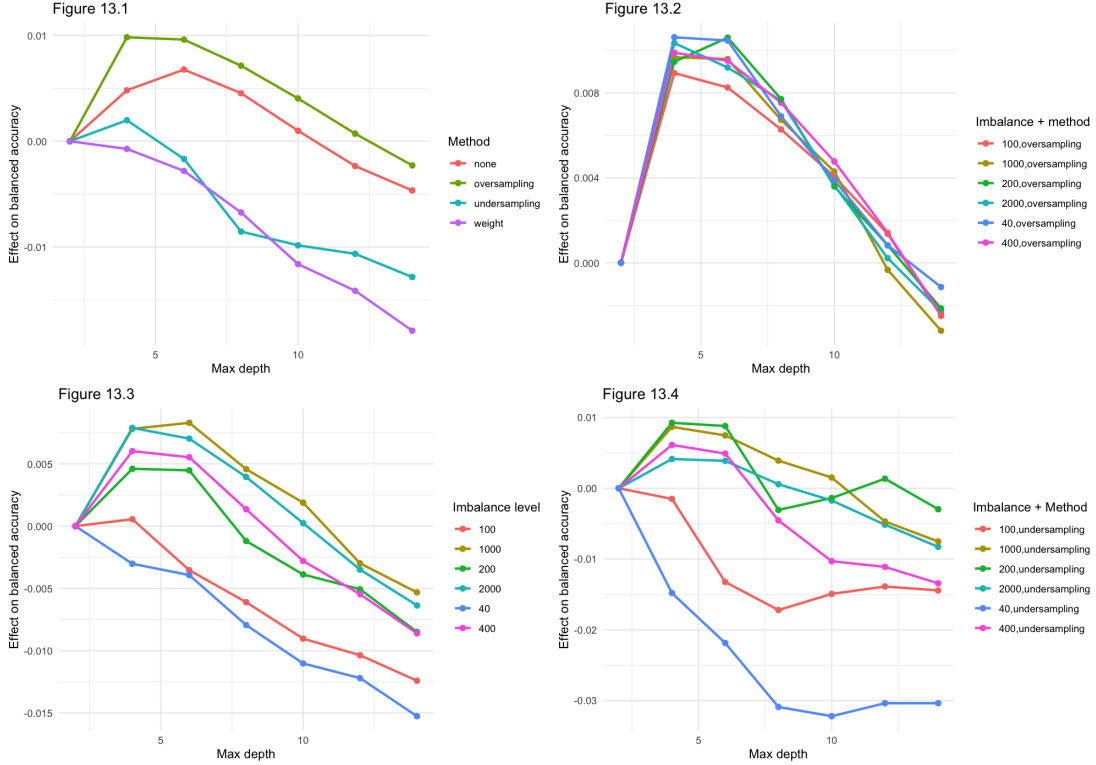


Figure 14: Effect of maximum depth on balanced accuracy compared to baseline level 2, plotted against maximum depth for different combinations of imbalance and imbalance methods, for the handwritten digits dataset. In figure 14.1 the mean effect on balanced accuracy across the 6 different imbalance levels is shown for each imbalance method. In figure 14.2 all results with oversampling have been plotted. In figure 14.3 the mean effect on balanced accuracy across the 4 different imbalance methods is shown for each imbalance level. In figure 14.4 all results with undersampling have been plotted.

#### 4.2.2 Class imbalance

Overall, more balanced datasets generally lead to higher balanced accuracy. However, for oversampling there is no clear difference in balanced accuracy across imbalance levels, as shown in Figure 15.1. When using weights and small maximum depths, the highest balanced accuracy is obtained when the smallest class size is 1000 rather than 2000, as shown in Figure 15.2. For undersampling, the effect of class imbalance is larger compared to the other methods, as illustrated in Figure 15.1. This effect also increases with larger maximum depth, as shown in Figure 15.3. However, this pattern is not observed for oversam-

pling, as shown in Figure 15.4. For the F1-score, a systematic difference is that when using weights, larger maximum depths do not consistently correspond to a stronger effect of class imbalance. Since this differs from the results for balanced accuracy, this finding should be interpreted with more caution.

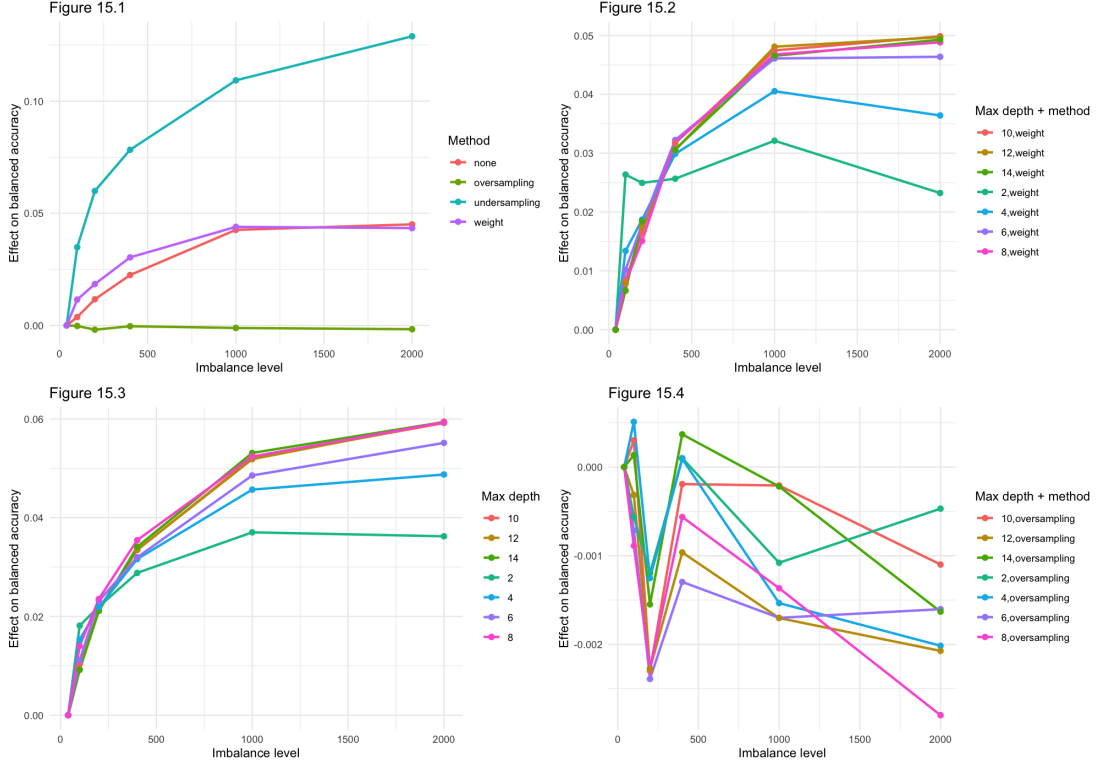


Figure 15: Effect of imbalance level on balanced accuracy compared to baseline level 40, plotted against imbalance level for different combinations of maximum depth and imbalance methods, for the forest cover type dataset. In figure 15.1 the mean effect on balanced accuracy across the 7 different maximum depths is shown for each imbalance method. In figure 15.2 all results with weights have been plotted. In figure 15.3 the mean effect on balanced accuracy across the 4 different imbalance method is shown for each maximum depth. In figure 15.4 the results with oversampling have been plotted.

The results in Figure 16 are broadly similar to those in Figure 15. One difference is that the effect of changing imbalance level under undersampling is substantially larger in Figure 15.1 than in Figure 16.1. In addition, Figure 16.4 suggests that imbalance levels of 200 or 400 with oversampling may result in slightly lower balanced accuracy than the other imbalance levels. Apart from these differences, the results in Figures 15 and 16 are largely consistent.

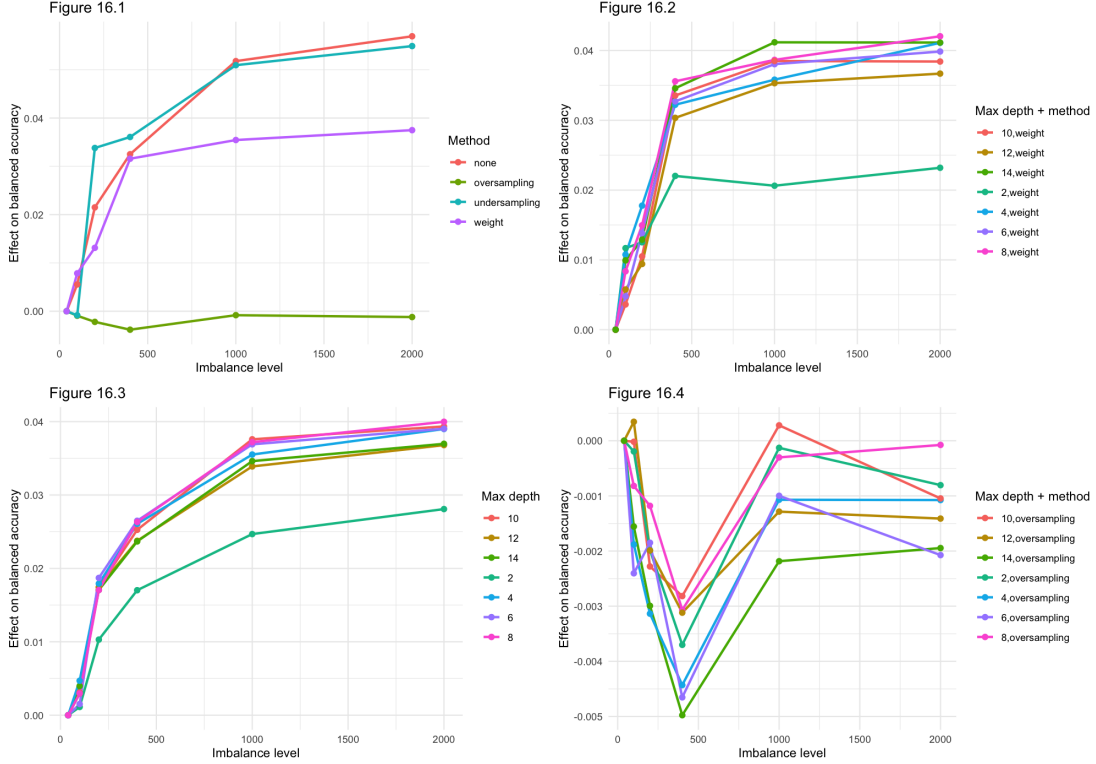


Figure 16: Effect of imbalance level on balanced accuracy compared to baseline level 40, plotted against imbalance level for different combinations of maximum depth and imbalance methods, for the handwritten digits dataset. In figure 16.1 the mean effect on balanced accuracy across the 7 different maximum depths is shown for each imbalance method. In figure 16.2 all results with weights have been plotted. In figure 16.3 the mean effect on balanced accuracy across the 4 different imbalance method is shown for each maximum depth. In figure 16.4 the results with oversampling have been plotted.

#### 4.2.3 Imbalance method

A major limitation of these results is that the results for the completely balanced datasets are less informative, since the observed differences mainly depend on the test set not being perfectly balanced, even though the full dataset is balanced. As shown in Figures 17.1 and 17.2, oversampling generally yields the highest balanced accuracy, while undersampling generally yields the lowest, often below the performance of using no imbalance method. Weights typically achieve higher balanced accuracy than using no imbalance method, as also seen in Figures 17.1 and 17.2. Figure 17.1 shows that for a maximum depth of 2, weights and oversampling achieve approximately the same balanced accuracy. For smaller maximum depths, the difference between these two methods is small, whereas for larger maximum depths oversampling performs substantially better than weights. The advantage of oversampling also increases with larger maxi-

mum depths, as shown in Figure 17.1. In addition, Figure 17.2 indicates that the benefit of oversampling grows as datasets become more imbalanced. In contrast, the disadvantage of undersampling increases both with higher imbalance levels and larger maximum depths, as shown in Figures 17.1 and 17.2.

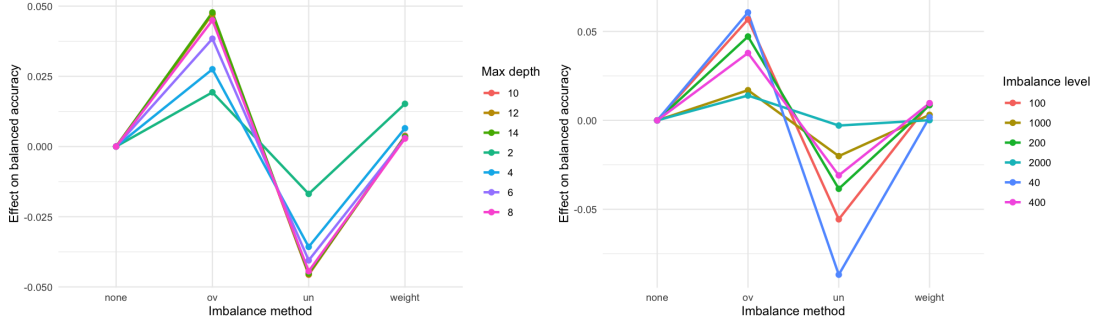


Figure 17: Effect of imbalance method on balanced accuracy compared to baseline level none, plotted against imbalance method for different combinations of imbalance and maximum depths, for the forest cover type dataset. In figure 17.1 the mean effect on balanced accuracy across the 6 imbalance levels is shown for each maximum depth. In figure 17.2 the mean effect on balanced accuracy across the 7 maximum depths is shown for each imbalance level.

When comparing Figure 17 with Figure 18, several differences emerge. In Figures 18.1 and 18.2, undersampling and using no imbalance method yield very similar balanced accuracy, which contrasts with Figures 17.1 and 17.2. Figure 18.1 further suggests that for smaller maximum depths, both weights and undersampling provide a larger positive effect on balanced accuracy than for larger maximum depths. In addition, Figure 18 indicates that the positive effect of weights is somewhat stronger for the handwritten digits dataset than for the forest cover type dataset. Figure 18.2 also suggests that the benefit of oversampling increases with higher imbalance levels. Overall, the patterns in Figure 18 are consistent with those in Figure 17, although differences between imbalance methods are less pronounced for the handwritten digits dataset.

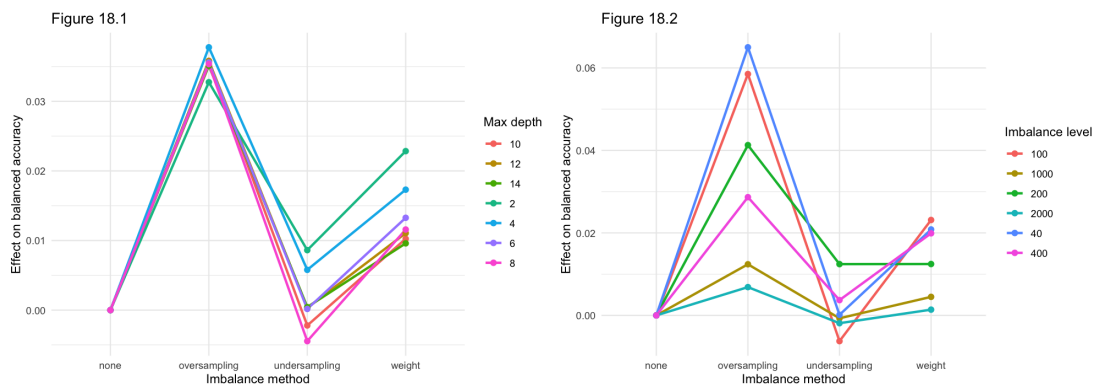


Figure 18: Effect of imbalance method on balanced accuracy compared to baseline level none, plotted against imbalance method for different combinations of imbalance and maximum depths, for the handwritten digits dataset. In figure 18.1 the mean effect on balanced accuracy across the 6 imbalance levels is shown for each maximum depth. In figure 18.2 the mean effect on balanced accuracy across the 7 maximum depths is shown for each imbalance level.

