Bachelors thesis 15 HP
June 2023

Uppsala University

Predicting periodontitis

An in depth study aiming to make accurate predictions of periodontitis using regularized gradient boost - XGBoost

Jacob Hoffstedt
Erik Lillrank

Contents

1 Introduction

Chronic periodontitis is a disease that affects roughly 35 percent of the population [20, 21, 2]. Periodontal disease is an infection of the gums. The symptoms of the infection are swollen gums, loose teeth and, if left untreated, loss of teeth. It is caused by the patient not brushing and flossing their teeth properly which leads to build up of plaque which leads to infection. If the teeth are not cleaned properly the plaque will build up and harden into tartar which can only be removed by a clinician. Periodontal disease is treated by cleaning the teeth in an attempt to control the infection. The patient also has to change their habits as to not risk re-infection or worsening the infection.[6]

The subject of predicting periodontitis progression has been discussed by different researchers and several methods have been suggested. In a review of risk assessment tools Lang et al. [15] identified three tools described in the literature which have been used on longitudinal data, namely: Periodontal risk calculator (PRC) [25], periodontal risk assessment (PRA) [16], and dentition risk system (DRS) [18]. This thesis also uses longitudinal data making these earlier studies similar in that respect. The aim of these earlier risk assessment tools has been to create intuitive tools to be used by clinicians. These three risk assessment tools were constructed between 2002 and 2010, since then the field of statistics has grown and new methods and software has emerged. As new methods have emerged, models with higher accuracy can be made. The authors of this thesis are statisticians and not periodontal specialists, the aim is therefore to make accurate models more so than intuitive tools for clinicians. This approach does not seem to be the norm , making this thesis a valuable contribution to the existing literature.

The method chosen for this task is XGBoost which is a regularized gradient boosting machine. The reason for choosing this machine learning technique specifically is that it has proven to be an accurate model for making predictions both for numeric data and classification. This has been shown in machine learning competitions and challenges where some form of XGBoost has often been the technique used by the winning team [11]. There have been studies predicting periodontal disease using machine learning tools, the one being most similar to ours being Patel et al. [27] who in 2022 built an XGBoost model for predicting periodontitis. In their study they used a large data set consisting of ca: 27 000 patients and they built their model using 74 explanatory variables. Their model achieved an area under the ROC curve of 0.72 which should be interpreted as moderate to high accuracy.[27]. Their results are promising and give us an indication that predictive machine learning models of periodontal disease has potential.

Compared to Patel et al. [27] the models in this thesis are based on fewer observations

and explanatory variables and we are interested to see if we are able to achieve similar accuracy. One of the goals of this thesis is to build a model which is replicable on new data, for this goal to be met it is favourable to construct a model consisting of fewer variables that are accessible to a large amount of researchers. Furthermore Patel et al. labeled their study as a pilot study within the field, indicating that further research is necessary to construct useful models. The research question that we want to answer is; is it possible to make accurate risk predictions of chronic periodontitis using the supervised machine learning technique regularized gradient boosting machine using the XGBoost software?

In the thesis four models with varying amounts and types of explanatory variables are constructed using XGBoost with the purpose of predicting periodontitis progression.

In the next section the data is introduced. In Section 3, tree based learning leading up to the theories and tuning behind XGBoost is explained. The descriptive statistics are presented and the data washing is discussed in Section 4. Followed by the results in Section 5 and the discussion in Section 6. The conclusions are in Section 7.

2 Data

In this section, the data sets and variables are described. The description of the data includes number of observations and variables. Further description of data and data washing is in Section 4.

The data is given by an external client. The data was collected from 213 patients by periodontal specialists and dentists in five clinics in the Stockholm area. The patients that were included in the study were between the ages 30-65 with varying status of periodontitis. The data is longitudinal, it was collected with a first appointment performed during December in 1998 and March of 1999, and the follow up appointments were performed in 2002. At the follow up appointments 30 of the original 213 patients were not available for examination, leaving us with 183 patients.

The data is received as two separate data sets: One data set, *patient* (See Table 1) which has information about the patients, and another data set *tooth* (See Table 2) consisting of information about each of the teeth belonging to patients. As XGBoost can not take hierarchy into account, a downside of using the variables that relate to each tooth of patients is correlation between teeth (observations) due to teeth belonging to the same patients. An alternative solution is aggregating the teeth data for each patient, however this would lead to few observations, 183 which is low for machine learning. The correlation between teeth, which the models will not be able to control for, is therefore accepted even though it can lead to worse performance.

We join *patient* and *tooth* to one data set containing 3408 observations and 30 explanatory variables. The response variable is *progression_yes_no* which is 1 if there is periodontitis progression and 0 if there is not. Progression of periodontal disease was measured between the two appointments, between 1998-1999 and 2002. If the patient there has had progression of periodontitis between these two appointments the outcome variable has the value 1.

Table 1: Variables in patient-data set

Variable	Description
Age	Age (years)
Mother suffers from periodontitis	1 if mother suffers from periodontitis, 0 if not, 2 if patient doesn't know.
Father suffers from periodontitis	1 if father suffers from periodontitis, 0 if not, 2 if patient doesn't know.
Pregnancy	1 if pregnancy, 0 if not.
Diabetes mellitus, not controlled	1 if Diabetus mellitus, not controlled, 0 otherwise.
Diabetes mellitus, well controlled	1 if Diabetes mellitus, well controlled, 0 otherwise.
Blood or immune disease	1 if blood or immune diseases, 0 otherwise.
Monogen genetic disturbances	1 if monogen genetic disturbances (e.g. Down Syndrome), 0 otherwise.
Granulomatous diseases	1 if Granulomatous diseases, 0 otherwise.
Drug induced periodontal disease	1 if Drug induced parodontal, 0 otherwise.
Osteoporosis	1 if Osteoperosis, 0 otherwise.
Malnutrition	1 if Malnutrition, 0 otherwise.
Sjögrens syndrom	1 if Sjögrens syndom, 0 otherwise.
Result of the Skin Prick Test	number of negative reactions (0-3).
Socio-economic factors	0 if none, 1 if negative stress, 2 if poor economy, 3 if both.
Current smoking	0 if no smoking, 1 if $<$ 10 cigarettes per day, 2 if 10-20 cigarettes per day, 3 if $>$ 20 cigarettes per day.
Previous smoking	0 if never smoked or stopped $> 2 year sago$, 1 if smoked $>$ 15/day and stopped $<$ 2 years ago, 2 if smoker (see above).
Operator experience of advanced periodontal treatment	1 if little, 2 if some (general clinics), 3 if extensive (special clinics).
Awareness and interest of disease after information	0 if none, 1 if little, 2 if high.

There are some terms in tables 1 and 2 which needs explanation: *Monogen genetic disturbances* are diseases caused by mutations in specific genes [30]. *Granulomatous*

Table 2: Variables in tooth-data set

Variable	Description
Probing plaque	1 if plaque on tooth, 0 otherwise
Endodontic factors	0 if endodontically intact or root filled and periapically intact, 1 if periapical destruction or root filled but periapical destruction
Furcation involvment	0 if none of the surfaces has furcation, 1 if diagnostable $< 2mm$ on any of the two surfaces but none $> 2mm$, 2 if $> 2mm$ on any of the two surfaces.
Vertical destruction	1 if vertical destruction on any of the two surfaces, 0 otherwise
Initial bone level	mm, average of the two surfaces.
Pocket depth	mm, average of the two surfaces.
Bleeding on probing	1 if bleeding on any of two surfaces, 0 otherwise
Restored surface	0 if no restoration, 1 if restoration in crown only, 2 if restoration is extending into the root on any of the two surfaces, 3 if overhanging restoration on any of the two surfaces.
Mobility	1 if mobility, 0 otherwise.
Abutment	1 if abutement, 0 otherwise .

diseases are a disease which makes the white blood cells unable to defend the body from bacteria and fungi which can lead to infections [22]. *Osteoporosis* is a condition which makes the bones weak and brittle [23]. *Sjögrens syndrom* is a disease which makes the mouth and eyes of the patient dry [1]. *Result of the Skin Prick Test* is a test conducted by the external client we are working with which tests the reaction to lipid A, this is hypothesised to be related to periodontal disease. *Endodontic factors* are factors which are related to the root of the tooth, if it is intact or not. *Furcation involvement* in this context is how much of the root of the tooth is showing due to breakdown of the gums, furcation is the area where the tooth splits in to two or three roots [28]. *Vertical destruction* is bone destruction in a vertical or angular direction. *Initial bone level* is measured by probing the gums by the teeth and measuring how far down the probe can be inserted, if it goes down deep this is an indication of destruction of the tooth [24]. *Pocket depth* measures the depth of the pocket between the gum and the tooth [36]. *Bleeding on probing* refers to probing the gums. *Restored surface* is a categorical variable of whether the teeth has been restored by a dentist in some way. *Mobility* refers to if the tooth is mobile, if it moves when touched. *Abutment* is a type of tooth replacement [8].

3 Method

In the first subsection the choice of method and alternative methods are discussed. In the next five subsections decision trees, boosting, gradient descent, gradient boosting, and bias-variance are presented concluded by an explanation of XGBoost theory. The theoretical explanations are followed by the construction of the models and hyperparameter tuning.

3.1 Choice of method and alternative methods

When deciding which method to use, mainly popular machine learning methods are considered. The goal for us is to make as accurate predictions as possible and to achieve this we consider what is used in competitions to get a sense of which type of models are most popular. Many of the winning solutions in Kaggle competitions are either neural networks, gradient boosting machines or a combination of the two [7]. However, these models win when utilizing large data sets, the data set in this thesis is relatively small. None the less, these winners do give an indication of which methods are able to produce accurate results. While both models would work for classification, our understanding by reviewing many competitions and papers is that gradient boosting machines are more popular, which is why we are interested in the method. Also, gradient boosting methods have been shown to produce more accurate results when using a smaller data set [39]. The most popular boosting methods are XGBoost, LightGBM [4] and CatBoost [19]. The three methods yield similar results. The advantage of CatBoost and LightGBM is that they perform faster than XGBoost.[5] The disadvantage of CatBoost and LightGBM is that both methods are relatively new as compared to XGBoost. The documentation and resources available for the methods are therefore limited. XGBoost on the contrary has been widely used for a longer time, as a consequence, there are more resources available. A deeper theoretical understanding of XGBoost can therefore be achieved. Also, the available resources for hyperparameter tuning in XGBoost are many. It has therefore been reasoned by the authors that using XGBoost can produce a thesis that includes deep subject knowledge and also accurate models as a result of robust hyperparameter tuning.

Besides neural networks and gradient boosting machines, there are other methods that can be used. Examples of other methods are logistic regression, other decision tree methods such as random forest, or support vector machines. We choose not to use any of these because we are interested in using a newer framework as the field of statistics and machine learning has progressed during the last twenty years. It would be interesting to do a comparative study of different methods when predicting periodontitis but we believe this to be outside the scope of this thesis.

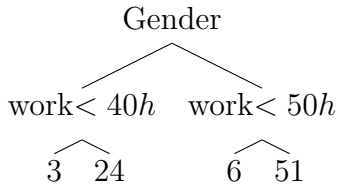

Figure 1: Fictional regression tree. Predicting the number of pizzas a person eats each year based on gender and how many hours they work in a week. Gender is a dummy variable equal to 1 if the observation is female and 0 if the observation is male, and work is a continous variable.

3.2 Decision trees

Decision trees can be used for predicting both qualitative and quantitative response variables, classification trees and regression trees. In this thesis we will focus on regression trees. This might seem counter intuitive since our response variable is binary, but regression trees is used in *Gradient Boosting* which will become important in this thesis.

Decision trees are normally drawn upside down with the root at the top and the leafs at the bottom. From the root containing the whole data set it is split into internal nodes connected with branches and ending in the last splits, the leafs, an example of this can be seen in Figure 1. The data is split based values on certain variables, for example a split may be gender = man or woman, splitting the data in to women and men. In figure 1 we see a simple example of how a regression tree can look. Using this tree, a woman who works 40 hours or more, we would predict eating 24 pizzas a year.