### 4.3 Comparison between models

In general, oversampling and larger maximum depths yield the highest balanced accuracy for both random forest and boosting, as shown in Table 2. However, for more balanced datasets, larger maximum depth appears to be more strongly associated with high balanced accuracy than oversampling. In these cases, the different imbalance methods perform similarly to oversampling as long as a sufficiently large maximum depth is used. For more imbalanced datasets, oversampling instead shows a stronger association with high balanced accuracy than maximum depth.

| Imbalance level 40   |                 |                 |           |            |           | Imbalance level 100  |    |                  |                 |           |            |           |              |
|----------------------|-----------------|-----------------|-----------|------------|-----------|----------------------|----|------------------|-----------------|-----------|------------|-----------|--------------|
|                      | desc_rf         | desc_boost      | mean_rf   | mean_boost | median_rf | median_boost         |    | desc_rf          | desc_boost      | mean_rf   | mean_boost | median_rf | median_boost |
| 2                    | 26,oversampling | 12,oversampling | 0.8973743 | 0.9318160  | 0.8978393 | 0.9326332            | 22 | 26,oversampling  | 14,oversampling | 0.8973449 | 0.9319079  | 0.8979774 | 0.9311684    |
| 8                    | 23,oversampling | 14,oversampling | 0.8961439 | 0.9317728  | 0.8964314 | 0.9325324            | 18 | 23,oversampling  | 12,oversampling | 0.8966558 | 0.9315005  | 0.8972164 | 0.9315133    |
| 4                    | 20,oversampling | 10,oversampling | 0.8934150 | 0.9304497  | 0.8943069 | 0.9314301            | 14 | 20,oversampling  | 10,oversampling | 0.8927125 | 0.9307519  | 0.8933358 | 0.9302423    |
| 0                    | 17,oversampling | 8,oversampling  | 0.8881530 | 0.9278045  | 0.8892473 | 0.9287529            | 10 | 17,oversampling  | 8,oversampling  | 0.8876658 | 0.9269169  | 0.8896653 | 0.9267430    |
| 6                    | 14,oversampling | 6,oversampling  | 0.8786327 | 0.9178087  | 0.8787032 | 0.9179282            | 6  | 14,oversampling  | 6,oversampling  | 0.8786533 | 0.9170948  | 0.8789601 | 0.9183121    |
| 2                    | 11,oversampling | 4,oversampling  | 0.8643922 | 0.8930267  | 0.8647426 | 0.8930044            | 2  | 11,oversampling  | 4,oversampling  | 0.8634564 | 0.8935378  | 0.8636213 | 0.8932749    |
| 0                    | 23,weight       | 14,weight       | 0.8466829 | 0.8618016  | 0.8438322 | 0.8603020            | 8  | 14,weight        | 10,weight       | 0.8548571 | 0.8692089  | 0.8610063 | 0.8691948    |
| 6                    | 20,weight       | 12,weight       | 0.8454206 | 0.8611705  | 0.8420381 | 0.8602329            | 16 | 20,weight        | 12,weight       | 0.8529011 | 0.8689771  | 0.8548609 | 0.8678178    |
| 4                    | 26,weight       | 10,weight       | 0.8448922 | 0.8608428  | 0.8424341 | 0.8608823            | 12 | 17,weight        | 14,weight       | 0.8528188 | 0.8684443  | 0.8537645 | 0.8678001    |
| 2                    | 17,weight       | 10,none         | 0.8446159 | 0.8606871  | 0.8388069 | 0.8608786            | 20 | 23,weight        | 8,weight        | 0.8497613 | 0.8672989  | 0.8500601 | 0.8640624    |
| Imbalance level 200  |                 |                 |           |            |           | Imbalance level 400  |    |                  |                 |           |            |           |              |
|                      | desc_rf         | desc_boost      | mean_rf   | mean_boost | median_rf | median_boost         |    | desc_rf          | desc_boost      | mean_rf   | mean_boost | median_rf | median_boost |
| 2                    | 26,oversampling | 14,oversampling | 0.8942068 | 0.9302236  | 0.8934919 | 0.9304773            | 22 | 26,oversampling  | 14,oversampling | 0.8980748 | 0.9321418  | 0.8973804 | 0.9330242    |
| 8                    | 23,oversampling | 12,oversampling | 0.8936810 | 0.9295420  | 0.8927746 | 0.9302510            | 18 | 23,oversampling  | 12,oversampling | 0.8955502 | 0.9308534  | 0.8957442 | 0.9311195    |
| 4                    | 20,oversampling | 10,oversampling | 0.8915404 | 0.9281505  | 0.8897455 | 0.9276724            | 14 | 20,oversampling  | 10,oversampling | 0.8941140 | 0.9302575  | 0.8939551 | 0.9304122    |
| 0                    | 17,oversampling | 8,oversampling  | 0.8845822 | 0.9255312  | 0.8825864 | 0.9243852            | 10 | 17,oversampling  | 8,oversampling  | 0.8882534 | 0.9272413  | 0.8880044 | 0.9295172    |
| 6                    | 14,oversampling | 6,oversampling  | 0.8752698 | 0.9154186  | 0.8747185 | 0.9153903            | 20 | 23,weight        | 6,oversampling  | 0.8808286 | 0.9165128  | 0.8828263 | 0.9161103    |
| 2                    | 17,weight       | 4,oversampling  | 0.8646538 | 0.8917734  | 0.8671478 | 0.8907180            | 6  | 14,oversampling  | 4,oversampling  | 0.8800651 | 0.8931177  | 0.8800487 | 0.8925446    |
| 6                    | 20,weight       | 14,weight       | 0.8636696 | 0.8802149  | 0.8664802 | 0.8787476            | 24 | 26,weight        | 10,weight       | 0.8788712 | 0.8925324  | 0.8810027 | 0.8960214    |
| 4                    | 26,weight       | 12,weight       | 0.8623590 | 0.8786619  | 0.8635655 | 0.8793529            | 16 | 20,weight        | 14,weight       | 0.8782085 | 0.8924428  | 0.8798163 | 0.8933790    |
| 0                    | 23,weight       | 10,weight       | 0.8610266 | 0.8773826  | 0.8578192 | 0.8752517            | 12 | 17,weight        | 12,weight       | 0.8746919 | 0.8916932  | 0.8771639 | 0.8954639    |
| 2                    | 11,oversampling | 8,weight        | 0.8607753 | 0.8743833  | 0.8598177 | 0.8734471            | 8  | 14,weight        | 8,weight        | 0.8711961 | 0.8911373  | 0.8718465 | 0.8921789    |
| Imbalance level 1000 |                 |                 |           |            |           | Imbalance level 2000 |    |                  |                 |           |            |           |              |
|                      | desc_rf         | desc_boost      | mean_rf   | mean_boost | median_rf | median_boost         |    | desc_rf          | desc_boost      | mean_rf   | mean_boost | median_rf | median_boost |
| 2                    | 26,oversampling | 14,oversampling | 0.8956844 | 0.9315538  | 0.8946377 | 0.9310273            | 22 | 26,oversampling  | 14,oversampling | 0.8971327 | 0.9301416  | 0.8982745 | 0.9296913    |
| 8                    | 23,oversampling | 10,oversampling | 0.8940173 | 0.9302421  | 0.8919968 | 0.9296119            | 18 | 23,oversampling  | 12,oversampling | 0.8956727 | 0.9297424  | 0.8965030 | 0.9308386    |
| 4                    | 20,oversampling | 12,oversampling | 0.8913373 | 0.9301128  | 0.8915530 | 0.9296676            | 14 | 20,oversampling  | 10,oversampling | 0.8932283 | 0.9293505  | 0.8931368 | 0.9303487    |
| 4                    | 26,weight       | 8,oversampling  | 0.8881062 | 0.9264379  | 0.8867435 | 0.9263997            | 10 | 17,oversampling  | 8,oversampling  | 0.8866440 | 0.9250052  | 0.8874845 | 0.9254809    |
| 0                    | 23,weight       | 6,oversampling  | 0.8869930 | 0.9161078  | 0.8871667 | 0.9166822            | 21 | 26,none          | 6,oversampling  | 0.8850675 | 0.9162053  | 0.8841636 | 0.9167628    |
| 0                    | 17,oversampling | 12,weight       | 0.8863072 | 0.9092654  | 0.8863325 | 0.9064284            | 24 | 26,weight        | 14,weight       | 0.8847694 | 0.9110928  | 0.8836824 | 0.9105914    |
| 6                    | 20,weight       | 14,weight       | 0.8854978 | 0.9083457  | 0.8862164 | 0.9083720            | 17 | 23,none          | 12,weight       | 0.8847494 | 0.9108923  | 0.8851343 | 0.9108902    |
| 2                    | 17,weight       | 10,weight       | 0.8825991 | 0.9083075  | 0.8828398 | 0.9081682            | 20 | 23,weight        | 14,none         | 0.8835174 | 0.9108749  | 0.8842780 | 0.9099162    |
| 8                    | 14,weight       | 14,none         | 0.8763382 | 0.9081972  | 0.8759878 | 0.9096693            | 13 | 20,none          | 10,weight       | 0.8829964 | 0.9107124  | 0.8819908 | 0.9102227    |
| 6                    | 14,oversampling | 12,none         | 0.8758277 | 0.9070277  | 0.8758313 | 0.9074308            | 23 | 26,undersampling | 12,none         | 0.8827243 | 0.9104474  | 0.8813355 | 0.9106859    |

Table 2: The top 10 combinations of maximum depth and imbalance method in balanced accuracy for both boosting and random forest for each level of imbalance, for the forest cover type dataset.

For the handwritten digits dataset, oversampling also generally yields the highest balanced accuracy, as shown in Table 3. However, for more balanced datasets, the difference between oversampling and weights is very small. Boosting and random forest also perform similarly across all imbalance levels, as shown in Table 3. In addition, for more balanced datasets, the differences in balanced accuracy between the top 10 models are generally small for both methods, and overall smaller than those observed in Table 2. As shown in Table 3, oversampling appears more important than selecting the optimal maximum depth for achieving high balanced accuracy in more imbalanced datasets. However, for boosting, selecting an appropriate maximum depth still has a noticeable effect on balanced accuracy.

Imbalance level 40

| desc_rf          | desc_boost      | mean_rf   | mean_boost | median_rf | median_boost |
|------------------|-----------------|-----------|------------|-----------|--------------|
| 17,oversampling  | 4,oversampling  | 0.8128517 | 0.8141750  | 0.8131244 | 0.8133163    |
| 20,oversampling  | 6,oversampling  | 0.8123236 | 0.8140194  | 0.8128184 | 0.8155629    |
| 14,oversampling  | 8,oversampling  | 0.8122322 | 0.8104609  | 0.8127540 | 0.8114287    |
| 23,oversampling  | 10,oversampling | 0.8121848 | 0.8074421  | 0.8117326 | 0.8089392    |
| 26,oversampling  | 12,oversampling | 0.8118665 | 0.8043794  | 0.8118334 | 0.8053834    |
| 11,oversampling  | 2,oversampling  | 0.8110945 | 0.8035537  | 0.8103852 | 0.8026031    |
| 8,oversampling   | 14,oversampling | 0.8065054 | 0.8024124  | 0.8060220 | 0.8030172    |
| 11,undersampling | 2,weight        | 0.7882205 | 0.7775007  | 0.7891118 | 0.7788670    |
| 5,oversampling   | 4,weight        | 0.7867003 | 0.7688425  | 0.7858453 | 0.7678209    |
| 8,weight         | 6,weight        | 0.7823705 | 0.7681674  | 0.7794105 | 0.7704112    |

Imbalance level 100

| desc_rf         | desc_boost      | mean_rf   | mean_boost | median_rf | median_boost |
|-----------------|-----------------|-----------|------------|-----------|--------------|
| 17,oversampling | 4,oversampling  | 0.8106128 | 0.8122948  | 0.8102240 | 0.8113184    |
| 20,oversampling | 6,oversampling  | 0.8105343 | 0.8116155  | 0.8097095 | 0.8105277    |
| 23,oversampling | 8,oversampling  | 0.8103730 | 0.8096418  | 0.8103276 | 0.8085821    |
| 26,oversampling | 10,oversampling | 0.8099609 | 0.8074238  | 0.8095783 | 0.8065618    |
| 11,oversampling | 12,oversampling | 0.8093310 | 0.8047238  | 0.8091801 | 0.8034157    |
| 6,oversampling  | 2,oversampling  | 0.8091542 | 0.8033597  | 0.8087195 | 0.8030195    |
| 8,oversampling  | 14,oversampling | 0.8049061 | 0.8008567  | 0.8047192 | 0.7997120    |
| 8,weight        | 2,weight        | 0.7953173 | 0.7891868  | 0.7972093 | 0.7916264    |
| 11,weight       | 4,weight        | 0.7910589 | 0.7796047  | 0.7970123 | 0.7810455    |
| 26,weight       | 6,weight        | 0.7895184 | 0.7729518  | 0.7898738 | 0.7809360    |

Imbalance level 200

| desc_rf         | desc_boost      | mean_rf   | mean_boost | median_rf | median_boost |
|-----------------|-----------------|-----------|------------|-----------|--------------|
| 17,oversampling | 6,oversampling  | 0.8102680 | 0.8121701  | 0.8104210 | 0.8134283    |
| 23,oversampling | 4,oversampling  | 0.8102377 | 0.8110386  | 0.8107545 | 0.8124966    |
| 14,oversampling | 8,oversampling  | 0.8101860 | 0.8092807  | 0.8103728 | 0.8092848    |
| 20,oversampling | 10,oversampling | 0.8096112 | 0.8051629  | 0.8094021 | 0.8062418    |
| 26,oversampling | 12,oversampling | 0.8095344 | 0.8023903  | 0.8091598 | 0.8029636    |
| 11,oversampling | 2,oversampling  | 0.8084270 | 0.8015671  | 0.8078530 | 0.8023641    |
| 8,oversampling  | 14,oversampling | 0.8051598 | 0.7994189  | 0.8039565 | 0.7995888    |
| 11,weight       | 2,weight        | 0.7969870 | 0.7900300  | 0.8017648 | 0.7917135    |
| 26,weight       | 4,weight        | 0.7956928 | 0.7866097  | 0.7999843 | 0.7870372    |
| 17,weight       | 4,undersampling | 0.7955759 | 0.7845467  | 0.8001258 | 0.7756991    |

Imbalance level 400

| desc_rf         | desc_boost      | mean_rf   | mean_boost | median_rf | median_boost |
|-----------------|-----------------|-----------|------------|-----------|--------------|
| 14,oversampling | 4,oversampling  | 0.8088191 | 0.8097441  | 0.8081693 | 0.8093263    |
| 17,oversampling | 6,oversampling  | 0.8087885 | 0.8093658  | 0.8087320 | 0.8088885    |
| 23,oversampling | 8,oversampling  | 0.8086599 | 0.8073934  | 0.8088976 | 0.8071664    |
| 26,oversampling | 10,oversampling | 0.8084141 | 0.8046249  | 0.8089623 | 0.8046356    |
| 20,oversampling | 12,oversampling | 0.8082794 | 0.8012615  | 0.8086652 | 0.8011105    |
| 11,oversampling | 4,weight        | 0.8073503 | 0.8010730  | 0.8072377 | 0.8021920    |
| 11,weight       | 6,weight        | 0.8060030 | 0.8008775  | 0.8066096 | 0.7992057    |
| 26,weight       | 2,oversampling  | 0.8054833 | 0.7998522  | 0.8057299 | 0.7999056    |
| 23,weight       | 2,weight        | 0.8051794 | 0.7995320  | 0.8058713 | 0.8009927    |
| 20,weight       | 8,weight        | 0.8051063 | 0.7981031  | 0.8052532 | 0.7996428    |

Imbalance level 1000

| desc_rf         | desc_boost      | mean_rf   | mean_boost | median_rf | median_boost |
|-----------------|-----------------|-----------|------------|-----------|--------------|
| 20,oversampling | 4,oversampling  | 0.8106978 | 0.8131053  | 0.8100164 | 0.8130951    |
| 14,oversampling | 6,oversampling  | 0.8106914 | 0.8130215  | 0.8098072 | 0.8129971    |
| 17,oversampling | 8,oversampling  | 0.8100862 | 0.8101591  | 0.8097092 | 0.8104707    |
| 26,oversampling | 10,oversampling | 0.8099613 | 0.8077221  | 0.8096984 | 0.8063890    |
| 23,oversampling | 6,weight        | 0.8095190 | 0.8062130  | 0.8090058 | 0.8064718    |
| 11,oversampling | 4,weight        | 0.8092020 | 0.8046637  | 0.8089947 | 0.8064480    |
| 14,weight       | 2,oversampling  | 0.8064337 | 0.8034259  | 0.8079214 | 0.8040737    |
| 23,weight       | 12,oversampling | 0.8055566 | 0.8030948  | 0.8069987 | 0.8023158    |
| 20,weight       | 4,undersampling | 0.8054096 | 0.8014849  | 0.8066767 | 0.8033650    |
| 8,oversampling  | 8,weight        | 0.8051551 | 0.8011733  | 0.8040862 | 0.8027038    |

Imbalance level 2000

| desc_rf         | desc_boost      | mean_rf   | mean_boost | median_rf | median_boost |
|-----------------|-----------------|-----------|------------|-----------|--------------|
| 23,oversampling | 4,oversampling  | 0.8096005 | 0.8130999  | 0.8091228 | 0.8118149    |
| 14,oversampling | 6,oversampling  | 0.8094235 | 0.8119460  | 0.8088032 | 0.8113921    |
| 20,oversampling | 8,oversampling  | 0.8094123 | 0.8103854  | 0.8087975 | 0.8100791    |
| 17,oversampling | 4,weight        | 0.8092510 | 0.8099453  | 0.8090958 | 0.8115877    |
| 26,oversampling | 6,weight        | 0.8091985 | 0.8080263  | 0.8089882 | 0.8075738    |
| 11,oversampling | 4,none          | 0.8091188 | 0.8064085  | 0.8075386 | 0.8034170    |
| 14,weight       | 10,oversampling | 0.8083826 | 0.8063960  | 0.8085894 | 0.8071253    |
| 26,weight       | 6,none          | 0.8078328 | 0.8062690  | 0.8079685 | 0.8051401    |
| 23,weight       | 8,weight        | 0.8076420 | 0.8045560  | 0.8083070 | 0.8036290    |
| 17,weight       | 4,undersampling | 0.8075721 | 0.8031784  | 0.8077182 | 0.7999862    |

Table 3: The top 10 combinations of maximum depth and imbalance method in balanced accuracy for both boosting and random forest for each level of imbalance, for the handwritten digits dataset.

## 4.4 Class differences

In Table 4, it can be seen that for both boosting and random forest, class 4 is correctly classified more frequently than the other classes, while class 0 and class 1 are correctly classified less frequently than the other classes.

| class | mean      | median    | class | mean      | median    |
|-------|-----------|-----------|-------|-----------|-----------|
| 0     | 0.8059839 | 0.8303637 | 0     | 0.849214  | 0.84748   |
| 1     | 0.8235807 | 0.8314327 | 1     | 0.8475985 | 0.8487964 |
| 2     | 0.8545264 | 0.8632182 | 2     | 0.9029364 | 0.9020754 |
| 3     | 0.8819683 | 0.8909191 | 3     | 0.911813  | 0.9133262 |
| 4     | 0.955506  | 0.9579034 | 4     | 0.9707333 | 0.9726695 |

### Random Forest

### Boosting

Table 4: Average balanced accuracy for different classes for random forest and boosting across all models trained on a balanced dataset, for the forest cover type dataset

When comparing Table 4 and Table 5, it can be seen that the differences in balanced accuracy between classes are considerably smaller for the handwritten digits dataset. However, class 2 and class 3 appear to be slightly less frequently correctly classified than the other classes, as can be seen in Table 5.

| class | mean      | median    | class | mean      | median    |
|-------|-----------|-----------|-------|-----------|-----------|
| 0     | 0.8455448 | 0.8457286 | 0     | 0.8401414 | 0.8402744 |
| 1     | 0.8208101 | 0.824295  | 1     | 0.8377845 | 0.8369577 |
| 2     | 0.7649072 | 0.7623824 | 2     | 0.7614971 | 0.7621155 |
| 3     | 0.7722787 | 0.774034  | 3     | 0.7618945 | 0.7623585 |
| 4     | 0.8168717 | 0.8197541 | 4     | 0.8062258 | 0.8072446 |

### Random Forest

### Boosting

Table 5: Average balanced accuracy for different classes for random forest and boosting across all models trained on a balanced dataset, for the handwritten digits dataset

## 5 Discussion

In this section, the results and the chosen method are discussed.

### 5.1 Limitations

In order to interpret the results correctly, it is important to consider the limitations of the chosen method. A major limitation is that only two datasets were used, which means that the results cannot be interpreted as general findings, but rather as dataset-specific results for the forest cover type dataset and the handwritten digits dataset, even though some shared results might hint about general results. It is therefore important to consider the characteristics of these

datasets, as they may have influenced the results.

One characteristic of the forest cover type dataset is that, even in balanced settings, classes 0 and 1 are generally classified less accurately than the other classes, whereas class 4 is classified more accurately. In the handwritten digits dataset, classes 2 and 3 are slightly more difficult to classify, although the differences between classes are considerably smaller.

The forest cover type dataset also contains a large number of predictor variables (54 in total), and there appears to be a relatively strong relationship between predictors and response, as the best models achieve balanced accuracy above 0.9. In contrast, the handwritten digits dataset contains only 13 predictor variables, and the relationship between predictors and response appears weaker, with best models achieving around 0.8 in balanced accuracy.

The main performance measure used is average balanced accuracy across all classes, supplemented by average F1-score. Since all classes are weighted equally, this measure directly addresses class imbalance. However, unlike standard accuracy, balanced accuracy is less intuitive, and high balanced accuracy does not necessarily imply high standard accuracy. Since balanced accuracy is the mean of sensitivity and specificity, a high value in one can partially compensate for a lower value in the other.

Another limitation is that only a few combinations of maximum depth and minimum node size were explored.

In addition, the level of imbalance refers to the full dataset, but small imbalances may still arise in the test sets due to random sampling. However, since results are averaged over eight repetitions, these effects are expected to largely cancel out.

Finally, there may be other dataset specific characteristics affecting the results that were not explicitly considered.

## 5.2 Result discussion

The results related to minimum node size suggest that its effect is generally small, particularly for the handwritten digits dataset where it is essentially negligible. Since maximum depth was fixed at 26 in this analysis, minimum node size becomes more relevant only when it is large relative to the dataset size. This is particularly evident in the undersampling setting, where the effective training set size becomes very small at high imbalance levels. In such cases, a minimum node size of 30 acts as a strong constraint compared to the full dataset. Overall, minimum node size appears to matter mainly when it is large relative to the training set, while maximum depth remains the dominant parameter for controlling model complexity.

A key observation is that larger maximum depth generally leads to higher balanced accuracy in the forest cover type dataset. This can be explained by the bias variance trade off, deeper trees reduce bias but increase variance. The dataset appears to contain complex relationships that require deep trees to be properly captured, which is consistent with the large number of predictor variables and the fact that many of them are binary. In addition, the strong signal between predictors and response suggests that deeper trees can exploit the available information more effectively.

This effect is most pronounced for random forest. That is not surprising, since random forest is designed to handle high variance through averaging, whereas boosting relies on weak learners and may become more sensitive to overfitting when the trees are too deep.

For the handwritten digits dataset, the effect of maximum depth is weaker, and for boosting it even becomes negative at depths above 6. This is consistent with the dataset having fewer predictors, all of which are quantitative, as well as a weaker signal between predictors and response. In general, datasets with many predictor variables, especially when many are binary and the signal is strong, seem to benefit from larger maximum depths. Datasets with the opposite characteristics appear to perform better with slightly smaller maximum depths. For random forest, larger maximum depths do not seem to harm performance, whereas for boosting a too large maximum depth can substantially reduce balanced accuracy when the dataset is prone to overfitting.

The difference in how the two datasets respond to imbalance methods also reflects their underlying characteristics. For the forest cover type dataset, oversampling performs very well, which is reasonable since its main drawback is overfitting, and that does not appear to be a major issue here. Undersampling performs poorly, likely because it removes a large amount of information, which means that the complex relationships in the forest cover type dataset can not be discovered. Weights perform moderately well, but for boosting they barely improve performance relative to using no imbalance method.

For the handwritten digits dataset, undersampling performs as well as or even better than using no imbalance method, which is consistent with the dataset being more prone to overfitting. Oversampling still outperforms weights, although the difference is smaller than for the forest cover type dataset. Weights also do not appear to improve boosting very much in this dataset.

An additional characteristic of the forest cover type dataset is that classes 0 and 1 are consistently harder to classify, while class 4 is easier, even in balanced settings. This suggests that class imbalance is not only a matter of class frequencies, but also of how distinct the classes are. Minority classes that are also difficult to classify may have a disproportionate effect on balanced accu-

racy, which may help explain why oversampling is particularly effective for this dataset. For weights and undersampling, this class-specific structure may partly distort the results.

In general, oversampling performs better than weights, especially for more imbalanced datasets. Undersampling generally performs worse than both methods, although the difference is smaller for datasets that are more prone to overfitting. For datasets more prone to underfitting, undersampling performs even worse than using no imbalance method. Weights tend to perform better relative to no imbalance method in random forest than in boosting, although this may also reflect that boosting performs well without any imbalance method rather than weights being less effective.

When considering combinations of maximum depth, imbalance level, and imbalance method, some clear patterns emerge. For both no imbalance method and oversampling, increasing maximum depth generally improves balanced accuracy more than for weights and undersampling. However, in the handwritten digits dataset with boosting, intermediate depths (4–6) often yield the best performance, although higher depths still improve performance relative to the baseline more for oversampling and no imbalance method than for weights and undersampling. This suggests that model complexity is more important when using oversampling or no imbalance method, compared to weights and undersampling.

Oversampling shows very little variation in balanced accuracy across imbalance levels in both datasets, while weights only show small differences between moderate imbalance levels (400, 1000, and 2000). This indicates that oversampling is more effective at compensating for class imbalance, whereas weights are more sensitive to the degree of imbalance.

For the forest cover type dataset, boosting consistently outperforms random forest across all imbalance levels. While this is not unexpected, it highlights the strength of boosting for this dataset. In this dataset, oversampling becomes increasingly important as imbalance grows, while larger maximum depth is more important for more balanced settings. This suggests that oversampling primarily reduces bias in highly imbalanced cases, whereas deeper trees contribute more when the class distribution is already relatively balanced.

In contrast, for the handwritten digits dataset, there is little difference between random forest and boosting. This is consistent with the dataset being more prone to overfitting, where both methods behave similarly. For highly imbalanced cases, oversampling is crucial for achieving high balanced accuracy regardless of model type, while in more balanced settings, weights perform similarly to oversampling. For random forest, differences between imbalance methods and maximum depth are generally small, whereas boosting is more sensitive to the choice of maximum depth. Overall, the difference between the best per-

forming models is smaller for the handwritten digits dataset than for the forest cover type dataset.

## 6 Conclusion

This thesis aimed to investigate how different levels of class imbalance affect the performance of random forest and gradient boosting in a five-class classification problem, how different hyperparameters influence model performance, and how different imbalance methods affect the results. A further aim was to identify whether any systematic patterns emerge across models, datasets and imbalance settings.

Overall, class imbalance was found to have a clear impact on model performance, but the magnitude and direction of this effect depended strongly on the dataset and modelling choices. More imbalanced datasets generally led to lower balanced accuracy, although this effect varied depending on the imbalance method and model type. Oversampling consistently gave the best performance across most settings, particularly for highly imbalanced datasets, indicating that it is the most effective method for handling class imbalance in this study. Weights generally improved performance compared to using no imbalance method, but the improvement was smaller and less consistent, especially when imbalance was large. Undersampling generally performed worst, particularly for datasets that appear more prone to underfitting.

Regarding hyperparameters, maximum depth had a substantially larger effect on performance than minimum node size. Increasing maximum depth generally improved balanced accuracy for the forest cover type dataset, particularly for random forest, indicating that deeper trees were needed to capture complex relationships in the data. For the handwritten digits dataset, the effect of maximum depth was weaker, and for boosting intermediate depths (approximately 4–6) often gave the best performance. Random forest was generally more robust to large maximum depths, while boosting was more sensitive to overfitting when trees were too deep. Minimum node size had only a minor effect overall, except in cases where it was large relative to the effective training set size, such as under strong undersampling.

When comparing models, boosting generally outperformed random forest for the forest cover type dataset across most imbalance levels, while the difference between the methods was small for the handwritten digits dataset. This suggests that boosting benefits more from the stronger signal and higher complexity of the forest cover type dataset, whereas both methods behave similarly in the more limited and lower-dimensional handwritten digits dataset.

In terms of interactions between hyperparameters and imbalance methods, oversampling and the use of no imbalance method benefited more from increased

maximum depth than weights and undersampling. This indicates that model complexity is more important in combination with these methods. In contrast, undersampling showed weaker gains from increasing depth, particularly in more imbalanced settings.

## 7 References

- Blackard, J. (1998). *Coverttype* [Data set]. UCI Machine Learning Repository. <https://archive.ics.uci.edu/dataset/31/coverttype>
- Brodersen, K. H., Ong, C. S., Stephan, K. E., & Buhmann, J. M. (2010). The balanced accuracy and its posterior distribution. In *2010 International Conference on Pattern Recognition (ICPR)* (pp. —). IEEE. <http://www.ong-home.my/papers/brodersen10post-balacc.pdf>
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). Association for Computing Machinery. <https://dl.acm.org/doi/epdf/10.1145/2939672.2939785>
- Cui, Y., Jia, M., Lin, T.-Y., Song, Y., & Belongie, S. (2019). Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 9268–9277). IEEE. [https://openaccess.thecvf.com/content\\_CVPR2019/papers/Cui\\_Class-Balanced\\_Loss\\_Based\\_on\\_Effective\\_Number\\_of\\_Samples\\_CVPR2019\\_paper.pdf](https://openaccess.thecvf.com/content_CVPR2019/papers/Cui_Class-Balanced_Loss_Based_on_Effective_Number_of_Samples_CVPR2019_paper.pdf)
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5), 1189–1232. [https://www.researchgate.net/publication/2424824\\_Greedy\\_Function\\_Approximation\\_A\\_Gradient\\_Boosting\\_Machine](https://www.researchgate.net/publication/2424824_Greedy_Function_Approximation_A_Gradient_Boosting_Machine)
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning* (2nd ed.). Springer.
- Lindholm, A., Wahlström, N., Lindsten, F., & Schön, T. B. (2022). *Machine learning: A first course for engineers and scientists*. Cambridge University Press.
- Lyu, S.-H., Wu, J.-H., Zheng, Q.-C., & Ye, B. (2024). The role of depth, width, and tree size in expressiveness of deep forest. In *Proceedings of the 27th European Conference on Artificial Intelligence (ECAI 2024)* (pp. 2042–2049). <https://doi.org/10.48550/arXiv.2407.05108>

Megahed, F. M., Chen, Y. J., Megahed, A., Ong, Y., Altman, N., & Krzywinski, M. (2021). The class imbalance problem. *Nature Methods*, 18, 1270–1272. <https://www.nature.com/articles/s41592-021-01302-4>

MNIST Database of Handwritten Digits. (2021). *UCI Machine Learning Repository*. <https://archive.ics.uci.edu/dataset/683/mnist+database+of+handwritten+digits>

## A Random forest

### A.1 Maximum depth

| desc        | X5        | X8         | X11        | X14        | X17        | X20        | X23        | X26        | mean_median | larger_or_not | class | measure |
|-------------|-----------|------------|------------|------------|------------|------------|------------|------------|-------------|---------------|-------|---------|
| 2000,none   | 0.8005072 | 0.03448005 | 0.05810547 | 0.07231388 | 0.07848771 | 0.08248915 | 0.08424220 | 0.08456034 |             |               | avg   | bal_acc |
| 1000,none   | 0.7295205 | 0.06878797 | 0.10079473 | 0.12585195 | 0.13578118 | 0.14250486 | 0.14281173 | 0.14367352 |             |               | avg   | bal_acc |
| 400,none    | 0.7133836 | 0.06847307 | 0.09493225 | 0.11181364 | 0.12407598 | 0.12789435 | 0.12980669 | 0.13101125 |             |               | avg   | bal_acc |
| 200,none    | 0.7132667 | 0.06815706 | 0.09416408 | 0.10663541 | 0.11529497 | 0.11865615 | 0.12048220 | 0.12152799 |             |               | avg   | bal_acc |
| 100,none    | 0.7147896 | 0.06270139 | 0.09018848 | 0.10485919 | 0.11174003 | 0.11461038 | 0.11692470 | 0.11616175 |             |               | avg   | bal_acc |
| 40,none     | 0.7104489 | 0.06986183 | 0.09603578 | 0.10905982 | 0.11649978 | 0.12113641 | 0.12325739 | 0.12252834 |             |               | avg   | bal_acc |
| 2000,un     | 0.7965845 | 0.03306061 | 0.05760268 | 0.06929224 | 0.07879588 | 0.08242382 | 0.08435529 | 0.08613979 |             |               | avg   | bal_acc |
| 1000,un     | 0.7975518 | 0.03277111 | 0.04963759 | 0.06081435 | 0.06565834 | 0.06882241 | 0.06963279 | 0.06969588 |             |               | avg   | bal_acc |
| 400,un      | 0.7967117 | 0.02468572 | 0.04104224 | 0.04530550 | 0.05078642 | 0.05196908 | 0.05076189 | 0.04931550 |             |               | avg   | bal_acc |
| 200,un      | 0.7949822 | 0.01925639 | 0.03609581 | 0.03580770 | 0.03384813 | 0.03672584 | 0.03779573 | 0.03742404 |             |               | avg   | bal_acc |
| 100,un      | 0.7747263 | 0.02156975 | 0.02616201 | 0.03007625 | 0.03130157 | 0.02898234 | 0.03231835 | 0.03727664 | 5,17        | -,+           | avg   | bal_acc |
| 40,un       | 0.7530132 | 0.02685878 | 0.03487123 | 0.04156658 | 0.04363734 | 0.04016745 | 0.02503619 | 0.03458871 | 5,8,20,23   | -,+,+         | avg   | bal_acc |
| 2000,ov     | 0.8020739 | 0.03564233 | 0.06194437 | 0.07779509 | 0.08457012 | 0.09115435 | 0.09359873 | 0.09505873 |             |               | avg   | bal_acc |
| 1000,ov     | 0.8010245 | 0.03582001 | 0.06124495 | 0.07480325 | 0.08528271 | 0.09031285 | 0.09299279 | 0.09465994 |             |               | avg   | bal_acc |
| 400,ov      | 0.8054013 | 0.03478467 | 0.05968073 | 0.07466390 | 0.08285213 | 0.08871272 | 0.09014893 | 0.09267352 |             |               | avg   | bal_acc |
| 200,ov      | 0.8004910 | 0.03801385 | 0.06028434 | 0.07477877 | 0.08409118 | 0.09104946 | 0.09319006 | 0.09371577 |             |               | avg   | bal_acc |
| 100,ov      | 0.8028061 | 0.03674648 | 0.06065028 | 0.07584717 | 0.08485963 | 0.08990640 | 0.09384972 | 0.09453875 |             |               | avg   | bal_acc |
| 40,ov       | 0.7996588 | 0.03956658 | 0.06473343 | 0.07897394 | 0.08849424 | 0.09375619 | 0.09648512 | 0.09771555 |             |               | avg   | bal_acc |
| 2000,weight | 0.8012028 | 0.03517162 | 0.05748731 | 0.07091319 | 0.07735809 | 0.08035820 | 0.08231453 | 0.08356659 |             |               | avg   | bal_acc |
| 1000,weight | 0.8071618 | 0.03149216 | 0.05582571 | 0.06917645 | 0.07543731 | 0.07833607 | 0.07983129 | 0.08094445 |             |               | avg   | bal_acc |
| 400,weight  | 0.8118856 | 0.03333813 | 0.04965797 | 0.05931059 | 0.06280635 | 0.06632298 | 0.06894308 | 0.06698561 |             |               | avg   | bal_acc |
| 200,weight  | 0.8083070 | 0.02985716 | 0.04651149 | 0.05216932 | 0.05634680 | 0.05536255 | 0.05271954 | 0.05405201 |             |               | avg   | bal_acc |
| 100,weight  | 0.8046596 | 0.03010435 | 0.04368089 | 0.05019752 | 0.04815923 | 0.04824153 | 0.04510175 | 0.04280509 | 14          | -             | avg   | bal_acc |
| 40,weight   | 0.7846641 | 0.03168311 | 0.04627857 | 0.05285445 | 0.05995179 | 0.06075649 | 0.06201881 | 0.06022813 | 17          | +             | avg   | bal_acc |