Regression trees are built in two steps, the first step is to divide the predictor space. The second step is to predict the value of the observations within the same region, simply by predicting the mean of the observations within that region. Dividing the predictor space is done by dividing the values of our predictors $X_1, X_2, ..., X_J$ - into J distinct and non-overlapping regions $R_1, R_2, ..., R_J$. We construct the regions with the goal of minimizing:

$$RSS = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y_{R_j}})^2.$$

Meaning that we minimize the the difference between the observed value,y_i , and the mean of the leafs at each possible split, $\hat{y_{R_j}}$. This is done by using a top down, greedy approach. That is, at each split making the best split moving our way down the tree. It is greedy in the sense that we do not take into consideration the whole tree when making our steps, we simply make the best split at that point, then move on to the next split to do the same. The splits are made by choosing our predictor X_j and

splitting the observations based on a cutoff-point s: $(X|X_j < s)$ and $(X|X_j \geq s)$. The cutoff-point and the predictor is chosen so that it minimizes the RSS. This process is repeated until a stopping criterion is met, such as max amount of levels our minimum amount of observations in each leaf. ([13] pp.327-340)

3.3 Boosting

Boosting is a method for improving the predictive results of a decision tree. In boosting we grow multiple trees sequentially using information on the previously grown trees. The trees are then combined into one predictive model. The idea is to create a first decision tree and then fitting a new tree to the residuals of the first tree, then we fit another tree to the residuals of that tree and so on. This creates a slow learning model which prevents overfitting and slowly improves the predictive ability of the combined model. Generally slow learning models outperform fast learning models. The basic idea of boosting is that several weak learners can be combined into one strong learner. ([13] pp.345-348)

3.4 Gradient descent

An example that is often used when explaining gradient descent is the scenario that you are stuck on the top of a mountain without being able to see. To get down, the technique of gradient descent, would suggest swinging your foot around looking for the steepest direction, take a small step towards the steepest direction, and repeat until the ground is flat. Then you will have arrived at the bottom of the mountain. This section is inspired by an article written by Terrence Parr and Jeremy Howard [26].

Mathematically we calculate the next position of $\boldsymbol{x}$ at the step t, the update function is given by:

$$\boldsymbol{x_t} = \boldsymbol{x_{t-1}} + (- \bigtriangledown f(\boldsymbol{x_{t-1}}))$$

$\boldsymbol{x}$ is a vector of all parameters of our function $f(\boldsymbol{x})$ and $\bigtriangledown f(\boldsymbol{x})$ is a vector of the partial derivatives of our function with respect to our parameters $\boldsymbol{x}$, also called the *gradient*. As to not take to large steps, in practice we also include the regularization parameter η: The learning rate, which has a value between 0-1.

$$\boldsymbol{x_t} = \boldsymbol{x_{t-1}} - \eta \bigtriangledown f(\boldsymbol{x_{t-1}}).$$

The reason for discussing gradient descent is that gradient boosting uses gradient descent to add weak learners to minimize the *loss function*. The loss function is a

$F_0(\mathbf{x}) = \frac{1}{2}\log\frac{1+\bar{y}}{1-\bar{y}}$
For $m = 1$ to M do:
$\tilde{y}_i = 2y_i / (1 + \exp(2y_i F_{m-1}(\mathbf{x}_i))),\ i = 1, N$ (A)
$\{R_{jm}\}_1^J = J$–terminal node $tree(\{\tilde{y}_i, \mathbf{x}_i\}_1^N)$ (B)
$\gamma_{jm} = \sum_{\mathbf{x}_i \in R_{jm}} \tilde{y}_i / \sum_{\mathbf{x}_i \in R_{jm}} |\tilde{y}_i|\,(2 - |\tilde{y}_i|),\ j = 1, J$ (C)
$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \sum_{j=1}^{J} \gamma_{jm} 1(\mathbf{x} \in R_{jm})$ (D)
endFor
end Algorithm

Figure 2: Algorithm proposed by Friedman for binary classification using gradient boost [9].

function which in machine learning compare the predictions made by the model to the observed value and measures how well the model performs. In classification tasks a common loss function is cross entropy loss, also known as log loss. In gradient boosting we nudge our predicted value $\hat{y}$ closer and closer to the real value y, hence the residual $y - \hat{y}$ is actually a direction vector.

3.5 Gradient boosting

In *gradient boosting* we want to minimize the *loss function* by adding weak learners using gradient descent. Gradient boosting machines (GBM) where introduced in 2001 by Jerome H. Friedman [9]. Gradient boosting machines fit a regression tree to the current pseudo residuals each iteration. Here the pseudo residuals are the gradient of the minimized loss function [10]. Friedman suggests the negative binomial log-likelihood as the loss function for binary classification, we chose to use the binary cross entropy loss (also known as log loss) instead. The reason for this is that this is the loss function that we will use when running our XGBoost, so this way the paper will be more coherent. Since we use a different loss function it looks slightly different, the calculations can be found in Appendix 1. Our loss function with y_i=observed value and p_i=predicted probability can be written as:

$$\sum_i l(p_i, y_i) = -\sum_{i=1}^{n} (y_i * log(p_i) + (1 - y_i) * log(1 - p_i)).$$

By looking at only one observation we simplify the loss function as $y_i * log(odds) + log(1 + e^{log(odds)})$. The loss function is differentiable with respect to log(odds) which is equal to $-y_i + p_i$. Now that we have our loss function we can start with the algorithm presented in Figure 2. As our loss function is different the steps will also look slightly

different

The first step: Initialize the model with a constant value: $F_0(x) = \underset{\gamma}{\text{argmin}} \sum_{i=1}^{n} L(y_i, \gamma)$. With $\gamma = log(odds)$ and what we will do is find a log(odds) that minimizes the loss function so we will derive the loss function with log(odds) and set it equal to zero, then solve for p. $p = log(1 + e^{log(odds)})$.

$$F_0(X) = \underset{\gamma}{\text{argmin}} \sum_{i=1}^{n} (-Observed * log(odds) + log(1 + e^{log(odds)}) = 0.$$

Simplifying this we get $p = \frac{\sum_{i=1}^{n} y_i}{n}$ and $log(odds) = \frac{p}{1-p}$.

The second step: For m=1 to M there are four steps which we will denote A-D. In short the steps are: (A) Calculate the pseudo residuals of the initial model, (B) fit a regression tree to the pseudo residuals, (C) calculate the output of the new regression tree, and (D) update the model for M iterations.

(A) From the initial model we calculate the negative gradient which is the same as we did in gradient descent. We will denote the negative gradient as *pseudo residuals* in this section. Compute $r_{im} = -[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}]_{F(x)=F_{m-1}(x)}$ This is the derivative of the Loss function with respect to the log(odds) which we have already calculated. With i=sample number and m=tree that we are building. Hence we can calculate the pseudo residuals by:

$$\text{Pseudo residual}=r_{im} = y_i - \frac{e^{log(odds)}}{1+e^{log(odds)}}.$$

(B) Fit a regression tree to the r_{im} values and create terminal regions R_{jm}, $j = 1, ..., J_m$. In simpler terms, we build a decision tree of the pseudo residuals that we calculated in step (A).

(C) For $j = 1, ..., J_m$ compute $r_{jm} = \underset{\gamma}{\text{argmin}} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma) =$

$$= \underset{\gamma}{\text{argmin}} \sum_{x_i \in R_{ij}} -y_i * [F_{m-1} + \gamma] + log(1 + e^{F_{m-i}(x_i)+\gamma})).$$

Since this will be difficult to derive we can instead approximate the function with the second order Taylor polynomial. The calculations for this can be found in the Appendix 1, and the results from the calculations is:

$$\gamma_{jm} = \frac{\sum_{x_i \in R_{ij}} r_{im}}{\sum_{x_i \in R_{ij}} p_i(1 - p_i)}.$$

In step (C) we calculate the output values (γ_{jm}) for each leaf in the new tree.

(D) Update $F_m(x) = F_{m-1} + \eta \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$, here η =learning rate and M=total amount of trees. The output of the current tree is then the output of the last tree (F_{m-1}) plus the sum of the outputs from the tree fitted to the psuedo residuals (γ_{jm}) times the learning rate (η). The output of our final model is $F_M(x_i)$ for an observation x_i. This model gives us the log(odds) which we can use to calculate predicted probabilities with $\frac{e^{log(odds)}}{1+e^{log(odds)}}$.

In short the gradient boosting machine works by first fitting an initial regression tree to the training data, then we fit a new regression tree to the pseudo residuals r_{im} and create terminal regions R_{jm}. We calculate γ_{jm} which is the optimal constant update in each terminal node region. This then leads us to our model which is the results from the last function plus the sum of γ_{jm} in the terminal regions of each node times the learning rate. [9] [10]

3.6 Bias-Variance

In machine learning, the trade-off between bias and variance is an important concept. The bias refers to the difference between the model prediction and the actual value, in other words the error. The variance refers to the difference in accuracy when using other data. If one would attempt approximating a complicated problem with a simple machine learning model the bias would likely be large. On the other hand the variance would probably be low since since the prediction would not change much when using the model with other data, the error would likely still be large. As the complexity of a machine learning model increases, the bias will generally decrease faster than the variance is increasing , assuming that the model is approximating the real world problem better and better. However, at one point as the model becomes increasingly complex, the variance will start increasing faster than the decrease in bias. A model that is too complex can therefore have a low bias but a high variance which leads to overfitting. The training data of an overfitted model can therefore have low errors and high predictive power but as the model predicts out of sample data, the errors are large since the variance is high. ([13] pp. 205-206). The external validity of the model, i.e the validity of using the model on other data and settings can therefore not be ensured. In order to prevent overfitting and to attempt creating a model that has low bias and low variance, XGBoost uses several regularization techniques as described in Subsection 3.7.1.

3.7 XGBoost

The aim of this section is to explain how the XGBoost algorithm works and why it in most cases gives accurate predictions. In essence XGBoost is a framework for regularized gradient boosting with optimizations meant to save on computational power. Since it is a form of GBM it works very similarly to the GBM introduced by Friedman [9]. It also builds regression trees to the pseudo residuals of the previous tree, making an ensemble model made up of the weak learners, the regression trees. Comparing it to the gradient boosting machine, the two main advantages of XGBoost compared to GBM is that it introduces more regularization to prevent overfitting, and that it is optimized to use a small amount of computational power. As the computational power needed is not relevant to the accuracy of the model, it will not be the focus of this thesis. Instead we start by comparing the level of regularization between GBM and XGBoost, which is the main reason for choosing XGBoost for this thesis. [3]

3.7.1 Regularization

The goal of regularization is to minimize overfitting the model to the data as to minimize the out of sample error of the model. XGBoost uses different forms of regularization in order to achieve this. In GBM we minimize the loss function, for XGBoost we instead minimize this objective function which contains both the loss function and regularization terms [3].

$$\mathcal{L}(\phi) = \sum_i l(p_i, y_i) + \sum_k \Omega(f_k)$$

where

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \parallel \omega \parallel^2 .$$

The first row is the objective function where the left expression represents a loss function which measures the difference between the observed y_i and the predicted probability p_i. The second term $\sum_k \Omega(f_k)$ consist of the regularization of the model and *weighted leaf*: ω of the explanatory variable k. As can be seen from the formula, the objective function contains the regularization terms *gamma* and *lambda*.

Gamma γ encourages pruning of the regression trees. Pruning means cutting off leaves and thereby ignoring splits that do not add enough gain at each split. Gain is the measure of how much better the optimal values of the leaf is due to splitting. If the added gain score of a potential split does not exceed the value of the γ parameter, the leaves of the potential split is pruned and ignored. The pruning mechanism can be seen mathematically in the Gain formula used for evaluating splits in the Appendix 2.