Table 6: Table of results for maximum depth for random forest with measurement balanced accuracy

| desc        | X5        | X8         | X11        | X14        | X17        | X20        | X23        | X26        | mean_median  | larger_or_not | class | measure |
|-------------|-----------|------------|------------|------------|------------|------------|------------|------------|--------------|---------------|-------|---------|
| 2000,none   | 0.6717708 | 0.06078858 | 0.10003299 | 0.12318119 | 0.13315794 | 0.13956479 | 0.14225490 | 0.14268937 |              |               | avg   | F1      |
| 1000,none   | NA        | NA         | NA         | NA         | NA         | NA         | NA         | NA         |              |               | avg   | F1      |
| 400,none    | NA        | NA         | NA         | NA         | NA         | NA         | NA         | NA         | 17           | +             | avg   | F1      |
| 200,none    | NA        | NA         | NA         | NA         | NA         | NA         | NA         | NA         | 26           | +             | avg   | F1      |
| 100,none    | NA        | NA         | NA         | NA         | NA         | NA         | NA         | NA         |              |               | avg   | F1      |
| 40,none     | NA        | NA         | NA         | NA         | NA         | NA         | NA         | NA         |              |               | avg   | F1      |
| 2000,un     | 0.6617967 | 0.06128483 | 0.10254106 | 0.12168299 | 0.13691374 | 0.14273737 | 0.14571776 | 0.14864743 | 8            | +             | avg   | F1      |
| 1000,un     | 0.6661770 | 0.05944286 | 0.08715961 | 0.10545998 | 0.11310546 | 0.11817880 | 0.11937968 | 0.11946319 |              |               | avg   | F1      |
| 400,un      | 0.6615064 | 0.04731491 | 0.07487128 | 0.08183283 | 0.09041068 | 0.09283291 | 0.09027807 | 0.08854850 |              |               | avg   | F1      |
| 200,un      | 0.6532753 | 0.03829094 | 0.06705196 | 0.06729259 | 0.06390790 | 0.06956683 | 0.07065461 | 0.06981994 | 5            | +             | avg   | F1      |
| 100,un      | 0.6189078 | 0.04266115 | 0.05287671 | 0.06153863 | 0.06241720 | 0.05733073 | 0.06249397 | 0.07121619 | 5,8,17,23,26 | -,+,+,-,-     | avg   | F1      |
| 40,un       | 0.5713699 | 0.05535381 | 0.06975399 | 0.07806229 | 0.08265430 | 0.07960582 | 0.05618410 | 0.06981759 | 5,8,11,17,23 | +,+,-,-,+     | avg   | F1      |
| 2000,ov     | 0.6746135 | 0.06233774 | 0.10596565 | 0.13163170 | 0.14257224 | 0.15317261 | 0.15706466 | 0.15936032 |              |               | avg   | F1      |
| 1000,ov     | 0.6721924 | 0.06393738 | 0.10576672 | 0.12753345 | 0.14454081 | 0.15255349 | 0.15673710 | 0.15938038 |              |               | avg   | F1      |
| 400,ov      | 0.6827725 | 0.05873725 | 0.09977743 | 0.12396739 | 0.13721063 | 0.14658033 | 0.14883935 | 0.15291549 |              |               | avg   | F1      |
| 200,ov      | 0.6703568 | 0.06827480 | 0.10532867 | 0.12870376 | 0.14365786 | 0.15477337 | 0.15815770 | 0.15891532 |              |               | avg   | F1      |
| 100,ov      | 0.6740500 | 0.06596534 | 0.10571390 | 0.13034168 | 0.14488176 | 0.15293027 | 0.15914918 | 0.16033248 |              |               | avg   | F1      |
| 40,ov       | 0.6701195 | 0.06994969 | 0.11120251 | 0.13436720 | 0.14966230 | 0.15805969 | 0.16241289 | 0.16436071 |              |               | avg   | F1      |
| 2000,weight | 0.6742973 | 0.06077926 | 0.09795284 | 0.11976478 | 0.13006593 | 0.13485907 | 0.13794596 | 0.14002995 |              |               | avg   | F1      |
| 1000,weight | 0.6793707 | 0.05298243 | 0.09362998 | 0.11626217 | 0.12720354 | 0.13282747 | 0.13528054 | 0.13748400 |              |               | avg   | F1      |
| 400,weight  | 0.6559581 | 0.05965671 | 0.09175568 | 0.11258353 | 0.12360046 | 0.13297683 | 0.13890058 | 0.13639162 | 5,8          | +,-           | avg   | F1      |
| 200,weight  | 0.6497630 | 0.04675922 | 0.08129359 | 0.10000205 | 0.11524674 | 0.11468090 | 0.10848145 | 0.11269801 | 20,23        | -,+           | avg   | F1      |
| 100,weight  | 0.6158095 | 0.05414293 | 0.09063892 | 0.11490225 | 0.11801126 | 0.13128526 | 0.12856001 | 0.11917882 | 5,8          | +,-           | avg   | F1      |
| 40,weight   | 0.5851592 | 0.06260178 | 0.10289902 | 0.13517010 | 0.15981500 | 0.16094589 | 0.17125932 | 0.17296839 | 5            | -             | avg   | F1      |

Table 7: Table of results for maximum depth for random forest with measurement F1

## A.2 Minimum node size

| desc        | X2        | X3            | X5            | X7            | X10           | X14           | X19          | X24           | X30          | mean_median         | larger_or_not     | class | measure |
|-------------|-----------|---------------|---------------|---------------|---------------|---------------|--------------|---------------|--------------|---------------------|-------------------|-------|---------|
| 2000,none   | 0.8822322 | 7.128164e-04  | 0.0004169713  | -0.0013584179 | -0.0018863089 | -0.0039416932 | -0.005791471 | -0.0064771697 | -0.008078800 |                     |                   | avg   | bal_acc |
| 1000,none   | 0.8737488 | 5.521727e-05  | -0.0010685086 | -0.0017740141 | -0.0024410877 | -0.0040166083 | -0.006834423 | -0.0097286123 | -0.010781911 |                     |                   | avg   | bal_acc |
| 400,none    | 0.8420772 | -6.347633e-04 | -0.0009212023 | -0.0015187581 | -0.0053357438 | -0.0064011146 | -0.007518328 | -0.0092833683 | -0.012098740 |                     |                   | avg   | bal_acc |
| 200,none    | 0.8351490 | -8.315646e-04 | -0.0016102561 | -0.0009445725 | -0.0026063243 | -0.0020114047 | -0.004431341 | -0.0061064071 | -0.005867977 |                     |                   | avg   | bal_acc |
| 100,none    | 0.8324740 | -6.093469e-04 | -0.0007146415 | -0.0003857403 | -0.0011294203 | -0.0017350240 | -0.002824610 | -0.0039603046 | -0.005461404 |                     |                   | avg   | bal_acc |
| 40,none     | 0.8353853 | -5.995387e-04 | -0.0019767598 | -0.0021459077 | -0.0036393056 | -0.0020521762 | -0.005063720 | -0.0059998981 | -0.006780509 |                     |                   | avg   | bal_acc |
| 2000,un     | 0.8827730 | -5.597489e-04 | -0.0008207603 | -0.0027936254 | -0.0019917214 | -0.0032616730 | -0.005388333 | -0.0064707859 | -0.009538514 |                     |                   | avg   | bal_acc |
| 1000,un     | 0.8734957 | 2.801720e-04  | -0.0012731763 | -0.0021385841 | -0.0023317496 | -0.0033372789 | -0.007841225 | -0.0086176677 | -0.010803989 |                     |                   | avg   | bal_acc |
| 400,un      | 0.8508802 | -5.061827e-03 | -0.0018772549 | -0.0057555542 | -0.0041873167 | -0.0067911163 | -0.009677595 | -0.0088429336 | -0.013233135 | 24                  | -                 | avg   | bal_acc |
| 200,un      | 0.8263256 | -1.506960e-03 | -0.0011491080 | -0.0004486764 | -0.0004953788 | -0.0058894014 | -0.004071319 | -0.0111008159 | -0.008192784 |                     |                   | avg   | bal_acc |
| 100,un      | 0.7935341 | -2.340242e-03 | -0.0037374972 | -0.0044230186 | -0.0028028362 | -0.0035841814 | -0.010078901 | -0.0090734203 | -0.004154915 | 5                   | +                 | avg   | bal_acc |
| 40,un       | 0.7793110 | 3.506058e-03  | -0.0030513453 | -0.0104088140 | -0.0116987561 | -0.0218823391 | -0.015373641 | -0.0123374670 | -0.029471318 | 2,3,5,7,10,14,24,30 | -,+,-,-,-,-,-,-,- | avg   | bal_acc |
| 2000,ov     | 0.8960962 | 7.810148e-04  | -0.0004509375 | -0.0011419686 | -0.0018359167 | -0.0030138809 | -0.005202423 | -0.0066502039 | -0.008799194 |                     |                   | avg   | bal_acc |
| 1000,ov     | 0.8955725 | 3.642740e-05  | -0.0008336561 | -0.0007278421 | -0.0027950462 | -0.0043296263 | -0.004971197 | -0.0070022920 | -0.009065185 |                     |                   | avg   | bal_acc |
| 400,ov      | 0.8954327 | 2.663716e-04  | 0.0001314350  | -0.0018310197 | -0.0027393562 | -0.0036554060 | -0.004476151 | -0.0073514634 | -0.008932355 |                     |                   | avg   | bal_acc |
| 200,ov      | 0.8961686 | -7.793934e-04 | -0.0021369660 | -0.0022766529 | -0.0026457554 | -0.0050319999 | -0.005836186 | -0.0070221219 | -0.009164284 |                     |                   | avg   | bal_acc |
| 100,ov      | 0.8958287 | -7.658837e-04 | -0.0009740545 | -0.0019575251 | -0.0025736278 | -0.0040909197 | -0.005779431 | -0.0069319849 | -0.008130195 |                     |                   | avg   | bal_acc |
| 40,ov       | 0.8979181 | -1.711223e-04 | -0.0005145885 | -0.0013817218 | -0.0021782173 | -0.0036572663 | -0.005444529 | -0.0070465919 | -0.008930421 |                     |                   | avg   | bal_acc |
| 2000,weight | 0.8799150 | -1.923961e-04 | -0.0008748376 | -0.0005852243 | -0.0013668028 | -0.0025031312 | -0.007092826 | -0.0075281969 | -0.009742429 |                     |                   | avg   | bal_acc |
| 1000,weight | 0.8876500 | -4.628273e-04 | -0.0018470605 | -0.0027163699 | -0.0037286772 | -0.0052760127 | -0.006709038 | -0.0079285461 | -0.010119449 |                     |                   | avg   | bal_acc |
| 400,weight  | 0.8755948 | -3.987347e-04 | -0.0009977874 | 0.0012947749  | -0.0002885136 | -0.0009734527 | -0.001310795 | -0.0026027619 | -0.003306787 |                     |                   | avg   | bal_acc |
| 200,weight  | 0.8767909 | -1.723330e-03 | -0.0037782276 | -0.0033630566 | -0.0029660333 | -0.0034389323 | -0.004954210 | -0.0042638237 | -0.006844788 |                     |                   | avg   | bal_acc |
| 100,weight  | 0.8557476 | 2.093064e-03  | 0.0011087842  | 0.0021520740  | -0.0005120103 | 0.0003886765  | 0.001273572  | -0.0008701648 | -0.001287034 |                     |                   | avg   | bal_acc |
| 40,weight   | 0.8445186 | -2.444631e-04 | -0.0002911487 | -0.0005984520 | -0.0017694349 | -0.0025156963 | -0.005322717 | -0.0043341924 | -0.006697577 |                     |                   | avg   | bal_acc |

Table 8: Table of results for minimum node size for random forest with measurement balance accuracy

| desc        | X2        | X3            | X5            | X7            | X10           | X14          | X19          | X24         | X30          | mean_median         | larger_or_not | class | measure |
|-------------|-----------|---------------|---------------|---------------|---------------|--------------|--------------|-------------|--------------|---------------------|---------------|-------|---------|
| 2000,none   | 0.8095942 | 1.212409e-03  | 0.0007780697  | -0.0020385564 | -2.917133e-03 | -0.006109184 | -0.009028506 | -0.01011349 | -0.012724692 | 19                  | -             | avg   | F1      |
| 1000,none   | 0.7973519 | -5.022051e-05 | -0.0016638700 | -0.0028934690 | -3.926968e-03 | -0.007026013 | -0.011939034 | -0.01749403 | -0.018915173 |                     |               | avg   | F1      |
| 400,none    | 0.7266562 | -2.396463e-03 | -0.0020265428 | -0.0042809318 | -1.304863e-02 | -0.014925808 | -0.019094691 | -0.02276497 | -0.031330400 |                     |               | avg   | F1      |
| 200,none    | 0.7126042 | -3.093423e-03 | -0.0071481578 | -0.0044561557 | -5.741275e-03 | -0.004990451 | 0.008595727  | -0.01874133 | -0.017258976 | 7                   | -             | avg   | F1      |
| 100,none    | NA        | NA            | NA            | NA            | NA            | NA           | NA           | NA          | NA           |                     |               | avg   | F1      |
| 40,none     | NA        | NA            | NA            | NA            | NA            | NA           | NA           | NA          | NA           |                     |               | avg   | F1      |
| 2000,un     | 0.8106825 | -8.779433e-04 | -0.0012381134 | -0.0043375430 | -3.054133e-03 | -0.005027055 | -0.008541374 | -0.01014748 | -0.015163347 |                     |               | avg   | F1      |
| 1000,un     | 0.7954291 | 5.370680e-04  | -0.0023684081 | -0.0035826655 | -3.620121e-03 | -0.005341441 | -0.012445371 | -0.01376591 | -0.017338118 | 2,5,7,10,14,30      | +++++         | avg   | F1      |
| 400,un      | 0.7568117 | -8.123518e-03 | -0.0024978461 | -0.0089424175 | -6.041391e-03 | -0.010420161 | -0.015702036 | -0.01387942 | -0.021383850 | 2,19,24             | +,-,-         | avg   | F1      |
| 200,un      | 0.7179349 | -2.531788e-03 | -0.0022424152 | -0.0003584980 | 3.785034e-05  | -0.009641677 | -0.006082691 | -0.01766450 | -0.013296344 |                     |               | avg   | F1      |
| 100,un      | 0.6587385 | -5.339571e-03 | -0.0121102649 | -0.0079604895 | -5.568443e-03 | -0.006496219 | -0.016939399 | -0.01702825 | -0.007604818 | 3,5,14,30           | +,+,-,-       | avg   | F1      |
| 40,un       | 0.6164276 | 9.832020e-03  | -0.0104614274 | -0.0204784421 | -1.995248e-02 | -0.038350100 | -0.030585178 | -0.02495907 | -0.055204143 | 2,3,5,7,10,14,19,24 | ++++++-       | avg   | F1      |
| 2000,ov     | 0.8323903 | 1.214534e-03  | -0.0007165575 | -0.0017686469 | -2.862030e-03 | -0.004774316 | -0.008281916 | -0.01056898 | -0.014008894 |                     |               | avg   | F1      |
| 1000,ov     | 0.8314504 | 7.278138e-05  | -0.0012238231 | -0.0011284091 | -4.518717e-03 | -0.006853350 | -0.007994612 | -0.01113731 | -0.014455519 |                     |               | avg   | F1      |
| 400,ov      | 0.8313629 | 3.742820e-04  | 0.0001775170  | -0.0029874010 | -4.451512e-03 | -0.005916188 | -0.007142685 | -0.01173846 | -0.014317218 |                     |               | avg   | F1      |
| 200,ov      | 0.8322836 | -1.269806e-03 | -0.0033928774 | -0.0035752850 | -4.138685e-03 | -0.008067614 | -0.009290286 | -0.01111019 | -0.014618500 |                     |               | avg   | F1      |
| 100,ov      | 0.8319454 | -1.248595e-03 | -0.0015743911 | -0.0031057055 | -4.079687e-03 | -0.006496737 | -0.009185379 | -0.01100091 | -0.012924884 |                     |               | avg   | F1      |
| 40,ov       | 0.8350960 | -2.606612e-04 | -0.0008172280 | -0.0022037035 | -3.466138e-03 | -0.005800920 | -0.008627677 | -0.01126549 | -0.014199059 |                     |               | avg   | F1      |
| 2000,weight | 0.8062732 | -3.494678e-04 | -0.0013333621 | -0.0009704505 | -2.153953e-03 | -0.003830961 | -0.011325478 | -0.01195935 | -0.015659637 |                     |               | avg   | F1      |
| 1000,weight | 0.8154488 | -5.175629e-04 | -0.0030487001 | -0.0044090627 | -6.378889e-03 | -0.009359117 | -0.011929710 | -0.01463346 | -0.018309494 |                     |               | avg   | F1      |
| 400,weight  | 0.7863558 | -1.315269e-04 | -0.0012010037 | 0.0010129419  | -3.067252e-03 | -0.004965052 | -0.007544672 | -0.01078711 | -0.014097381 |                     |               | avg   | F1      |
| 200,weight  | 0.7861808 | -2.470906e-03 | -0.0073276925 | -0.0065305005 | -7.938590e-03 | -0.009439302 | -0.015090158 | -0.01598898 | -0.023471010 |                     |               | avg   | F1      |
| 100,weight  | 0.7592881 | 1.344891e-02  | -0.0059457031 | -0.0078090879 | -8.980578e-03 | -0.008053173 | -0.012069982 | -0.01813475 | -0.018720194 |                     |               | avg   | F1      |
| 40,weight   | 0.7639147 | 8.214673e-04  | 0.0020275014  | -0.0048790587 | -2.786646e-03 | -0.004753552 | -0.007380907 | -0.01833853 | -0.013005982 |                     |               | avg   | F1      |

Table 9: Table of results for minimum node size for random forest with measurement F1

### A.3 Class imbalance

| desc      | X40       | X100          | X200          | X400          | X1000        | X2000         | mean_median | larger_or_not | class | measure |
|-----------|-----------|---------------|---------------|---------------|--------------|---------------|-------------|---------------|-------|---------|
| 5,none    | 0.7104489 | 4.340686e-03  | 2.817840e-03  | 0.0029346709  | 0.019071580  | 0.0900583200  |             |               | avg   | bal_acc |
| 8,none    | 0.7803107 | -2.819755e-03 | 1.113066e-03  | 0.0015459027  | 0.017997712  | 0.0546765374  |             |               | avg   | bal_acc |
| 11,none   | 0.8064847 | -1.506611e-03 | 9.461377e-04  | 0.0018311467  | 0.023830532  | 0.0521280120  |             |               | avg   | bal_acc |
| 14,none   | 0.8195087 | 1.400562e-04  | 3.934299e-04  | 0.0056884942  | 0.035863710  | 0.0533123747  |             |               | avg   | bal_acc |
| 17,none   | 0.8269487 | -4.190613e-04 | 1.613029e-03  | 0.0105108758  | 0.038352982  | 0.0520462495  |             |               | avg   | bal_acc |
| 20,none   | 0.8315853 | -2.185343e-03 | 3.375830e-04  | 0.0096926082  | 0.040440029  | 0.0514110560  |             |               | avg   | bal_acc |
| 23,none   | 0.8337063 | -1.992010e-03 | 4.264836e-05  | 0.0094839666  | 0.038625920  | 0.0510431234  |             |               | avg   | bal_acc |
| 26,none   | 0.8329772 | -2.025901e-03 | 1.817494e-03  | 0.0114175875  | 0.040216763  | 0.0520903248  |             |               | avg   | bal_acc |
| 5,un      | 0.7530132 | 2.171307e-02  | 4.196890e-02  | 0.0436984363  | 0.044538555  | 0.0435712299  | 40,100      | -,            | avg   | bal_acc |
| 8,un      | 0.7798720 | 1.642404e-02  | 3.436652e-02  | 0.0415253776  | 0.050450888  | 0.0497730597  | 40          | -             | avg   | bal_acc |
| 11,un     | 0.7878845 | 1.300386e-02  | 4.319348e-02  | 0.0498694490  | 0.059304915  | 0.0663026778  |             |               | avg   | bal_acc |
| 14,un     | 0.7945798 | 1.022274e-02  | 3.621002e-02  | 0.0474373643  | 0.063786325  | 0.0712968974  |             |               | avg   | bal_acc |
| 17,un     | 0.7966506 | 9.377304e-03  | 3.217970e-02  | 0.0508475165  | 0.066559556  | 0.0787297740  | 100         | +             | avg   | bal_acc |
| 20,un     | 0.7931807 | 1.052797e-02  | 3.852729e-02  | 0.0555000707  | 0.073193517  | 0.0858276029  | 40          | -             | avg   | bal_acc |
| 23,un     | 0.7780494 | 2.899523e-02  | 5.472844e-02  | 0.0694241360  | 0.089135154  | 0.1028903249  | 40          | +             | avg   | bal_acc |
| 26,un     | 0.7876020 | 2.440101e-02  | 4.480423e-02  | 0.0584252311  | 0.079645725  | 0.0951223077  |             |               | avg   | bal_acc |
| 5,ov      | 0.7996588 | 3.147330e-03  | 8.322007e-04  | 0.0057424603  | 0.001365682  | 0.0024151346  |             |               | avg   | bal_acc |
| 8,ov      | 0.8392254 | 3.272287e-04  | -7.205331e-04 | 0.0009605496  | -0.002380888 | -0.0015091155 |             |               | avg   | bal_acc |
| 11,ov     | 0.8643922 | -9.358251e-04 | -3.616886e-03 | 0.0006897627  | -0.002122796 | -0.0003739277 |             |               | avg   | bal_acc |
| 14,ov     | 0.8786327 | 2.056289e-05  | -3.362972e-03 | 0.0014324197  | -0.002805004 | 0.0012362852  |             |               | avg   | bal_acc |
| 17,ov     | 0.8881530 | -4.872746e-04 | -3.570855e-03 | 0.0001003539  | -0.001845841 | -0.0015089828 |             |               | avg   | bal_acc |
| 20,ov     | 0.8934150 | -7.024645e-04 | -1.874533e-03 | 0.0006989876  | -0.002077656 | -0.0001867076 |             |               | avg   | bal_acc |
| 23,ov     | 0.8961439 | 5.119201e-04  | -2.462866e-03 | -0.0005937316 | -0.002126653 | -0.0004712619 |             |               | avg   | bal_acc |
| 26,ov     | 0.8973743 | -2.947647e-05 | -3.167580e-03 | 0.0007004271  | -0.001689929 | -0.0002416934 |             |               | avg   | bal_acc |
| 5,weight  | 0.7846641 | 1.999544e-02  | 2.364293e-02  | 0.0272214423  | 0.022497648  | 0.0165387231  |             |               | avg   | bal_acc |
| 8,weight  | 0.8163472 | 1.841668e-02  | 2.181698e-02  | 0.0288764629  | 0.022306699  | 0.0200272313  |             |               | avg   | bal_acc |
| 11,weight | 0.8309427 | 1.739777e-02  | 2.387585e-02  | 0.0306008413  | 0.032044796  | 0.0277474675  |             |               | avg   | bal_acc |
| 14,weight | 0.8375186 | 1.733851e-02  | 2.295779e-02  | 0.0336775840  | 0.038819644  | 0.0345974671  | 100         | -             | avg   | bal_acc |
| 17,weight | 0.8446159 | 8.202885e-03  | 2.003794e-02  | 0.0300760022  | 0.037983171  | 0.0339450256  | 40          | +             | avg   | bal_acc |
| 20,weight | 0.8454206 | 7.480487e-03  | 1.824899e-02  | 0.0327879369  | 0.040077226  | 0.0361404386  |             |               | avg   | bal_acc |
| 23,weight | 0.8466829 | 3.078377e-03  | 1.434365e-02  | 0.0341457029  | 0.040310119  | 0.0368344362  |             |               | avg   | bal_acc |
| 26,weight | 0.8448922 | 2.572407e-03  | 1.746681e-02  | 0.0339789221  | 0.043213970  | 0.0398771800  |             |               | avg   | bal_acc |

Table 10: Table of results for imbalance level for random forest with measurement balanced accuracy

| desc      | X40       | X100          | X200          | X400          | X1000        | X2000         | mean_median | larger_or_not | class | measure |
|-----------|-----------|---------------|---------------|---------------|--------------|---------------|-------------|---------------|-------|---------|
| 5,none    | NA        | NA            | NA            | NA            | NA           | NA            |             |               | avg   | F1      |
| 8,none    | NA        | NA            | NA            | NA            | NA           | NA            |             |               | avg   | F1      |
| 11,none   | NA        | NA            | NA            | NA            | NA           | NA            |             |               | avg   | F1      |
| 14,none   | NA        | NA            | NA            | NA            | NA           | NA            |             |               | avg   | F1      |
| 17,none   | NA        | NA            | NA            | NA            | NA           | NA            | 400         | +             | avg   | F1      |
| 20,none   | NA        | NA            | NA            | NA            | NA           | NA            |             |               | avg   | F1      |
| 23,none   | NA        | NA            | NA            | NA            | NA           | NA            |             |               | avg   | F1      |
| 26,none   | NA        | NA            | NA            | NA            | NA           | NA            | 200         | +             | avg   | F1      |
| 5,un      | 0.5713699 | 4.753796e-02  | 0.0819054289  | 0.0901364990  | 0.094807082  | 0.0904268445  | 40,100,200  | +, -, +       | avg   | F1      |
| 8,un      | 0.6267237 | 3.484529e-02  | 0.0648425599  | 0.0820975939  | 0.098896124  | 0.0963578651  | 40,100,2000 | -, -, +       | avg   | F1      |
| 11,un     | 0.6411239 | 3.066067e-02  | 0.0792033957  | 0.0952537851  | 0.112212701  | 0.1232139140  | 40          | -             | avg   | F1      |
| 14,un     | 0.6494322 | 3.101429e-02  | 0.0711357299  | 0.0939070381  | 0.122204776  | 0.1340475408  |             |               | avg   | F1      |
| 17,un     | 0.6540242 | 2.730085e-02  | 0.0631590286  | 0.0978928742  | 0.125258243  | 0.1446862776  | 40,100      | -, +          | avg   | F1      |
| 20,un     | 0.6509757 | 2.526286e-02  | 0.0718664315  | 0.1033635854  | 0.133380062  | 0.1535583869  |             |               | avg   | F1      |
| 23,un     | 0.6275540 | 5.384782e-02  | 0.0963759311  | 0.1242304593  | 0.158002656  | 0.1799605038  | 40,100      | +, -          | avg   | F1      |
| 26,un     | 0.6411875 | 4.893655e-02  | 0.0819077754  | 0.1088674089  | 0.144452678  | 0.1692566808  | 100         | -             | avg   | F1      |
| 5,ov      | 0.6701195 | 3.930494e-03  | 0.0002372901  | 0.0126530333  | 0.002072944  | 0.0044940219  |             |               | avg   | F1      |
| 8,ov      | 0.7400692 | -5.385612e-05 | -0.0014376049 | 0.0014405848  | -0.003939367 | -0.0031179314 |             |               | avg   | F1      |
| 11,ov     | 0.7813220 | -1.558114e-03 | -0.0056365530 | 0.0012279530  | -0.003362841 | -0.0007428342 |             |               | avg   | F1      |
| 14,ov     | 0.8044867 | -9.502515e-05 | -0.0054261555 | 0.0022532197  | -0.004760809 | 0.0017585169  |             |               | avg   | F1      |
| 17,ov     | 0.8197818 | -8.500445e-04 | -0.0057671498 | 0.0002013685  | -0.003048545 | -0.0025960370 |             |               | avg   | F1      |
| 20,ov     | 0.8281792 | -1.198926e-03 | -0.0030490316 | 0.0011736787  | -0.003433255 | -0.0003930580 |             |               | avg   | F1      |
| 23,ov     | 0.8325324 | 6.667818e-04  | -0.0040179012 | -0.0009205022 | -0.003602840 | -0.0008542043 |             |               | avg   | F1      |
| 26,ov     | 0.8344802 | -9.773808e-05 | -0.0052081058 | 0.0012078079  | -0.002907389 | -0.0005063742 |             |               | avg   | F1      |
| 5,weight  | 0.5851592 | 3.065030e-02  | 0.0646037924  | 0.0707988141  | 0.094211476  | 0.0891380853  | 40,100,400  | -, +, +       | avg   | F1      |
| 8,weight  | 0.6477610 | 2.219145e-02  | 0.0487612341  | 0.0677627394  | 0.084592123  | 0.0873155616  | 100,400     | -, -          | avg   | F1      |
| 11,weight | 0.6880583 | 1.839020e-02  | 0.0429983590  | 0.0596554724  | 0.084942433  | 0.0841918992  |             |               | avg   | F1      |
| 14,weight | 0.7203294 | 1.038244e-02  | 0.0294357429  | 0.0482122453  | 0.075303547  | 0.0737327659  |             |               | avg   | F1      |
| 17,weight | 0.7449743 | -1.115345e-02 | 0.0200355278  | 0.0345842736  | 0.061600014  | 0.0593890136  |             |               | avg   | F1      |
| 20,weight | 0.7461051 | 9.896757e-04  | 0.0183388074  | 0.0428297516  | 0.066093052  | 0.0630512637  | 200         | -             | avg   | F1      |
| 23,weight | 0.7564186 | -1.204900e-02 | 0.0018259301  | 0.0384400826  | 0.058232699  | 0.0558247302  | 200         | +             | avg   | F1      |
| 26,weight | 0.7581276 | -2.313926e-02 | 0.0043334138  | 0.0342220459  | 0.058727089  | 0.0561996492  |             |               | avg   | F1      |

Table 11: Table of results for imbalance level for random forest with measurement F1