Lambda, λ, is a regularization term which penalizes complexity of the model. Lambda is used in conjunction with ω^2 which is the square of the weight of the explanatory variables that was used to make the trees, a complex tree uses a larger amount of explanatory variables which is penalized by lambda, also known as ridge regularization. [3] However, the caret package in R is used in this thesis, in the caret package lambda is included in the linear XGBoost method, but not in the XGBtree method. [37] Since this thesis uses the XGBtree method, lambda is not included as a hyperparameter used for tuning the models. Other hyperparameters are used in preventing overfitting, the full list of hyperparameters used are included in Subsection 3.9, and are explained in detail below.

XGBoost includes the *learning rate*, η , or shrinkage as a regularization term. It shrinks the size of the steps of the boosting algorithm, slowing down the rate at which the model learns. Generally slower models perform better then fast learners as faster learners tend to sometimes take large steps in the wrong direction, while slower models take many small steps instead to prevent this. A mathematical example of η can be seen in Appendix 3.

XGBoost also has regularization techniques regarding the building of the trees. *Max depth* is the maximum number of levels the regression trees are allowed to have in our model. Deeper trees can lead to overfitting. *Nrounds* is the number of trees constructed, this is related to the size of eta as a lower learning rate requires a larger amount of iterations. *Subsample* and *colsample by tree* is another method of preventing overfitting by introducing randomness into the model. When XGBoost subsamples it randomly chooses a proportion of rows of data to train in the iteration, while subsample by tree means it chooses a proportion of columns (variables) to train. *Min child weight* is the minimum sum of weights in each leaf. The weight is calculated as the *cover* in the Appendix 4. Intuitively, if the previous probabilities for observations having periodontitis progression in a leaf are close to 1 and 0 the sum of the weights will be lower. If they are 0.5, the sum of the weights will be higher. Therefore, XGBoost at each iteration prefers including observations that have not yet been learned. Cover also increases as the amount of observations increase. In conclusion, setting min child weight will decide to which degree XGBoost will learn from weak learners and how many observations to be included in the leaf. [17].

Discussion of how we choose regularization parameters and their values are in Subsection 3.9 *Choosing hyperparameters*. The full list of parameters for the XGBoost can be found in the documentation of the XGBoost package [38] which is also the basis for this subsection in conjunction with the original manuscript of XGBoost [3].

3.7.2 XGBoost tree building

The purpose of XGBoost tree building is finding splits that result in leaves that are as homogenous as possible with respects to the variable that is predicted. In other words, when XGBoost is creating a tree with the purpose of classifying periodontitis progression which is binary, it is finding explanatory variable splits that results in as many periodontitis progression 1:s as possible in one leaf, and 0:s in the other. Using the exact greedy algorithm explained in Subsection 3.7.4, XGBoost searches for the best possible split across all explanatory variables. XGBoost evaluates the splits by calculating a score for every possible split and the highest scoring potential split is chosen. [3] [34] An example of how XGBoost builds trees is included in the Appendix 2.

3.7.3 XGBoost Predictions

When XGBoost has created the tree using the split that received the highest score it can make new predictions. Relating to periodontitis progression, XGBoost will predict the probability of periodontitis progression for observations that are allocated to the leaves. All observations in the same end leaf receives the same probability of periodontitis. When predicting periodontitis for observations in each leaf, scores for each of the end leaves in the tree are calculated. The scores for each leaf are summed with the probability of periodontitis in the previous iteration (which is set to 0.5 in the first iteration) times a learning rate η. The purpose of making new predictions is to for each iteration making increasingly more accurate predictions until the learning stops. The final XGBoost model is then used to predict out of sample observations to evaluate the performance. [3][9][34] An example of how XGBoost predicts is included in the Appendix 3.

3.7.4 Exact greedy algorithm

The XGBoost framework offers two methods of tree building, the appropriate method is chosen based on the size of the data set and the constellation of explanatory variables. The methods are the exact greedy algorithm and the approximate greedy approach. The exact greedy algorithm searches all possible splits, the split is based on the max score as seen in the Appendix 5. The exact greedy method is computationally expensive, especially when working with a large amount of continuous explanatory variables as it has to compute each possible split. When the exact greedy method is too computationally expensive, the software instead uses the approximate greedy method, which instead of searching all possible splits, divides the explanatory variables into quantiles containing the same amount of observations. Instead of searching for splits all along the continuous variables it now only has to calculate the scores of the amount of quantiles. This method can significantly decrease the run

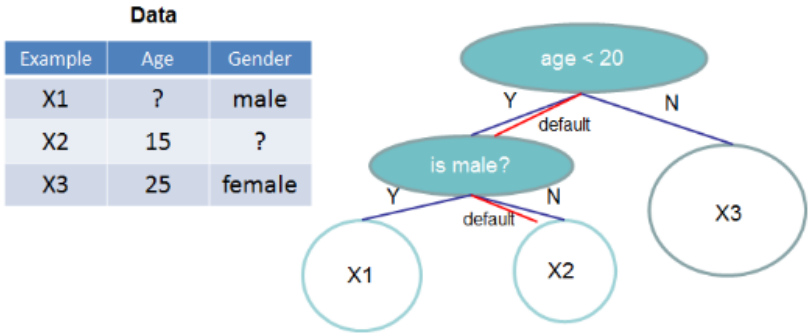

Figure 3: XGBoost missing values [3].

time of complex models. [3]

As our data consists almost exclusively of categorical explanatory variables there is no need to use the approximate greedy algorithm as there is likely no difference in computational power.

3.7.5 Missing values

Another advantage of using XGBoost is it's handling of sparse or missing data. XGBoost handles sparse data by using an algorithm that distributes the missing data to a default leaf at each node.

XGBoost decides the default leaf by setting the missing values to zero. Based on the initially set probability, XGBoost calculates the residuals. The gain of adding the residuals to the left leaf is calculated. The same is done for the right leaf. The gain is a tree building score explained in depth in Appendix 2 and 3. The leaf with the highest gain is chosen as the default option. When predicting, XGBoost will allocate missing value observations to the leaf that was chosen. In figure 3, the gender value for the X2 observation is missing, XGBoost calculates the scores of allocating X2 to the right leaf and left leaf and chooses the highest score as the default direction, "is male?" no (right leaf) was chosen as the default leaf. [3] The algorithm for sparsity aware split finding can be seen in Appendix 6.

The algorithm allows XGBoost to handle sparse data well. This is beneficial as it makes it possible to include observations with missing values, leading to a larger sample to learn from. However missing values still has to be considered when building our model. A large amount of missing values means that the model has less data to

learn from, leading to a weaker model.

3.8 Models

In this thesis we have been asked by the external client to create three separate models. One which includes all variables in the patient and tooth data sets, denoted as the *large model*, and two models including variables specified by the client, denoted as the *medium model* and *small model*. The large, medium and small models are specified in Tables 3, 4 and 5. The models here are presented as described by the client. In Section 4 we show the descriptive statistics and wash the data. After washing the data some variables are left out of the final models due to all of the observations falling in to the same category, making them not useful. The variables that are left out in are marked as "Excluded" in tables 3, 4 and 5.

We will also create a fourth model which we suggest instead of the large or medium sized models. This will be a smaller model made up of the most important explanatory variables of the large model. The fourth model will be built based on the Variable importance score table from the large model. The reason for this is that both the medium and large models include a large amount of explanatory variables. While a large amount of explanatory variables can be beneficial it can also lead to a weaker model, either because the explanatory variables have a negligible effect on the response variable in the population or if the sample is too small to see the relationship between the explanatory variable and response variable. In those cases it can be beneficial to exclude explanatory variables that are less important to reduce noise, doing this can lead to a stronger model.

3.9 Choosing hyperparameters

One of the appealing parts of XGBoost is the many hyperparameters available which can be used to regularize the model. Choosing the right hyperparameters and their values is a vital part of building an accurate model. The parameters used in the models are shown in table 6. η is the learning rate. γ encourages pruning of the regression trees. Max depth decides the depth of the tree. Min child weight is the minimum sum of weights in each leaf. Nrounds decides the amount of trees to be constructed (iterations). Subsample and subsample by tree decides which proportion of the rows

Table 3: Variables in large model

Variable	Name in data set	Type	Excluded
Age	age	Continous	
Mother suffers from periodontitis	DTMOTHER	Categorical	
Father suffers from periodontitis,	DTFATHER	Categorical	
Diabetes mellitus, not controlled	diab.notcontr	Categorical	X
Diabetes mellitus, well controlled,	diab.contr	Categorical	
Blood or immune disease	blood	Categorical	X
Monogen genetic disturbances	monogen	Categorical	X
Granulomatous diseases	granulom	Categorical	X
Drug induced parodontal disease	drug	Categorical	
Osteoporosis	osteop	Categorical	X
Malnutrition	malnutr	Categorical	X
Sjögrens syndrom	sjogren	Categorical	X
Result of the Skin Prick Test	antal3	Discrete	
Current smoking	skhab	Categorical	
Previous smoking	skhis	Categorical	
Operator experience of advanced periodontal treatment	opexp	Categorical	
Awareness and interest of disease after information:	dtintdis	Categorical	
Probing plaque	plaque	Categorical	
Endodontic factors	dtedont	Categorical	
Initial bone level	bonelevel	Categorical	
Pocket depth	depth	Categorical	
Bleeding on probing	bleed	Categorical	
Restored surface	restorat	Categorical	
Mobility	mobility	Categorical	
Abutment	brotand	Categorical	
Socio-economic status	socec	Categorical	
Vertical destruction	destr	Categorical	
Furcation involvement	furcat	Categorical	

and columns respectively XGBoost randomly samples from. The hyperparameters are explained in depth in Section 3.7.1.

The hyperparameters used in the models are chosen by looking at what seems to be the consensus among statisticians and data scientists and the ones chosen are the ones

Table 4: Variables in medium model

Variable	Name in data set	Type	Excluded
Age	age	Continous	
Mother suffers from periodontitis	DTMOTHER	Categorical	
Father suffers from periodontitis,	DTFATHER	Categorical	
Diabetes mellitus, not controlled	diab.notcontr	Categorical	X
Diabetes mellitus, well controlled,	diab.contr	Categorical	
Blood or immune disease	blood	Categorical	X
Monogen genetic disturbances	monogen	Categorical	X
Granulomatous diseases	granulom	Categorical	X
Drug induced parodontal disease	drug	Categorical	
Osteoporosis	osteop	Categorical	X
Malnutrition	malnutr	Categorical	X
Sjögrens syndrom	sjogren	Categorical	X
Result of the Skin Prick Test	antal3	Discrete	
Current smoking	skhab	Categorical	
Previous smoking	skhis	Categorical	
Operator experience of advanced periodontal treatment	opexp	Categorical	
Awareness and interest of disease after information:	dtintdis	Categorical	
Probing plaque	plaque	Categorical	
Endodontic factors	dtedont	Categorical	
Initial bone level	bonelevel	Categorical	
Pocket depth	depth	Categorical	
Bleeding on probing	bleed	Categorical	
Restored surface	restorat	Categorical	
Mobility	mobility	Categorical	
Abutment	brotand	Categorical	
Socio-economic status	socec	Categorical	

most commonly used. *Grid search cross validation* is used to find the hyperparameter values of our XGBoost models. For the full list of parameters we suggest the XGBoost documentation of parameters for further reading [38].