## A.4 Imbalance method

| desc    | none      | ov          | un            | weight        | mean_median | larger_or_not | class | measure |
|---------|-----------|-------------|---------------|---------------|-------------|---------------|-------|---------|
| 5,2000  | 0.8005072 | 0.001566719 | -0.0039227288 | 6.956280e-04  |             |               | avg   | bal_acc |
| 8,2000  | 0.8349873 | 0.002729002 | -0.0053421724 | 1.387193e-03  |             |               | avg   | bal_acc |
| 11,2000 | 0.8586127 | 0.005405618 | -0.0044255215 | 7.746942e-05  |             |               | avg   | bal_acc |
| 14,2000 | 0.8728211 | 0.007047932 | -0.0069443614 | -7.050534e-04 |             |               | avg   | bal_acc |
| 17,2000 | 0.8789949 | 0.007649130 | -0.0036145547 | -4.339898e-04 |             |               | avg   | bal_acc |
| 20,2000 | 0.8829964 | 0.010231922 | -0.0039880537 | -1.435314e-03 |             |               | avg   | bal_acc |
| 23,2000 | 0.8847494 | 0.010923251 | -0.0038096370 | -1.232041e-03 |             |               | avg   | bal_acc |
| 26,2000 | 0.8850675 | 0.012065103 | -0.0023432854 | -2.981294e-04 |             |               | avg   | bal_acc |
| 5,1000  | 0.7295205 | 0.071504007 | 0.0680313370  | 7.764129e-02  |             |               | avg   | bal_acc |
| 8,1000  | 0.7983084 | 0.038536054 | 0.0320144808  | 4.034549e-02  |             |               | avg   | bal_acc |
| 11,1000 | 0.8303152 | 0.031954229 | 0.0168741960  | 3.267228e-02  |             |               | avg   | bal_acc |
| 14,1000 | 0.8553724 | 0.020455307 | 0.0029937306  | 2.096579e-02  |             |               | avg   | bal_acc |
| 17,1000 | 0.8653016 | 0.021005539 | -0.0020915047 | 1.729742e-02  |             |               | avg   | bal_acc |
| 20,1000 | 0.8720253 | 0.019312001 | -0.0056511126 | 1.347250e-02  |             |               | avg   | bal_acc |
| 23,1000 | 0.8723322 | 0.021685064 | -0.0051476044 | 1.466085e-02  |             |               | avg   | bal_acc |
| 26,1000 | 0.8731940 | 0.022490429 | -0.0059463063 | 1.491222e-02  |             |               | avg   | bal_acc |
| 5,400   | 0.7133836 | 0.092017694 | 0.0833281267  | 9.850200e-02  |             |               | avg   | bal_acc |
| 8,400   | 0.7818566 | 0.058329302 | 0.0395407802  | 6.336706e-02  |             |               | avg   | bal_acc |
| 11,400  | 0.8083158 | 0.056766173 | 0.0294381151  | 5.322771e-02  |             |               | avg   | bal_acc |
| 14,400  | 0.8251972 | 0.054867947 | 0.0168199860  | 4.599894e-02  |             |               | avg   | bal_acc |
| 17,400  | 0.8374595 | 0.050793840 | 0.0100385615  | 3.723236e-02  |             |               | avg   | bal_acc |
| 20,400  | 0.8412779 | 0.052836065 | 0.0074028618  | 3.693063e-02  |             |               | avg   | bal_acc |
| 23,400  | 0.8431902 | 0.052359938 | 0.0042833309  | 3.763838e-02  |             |               | avg   | bal_acc |
| 26,400  | 0.8443948 | 0.053679961 | 0.0016323752  | 3.447635e-02  |             |               | avg   | bal_acc |
| 5,200   | 0.7132667 | 0.087224265 | 0.0817154251  | 9.504031e-02  |             |               | avg   | bal_acc |
| 8,200   | 0.7814238 | 0.057081056 | 0.0328147595  | 5.674041e-02  |             |               | avg   | bal_acc |
| 11,200  | 0.8074308 | 0.053344534 | 0.0236471542  | 4.738772e-02  |             |               | avg   | bal_acc |
| 14,200  | 0.8199021 | 0.055367620 | 0.0108877108  | 4.057422e-02  |             |               | avg   | bal_acc |
| 17,200  | 0.8285617 | 0.056020478 | 0.0002685894  | 3.609215e-02  |             |               | avg   | bal_acc |
| 20,200  | 0.8319229 | 0.059617570 | -0.0002148928 | 3.174671e-02  |             |               | avg   | bal_acc |
| 23,200  | 0.8337489 | 0.059932122 | -0.0009710510 | 2.727765e-02  |             |               | avg   | bal_acc |
| 26,200  | 0.8347947 | 0.059412047 | -0.0023885309 | 2.756433e-02  |             |               | avg   | bal_acc |
| 5,100   | 0.7147896 | 0.088016548 | 0.0599367485  | 8.986998e-02  | un          | -             | avg   | bal_acc |
| 8,100   | 0.7774910 | 0.062061638 | 0.0188051031  | 5.727294e-02  |             |               | avg   | bal_acc |
| 11,100  | 0.8049781 | 0.058478344 | -0.0040897196 | 4.336239e-02  |             |               | avg   | bal_acc |
| 14,100  | 0.8196488 | 0.059004528 | -0.0148461960 | 3.520831e-02  | weight      | -             | avg   | bal_acc |
| 17,100  | 0.8265296 | 0.061136149 | -0.0205017141 | 2.628918e-02  | un          | +             | avg   | bal_acc |
| 20,100  | 0.8294000 | 0.063312565 | -0.0256912886 | 2.350113e-02  |             |               | avg   | bal_acc |
| 23,100  | 0.8317143 | 0.064941567 | -0.0246695941 | 1.804703e-02  |             |               | avg   | bal_acc |
| 26,100  | 0.8309513 | 0.066393546 | -0.0189483582 | 1.651332e-02  |             |               | avg   | bal_acc |
| 5,40    | 0.7104489 | 0.089209905 | 0.0425643613  | 7.421522e-02  | un          | -             | avg   | bal_acc |
| 8,40    | 0.7803107 | 0.058914655 | -0.0004386947 | 3.603650e-02  | un          | -             | avg   | bal_acc |
| 11,40   | 0.8064847 | 0.057907557 | -0.0186001873 | 2.445801e-02  |             |               | avg   | bal_acc |
| 14,40   | 0.8195087 | 0.059124021 | -0.0249288841 | 1.809285e-02  |             |               | avg   | bal_acc |
| 17,40   | 0.8269487 | 0.061204362 | -0.0302980792 | 1.766723e-02  | weight      | +             | avg   | bal_acc |
| 20,40   | 0.8315853 | 0.061829686 | -0.0384046007 | 1.383530e-02  | un          | -             | avg   | bal_acc |
| 23,40   | 0.8337063 | 0.062437637 | -0.0556568384 | 1.297665e-02  | un          | +             | avg   | bal_acc |
| 26,40   | 0.8329772 | 0.064397121 | -0.0453752684 | 1.191502e-02  |             |               | avg   | bal_acc |

Table 12: Table of results for imbalance method for random forest with measurement balanced accuracy

## B Gradient boosting

### B.1 Maximum depth

| desc        | X2        | X4            | X6           | X8          | X10          | X12          | X14          | mean_median | larger_or_not | class | measure |
|-------------|-----------|---------------|--------------|-------------|--------------|--------------|--------------|-------------|---------------|-------|---------|
| 2000,none   | 0.8521394 | 0.0339563617  | 0.050228664  | 0.054979166 | 0.056569078  | 0.058308018  | 0.058735547  |             |               | avg   | bal_acc |
| 1000,none   | 0.8530400 | 0.0311420643  | 0.046069256  | 0.050146509 | 0.052891907  | 0.053987752  | 0.055157203  |             |               | avg   | bal_acc |
| 400,none    | 0.8358426 | 0.0312243235  | 0.044920358  | 0.048649393 | 0.047400929  | 0.048455286  | 0.048148365  |             |               | avg   | bal_acc |
| 200,none    | 0.8238439 | 0.0342005730  | 0.046813467  | 0.049251670 | 0.050117483  | 0.048767735  | 0.047940066  |             |               | avg   | bal_acc |
| 100,none    | 0.8223678 | 0.0258465957  | 0.037558250  | 0.040298068 | 0.041100566  | 0.043503082  | 0.043626399  |             |               | avg   | bal_acc |
| 40,none     | 0.8209780 | 0.0242762926  | 0.035033753  | 0.037753723 | 0.039709070  | 0.039383582  | 0.039363833  |             |               | avg   | bal_acc |
| 2000,un     | 0.8492884 | 0.0327330627  | 0.050814253  | 0.054228633 | 0.057203858  | 0.059036654  | 0.058384114  |             |               | avg   | bal_acc |
| 1000,un     | 0.8433106 | 0.0237797004  | 0.034032672  | 0.036966643 | 0.039252756  | 0.039398048  | 0.043320047  |             |               | avg   | bal_acc |
| 400,un      | 0.8329201 | 0.0040427894  | 0.009909242  | 0.012416385 | 0.013345178  | 0.015934627  | 0.017138839  |             |               | avg   | bal_acc |
| 200,un      | 0.8195897 | -0.0001683832 | 0.013278730  | 0.007989753 | 0.004176373  | 0.007302107  | 0.004934429  | 12          | -             | avg   | bal_acc |
| 100,un      | 0.8036146 | 0.0030780211  | -0.001815532 | 0.001938715 | -0.007114754 | -0.011283389 | -0.010987187 | 8,12        | -,+           | avg   | bal_acc |
| 40,un       | 0.7582620 | 0.0040040790  | 0.012364705  | 0.002332950 | 0.008292295  | 0.010084410  | 0.010084410  | 4,10,12,14  | +,+,+,+       | avg   | bal_acc |
| 2000,ov     | 0.8540793 | 0.0369317556  | 0.062126050  | 0.070925875 | 0.075271253  | 0.075663108  | 0.076062301  |             |               | avg   | bal_acc |
| 1000,ov     | 0.8534686 | 0.0380235884  | 0.062639182  | 0.072969282 | 0.076773482  | 0.076644171  | 0.078085120  |             |               | avg   | bal_acc |
| 400,ov      | 0.8546487 | 0.0384690120  | 0.061864139  | 0.072592620 | 0.075608848  | 0.076204701  | 0.077493162  |             |               | avg   | bal_acc |
| 200,ov      | 0.8533513 | 0.0384221048  | 0.062067245  | 0.072179818 | 0.074799196  | 0.076190645  | 0.076872259  |             |               | avg   | bal_acc |
| 100,ov      | 0.8539857 | 0.0395520885  | 0.063109108  | 0.072931143 | 0.076766145  | 0.077514784  | 0.077922147  |             |               | avg   | bal_acc |
| 40,ov       | 0.8545481 | 0.0384785490  | 0.063260594  | 0.073256368 | 0.075901513  | 0.077267872  | 0.077224701  |             |               | avg   | bal_acc |
| 2000,weight | 0.8510801 | 0.0334937869  | 0.050957917  | 0.057061495 | 0.059632367  | 0.059812272  | 0.060012791  |             |               | avg   | bal_acc |
| 1000,weight | 0.8599720 | 0.0287289233  | 0.041782731  | 0.046114572 | 0.048335543  | 0.049293378  | 0.048373734  |             |               | avg   | bal_acc |
| 400,weight  | 0.8535117 | 0.0245369272  | 0.034383253  | 0.037625591 | 0.039020678  | 0.038181476  | 0.038931099  |             |               | avg   | bal_acc |
| 200,weight  | 0.8528019 | 0.0140650352  | 0.020782172  | 0.021581428 | 0.024580775  | 0.025860029  | 0.027413045  |             |               | avg   | bal_acc |
| 100,weight  | 0.8542187 | 0.0073398997  | 0.011609891  | 0.013080221 | 0.014990161  | 0.014758429  | 0.014225562  |             |               | avg   | bal_acc |
| 40,weight   | 0.8278537 | 0.0203096273  | 0.027803149  | 0.031446695 | 0.032989097  | 0.033316728  | 0.033947873  |             |               | avg   | bal_acc |

Table 14: Table of results for maximum depth for boosting with measurement balanced accuracy

| desc    | none      | ov          | un           | weight        | mean_median | larger_or_not | class | measure |
|---------|-----------|-------------|--------------|---------------|-------------|---------------|-------|---------|
| 5,2000  | 0.6717708 | 0.002842740 | -0.009974058 | 0.0025265491  |             |               | avg   | F1      |
| 8,2000  | 0.7325594 | 0.004391902 | -0.009477802 | 0.0025172270  | un          | +             | avg   | F1      |
| 11,2000 | 0.7718038 | 0.008775402 | -0.007465988 | 0.0004463967  |             |               | avg   | F1      |
| 14,2000 | 0.7949520 | 0.011293253 | -0.011472257 | -0.0008898527 |             |               | avg   | F1      |
| 17,2000 | 0.8049287 | 0.012257037 | -0.006218264 | -0.0005654625 |             |               | avg   | F1      |
| 20,2000 | 0.8113356 | 0.016450562 | -0.006801478 | -0.0021791685 |             |               | avg   | F1      |
| 23,2000 | 0.8140257 | 0.017652499 | -0.006511198 | -0.0017823952 |             |               | avg   | F1      |
| 26,2000 | 0.8144602 | 0.019513691 | -0.004015997 | -0.0001328646 |             |               | avg   | F1      |
| 5,1000  | NA        | NA          | NA           | NA            |             |               | avg   | F1      |
| 8,1000  | 0.6354315 | 0.100698287 | 0.090188277  | 0.0969216081  |             |               | avg   | F1      |
| 11,1000 | 0.7113304 | 0.066628774 | 0.042006177  | 0.0616703091  |             |               | avg   | F1      |
| 14,1000 | 0.7638987 | 0.035827227 | 0.007738278  | 0.0317342273  |             |               | avg   | F1      |
| 17,1000 | 0.7820687 | 0.034664548 | -0.002786280 | 0.0245055575  |             |               | avg   | F1      |
| 20,1000 | 0.7943061 | 0.030439855 | -0.009950314 | 0.0178921098  |             |               | avg   | F1      |
| 23,1000 | 0.7948694 | 0.034060142 | -0.009312767 | 0.0197818519  |             |               | avg   | F1      |
| 26,1000 | 0.7962184 | 0.035354460 | -0.010578216 | 0.0206363590  |             |               | avg   | F1      |
| 5,400   | NA        | NA          | NA           | NA            | weight      | +             | avg   | F1      |
| 8,400   | 0.6168187 | 0.124691038 | 0.092002546  | 0.0987050246  | weight      | -             | avg   | F1      |
| 11,400  | 0.6609763 | 0.121573667 | 0.075401361  | 0.0867374478  |             |               | avg   | F1      |
| 14,400  | 0.6933216 | 0.113418363 | 0.050017648  | 0.0752200342  |             |               | avg   | F1      |
| 17,400  | 0.7219613 | 0.098021919 | 0.029955809  | 0.0575972741  | none        | +             | avg   | F1      |
| 20,400  | 0.7292093 | 0.100143574 | 0.025129996  | 0.0597255947  |             |               | avg   | F1      |
| 23,400  | 0.7320342 | 0.099577678 | 0.019750234  | 0.0628244341  |             |               | avg   | F1      |
| 26,400  | 0.7360241 | 0.099663930 | 0.014030788  | 0.0563255889  |             |               | avg   | F1      |
| 5,200   | NA        | NA          | NA           | NA            | un          | +             | avg   | F1      |
| 8,200   | NA        | NA          | NA           | NA            |             |               | avg   | F1      |
| 11,200  | 0.6673256 | 0.108359873 | 0.053001683  | 0.0637310461  |             |               | avg   | F1      |
| 14,200  | 0.6827562 | 0.116304386 | 0.037811737  | 0.0670089295  |             |               | avg   | F1      |
| 17,200  | 0.6996770 | 0.114337636 | 0.017506199  | 0.0653327638  |             |               | avg   | F1      |
| 20,200  | 0.7035211 | 0.121609095 | 0.019321073  | 0.0609228813  | weight      | -             | avg   | F1      |
| 23,200  | 0.7084080 | 0.120106541 | 0.015521968  | 0.0498365438  | weight      | +             | avg   | F1      |
| 26,200  | 0.7115756 | 0.117696518 | 0.011519657  | 0.0508854589  | none        | +             | avg   | F1      |
| 5,100   | NA        | NA          | NA           | NA            | un,weight   | -,+           | avg   | F1      |
| 8,100   | NA        | NA          | NA           | NA            | un,weight   | -, -          | avg   | F1      |
| 11,100  | NA        | NA          | NA           | NA            |             |               | avg   | F1      |
| 14,100  | NA        | NA          | NA           | NA            |             |               | avg   | F1      |
| 17,100  | NA        | NA          | NA           | NA            | un          | +             | avg   | F1      |
| 20,100  | NA        | NA          | NA           | NA            |             |               | avg   | F1      |
| 23,100  | NA        | NA          | NA           | NA            | un          | -             | avg   | F1      |
| 26,100  | NA        | NA          | NA           | NA            | un          | -             | avg   | F1      |
| 5,40    | NA        | NA          | NA           | NA            | un,weight   | +,-           | avg   | F1      |
| 8,40    | NA        | NA          | NA           | NA            | un          | -             | avg   | F1      |
| 11,40   | NA        | NA          | NA           | NA            | un          | -             | avg   | F1      |
| 14,40   | NA        | NA          | NA           | NA            |             |               | avg   | F1      |
| 17,40   | NA        | NA          | NA           | NA            | un          | -             | avg   | F1      |
| 20,40   | NA        | NA          | NA           | NA            |             |               | avg   | F1      |
| 23,40   | NA        | NA          | NA           | NA            | un          | +             | avg   | F1      |
| 26,40   | NA        | NA          | NA           | NA            |             |               | avg   | F1      |

Table 13: Table of results for imbalance method for random forest with measurement F1

| desc        | X2        | X4           | X6           | X8           | X10          | X12         | X14          | mean_median      | larger_or_not | class | measure |
|-------------|-----------|--------------|--------------|--------------|--------------|-------------|--------------|------------------|---------------|-------|---------|
| 2000,none   | 0.7617314 | 0.054612917  | 0.080948741  | 0.088553708  | 0.091103494  | 0.09397202  | 0.094526226  |                  |               | avg   | F1      |
| 1000,none   | 0.7629472 | 0.049607825  | 0.073339625  | 0.080044405  | 0.084848591  | 0.08686986  | 0.088772301  |                  |               | avg   | F1      |
| 400,none    | 0.7301029 | 0.052651908  | 0.074528404  | 0.081159706  | 0.078097565  | 0.07929723  | 0.079079098  |                  |               | avg   | F1      |
| 200,none    | 0.6967694 | 0.062059997  | 0.083176574  | 0.087811459  | 0.089259496  | 0.08530726  | 0.082586588  | 10               | +             | avg   | F1      |
| 100,none    | 0.7031895 | 0.045977099  | 0.049678429  | 0.059398983  | 0.065600521  | 0.07131883  | 0.078190495  | 12               | -             | avg   | F1      |
| 40,none     | 0.7362073 | 0.038977138  | NA           | NA           | NA           | NA          | NA           |                  |               | avg   | F1      |
| 2000,un     | 0.7569250 | 0.052699979  | 0.081834222  | 0.087288987  | 0.092134543  | 0.09514569  | 0.094021562  |                  |               | avg   | F1      |
| 1000,un     | 0.7472194 | 0.038544324  | 0.054808318  | 0.059839061  | 0.063453373  | 0.06364172  | 0.069807161  | 8                | -             | avg   | F1      |
| 400,un      | 0.7296315 | 0.006849977  | 0.016220297  | 0.020823360  | 0.021849889  | 0.02614033  | 0.028131452  | 4,10,12          | +,,-,-        | avg   | F1      |
| 200,un      | 0.7048347 | -0.001283462 | 0.021289857  | 0.013310284  | 0.006512072  | 0.01132267  | 0.008201771  | 12               | -             | avg   | F1      |
| 100,un      | 0.6778912 | 0.006250384  | -0.002346113 | 0.003452413  | -0.011554760 | -0.01788816 | -0.018450931 | 4,8,14           | +,,-,-        | avg   | F1      |
| 40,un       | 0.6002771 | 0.004029472  | 0.016367561  | -0.002426619 | 0.007681287  | 0.01057946  | 0.010579463  | 2,4,6,8,10,12,14 | +,+,-,-,+,+,+ | avg   | F1      |
| 2000,ov     | 0.7649179 | 0.059366402  | 0.099894057  | 0.114106840  | 0.121163008  | 0.12180911  | 0.122460401  |                  |               | avg   | F1      |
| 1000,ov     | 0.7638635 | 0.061225592  | 0.100854371  | 0.117567484  | 0.123714430  | 0.12344871  | 0.125790943  |                  |               | avg   | F1      |
| 400,ov      | 0.7658446 | 0.061849566  | 0.099617445  | 0.116892099  | 0.121779918  | 0.12269509  | 0.124786763  |                  |               | avg   | F1      |
| 200,ov      | 0.7636468 | 0.061785447  | 0.099945440  | 0.116253323  | 0.120453315  | 0.12267461  | 0.123821836  |                  |               | avg   | F1      |
| 100,ov      | 0.7646656 | 0.063752474  | 0.101692960  | 0.117557291  | 0.123753303  | 0.12496497  | 0.125622695  |                  |               | avg   | F1      |
| 40,ov       | 0.7655994 | 0.062090252  | 0.101920293  | 0.117974135  | 0.122267896  | 0.12454174  | 0.124483298  |                  |               | avg   | F1      |
| 2000,weight | 0.7600631 | 0.054011805  | 0.082111517  | 0.091924845  | 0.096191416  | 0.09645485  | 0.096754412  |                  |               | avg   | F1      |
| 1000,weight | 0.7679950 | 0.048792663  | 0.070297698  | 0.077861379  | 0.081612722  | 0.08322229  | 0.081872738  | 12               | +             | avg   | F1      |
| 400,weight  | 0.7353360 | 0.054685986  | 0.075424904  | 0.082389788  | 0.085182474  | 0.08402482  | 0.085082648  | 10,12            | -,-           | avg   | F1      |
| 200,weight  | 0.7237731 | 0.046068143  | 0.059127906  | 0.059590368  | 0.066183805  | 0.06910346  | 0.071895382  |                  |               | avg   | F1      |
| 100,weight  | 0.7147900 | 0.042660530  | 0.049410465  | 0.054438376  | 0.055041154  | 0.05457717  | 0.052774089  | 6,8,12           | +,+,+         | avg   | F1      |
| 40,weight   | 0.6859924 | 0.081890589  | NA           | NA           | NA           | 0.09026768  | 0.093147009  |                  |               | avg   | F1      |

Table 15: Table of results for maximum depth for boosting with measurement F1

## B.2 Class imbalance

| desc      | X40       | X100          | X200         | X400          | X1000        | X2000        | mean_median | larger_or_not | class | measure |
|-----------|-----------|---------------|--------------|---------------|--------------|--------------|-------------|---------------|-------|---------|
| 2,none    | 0.8209780 | 0.0013898329  | 0.002865869  | 1.486457e-02  | 0.032061962  | 0.031161365  |             |               | avg   | bal_acc |
| 4,none    | 0.8452543 | 0.0029601360  | 0.012790149  | 2.181260e-02  | 0.038927733  | 0.040841434  |             |               | avg   | bal_acc |
| 6,none    | 0.8560118 | 0.0039143296  | 0.014645582  | 2.475117e-02  | 0.043097465  | 0.046356276  |             |               | avg   | bal_acc |
| 8,none    | 0.8587317 | 0.0039341782  | 0.014363816  | 2.576024e-02  | 0.044454747  | 0.048386808  |             |               | avg   | bal_acc |
| 10,none   | 0.8606871 | 0.0027813284  | 0.013274282  | 2.255643e-02  | 0.045244798  | 0.048021373  |             |               | avg   | bal_acc |
| 12,none   | 0.8603616 | 0.0055093331  | 0.012250022  | 2.393627e-02  | 0.046666132  | 0.050085801  |             |               | avg   | bal_acc |
| 14,none   | 0.8603418 | 0.0056523987  | 0.011442102  | 2.364910e-02  | 0.047855332  | 0.050533079  |             |               | avg   | bal_acc |
| 2,un      | 0.7582620 | 0.0453525724  | 0.061327680  | 7.465808e-02  | 0.085048572  | 0.091026331  |             |               | avg   | bal_acc |
| 4,un      | 0.7622661 | 0.0444265145  | 0.057155218  | 7.469679e-02  | 0.104824193  | 0.119755315  | 40          | +             | avg   | bal_acc |
| 6,un      | 0.7706267 | 0.0311723352  | 0.062241704  | 7.220261e-02  | 0.106716539  | 0.129475878  |             |               | avg   | bal_acc |
| 8,un      | 0.7605950 | 0.0449583368  | 0.066984482  | 8.474151e-02  | 0.119682265  | 0.142922013  | 100         | -             | avg   | bal_acc |
| 10,un     | 0.7665543 | 0.0299455225  | 0.057211758  | 7.971096e-02  | 0.116009032  | 0.139937893  | 40          | +             | avg   | bal_acc |
| 12,un     | 0.7683464 | 0.0239847730  | 0.058545377  | 8.050829e-02  | 0.114362210  | 0.139978574  | 40,100,200  | +,+,-         | avg   | bal_acc |
| 14,un     | 0.7683464 | 0.0242809752  | 0.056177698  | 8.171251e-02  | 0.118284209  | 0.139326034  | 40          | +             | avg   | bal_acc |
| 2,ov      | 0.8545481 | -0.0005624022 | -0.001196796 | 1.005461e-04  | -0.001079504 | -0.000468849 |             |               | avg   | bal_acc |
| 4,ov      | 0.8930267 | 0.0005111373  | -0.001253241 | 9.100914e-05  | -0.001534464 | -0.002015642 |             |               | avg   | bal_acc |
| 6,ov      | 0.9178087 | -0.0007138879 | -0.002390145 | -1.295909e-03 | -0.001700916 | -0.001603393 |             |               | avg   | bal_acc |
| 8,ov      | 0.9278045 | -0.0008876265 | -0.002273346 | -5.632011e-04 | -0.001366590 | -0.002799342 |             |               | avg   | bal_acc |
| 10,ov     | 0.9304497 | 0.0003022297  | -0.002299114 | -1.921188e-04 | -0.000207534 | -0.001099109 |             |               | avg   | bal_acc |
| 12,ov     | 0.9318160 | -0.0003154898 | -0.002274023 | -9.626250e-04 | -0.001703204 | -0.002073613 |             |               | avg   | bal_acc |
| 14,ov     | 0.9317728 | 0.0001350438  | -0.001549238 | 3.690075e-04  | -0.000219084 | -0.001631249 |             |               | avg   | bal_acc |
| 2,weight  | 0.8278537 | 0.0263649649  | 0.024948123  | 2.565801e-02  | 0.032118250  | 0.023226326  |             |               | avg   | bal_acc |
| 4,weight  | 0.8481634 | 0.0133952373  | 0.018703531  | 2.988531e-02  | 0.040537546  | 0.036410486  |             |               | avg   | bal_acc |
| 6,weight  | 0.8556569 | 0.0101717064  | 0.017927145  | 3.223812e-02  | 0.046097832  | 0.046381094  |             |               | avg   | bal_acc |
| 8,weight  | 0.8593004 | 0.0079984904  | 0.015082856  | 3.183691e-02  | 0.046786127  | 0.048841125  |             |               | avg   | bal_acc |
| 10,weight | 0.8608428 | 0.0083660293  | 0.016539801  | 3.168960e-02  | 0.047464697  | 0.049869596  |             |               | avg   | bal_acc |
| 12,weight | 0.8611705 | 0.0078066662  | 0.017491424  | 3.052276e-02  | 0.048094901  | 0.049721870  |             |               | avg   | bal_acc |
| 14,weight | 0.8618016 | 0.0066426543  | 0.018413295  | 3.064124e-02  | 0.046544112  | 0.049291244  |             |               | avg   | bal_acc |

Table 16: Table of results for imbalance level for boosting with measurement balanced accuracy

| desc      | X40       | X100          | X200         | X400          | X1000         | X2000         | mean_median  | larger_or_not | class | measure |
|-----------|-----------|---------------|--------------|---------------|---------------|---------------|--------------|---------------|-------|---------|
| 2,none    | 0.7362073 | -0.0330177795 | -0.039437905 | -6.104379e-03 | 0.0267399540  | 0.0255241350  |              |               | avg   | F1      |
| 4,none    | 0.7751844 | -0.0260178180 | -0.016355046 | 7.570392e-03  | 0.0373706407  | 0.0411599139  |              |               | avg   | F1      |
| 6,none    | NA        | NA            | NA           | NA            | NA            | NA            |              |               | avg   | F1      |
| 8,none    | NA        | NA            | NA           | NA            | NA            | NA            |              |               | avg   | F1      |
| 10,none   | NA        | NA            | NA           | NA            | NA            | NA            | 200          | +             | avg   | F1      |
| 12,none   | NA        | NA            | NA           | NA            | NA            | NA            | 100          | -             | avg   | F1      |
| 14,none   | NA        | NA            | NA           | NA            | NA            | NA            |              |               | avg   | F1      |
| 2,un      | 0.6002771 | 0.0776140600  | 0.104557540  | 1.293544e-01  | 0.1469422350  | 0.1566479010  | 40           | +             | avg   | F1      |
| 4,un      | 0.6043066 | 0.0798349722  | 0.099244606  | 1.321749e-01  | 0.1814570877  | 0.2053184082  | 40,100,400   | +,+,+         | avg   | F1      |
| 6,un      | 0.6166447 | 0.0589003864  | 0.109479836  | 1.292071e-01  | 0.1853829921  | 0.2221145618  | 40           | -             | avg   | F1      |
| 8,un      | 0.5978505 | 0.0834930914  | 0.120294443  | 1.526044e-01  | 0.2092079150  | 0.2463635070  | 40,100,1000  | +,-,-         | avg   | F1      |
| 10,un     | 0.6079584 | 0.0583780138  | 0.103388325  | 1.435230e-01  | 0.2027143211  | 0.2411011576  | 40,400       | +,-           | avg   | F1      |
| 12,un     | 0.6108566 | 0.0491464329  | 0.105300751  | 1.449152e-01  | 0.2000044876  | 0.2412141281  | 40,200,400   | +,-,-         | avg   | F1      |
| 14,un     | 0.6108566 | 0.0485836655  | 0.102179847  | 1.469064e-01  | 0.2061699322  | 0.2400899994  | 40,100       | +,-           | avg   | F1      |
| 2,ov      | 0.7655994 | -0.0009337660 | -0.001952547 | 2.452172e-04  | -0.0017358762 | -0.0006815121 |              |               | avg   | F1      |
| 4,ov      | 0.8276896 | 0.0007284560  | -0.002257352 | 4.530466e-06  | -0.0026005364 | -0.0034053627 |              |               | avg   | F1      |
| 6,ov      | 0.8675197 | -0.0011610985 | -0.003927400 | -2.057631e-03 | -0.0028017984 | -0.0027077481 |              |               | avg   | F1      |
| 8,ov      | 0.8835735 | -0.0013506102 | -0.003673359 | -8.368191e-04 | -0.0021425273 | -0.0045488067 |              |               | avg   | F1      |
| 10,ov     | 0.8878673 | 0.0005516416  | -0.003767128 | -2.427601e-04 | -0.0002893421 | -0.0017864001 |              |               | avg   | F1      |
| 12,ov     | 0.8901411 | -0.0005105373 | -0.003819677 | -1.601434e-03 | -0.0028289072 | -0.0034141497 |              |               | avg   | F1      |
| 14,ov     | 0.8900827 | 0.0002056309  | -0.002614009 | 5.486822e-04  | -0.0004282308 | -0.0027044094 |              |               | avg   | F1      |
| 2,weight  | 0.6859924 | 0.0287976124  | 0.037780698  | 4.934359e-02  | 0.0820025857  | 0.0740706310  |              |               | avg   | F1      |
| 4,weight  | 0.7678830 | -0.0104324461 | 0.001958252  | 2.213899e-02  | 0.0489046598  | 0.0461918476  |              |               | avg   | F1      |
| 6,weight  | NA        | NA            | NA           | NA            | NA            | NA            | 100          | +             | avg   | F1      |
| 8,weight  | NA        | NA            | NA           | NA            | NA            | NA            | 100          | +             | avg   | F1      |
| 10,weight | NA        | NA            | NA           | NA            | NA            | NA            | 400          | -             | avg   | F1      |
| 12,weight | 0.7762601 | -0.0068928972 | 0.016616479  | 4.310073e-02  | 0.0749572000  | 0.0802578010  | 100,400,1000 | +,-,+         | avg   | F1      |
| 14,weight | 0.7791394 | -0.0115753075 | 0.016529071  | 4.127923e-02  | 0.0707283149  | 0.0776780344  |              |               | avg   | F1      |