Grid search cross validation is an exhaustive method of choosing hyperparameter values. Before training our model a grid is specified, containing different values for all our hyperparameters which we wish to test. For example, we might want to test if the best value for eta is either 0.05, 0.1 or 0.15 and gamma 0.05, 0.2 or 0.4, we do

Table 5: Variables in small model

Variable	Name in data set	Type	Excluded
Age	age	Continous	
Mother suffers from periodontitis	DTMOTHER	Categorical	
Father suffers from periodontitis,	DTFATHER	Categorical	
Diabetes mellitus, not controlled	diab.notcontr	Categorical	X
Diabetes mellitus, well controlled,	diab.contr	Categorical	
Blood or immune disease	blood	Categorical	X
Monogen genetic disturbances	monogen	Categorical	X
Granulomatous diseases	granulom	Categorical	X
Drug induced parodontal disease	drug	Categorical	
Osteoporosis	osteop	Categorical	X
Malnutrition	malnutr	Categorical	X
Sjögrens syndrom	sjogren	Categorical	X
Bleeding on probing	bleed	Categorical	
Mobility	mobility	Categorical	
Current smoking	skhab	Categorical	
Previous smoking	skhis	Categorical	
Operator experience of advanced periodontal treatment	opexp	Categorical	
Awareness and interest of disease after information:	dtintdis	Categorical	
Socio-economic status	socec	Categorical	

this for all of the parameters we want to test different values for. When training our model the grid tests different combinations of hyperparameter values in order to find the most accurate model. It tests this by cross validation which means that it will partition the sample in to k folds, for example 5 folds splits the data into 5 parts with equal amounts of data in each part. It then trains the model on k-1 folds, testing on the omitted fold, repeating on so that all folds have been omitted once. This process can then be repeated multiple times for higher accuracy. Lastly, the output shows hyperparameters that resulted in the highest accuracy achieved when training. ([13] pp.203-205)

Theoretically this method could be used with endless grids of hyperparameters, folds and repeats of the crossvalidation. The problem as this is an exhaustive method is that it requires a large amount of computational power. As we do not have endless amount of time we instead use grids with three values for each parameter, starting with large differences for the values to get a sense of direction of what the optimal

Table 6: Hyperparameters of XGBoost

Hyperparameter	Range
η	$0-1$
γ	$0-\infty$
max depth	$0-\infty$
min child weight	$0-\infty$
nrounds	$0-\infty$
subsample	$0-1$
colsample bytree	$0-1$

values are, then use values closer and closer based on the output value of the last grid. This way we can move closer to the optimal values of our hyperparameters sequentially and use our computational power more effectively.([13] pp.203-205)

3.10 Evaluation metrics

In this section the metrics used when evaluating the models are discussed. The main five metrics used are accuracy, precision, recall, F1 score and area under the ROC-curve (AUC). Accuracy is the simplest of these metrics as it is the percent of correct predictions. When building our models we partition our data into a training set and test set. One for which we use to train the model and the other is used to test the model. The accuracy is the amount of correct predictions divided by the number of observations in the test set. For example, if the accuracy is 0.7, the model makes correct predictions 70% of the time.

Precision, recall and F1 score are related metrics. Precision, also known as positive predictive value (PPV), is the ratio of true positives to the predicted positives (see Table 7) [29]. Recall, also known as sensitivity, is the ratio of true positives to the number of actual positives in the data (see Table 7) [29]. In other words, recall is a measure of how well the model identifies positives in the data and precision is a measure of well the model differentiates between positives and negatives. F1 score combines these two measures [31]:

$$F1 = 2\frac{Precision * Recall}{Precision + Recall}.$$

A high F1 score means that the model has high precision and recall.

The last metric used when evaluating the models is Area Under the Curve (AUC).

Table 7: Confusion matrix

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

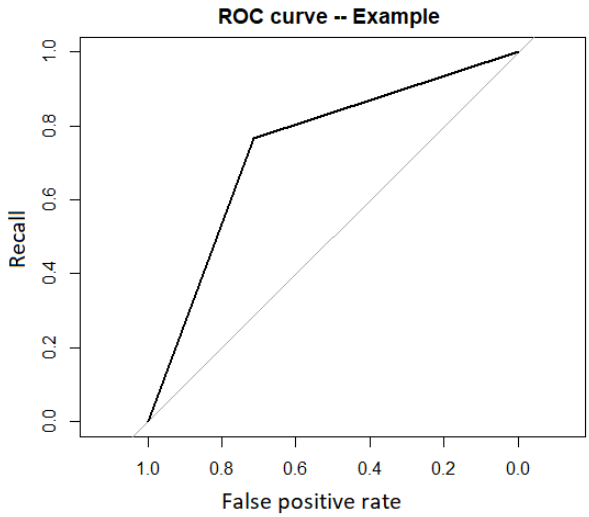

Figure 4: Example of ROC curve

The curve in question is the ROC curve (receiving operating characteristic curve) which plots the true positive rate and false positive rate of the model. In Figure 4 we show an example of what a ROC curve for an XGBoost classifier can look like. [12] On the Y-axis it measures Recall which is the rate of true positives and on the X-axis it measures the rate of false positives. The rate of the false positives are also known as the 1-specificity, the specificity is the rate of negatives that were predicted as negative (True Negative). It is expected that as the sensitivity increases/decreases, the specificity decreases/increases. [32] The AUC is the area below the plotted line in the ROC curve. As such the AUC is another measure of accuracy, if it is equal to 1 it predicts correctly 100% of the time and if it is 0 it predicts correctly 0% of the time. This is different from hit rate as it measures how well the model differentiate between positive and negative values, punishing false predictions. A model is better than chance if $AUC > 0.5$, indicated by the dotted line in the ROC curve. For a model with an AUC score of 1, the ROC curve would be 90 degrees and in the upper left corner, the lines being leveled at 1 for both recall and false positive rate. [12]

3.11 Software

The software that is used to build the models is R with RStudio. The package that was used is the Caret package. The reason for choosing this package rather than the XGBoost package is that Caret has the option of grid search cross validation which we decided was necessary to find the optimal values of our hyperparameters. Also, categorical variables that are not binary are handled by caret, automatically creating dummies when training.

4 Descriptive statistics and data washing

Our data consists of two data sets: patient data and tooth data. The patient data includes ten explanatory variables and the tooth data consists of 20 explanatory variables. The patient data consists of explanatory variables which are related to the patients health and background while the tooth data consists of explanatory variables related to the health of each tooth as well as the outcome variable *Periodontitis progression*. The data sets consist of information sampled from the same patients, but as the gathering of data was done on separate visits there are some patients that have dropped out of the study, these have been removed from the data set as they do not have a value for the response variable.

The patient data consists of 213 observations and the tooth data consists of 3408 observations, these where joined into one data set with 3408 observations. After removing the drop outs the data set consists of 183 patients and 2928 teeth. They were joined by matching patient ID which we were told by the client was the same for both data sets.

In Table 8 the explanatory variables in the data set is summarized, before any data washing has been executed. From this we have concluded that there are some variables that can be removed from the models, namely the dummy variables which only have observations with the value "No". These are: *Pregnancy*, *Diabetes mellitus not controlled*, *Blood or immune disease*, *Monogen genetic disturbance*, *Osteoperosis*, *Malnutrition* and *Sjögrens syndrom*. As these only have the value "No" the model can not learn anything from them. This removes all systematic diseases except *Diabetes mellitus controlled* and *Drug induced periodontal disease*, which are also questionable if they should be included as they have very few values equal to "Yes". We choose to include *Diabetes mellitus controlled* and *Drug induced periodontal disease*, despite that their contribution to the model might be small. The excluded variables are marked in tables 3, 4 and 5 as "Excluded".

The model that we are using is robust against missing values in explanatory variables which means that we are not required to omit a large amount of our observations.

Table 8: Summary of explanatory variables. Explanatory variables are written in cursive and the descriptive statistics are written in the cell bellow.

Age	*Mother suffers from periodontitis*	*Father suffers from periodontitis*	*Skin prick test*	*Awereness and interest of disease*
Min. :29.71 Median :50.52 Mean(std) :48.51 (10.14) Max. :76.35	No : 29% Yes : 21% Don't know :50%	No : 32% Yes : 16% Don't know :52%	Min. :0.00 Median :2.00 Mean(std) :1.88 (1.16) Max: 3.00	No: 2% Little : 27% High : 70%
Diabetes mellitus, controlled	*Blood or immune disease*	*Monogen genetic disturbances*	*Granulomatous disease*	*Drug induced periodontal disease*
No: 98% Yes: 2%	No: 100%	No: 100%	No: 100%	No: 99% Yes: 1%
Osteoperosis	*Malnutrition*	*Sjögrens syndrom*	*Pregnant*	*Diabetes mellitus, not controlled*
No: 100%	No: 100%	No: 100%	No: 100%	No: 100% %
Socioeconomic factors	*Current smoking*	*Previous smoking*	*Operator experience*	*Probing plaque*
None: 74% Stress: 24% Poor: 16%	No: 69% $<$ 10: 17% 10-20: 11%	No: 67% $> 2years$: 2% $< 2years$: 31% Smoker: 2%	Some: 60% Extensive:40%	No: 30% Yes: 55% NA's: 14%
Endodontic factors	*Furcation involvement*	*Vertical destruction*	*Initial bone level*	*Pocket depth*
Intact: 38% Destruct.: 1% NA's: 60%	None: 25% $< 2mm$: 2% $> 2mm$: 3% NA's: 70%	No: 74% Yes: 10% NA's: 16%	Min: 0.75 Median: 2.80 Mean(std) : 3.29 (1.69) Max. :12.00 NA's :729	Min: 3.00 Median: 3.50 Mean(std) : 3.79 (1.01) Max. :11.00 NA's :903
Bleeding on probing	*Restored surface*	*Mobility*	*Abutment*	
No: 32% Yes: 54% NA's: 14%	No: 31% Crown: 33% Root: 20% NA's: 16%	No: 57% Yes: 6% NA's: 36%	Yes: 4% NA's: 96%	

Table 9: Summary of dependent variable

Progression of periodontitis		
No: 40%	Yes: 45%	NA's: 15%

The way the model handles missing values is described in Section 3.7.5. However, large amount of missing values will lead to a weaker model.

The variable of interest in this study is *Progression of periodontitis*, seen in Table 9. This variable has a good spread between "yes" and "no" which should benefit us when training our model. As this is the dependent variable used to train the model it can not include missing values. As we are not confident whether there is an underlying pattern to the missing value or not, we have chosen not to compute any values for this variable. Instead we choose to remove the observations which has missing values for *progression_yes_no*. This leaves us with a washed data set consisting of 2485 observations. After removing the explanatory variables that do not contribute to the model we are left with 22 explanatory variables. We choose to divide our data set into two separate data sets, our training data and our test data. The training data is used to train the models and the test data is used when evaluating the performance of the model. The data was split with 70% in the training set and 30% in the test set. Meaning that we are training the models on 1740 observations and test the models on 745 observations.

5 Results

In this section the results of the different models are presented. The shown outputs are the hyperparameter values found by grid search cross validation, and metrics of the model performance. Besides the output, tables of variable importance in the model are included to show the most prominent predictors, this information will also be the foundation for the model suggested by the authors.

5.1 Small, medium and large models

The results of the three models suggested by the client can be seen in Table 9. All three models are trained on a data set consisting of 1740 observations (70% of the washed data set) and tested on a new data set consisting of 745 observations (30% of the washed data set). The small model consists of 12 explanatory variables (see table 5), the medium model consists of 19 explanatory variables (see table 4), and the large model consists of 22 explanatory variables (see table 3). All three models are firstly trained with 5-fold cross validation with 2 repeats to find the optimal values for the hyperparameters. After optimal values are found the models are trained again using 5-fold cross validation with 10 repeats. Different number of folds and repeats for our

Table 10: Small, medium and large models

	Models:		
	Small	Medium	Large
η	0.05	0.04	0.05
γ	0.125	0.7	1.4
Max depth	6	8	3
min child weight	0.2	0.6	2.5
nrounds	1800	500	600
subsample	0.4	0.4	0.75
colsample bytree	0.2	0.2	0.65
Accuracy	0.694	0.7074	0.745
Precision	0.6875	0.7079	0.7575
Recall	0.7778	0.7677	0.7652
Specificity	0.5989	0.6418	0.7221
F1 score	0.7299	0.7354	0.7613
AUC	0.6883	0.7035	0.7436

cross validation are tested but the procedures only makes a small difference in the results which is why we decide on five folds with ten repeats.

In Table 10 we see that the models have an accuracy of 69% for the small model, 71% for the medium model and 75% for the large model. The small and medium models achieve similar results in all evaluation metrics and the large model performs better in all metrics except recall.