Table 17: Table of results for imbalance level for boosting with measurement F1



### B.3 Imbalance method

| desc    | none      | ov           | un           | weight        | mean_median | larger_or_not | class | measure |
|---------|-----------|--------------|--------------|---------------|-------------|---------------|-------|---------|
| 2,2000  | 0.8521394 | 0.0019399116 | -0.002851019 | -0.0010593251 |             |               | avg   | bal_acc |
| 4,2000  | 0.8860957 | 0.0049153056 | -0.004074318 | -0.0015218999 |             |               | avg   | bal_acc |
| 6,2000  | 0.9023680 | 0.0138372969 | -0.002265431 | -0.0003300722 |             |               | avg   | bal_acc |
| 8,2000  | 0.9071185 | 0.0178866201 | -0.003601552 | 0.0010230035  |             |               | avg   | bal_acc |
| 10,2000 | 0.9087085 | 0.0206420868 | -0.002216239 | 0.0020039638  |             |               | avg   | bal_acc |
| 12,2000 | 0.9104474 | 0.0192950026 | -0.002122383 | 0.0004449292  |             |               | avg   | bal_acc |
| 14,2000 | 0.9108749 | 0.0192666659 | -0.003202452 | 0.0002179192  |             |               | avg   | bal_acc |
| 2,1000  | 0.8530400 | 0.0004286605 | -0.009729375 | 0.0069320029  |             |               | avg   | bal_acc |
| 4,1000  | 0.8841820 | 0.0073101847 | -0.017091738 | 0.0045188618  |             |               | avg   | bal_acc |
| 6,1000  | 0.8991092 | 0.0169985858 | -0.021765959 | 0.0026454778  |             |               | avg   | bal_acc |
| 8,1000  | 0.9031865 | 0.0232514336 | -0.022909240 | 0.0029000667  |             |               | avg   | bal_acc |
| 10,1000 | 0.9059319 | 0.0243102363 | -0.023368525 | 0.0023756390  |             |               | avg   | bal_acc |
| 12,1000 | 0.9070277 | 0.0230850795 | -0.024319079 | 0.0022376284  |             |               | avg   | bal_acc |
| 14,1000 | 0.9081972 | 0.0233565775 | -0.021566531 | 0.0001485335  |             |               | avg   | bal_acc |
| 2,400   | 0.8358426 | 0.0188061048 | -0.002922475 | 0.0176691606  |             |               | avg   | bal_acc |
| 4,400   | 0.8670669 | 0.0260507933 | -0.030104009 | 0.0109817643  |             |               | avg   | bal_acc |
| 6,400   | 0.8807629 | 0.0357498856 | -0.037933591 | 0.0071320563  |             |               | avg   | bal_acc |
| 8,400   | 0.8844920 | 0.0427493320 | -0.039155483 | 0.0066453586  |             |               | avg   | bal_acc |
| 10,400  | 0.8832435 | 0.0470140240 | -0.036978226 | 0.0092889102  |             |               | avg   | bal_acc |
| 12,400  | 0.8842979 | 0.0465555198 | -0.035443133 | 0.0073953512  |             |               | avg   | bal_acc |
| 14,400  | 0.8839909 | 0.0481509020 | -0.033932001 | 0.0084518951  |             |               | avg   | bal_acc |
| 2,200   | 0.8238439 | 0.0295074606 | -0.004254174 | 0.0289579680  |             |               | avg   | bal_acc |
| 4,200   | 0.8580445 | 0.0337289924 | -0.038623130 | 0.0088224302  |             |               | avg   | bal_acc |
| 6,200   | 0.8706574 | 0.0447612394 | -0.037788911 | 0.0029266731  |             |               | avg   | bal_acc |
| 8,200   | 0.8730956 | 0.0524356091 | -0.045516091 | 0.0012877266  |             |               | avg   | bal_acc |
| 10,200  | 0.8739614 | 0.0541891730 | -0.050195284 | 0.0034212600  |             |               | avg   | bal_acc |
| 12,200  | 0.8726116 | 0.0569303710 | -0.045719802 | 0.0060502620  | un          | -             | avg   | bal_acc |
| 14,200  | 0.8717840 | 0.0584396535 | -0.047259811 | 0.0084309467  |             |               | avg   | bal_acc |
| 2,100   | 0.8223678 | 0.0316178907 | -0.018753245 | 0.0318508461  |             |               | avg   | bal_acc |
| 4,100   | 0.8482144 | 0.0453233835 | -0.041521820 | 0.0133441501  |             |               | avg   | bal_acc |
| 6,100   | 0.8599261 | 0.0571687488 | -0.058127027 | 0.0059024867  |             |               | avg   | bal_acc |
| 8,100   | 0.8626659 | 0.0642509656 | -0.057112599 | 0.0046329985  | un          | -             | avg   | bal_acc |
| 10,100  | 0.8634684 | 0.0672834698 | -0.066968566 | 0.0057404415  |             |               | avg   | bal_acc |
| 12,100  | 0.8658709 | 0.0656295934 | -0.073539716 | 0.0031061934  | un          | +             | avg   | bal_acc |
| 14,100  | 0.8659942 | 0.0659136383 | -0.073366831 | 0.0024500091  |             |               | avg   | bal_acc |
| 2,40    | 0.8209780 | 0.0335701258 | -0.062715985 | 0.0068757141  |             |               | avg   | bal_acc |
| 4,40    | 0.8452543 | 0.0477723822 | -0.082988199 | 0.0029090488  | un          | +             | avg   | bal_acc |
| 6,40    | 0.8560118 | 0.0617969662 | -0.085385033 | -0.0003548900 |             |               | avg   | bal_acc |
| 8,40    | 0.8587317 | 0.0690727703 | -0.098136758 | 0.0005686864  |             |               | avg   | bal_acc |
| 10,40   | 0.8606871 | 0.0697625684 | -0.094132760 | 0.0001557406  | un          | +             | avg   | bal_acc |
| 12,40   | 0.8603616 | 0.0714544162 | -0.092015156 | 0.0008088603  | un          | +             | avg   | bal_acc |
| 14,40   | 0.8603418 | 0.0714309932 | -0.091995408 | 0.0014597535  | un          | +             | avg   | bal_acc |

Table 18: Table of results for imbalance method for boosting with measurement  
balanced accuracy

| desc    | none      | ov           | un            | weight        | mean_median | larger_or_not | class | measure |
|---------|-----------|--------------|---------------|---------------|-------------|---------------|-------|---------|
| 2,2000  | 0.7617314 | 0.0031864501 | -0.0048063736 | -0.0016683497 |             |               | avg   | F1      |
| 4,2000  | 0.8163443 | 0.0079399349 | -0.0067193116 | -0.0022694613 |             |               | avg   | F1      |
| 6,2000  | 0.8426801 | 0.0221317658 | -0.0039208930 | -0.0005055739 |             |               | avg   | F1      |
| 8,2000  | 0.8502851 | 0.0287395824 | -0.0060710943 | 0.0017027878  |             |               | avg   | F1      |
| 10,2000 | 0.8528349 | 0.0332459642 | -0.0037753240 | 0.0034195724  |             |               | avg   | F1      |
| 12,2000 | 0.8557034 | 0.0310235315 | -0.0036327077 | 0.0008144724  |             |               | avg   | F1      |
| 14,2000 | 0.8562576 | 0.0311206246 | -0.0053110379 | 0.0005598363  |             |               | avg   | F1      |
| 2,1000  | 0.7629472 | 0.0009162668 | -0.0157278588 | 0.0050477859  |             |               | avg   | F1      |
| 4,1000  | 0.8125551 | 0.0125340344 | -0.0267913589 | 0.0042326242  |             |               | avg   | F1      |
| 6,1000  | 0.8362869 | 0.0284310124 | -0.0342591659 | 0.0020058587  |             |               | avg   | F1      |
| 8,1000  | 0.8429916 | 0.0384393462 | -0.0359332020 | 0.0028647601  | un          | -             | avg   | F1      |
| 10,1000 | 0.8477958 | 0.0397821055 | -0.0371230773 | 0.0018119169  |             |               | avg   | F1      |
| 12,1000 | 0.8498171 | 0.0374951229 | -0.0389559992 | 0.0014002204  | weight      | +             | avg   | F1      |
| 14,1000 | 0.8517195 | 0.0379349093 | -0.0346929991 | -0.0018517771 |             |               | avg   | F1      |
| 2,400   | 0.7301029 | 0.0357416931 | -0.0004713733 | 0.0052331210  |             |               | avg   | F1      |
| 4,400   | 0.7827548 | 0.0449393503 | -0.0462733050 | 0.0072671990  | un          | +             | avg   | F1      |
| 6,400   | 0.8046313 | 0.0608307340 | -0.0587794804 | 0.0061296207  |             |               | avg   | F1      |
| 8,400   | 0.8112626 | 0.0714740857 | -0.0608077198 | 0.0064632033  |             |               | avg   | F1      |
| 10,400  | 0.8082005 | 0.0794240465 | -0.0567190492 | 0.0123180296  | un,weight   | -, -          | avg   | F1      |
| 12,400  | 0.8094001 | 0.0791395556 | -0.0536282782 | 0.0099607142  | un,weight   | -, -          | avg   | F1      |
| 14,400  | 0.8091820 | 0.0814493584 | -0.0514190186 | 0.0112366718  |             |               | avg   | F1      |
| 2,200   | 0.6967694 | 0.0668774554 | 0.0080653056  | 0.0270037574  |             |               | avg   | F1      |
| 4,200   | 0.7588294 | 0.0666029054 | -0.0552781531 | 0.0110119039  |             |               | avg   | F1      |
| 6,200   | 0.7799459 | 0.0836463213 | -0.0538214115 | 0.0029550902  |             |               | avg   | F1      |
| 8,200   | 0.7845808 | 0.0953193187 | -0.0664358701 | -0.0012173342 |             |               | avg   | F1      |
| 10,200  | 0.7860289 | 0.0980712749 | -0.0746821180 | 0.0039280669  | none        | +             | avg   | F1      |
| 12,200  | 0.7820766 | 0.1042448112 | -0.0659192771 | 0.0107999580  | un          | -             | avg   | F1      |
| 14,200  | 0.7793560 | 0.1081127031 | -0.0663195120 | 0.0163125512  |             |               | avg   | F1      |
| 2,100   | 0.7031895 | 0.0614761106 | -0.0252983001 | 0.0116005462  |             |               | avg   | F1      |
| 4,100   | 0.7491666 | 0.0792514855 | -0.0650250156 | 0.0082839770  | un          | +             | avg   | F1      |
| 6,100   | 0.7528679 | 0.1134906421 | -0.0773228418 | 0.0113325826  | weight      | +             | avg   | F1      |
| 8,100   | 0.7625885 | 0.1196344184 | -0.0812448703 | 0.0066399389  | un,weight   | -, +          | avg   | F1      |
| 10,100  | 0.7687900 | 0.1196288925 | -0.1024535813 | 0.0010411792  |             |               | avg   | F1      |
| 12,100  | 0.7745083 | 0.1151222547 | -0.1145052921 | -0.0051411150 | none,weight | -, +          | avg   | F1      |
| 14,100  | 0.7813800 | 0.1089083105 | -0.1219397263 | -0.0138158601 | un          | -             | avg   | F1      |
| 2,40    | 0.7362073 | 0.0293920971 | -0.1359301397 | -0.0502148458 | un          | +             | avg   | F1      |
| 4,40    | 0.7751844 | 0.0525052115 | -0.1708778059 | -0.0073013949 | un          | +             | avg   | F1      |
| 6,40    | NA        | NA           | NA            | NA            | un          | -             | avg   | F1      |
| 8,40    | NA        | NA           | NA            | NA            | un          | +             | avg   | F1      |
| 10,40   | NA        | NA           | NA            | NA            | un          | +             | avg   | F1      |
| 12,40   | NA        | NA           | NA            | NA            | un          | +             | avg   | F1      |
| 14,40   | NA        | NA           | NA            | NA            | un          | +             | avg   | F1      |

Table 19: Table of results for imbalance method for boosting with measurement F1

## C Method comparison

| F1, 40    |            |           |            |           |              | F1, 100      |            |           |            |           |              |
|-----------|------------|-----------|------------|-----------|--------------|--------------|------------|-----------|------------|-----------|--------------|
| desc_rf   | desc_boost | mean_rf   | mean_boost | median_rf | median_boost | desc_rf      | desc_boost | mean_rf   | mean_boost | median_rf | median_boost |
| 26,ov     | 12,ov      | 0.8344802 | 0.8901411  | 0.8354830 | 0.8915350    | 22 26,ov     | 14,ov      | 0.8343825 | 0.8902883  | 0.8356170 | 0.8892185    |
| 23,ov     | 14,ov      | 0.8325324 | 0.8900827  | 0.8331794 | 0.8914043    | 18 23,ov     | 12,ov      | 0.8331992 | 0.8896306  | 0.8342345 | 0.8896386    |
| 20,ov     | 10,ov      | 0.8281792 | 0.8878673  | 0.8296215 | 0.8895448    | 14 20,ov     | 10,ov      | 0.8269803 | 0.8884189  | 0.8282004 | 0.8878220    |
| 17,ov     | 8,ov       | 0.8197818 | 0.8835735  | 0.8216259 | 0.8853108    | 10 17,ov     | 8,ov       | 0.8189318 | 0.8822229  | 0.8221097 | 0.8817047    |
| 14,ov     | 6,ov       | 0.8044867 | 0.8675197  | 0.8053564 | 0.8677893    | 6 14,ov      | 6,ov       | 0.8043917 | 0.8663586  | 0.8051592 | 0.8685784    |
| 11,ov     | 4,ov       | 0.7813220 | 0.8276896  | 0.7822565 | 0.8275550    | 2 11,ov      | 4,ov       | 0.7797639 | 0.8284181  | 0.7800373 | 0.8281996    |
| 26,weight | 14,weight  | 0.7581276 | 0.7791394  | 0.7581276 | 0.7791394    | 16 20,weight | 14,none    | 0.7470948 | 0.7813800  | 0.7506134 | 0.7783690    |
| 23,weight | 12,weight  | 0.7564186 | 0.7762601  | 0.7564186 | 0.7762601    | 20 23,weight | 12,none    | 0.7443696 | 0.7745083  | 0.7469420 | 0.7818687    |
| 20,weight | 4,none     | 0.7461844 | 0.7751844  | 0.7461051 | 0.7751844    | 30 8,ov      | 10,weight  | 0.7400153 | 0.7698312  | 0.7385293 | 0.7715068    |
| 17,weight | 4,weight   | 0.7449743 | 0.7678830  | 0.7449743 | 0.7678830    | 24 26,weight | 12,weight  | 0.7349884 | 0.7693672  | 0.7366080 | 0.7641834    |
| F1, 200   |            |           |            |           |              | F1, 400      |            |           |            |           |              |
| desc_rf   | desc_boost | mean_rf   | mean_boost | median_rf | median_boost | desc_rf      | desc_boost | mean_rf   | mean_boost | median_rf | median_boost |
| 26,ov     | 14,ov      | 0.8292721 | 0.8874687  | 0.8283776 | 0.8879509    | 22 26,ov     | 14,ov      | 0.8356880 | 0.8906313  | 0.8346339 | 0.8920557    |
| 23,ov     | 12,ov      | 0.8285145 | 0.8863214  | 0.8271391 | 0.8875694    | 18 23,ov     | 12,ov      | 0.8316119 | 0.8885397  | 0.8320070 | 0.8890450    |
| 20,ov     | 10,ov      | 0.8251302 | 0.8841001  | 0.8224557 | 0.8834620    | 14 20,ov     | 10,ov      | 0.8293529 | 0.8876245  | 0.8291422 | 0.8879726    |
| 17,ov     | 8,ov       | 0.8140147 | 0.8799001  | 0.8106550 | 0.8780649    | 10 17,ov     | 8,ov       | 0.8199832 | 0.8827367  | 0.8198017 | 0.8866334    |
| 14,ov     | 6,ov       | 0.7990606 | 0.8635923  | 0.7980346 | 0.8635799    | 6 14,ov      | 6,ov       | 0.8067399 | 0.8654620  | 0.8067373 | 0.8647802    |
| 11,ov     | 4,ov       | 0.7756855 | 0.8254323  | 0.7745509 | 0.8239899    | 20 23,weight | 4,ov       | 0.7948586 | 0.8276942  | 0.7966822 | 0.8269507    |
| 17,weight | 14,weight  | 0.7650098 | 0.7956685  | 0.7679292 | 0.7929337    | 24 26,weight | 10,weight  | 0.7923497 | 0.8205185  | 0.7949456 | 0.8278154    |
| 20,weight | 12,weight  | 0.7644439 | 0.7928766  | 0.7695075 | 0.7950718    | 16 20,weight | 14,weight  | 0.7889349 | 0.8204187  | 0.7899699 | 0.8219658    |
| 26,weight | 10,weight  | 0.7624611 | 0.7899569  | 0.7660222 | 0.7856888    | 2 11,ov      | 12,weight  | 0.7825500 | 0.8193608  | 0.7818696 | 0.8273449    |
| 23,weight | 10,none    | 0.7582445 | 0.7860289  | 0.7516359 | 0.7800028    | 12 17,weight | 8,weight   | 0.7795585 | 0.8177258  | 0.7789381 | 0.8202017    |
| F1, 1000  |            |           |            |           |              | F1, 2000     |            |           |            |           |              |
| desc_rf   | desc_boost | mean_rf   | mean_boost | median_rf | median_boost | desc_rf      | desc_boost | mean_rf   | mean_boost | median_rf | median_boost |
| 26,ov     | 14,ov      | 0.8315728 | 0.8896544  | 0.8294870 | 0.8885369    | 22 26,ov     | 14,ov      | 0.8339738 | 0.8873783  | 0.8361935 | 0.8867017    |
| 23,ov     | 10,ov      | 0.8289296 | 0.8875779  | 0.8254453 | 0.8864436    | 18 23,ov     | 12,ov      | 0.8316782 | 0.8867270  | 0.8328932 | 0.8885361    |
| 20,ov     | 12,ov      | 0.8247459 | 0.8873122  | 0.8250588 | 0.8863493    | 14 20,ov     | 10,ov      | 0.8277861 | 0.8860809  | 0.8276604 | 0.8878079    |
| 26,weight | 8,ov       | 0.8168547 | 0.8814310  | 0.8141874 | 0.8814401    | 10 17,ov     | 8,ov       | 0.8171858 | 0.8790247  | 0.8186455 | 0.8797313    |
| 17,ov     | 6,ov       | 0.8167333 | 0.8647179  | 0.8166844 | 0.8656864    | 21 26,none   | 6,ov       | 0.8144602 | 0.8648119  | 0.8133232 | 0.8658127    |
| 23,weight | 14,none    | 0.8146513 | 0.8517195  | 0.8153518 | 0.8528236    | 24 26,weight | 14,weight  | 0.8143273 | 0.8568175  | 0.8125662 | 0.8558556    |
| 20,weight | 12,weight  | 0.8121982 | 0.8512173  | 0.8128210 | 0.8460485    | 17 23,none   | 12,weight  | 0.8140257 | 0.8565179  | 0.8143750 | 0.8566422    |
| 17,weight | 14,weight  | 0.8065743 | 0.8498677  | 0.8074459 | 0.8498034    | 20 23,weight | 14,none    | 0.8122433 | 0.8562576  | 0.8138784 | 0.8547147    |
| 14,ov     | 12,none    | 0.7997259 | 0.8498171  | 0.7997492 | 0.8503045    | 13 20,none   | 10,weight  | 0.8113356 | 0.8562545  | 0.8101861 | 0.8555124    |
| 26,none   | 10,weight  | 0.7962184 | 0.8496077  | 0.7982523 | 0.8500660    | 23 26,un     | 12,none    | 0.8104442 | 0.8557034  | 0.8079322 | 0.8564160    |

Table 20: Table of results for model comparison with measurement F1