In Table 11 the 20 most important explanatory variables in the large model are shown. Importance score is the relative influence of each explanatory variable scaled between 0-100, 0 being not important and 100 being very important [14]. Here we see that *age*, *initial bone level* and *vertical destruction* are the three most important variables by a large margin. Most of the explanatory variables have an importance score below 10 meaning that they contribute less to the model relative to the variables with higher scores. Non-binary explanatory categorical variables in the variable importance table show different importance values depending on which category value that was split. These results are used to construct the model suggested by the authors.

5.2 Suggested model

The suggested model includes six explanatory variables: *Initial bone level*, *Age*, *Ver-*

Table 11: Variable importance large model

Explanatory variable	Importance score	Explanatory variable	Importance score
Initial bone level	100	Restored crown	7.060
Age	99.457	Father suffer periodontitis	5.909
Vertical destruction	79.880	Restored root	5.761
Pocket depth	21.699	Little awareness of diseas	5.582
Furcation >2mm	19.958	Mother suffers from periodontitis	5.281
Furcation <2mm	19.720	Don't know if mother suffers from periodontitis	4.993
Skin prick test	10.569	Extensive operator experience	4.673
Mobility	8.301	Current smoking <10	4.489
Endodontic destruction	7.482	Smoking previous <2years	4.423
Probing plaque	7.391	Bleed on probing	3.957

Table 12: Results of the suggested model

	Hyperparameters:
η	0.05
γ	0.125
max depth	6
min child weight	0.2
nrounds	1800
subsample	1
colsample by tree	0.2
	Metrics
Accuracy	0.7477
Precision	0.7574
Recall	0.7727
Specificity	0.7192
F1 score	0.7650
AUC	0.746

Observations training=1740, *explanatory variables*=6,
observations testing=745

tical destruction, *Pocket depth*, *Furcation involvement*, and *Results of skin prick test*. The explanatory variables are chosen based on the importance scores in Table 11. These six variables all have importance scores above ten which we choose as the cut off point. When building the model we have also tried with more explanatory variables and found that it either performed worse or marginally better than with the chosen variables. We have also tried creating a smaller model consisting of *Initial bone level*, *Age*, *Vertical destruction*, *Pocket depth* and *Furcation involvement*. This model performs slightly worse than the one we suggest. The accuracy of this model on the test set was 0.7356, the F1 score was 0.7565 and the AUC was 0.7331. As it performs slightly worse it is not the model suggested by the authors, in the case that the *Skin prick test* is not an available variable this model performs adequately. The details of this model can be seen in Appendix 7.

The results of the suggested model can be seen in Table 12. The final hyperparameter values are shown in the top section and the evaluation metrics are shown in the bottom section. The hyperparameters are chosen by 5 fold cross validation with 5 repeats, the final model is afterwards trained with 10 folds and 10 repeats. The suggested model makes correct predictions on the test set 75% of the time. And comparing table 10 and 12 we see that the suggested model performs equally well as the large model.

6 Discussion

In this section the results of the models and their limitations are discussed. Also, factors necessary to take into consideration should these models be used by dental clinicians and researchers are discussed.

6.1 Model evaluation

The models with the highest accuracy are the large model and the suggested model which have very similar values for all evaluation metrics. The small and medium model perform similarly to each other and have lower values for the evaluation metrics compared to the large and suggested model. The medium and large model have a very similar constellation of explanatory variables and we believe that the difference in performance is mostly due to the medium model not including the variables *Vertical destruction* and *Furcation involvement*. These are two of the most important explanatory variables in the large model but are not included in the medium model which could explain the difference in accuracy.

The large and the suggested model perform similarly which is reasonable as the suggested model is built based on the explanatory variables with the highest variable importance score in table 11. Since these two models perform very similarly it is also

an indication that the explanatory variables that were omitted when constructing the suggested model had very little impact on the performance of the large model. This could be due to the explanatory variables having weak relationships with the progression of periodontal disease, or that the data set is too small to recognise the relationship. The small and medium model are also similar, with the medium model performing slightly better. Looking at the variable importance score of these models in the Appendix Sections 8 and 9 it can be seen that *Age* is the most important variable in both of the models. In the medium model the variables *Initial bone level* and *Pocket depth* also have high importance scores, the inclusion of these two explanatory variables is likely to be the reason for slightly higher values in the evaluation metrics compared to the small model.

Interpreting the evaluation metrics of the models is not completely straight forward as it depends on the context of the study. There are no definitive good or bad results since the process of evaluating the metrics vary across fields. We will compare our results to the results of Patel et al. 2022 [27] as the method of their study is the most similar to ours. The results of their evaluation metrics can be found in Appendix 10. Compared to their results our models have higher precision, recall and F1-score than all their models. The model from their study which performs best is the "Combined model" which has an AUC of 0.72. The small and medium models which we built have a lower AUC but our large and suggested models have a higher AUC. It should however be mentioned that our results are only marginally better than the results of their study, comparing an AUC of 0.75 to their 0.72. Their study is constructed differently than ours in a few key factors, specifically that they use a larger data set with a larger amount of explanatory variables and also that their study does not use longitudinal data. This is a key difference as they predict individuals risk of periodontal disease in the present while our study predicts the risk of progression of periodontal disease 3-4 years into the future, which means that our results are more useful since we are both catching disease progression at an earlier stage, and using less information (fewer variables).

The washed data set only includes 183 unique patients which means that the model has very little data to learn from regarding how the variables related to patient health are indicators of progression of periodontal disease. As there are few observations for the model to learn from it is less likely that it will find the relationship between these explanatory variables and the progression of periodontitis. As can be seen in Table 11, 15 and 16 the variables related to patients health generally have lower variable importance scores than the variables relating to tooth health. We can not however draw definitive conclusions of whether the variables related to the patients health do not have a strong relationship with the outcome variable in the population or if it is due to the small sample, the model would have to be trained using a larger sample for us to get a definitive answer. It is possible that patient level variables have

stronger predictive power in the population than it has in this sample. If this is the case, it is likely that models with higher accuracy could be constructed using a larger sample. This is especially true for the small model which performed worst of the four constructed models. The small model consist mainly of explanatory variables related to patients health. If the variables related to patient health have a strong relationship with the outcome variable in the population it is likely that the small model can be improved by training on a larger sample. The medium and large model would also benefit from a larger data set, and likely achieve predictions with higher accuracy.

The variable importance metrics produced by the models shown in tables 11, 15, 16 and 17 could have inferential value. The reasonableness of the variables that showed large and small importance when classifying could benefit from being further investigated by dentistry professionals. Further, dental exams can be tailored based on which variables that showed large and small variable importance. The large model shows interesting information since it provides a comparison of the variable importance of all variables in table 11. Tooth level variables are shown to have higher variable importance compared to the patient level variables, the *Age* variable is the exception. This pattern can also be seen in the medium and suggested model in tables 15 and 16. For all models, patient level variables (except age) show a low variable importance. Across the models the explanatory variables *Initial bone level*, *Age*, *Vertical destruction*, *Pocket depth*, and *Furcation involvement diagnostable* show the largest variable importance.

6.2 Recall & Specificity

The recall metric of the models is similar across all models and decreases slightly as the specificity increases as expected. The specificity of the models however, varies more and increases rapidly as the complexity and accuracy of the models increase. The results show that the accuracy of the models are to a larger extent affected by the specificity of the models. Meaning that the models' ability to correctly classify individuals that do not have periodontitis influences the models' accuracy more than being able to correctly classify observations that do have periodontitis, given the chosen threshold of what classifies as a positive prediction.

In order to appropriately tune the threshold when classifying, professional dentists' expertise should be considered. The default threshold in XGBoost is 0.5. It can be altered to appropriately consider the risks involved in miss-classifying positives and negatives. The recall showing an average value of approximately 76% across the models means that on average approximately 24% of individuals will be predicted as having periodontitis when they in fact do not. Professionals should especially consider that the specificity of the models are on average approximately 67% meaning that on

average approximately 33% of individuals are predicted as not having periodontitis progression when they in fact do. The models not being able to predict periodontits progression poses a risk to patients' health. Based on professionals' knowledge of risk, dentists can determine the appropriate trade-off between using models that have a lower or higher risk of miss-classifying positives or miss-classifying negatives. The classifying threshold can thereafter be altered appropriately.

6.3 External validity

We can not be certain that the sample used in this thesis is a good representation of the larger population. An indication that this might not be the case is that the sample used to construct our models are patients that have visited the dentist for both the first and follow up appointment in which the data was collected. From this fact we can assume that the patients within the sample visits the dentist somewhat regularly. Should these models be used to predict the progression of periodontal disease for patients who do not regularly visit a dental clinician it is possible that the models produce inaccurate predictions.

The scope of this problem is hard to assess as the population is unknown. We do however believe that the models are most likely to be used to predict periodontal disease in patients who do regularly visit the dentist as the data would have to be collected by a clinician. It is more likely that the data needed to use the models for predictions are available for patients who regularly visits the dentist and as such we believe that the population, for the purpose of using these models for predictions, can be defined based on the sample. In the case that the models are used to predict periodontal disease for patients who do not regularly visit the dentist, we are unsure of the validity of the models.

6.4 Summary

The research question which we aimed to answer in this thesis is:

Is it possible to make accurate risk predictions of chronic periodontitis using the supervised machine learning technique regularized gradient boosting machine using the XGBoost software?.

To answer this question we have constructed four models using the XGBoost framework. The washed data that was used to construct the models consists of 183 patients with 2485 unique teeth with explanatory variables relating both to the health of the patient and the health of each individual tooth. The outcome variable of our research was *Progression of periodontitis*, the initial state of periodontal disease was measured during the first appointments which where conducted during 1998 and 1999,

the progression of their periodontal health status was then measured the second appointments which were conducted during 2002. As such the constructed models are designed to predict the progression of periodontal disease in the coming 3-4 years.

The models are labeled *small*, *medium*, *large* and *suggested*. The first three models are constructed based on the suggestions of the external client which we are working with, and the *suggested model* is constructed by us choosing explanatory variables based on their importance score in the large model.

7 Conclusion

We consider these results to be promising, but as statisticians we are not knowledgeable in the field of periodontology, and as such we do not know how accurate the models needs to be for them to be useful in the setting of a dental clinic. Because of this we leave the interpretation of these results to the professionals within the field of dentistry and periodontology.

We do however believe that these results are promising as we achieved higher values for our evaluation metrics compared to a similar study that also used the XGBoost framework. Because of this we believe that the method has not yet been used to it's full potential in the subject of predicting periodontitis. With a larger sample we believe that it is possible to build models which achieve higher accuracy.

Our results indicate that there is potential for similar models to become useful tools in the field of dentistry. Because of this we encourage other researchers in the field to use XGBoost and other machine learning methods in future research aiming to predict periodontal disease. We believe that with larger samples these methods are likely to achieve impressive accuracy compared to earlier methods used to predict periodontitis.

References

[1] 1177 Vårdguiden. *Sjögrens syndrom*. Accessed: 2023-05-17. 2022. URL: https://www.1177.se/Uppsala-lan/sjukdomar--besvar/ogon-oron-nasa-och-hals/ogonbesvar/sjogrens-syndrom/.

[2] Brown, LJ and Löe, H. "Prevalence, extent, severity, and progression of periodontal disease". In: *Periodontol 2000* 2 (1993), pp. 57–71. DOI: 10.1111/j.1600-0757.1993.tb00220.x..

[3] Chen, Tianqi and Guestrin, Carlos. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 785–794. ISBN: 9781450342322. DOI: 10.1145/2939672.2939785. URL: https://doi.org/10.1145/2939672.2939785.

[4] Corporation, Microsoft. *LightGBM Documentation*. 2021. URL: https://lightgbm.readthedocs.io/en/v3.3.2/ (visited on 05/03/2023).

[5] Daoud, Al Essem. "Comparison between XGBoost, LightGBM, and Catboost Using a Home Credit Dataset". In: *International Journal of Computer and Information Engineering* 13.1 (2019). Accessed: 2023-03-26. DOI: doi.org/10.5281/zenodo.3607805. URL: https://publications.waset.org/10009954/pdf.

[6] Dental, National Institute of and Research, Craniofacial. *Gum Disease*. 2021. URL: https://www.nidcr.nih.gov/health-info/gum-disease.

[7] Farid, Mohammed. *Kaggle Solutions*. n.d. URL: https://farid.one/kaggle-solutions/ (visited on 05/03/2023).

[8] Folktandvården Sörmland. *Tandbroar*. Accessed: 2023-05-17. 2022. URL: https://www.folktandvardensormland.se/rad--tips/tandbroar/.

[9] Friedman H., Jerome. "Greedy Function Approximation: A Gradient Boosting Machine". In: *Ann. Statist.* 29.5 (2001), pp. 1189–1232. DOI: https://doi.org/10.1214/aos/1013203451.

[10] Friedman H., Jerome. "Stochastic Gradient Boosting". In: *Department of Statistics and Stanford Linear Accelerator Center.* (2001). DOI: https://doi.org/10.1016/S0167-9473(01)00065-2.

[11] Github. *Machine Learning Challenge Winning Solutions*. URL: https://github.com/dmlc/xgboost/tree/master/demo#machine-learning-challenge-winning-solutions. (accessed: 30.03.2023).

[12] Google. *ROC and AUC*. URL: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc (visited on 05/03/2023).

[13] James, Gareth et al. *An Introduction to Statistical Learning: with Applications in R*. New York: Springer, 2021.

[14] Kuhn, Max and Wing, Jed. *Variable Importance*. Accessed: 2023-05-04. topepo. 2021. URL: https://topepo.github.io/caret/variable-importance.html (visited on 05/04/2023).

[15] Lang, Niklaus P, Suvan, Jean E, and Tonetti, Maurizio S. "Risk factor assessment tools for the prevention of periodontitis progression a systematic review". In: *Journal of clinical periodontology* 42 (2015), S59–S70.

[16] Lang, Niklaus P and Tonetti, Maurizio S. "Periodontal risk assessment (PRA) for patients in supportive periodontal therapy (SPT)". In: *Oral Health Prev Dent* 1.1 (2003), pp. 7–16.

[17] Lemagnen, Kevin. *Hyperparameter Tuning in XGBoost*. Cambridge Spark Blog. Accessed: 2023-05-03. Oct. 2017. URL: https://blog.cambridgespark.com/hyperparameter-tuning-in-xgboost-4ff9100a3b2f.

[18] Lindskog, Sven et al. "Validation of an Algorithm for Chronic Periodontitis Risk Assessment and Prognostication: Risk Predictors,Explanatory Values, Measures of Quality, and Clinical Use". In: *J Periodontol* 81.4 (2010), pp. 584–93. DOI: 10.1902/jop.2010.090529.

[19] LLC, Yandex. *CatBoost Documentation*. 2021. URL: https://catboost.ai/en/docs/ (visited on 05/03/2023).

[20] Löe, H et al. "The natural history of periodontal disease in man". In: *J Clin Periodontol* 13.5 (1978), pp. 607–620. DOI: 10.1111/j.1600-051x.1986.tb01487.x..

[21] Löe, H et al. "The natural history of periodontal disease in man. Rapid, moderate and no loss of attachment in Sri Lankan labourers 14 to 46 years of age". In: *J Clin periodontol* 13.5 (1986), pp. 431–445. DOI: 10.1111/j.1600-051x.1986.tb01487.x.

[22] National Institute of Allergy and Infectious Diseases. *Chronic Granulomatous Disease (CGD)*. Accessed: 2023-05-17. National Institute of Allergy and Infectious Diseases. 2020. URL: https://www.niaid.nih.gov/diseases-conditions/chronic-granulomatous-disease-cgd.

[23] NHS. *Osteoporosis*. Accessed: 2023-05-17. National Health Service. 2022. URL: https://www.nhs.uk/conditions/osteoporosis/.

[24] Nichols, Lucy. *Understanding Your Gum Health: Bone Levels*. Accessed: 2023-05-17. 2019. URL: https://www.drlucynichols.com/blog/understanding-your-gum-health-bone-levels/.

[25] Page, Roy C et al. "Longitudinal validation of a risk calculator for periodontal disease". In: *Journal of clinical periodontology* 30.9 (2003), pp. 819–827.

[26] Parr, Terrence and Howard, Jeremy. *Gradient boosting performs gradient descent*. URL: https://xgboost.readthedocs.io/en/stable/parameter.html. (accessed: 13.03.2023).

[27] Patel, Jay et al. "Developing and testing a prediction model for periodontal disease using machine learning and big electronic dental record data". In: *Faculty/Researcher Works* (2022).

[28] PerioBASICS. *Furcation Involvement and Its Management*. Accessed: 2023-05-17. URL: https://periobasics.com/furcation-involvement-and-its-management/.

[29] Powers, David M W. "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation". In: *Journal of Machine Learning Technologies* 2.1 (2011), pp. 37–63. URL: https://www.jmlr.org/papers/volume2/powers01a/powers01a.pdf (visited on 11/14/2019).

[30] Raby, Bejnamin A. *Inheritance patterns of monogenic disorders: Mendelian and non-Mendelian*. Accessed: 2023-05-17. 2021. URL: https://www.uptodate.com/contents/inheritance-patterns-of-monogenic-disorders-mendelian-and-non-mendelian.

[31] Sasaki, Yutaka. *The Truth of the F-Measure*. 2007. URL: https://www.cs.odu.edu/~mukka/cs795sum09dm/Lecturenotes/Day3/F-measure-YS-26Oct07.pdf (visited on 05/03/2023).

[32] Shreffler, Jacob and Huecker, Martin R. *Diagnostics Testing: Accuracy Sensitivity Specificity Predictive values and likelihood ratios*. Accessed: 2023-05-09. StatPearls Publishing, Apr. 2023. URL: https://www.ncbi.nlm.nih.gov/books/NBK557491/.

[33] Starmer, Josh. *Gradient Boost Part 4 (of 4):Classification Details*. Accessed: 2023-03-28. Apr. 2019. URL: https://www.youtube.com/watch?v=StWY5QWMXCw&t=363s.

[34] Starmer, Josh. *XGBoost Part 2 (of 4):Classification*. Accessed: 2023-03-28. Jan. 2020. URL: https://www.youtube.com/watch?v=8b1JEDvenQU&t=1178s.

[35] Starmer, Josh. *XGBoost Part 3 (of 4):Mathematical Details*. Accessed: 2023-03-28. Feb. 2020. URL: https://www.youtube.com/watch?v=ZVFeW798-2I&t=169s.

[36] Vilardi, Mario A. *Understanding Periodontal Pockets*. Accessed: 2023-05-17. 2017. URL: https://www.deardoctor.com/articles/understanding-periodontal-pockets/.

[37] Wing, Jed et al. *Package "caret"*. Accessed: 2023-05-11. Mar. 2023. URL: https://cran.r-project.org/web/packages/caret/caret.pdf.

[38] *XGBoost documentation: XGBoost parameters*. URL: https://explained.ai/gradient-boosting/descent.html. (accessed: 21.04.2023).

[39] Zhou, Miao et al. "Optimized XGBoost Model with Small Dataset for Predicting Relative Density Of Ti-6Al-4v Parts Manufactured By Selective Laser Melting". In: *Material* 15 (2022). Accessed: 2023-06-05. DOI: https://doi.org/10.3390/ma15155298. URL: https://www.mdpi.com/1996-1944/15/15/5298.

Appendix

Appendix 1

Loss function for binary gradient boosting and showing how we calculate the output of the gradient boosting machine. The following mathematics have been derived by Josh Starmer BsC in Computer science and PhD in Biomathematics, Bioinformatics and computational biology. [33]
Input: Data $\{(x_i, y_i)\}_{i=1}^{n}$ and a differentiable loss function: $L(y_i, F(x_i))$. The data refers to the training data and the loss function is: log(likelihood of observed data given the prediction). With p_i=predicted probability and y_i=observed value it can be written as:

$$Log(likelihood\ of\ the\ observed\ data\ given\ the\ prediction) =$$

$$= -\sum_{i=1}^{n}(y_i * log(p_i) + (1 - y_i) * log(1 - p_i)).$$

For one response variable the loss function can be written as: $-[Observed * log(p) + (1 - Observed) * log(1 - p)]$, simplified step by step as:

$$1. -Observed * log(p) - (1 - Observed) * log(1 - p)$$

$$2. -Observed * log(p) - log(1 - p) + Observed * log(1 - p)$$

$$3. -Observed[log(p) - log(1 - p)] - log(1 - p).$$

Since: $log(p) - log(1 - p) = \frac{log(p)}{log(1-p)} = log(\frac{p}{1-p}) = log(odds)$

$$4. -Observed * log(odds) - log(1 - p).$$

Since:

$$log(1 - p) = log(1 - \frac{e^{log(odds)}}{1 + e^{log(odds)}}) = log(\frac{1 + e^{log(odds)}}{1 + e^{log(odds)}} - \frac{e^{log(odds)}}{1 + e^{log(odds)}})$$

$$= log(\frac{1}{1 + e^{log(odds)}} = log(1) - log(1 + e^{log(odds)} = -log(1 + e^{log(odds)}).$$

$$5. -Observed * log(odds) + log(1 + e^{log(odds)})$$

Before moving on we have to make sure that the loss function is differentiable.

$$\frac{d}{dlog(odds)}(-Observed * log(odds) + log(1 + e^{log(odds)}))$$

$$= -Observed + \frac{e^{log(odds)}}{1 + e^{log(odds)}} = -Observed + p.$$

Now that we have our loss function we can start building the gradient boosting model.

Step 1: Initialize the model with a constant value: $F_0(x) = \underset{\gamma}{\text{argmin}} \sum_{i=1}^{n} L(y_i, \gamma)$. $\gamma = log(odds)$ and what we will do is find a log(odds) that minimizes the loss function so we will derive the loss function with log(odds) and set it equal to zero, then solve for p. $p = log(1 + e^{log(odds)})$.

$$\underset{\gamma}{\text{argmin}} \sum_{i=1}^{n} (-Observed * log(odds) + log(1 + e^{log(odds)}) = 0$$

Simplifying this we get $p = \frac{\sum_{i=1}^{n} y_i}{n}$ and $log(odds) = \frac{p}{1-p}$.

Step 2: For m=1 to M:

(A) Compute $r_{im} = -[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}]_{F(x)=F_{m-1}(x)}$ This is the derivative of the Loss function with respect to the log(odds) which we have already calculated. i=sample number and m=tree that we are building. Hence we can calculate the pseudo residuals by:

Pseudo residual=$Observed - \frac{e^{log(odds)}}{1+e^{log(odds)}}$

(B) Fit regression tree to the r_{im} values and create terminal regions R_{jm}, $j = 1, ..., J_m$

(C) For $j = 1, ..., J_m$ compute $r_{jm} = \underset{\gamma}{\text{argmin}} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)$

$$= \underset{\gamma}{\text{argmin}} \sum -y_i * [F_{m-1} + \gamma] + log(1 + e^{F_{m-i}(x_i)+\gamma})).$$

Since this will be difficult to derive we can instead approximate the function with the second order Taylor polynomial.

$$\sum_{x_i \in R_{ij}} L(y_i, F_{m-i}(x_i) + \gamma)$$

$$\approx \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \frac{d}{dF()}(y_i, F_{m-1}(x_i))\gamma + \frac{1}{2} * \frac{d^2}{2dF()^2}(y_i, F_{m-1}(x_i)) * \gamma^2$$

$$\frac{d}{d\gamma} \sum_{x_i \in R_{ij}} L(y_i, F_{m-i}(x_i) + \gamma)$$

$$= \frac{d}{dF()}(y_i, F_{m-1}(x_i)) + * \frac{d^2}{2dF()^2}(y_i, F_{m-1}(x_i)) * \gamma = 0.$$

Solve for γ:

$$\gamma = -\sum_{x_i \in R_{ij}} \frac{\frac{d}{dF()}(y_i, F_{m-1}(x_i))}{\frac{d^2}{2dF()^2}(y_i, F_{m-1}(x_i))}$$

$$= \sum_{x_i \in R_{ij}} \frac{y_i - p}{\frac{d}{dlog(odds)} - y_i + \frac{e^{log(odds)}}{1+e^{log(odds)}}}.$$

Which can also be written as:

$$\gamma = \sum \frac{Residuals}{p(1-p)}.$$

(D) Update $F_m(x) = F_{m-1} + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$, ν =learning rate and M=total amount of trees. The output of our final model is $F_M(x_i)$ for an observation x_i. This model gives us the log(odds) which we can use to calculate predicted probabilities with $\frac{e^{log(odds)}}{1+e^{log(odds)}}$. [35]

Appendix 2

Lets create a fictitious data set using explanatory variables from the data set in the thesis and show how XGBoost creates the trees using the exact greedy method. The Age variable is the age of the individuals and the explanatory variable bleeding with the value 1 represents bleeding on any of two surfaces in the mouth, and 0 otherwise.

Table 13: Example data

Age	Bleed	Periodontitis
29	1	0
39	0	0
40	0	0
43	1	0
56	0	0
58	0	1
68	1	1
76	1	0
83	1	1
90	0	1

As mentioned, using the exact greedy method, XGBoost tests all possible splits across

all explanatory variables. Lets look at an example of how XGBoost would choose the best split and then predict new observations using this fictitious data set. If XGBoost tests splitting at $Age \leq 58$ it calculates a score to determine if it is the split that decreases the loss function the most. The score is calculated using the following formula:

$$Gain = \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{G_j^2}{H_j + \lambda} - \gamma.$$

Where the underscore L represents the left leaf and the underscore R represents the right leaf, and the underscore j represents the entire data set before splitting. Intuitively, XGBoost adds the score of the left leaf and right leaf, and subtracts the score of not splitting at all. This gain score is compared to the scores of all possible splits across all explanatory variables and the highest score is chosen as the nodesplit. The γ term is explained further in Subsection 3.7.1.

The formula used for scoring the leaves is derived as seen below, the formal derivation is included in the Appendix 4:

$$\begin{aligned} O_{Value} &= \frac{\sum (y_i - p_i)^2}{\sum (p_i * (1 - p_i)) + \lambda} \\ &= \frac{(\sum Residuals_i)^2}{\sum (Previousprobability_i * (1 - Previousprobability_i)) + \lambda} \\ &= \frac{G_{Leaf}^2}{H_{Leaf} + \lambda} \end{aligned}$$

where y_i denotes the predicted probability of periodontitis for an observation and p_i denotes the probability of periodontitis in the previous iteration (previous tree structure). Intuitively, the sum of residuals squared $(y_i - p_i)^2$ are calculated for all observations belonging to a leaf. The value is divided by the sum of the previous probabilities times 1 minus the previous probability for each observation. λ is a regularization term is explained further in subsection 3.7.1.

XGBoost calculating the score for the split $Age \leq 58$ would calculate the scores associated to each leaf as seen below. The values 1 or 0 for periodontits are put in the formulas based on the split. The initial previous probability is set to 0.5 as default by XGBoost.

$$\frac{G_L^2}{H_L+\lambda} = \frac{\sum(y_i - p_i)^2}{\sum(p_i * (1-p_i)) + \lambda}$$
$$= \frac{((0-0.5)+(0-0.5)+(0-0.5)+(0-0.5)+(0-0.5))^2}{0.5*(1-0.5)+0.5*(1-0.5)+0.5*(1-0.5)+0.5*(1-0.5)+0.5*(1-0.5)} = 5$$

$$\frac{G_R^2}{H_R+\lambda} = \frac{\sum(y_i - p_i)^2}{\sum(p_i * (1-p_i)) + \lambda}$$
$$= \frac{((1-0.5)+(1-0.5)+(1-0.5)+(1-0.5)+(0-0.5))^2}{0.5*(1-0.5)+0.5*(1-0.5)+0.5*(1-0.5)+0.5*(1-0.5)+0.5*(1-0.5)} = 1.8$$

$$\frac{G_j^2}{H_j+\lambda} = \frac{\sum(y_i - p_i)^2}{\sum(p_i * (1-p_i)) + \lambda} = \frac{(\sum_1^6(0-0.5)_i + \sum_1^4(1-0.5)_i)^2}{\sum_1^{10}(0.5*(1-0.5)_i)} = 0.4.$$

The graphical representation below (Figure 5) shows the residuals allocated to each leaf and the score calculated for each leaf.

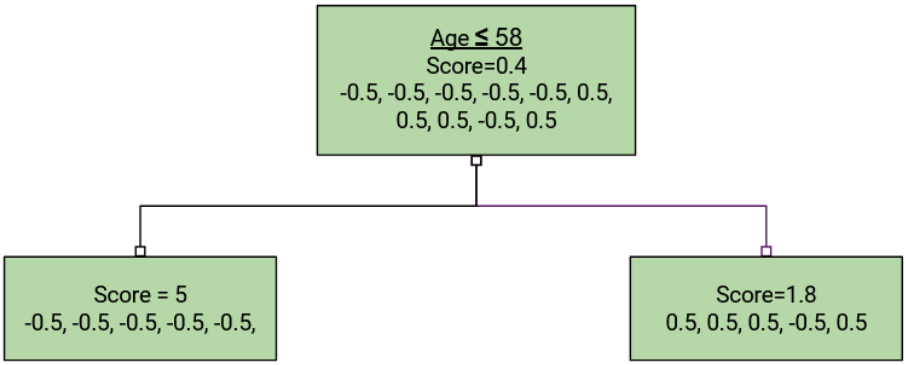

Figure 5: Example tree building

Plugging in the scores calculated for each leaf gives the following gain of splitting at $Age \leq 58$. Gamma is set to zero.

$$Gain = \frac{G_{split}^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G_j^2}{H_j+\lambda} - \gamma = 5 + 1.8 - 0.4 - 0 = 6.4.$$

After calculating the Gain score, XGBoost compares the score with scores calculated

across all explanatory variables for all values. This method is called the exact greedy algorithm and is explained in Subsection 3.7.4. The gain score for splitting the observations at $Age \leq 58$ can be compared to splitting the data based on the binary variable bleed. Putting the predicted probabilities for periodontits into the leaves based on the split gives the following:

$$\frac{G_L^2}{H_L + \lambda} = \frac{\sum(y_i - p_i)^2}{\sum(p_i * (1 - p_i)) + \lambda}$$
$$= \frac{((0 - 0.5) + (0 - 0.5) + (0 - 0.5) + (1 - 0.5) + (1 - 0.5))^2}{0.5 * (1 - 0.5) + 0.5 * (1 - 0.5) + 0.5 * (1 - 0.5) + 0.5 * (1 - 0.5) + 0.5 * (1 - 0.5)} = 0.2$$

$$\frac{G_R^2}{H_R + \lambda} = \frac{\sum(y_i - p_i)^2}{\sum(p_i * (1 - p_i)) + \lambda}$$
$$= \frac{((0 - 0.5) + (0 - 0.5) + (1 - 0.5) + (0 - 0.5) + (1 - 0.5))^2}{0.5 * (1 - 0.5) + 0.5 * (1 - 0.5) + 0.5 * (1 - 0.5) + 0.5 * (1 - 0.5) + 0.5 * (1 - 0.5)} = 0.2$$

$$\frac{G_j^2}{H_j + \lambda} = \frac{\sum(y_i - p_i)^2}{\sum(p_i * (1 - p_i)) + \lambda} = \frac{(\sum_1^5 (0 - 0.5)_i + \sum_1^5 (1 - 0.5)_i)^2}{\sum_1^{10} (0.5 * (1 - 0.5)_i)} = 0.4.$$

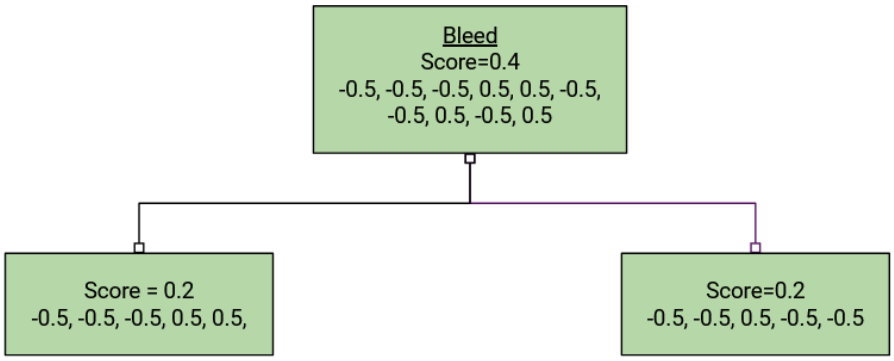

Figure 6: Example tree buildning

The Gain score for the potential split is calculated as follows (See also figure 6):

$$Gain = \frac{G_{split}^2}{H_split + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{G_j^2}{H_j + \lambda} - \gamma = 0.2 + 0.2 - 0.4 - 0 = 0.$$

The Gain score for splitting by the binary variable bleed is 0 which is lower than the gain score calculated when splitting the by $Age \leq 58$ which was 6.4. XGBoost would therefore choose $Age \leq 58$ as the split for the first node. Ofcourse, XGBoost does not only compare these two potential splits, but all potential splits across all explanatory variables for all values. In the second node XGBoost will again search for the optimal split across all explanatory variables and values. And the same for the third and the fourth node split. The optimal amount of splits per tree is a parameter that is found using grid search when running the model.

In conclusion, when XGBoost chooses the best possible split it can be seen that the explanatory variable splits that allocates homogeneous values for the predicted probability of periodontitis results in higher gain scores. This can be seen in table 13, where the risk of periodontitis increases with age and the split $Age \leq 58$ allocates only 0 values to the left leaf and mostly 1 values to the right leaf. Whereas the binary split for bleed allocates does not clearly allocate the values for periodontitis in a homogenous way. The purpose of XGBoost is to find explanatory variable splits that allocates as homogenous values as possible for the output variable into the leaves. [3] [34]

Appendix 3

Continuing on the example in the previous section, since the split $Age \leq 58$ lead to a higher gain score than when XGBoost attempted splitting by the binary variable bleed, the split $Age \leq 58$ is used as the first tree node. For illustrative purposes the tree depth is set to 1 and the tree is not split any further. In order to make new predictions, the optimal weights for each leaf are firstly calculated using a similar formula as for the scoring of the leaves. The difference is that in the formula for the output values, the residuals are not squared. The following formula is used to calculate optimal weights: [3] Lambda is set to 0.

$$w^* = \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} = \frac{G_{leaf}}{H_{leaf} + \lambda}$$

$$w^*_{leftleaf} = \frac{G_L}{H_L + \lambda} = \frac{\sum(y_i - p_i)}{\sum(p_i * (1 - p_i)) + \lambda}$$
$$= \frac{((0 - 0.5) + (0 - 0.5) + (0 - 0.5) + (0 - 0.5) + (0 - 0.5))^2}{0.5 * (1 - 0.5) + 0.5 * (1 - 0.5) + 0.5 * (1 - 0.5) + 0.5 * (1 - 0.5) + 0.5 * (1 - 0.5)}$$
$$= -2$$

$$w^*_{rightleaf} = \frac{G_R}{H_R + \lambda} = \frac{\sum(y_i - p_i)}{\sum(p_i * (1 - p_i)) + \lambda}$$
$$= \frac{((1 - 0.5) + (1 - 0.5) + (0 - 0.5) + (1 - 0.5) + (1 - 0.5))}{0.5 * (1 - 0.5) + +0.5 * (1 - 0.5) + 0.5 * (1 - 0.5) + 0.5 * (1 - 0.5) + 0.5 * (1 - 0.5)}$$
$$= 1.2.$$

The optimal weight for the left leaf is -2 and the optimal value for the right leaf is 1.2. A new prediction for each leaf can be made by applying the boosting formula introduced by Friedman that is described in Subsection 3.5.

$$F_m(x) = F_{m-1} + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm}).$$

In Friedmans paper for gradient tree boosting, the term $\sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$ represents the solution for a new leaf. [10] In the XGBoost model this represents the optimal weight scores. The ν term represents the learning rate as described as η in the XGBoost package. [17] Friedmans formula can therefore be rewritten intuitively as:

$$New\ prediction = Previous\ prediction + \eta * optimal\ weight$$

Since the variable for periodontitis is binary, the optimal weight scores are in log-odds. [9] The previous probability is therefore transformed into log-odds as well. The previous probability is the same as the initial probability set by XGBoost as 0.5 for the first iteration.

$$log(\frac{p}{1-p}) = log(\frac{0.5}{1 - 0.5}) = 0.$$

Calculating a new log-odds prediction using the formula for new predictions above, for observations that will allocate to the leaves: η is set to 0.05.

$$log\ odds_{left\ leaf} = log(0 + -2 * 0.05) = -0.1$$

$$log\ odds_{right\ leaf} = log(0 + 1.2 * 0.05) = 0.06.$$

The log-odds predictions are transformed into a probability by using the formula:

$$Probability = \frac{e^{log\ odds}}{1 + e^{log\ odds}}$$

$$Probability_{leftleaf} = \frac{e^{-0.1}}{1 + e^{-0.1}} = 0.4750208$$

$$Probability_{rightleaf} = \frac{e^{0.06}}{1 + e^{0.06}} = 0.5149555.$$

Since the predicted probilities for periodontitis for observations allocated to the left leaf is below 0.5, XGBoost will give an output of 0, thus predicting that the observation does not have periodontitis progression. Since the predicted probability for the right leaf is higher than 0,5, future observations that are allocated to the leaf will be predicted as 1 by XGBoost, i.e that there is periodontits progression. Individuals that are $Age \leq 58$ will therefore be classified as not having periodontitis progression, and individuals that are not $Age \leq 58$ will be classified as having periodontitis progression. [34]

Appendix 4

In this section we will show the process of how XGBoost build trees. We will show how it works for classification, however the algorithm is the same for regression using another loss function. The following mathematics have been derived by Josh Starmer BsC in Computer science and PhD in Biomathematics, Bioinformatics and computational biology. [35]

We will start of by defining or loss function, the negative log loss function. Here p_i =predicted probability, y_i = observed value and n=total number of observations.

$$L(p_i, y_i) = -[(y_i * log(p_i) + (1 - y_i) * log(1 - p_i))].$$

We use the loss function for building trees by minimizing:

$$[\sum_{i=1}^{n} L(p_i, y_i)] + \gamma T + \frac{1}{2}\lambda O_{Value}^2.$$

In this equation we see two regularization parameters, γ and λ. γ is a regularization parameter which encourage pruning of the tree and T is the number of leafs. We will set $\gamma = 0$ to make the calculations easier. Pruning takes place after building the tree so it is not necessary here. λ shrinks the optimal output value O_Value. We are then left with:

$$[\sum_{i=1}^{n} L(p_i, y_i)] + \frac{1}{2}\lambda O_{Value}^2.$$

We start of with looking at the first tree:

$$[\sum_{i=1}^{n} L(y_i, p_i^0 + O_{Value})] + \frac{1}{2}\lambda O_{Value}^2.$$

To find the optimal output value we approximate the loss function with a second order Taylor polynomial because it will make the calculations easier.

$$L(y_i, p_i + O_{Value}) \approx L(y_i, p_i) + [\frac{d}{dp_i} L(y_i, p_i)]O_{Value} + \frac{1}{2}[\frac{d^2}{dp_i^2} L(y_i, p_i)]O_{Value}.$$

As the first derivative is the gradient we will use the notation g_i and the second derivative is the heschen so we will use the notation h_i. We then get:

$$[\sum_{i=1}^{n} L(y_i, p_i^0 + O_{Value})] + \frac{1}{2}\lambda O_{Value}^2$$

$$= L(y_1, p_1) + g_1 O_{Value} + \frac{1}{2} h_1 O_{Value}^2 +$$

$$L(y_2, p_2) + g_2 O_{Value} + \frac{1}{2} h_2 O_{Value}^2 + \cdots +$$

$$L(y_n, p_n) + g_n O_{Value} + \frac{1}{2} h_n O_{Value}^2 + \frac{1}{2}\lambda O_{Value}^2.$$

As $L(y_i, p_i)$ does not contain the output value we can omit these terms and we are left with:

$$(g_1 + g_2 + \cdots + g_n)O_{Value} + \frac{1}{2}(h_1 + h_2 + \cdots + h_n + \lambda)O_{Value}^2.$$

Minimizing this expression with respect to O_Value:

$$\frac{d}{dO_{Value}}(g_1 + g_2 + \cdots + g_n)O_{Value} + \frac{1}{2}(h_1 + h_2 + \cdots + h_n + \lambda)O_{Value}^2 = 0$$

$$= (g_1 + g_2 + \cdots + g_n) + (h_1 + h_2 + \cdots + h_n + \lambda)O_{Value} = 0.$$

Solving for O_{Value}:

$$O_{Value} = \frac{-(g_1 + g_2 + \cdots + g_n)}{(h_1 + h_2 + \cdots + h_n + \lambda)}.$$

In Appendix 1 we calculated the gradient and the heschen so we will not show the calculations here. Simplified they are: $g_i = -(y_i - p_i)$ and $p_i(1 - p_i)$, so the optimal output value is:

$$O_{Value} = \frac{\sum Residuals_i}{\sum(Previousprobability_i * (1 - Previousprobability_i)) + \lambda}.$$

Now we move on to calculating the similarity score which is used when calculating the gain of the node and pruning the tree. We start of with:

$$-[(g_1 + g_2 + \cdots + g_n)O_{Value} + \frac{1}{2}(h_1 + h_2 + \cdots + h_n + \lambda)O^2_{Value}].$$

We then plug in the formula for O_{Value}, which when simplified is equal to:

$$SimilarityScore = \frac{1}{2}\frac{(g_1 + g_2 + \cdots + g_n)^2}{(h_1 + h_2 + \cdots + h_n + \lambda)}.$$

Since $\frac{1}{2}$ is a relative term we can omit this from the formula. Then we can see that the similarity score is:

$$SimilarityScore = \frac{(\sum Residuals_i)^2}{\sum(Previousprobability_i * (1 - Previousprobability_i)) + \lambda}.$$

And cover is equal to: $\sum(p_i(1 - p_i)$.

Appendix 5

Algorithm 1: Exact Greedy Algorithm for Split Finding

Input: I, instance set of current node
Input: d, feature dimension
$gain \leftarrow 0$
$G \leftarrow \sum_{i \in I} g_i$, $H \leftarrow \sum_{i \in I} h_i$
for $k = 1$ ***to*** m **do**
  $G_L \leftarrow 0$, $H_L \leftarrow 0$
  for j *in sorted*$(I,$ *by* $\mathbf{x}_{jk})$ **do**
    $G_L \leftarrow G_L + g_j$, $H_L \leftarrow H_L + h_j$
    $G_R \leftarrow G - G_L$, $H_R \leftarrow H - H_L$
    $score \leftarrow \max(score, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda})$
  end
end
Output: Split with max score

Figure 7: Exact greedy algorithm

Appendix 6

Algorithm 3: Sparsity-aware Split Finding

Input: I, instance set of current node
Input: $I_k = \{i \in I | x_{ik} \neq \text{missing}\}$
Input: d, feature dimension
Also applies to the approximate setting, only collect statistics of non-missing entries into buckets
$gain \leftarrow 0$
$G \leftarrow \sum_{i \in I}, g_i, H \leftarrow \sum_{i \in I} h_i$
for $k = 1$ ***to*** m **do**
  // enumerate missing value goto right
  $G_L \leftarrow 0,\ H_L \leftarrow 0$
  for j *in sorted*$(I_k,$ *ascent order by* $\mathbf{x}_{jk})$ **do**
    $G_L \leftarrow G_L + g_j,\ H_L \leftarrow H_L + h_j$
    $G_R \leftarrow G - G_L,\ H_R \leftarrow H - H_L$
    $score \leftarrow \max(score, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda})$
  end
  // enumerate missing value goto left
  $G_R \leftarrow 0,\ H_R \leftarrow 0$
  for j *in sorted*$(I_k,$ *descent order by* $\mathbf{x}_{jk})$ **do**
    $G_R \leftarrow G_R + g_j,\ H_R \leftarrow H_R + h_j$
    $G_L \leftarrow G - G_R,\ H_L \leftarrow H - H_R$
    $score \leftarrow \max(score, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda})$
  end
end
Output: Split and default directions with max gain

Figure 8: XGBoost sparsity aware split finding [3].

Appendix 7

Table 14: Alternative to the suggested model

	Hyperparameters:
η	0.05
γ	0.5
max depth	8
min child weight	0.8
nrounds	500
subsample	1
colsample by tree	0.4
	Metrics
Accuracy	0.7356
Precision	0.7409
Recall	0.7727
F1 score	0.7565
AUC	0.7331

Observations=1740, *explanatory variables*=6

Appendix 8

Table 15: Variable importance small model

explanatory variable	Importance score	explanatory variable	Importance score
Age	100	Father suffers from periodontitis	3.2453
Mobility	8.1868	Current smoking $<$ 10 cigarettes per day	3.0351
Extensive special clinic experience of advanced periodontal treatment	4.8860	Current smoking $10-20$ cigarettes per day	2.5857
Bleeding on probing	4.7233	Diabetes mellitus, well controlled	1.7703
Little awareness and interest of disease after information	4.2210	Previous smoking $>$ 15 cigarettes per day and stopped $<$ 2 years ago	1.5871
Patient does not know if father suffers from periodontitis	3.9727	Current smoking $>$ 20 cigarettes per day	1.3421
Mother suffers from periodontitis	3.8481	Socio economic factors, poor economy	0.5229
Socioeconomic factors, negative stress	3.5143	Drug induced periodontal disease	0
High awareness and interest of disease after information	3.3002		

Appendix 9

Table 16: Variable importance medium model

Explanatory variable	Importance score	Explanatory variable	Importance score
Age	100	Extensive special clinic experience of advanced periodontal treatment	6.230
Initial bone level	84.015	Socio-economic factors, negative stress	6.196
Pocket depth	29.269	Probing plaque	5.806
Result of the skin prick test, three negative reactions	13.908	Little awareness and interest of disease after information	5.616
Mobility	11.935	Patient does not know if mother suffers from periodontitis	5.590
Endodontic factors	8.740	Previous smoking, is smoker	5.540
Restored surface in crown only	8.250	Mother suffers from periodontitis	5.471
Bleeding on probing	8.215	Current smoking $<$ 10 cigarettes per day	4.959
Restored surface, restoration is extnding into the root on any of the two surfaces	8.167	High awareness and interest of disease after information	4.857
Patient does not know if father suffers from periodontitis	6.483	Father suffers from periodontitis	4.757

Appendix 10

	AUC	Precision	Recall	F1-score
Healthy vs. others (PD)	0.69	0.56	0.81	0.66
Mild vs. others	0.59	0.40	0.28	0.33
Severe vs. others	0.71	0.49	0.22	0.30
Combined model	0.72	0.501	0.521	0.481

AUC, Area under the receiver operating characteristic curve; PD, Periodontal disease. Precision, Recall, and F1-score of the combined model were calculated as weighted average of the three base models.

Figure 9: Results of Patel et al. 2022 [27].

Appendix 11

Table 17: Variable importance suggested model

Explanatory variable	Importance score	Explanatory variable	Importance score
Age	100	furcat2	10.54
Initial bone level	49.91	depth	8.83
Furcation involvement diagnostable $< 2mm$ on any of the two surfaces	11.44	Result of the skin prick test, three negative reactions	0
Furcation inolvement $> 2mm$ on any of the two surfaces	4.7233		