Stockholms universitet

Finding the high-retaining customer

A study of boosted decision trees to classify user retention on an audio streaming service

Carl Samuelsson

Masteruppsats i matematisk statistik
Master Thesis in Mathematical Statistics

Masteruppsats 2020:7
Matematisk statistik
Juni 2020

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Stockholm University

Mathematical Statistics
Stockholm University
Master Thesis **2020:7**
http://www.math.su.se

Finding the high-retaining customer

A study of boosted decision trees to classify user retention on an audio streaming service

Carl Samuelsson*

June 2020

Abstract

In an audio streaming service where its success is greatly dependent on the degree to which its users retain on the service, it is of evident interest to predict who will stay and not for proactive measures. Moreover, to optimize the service towards the preferences of its users, inferential work is needed towards obtaining a better understanding of the underlying processes of why certain users decide to stay on the service or not. This work centers around the study and evaluation of boosted tree models on predictive and inferential grounds. Empirical results shows however only a marginal predictive benefit of using these rather complex models in comparison with a baseline logistic regression. However, more exhaustive future work which utilizes a larger degree of the vast amount of information available could prove the boosted tree models justice for the task of interest.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: c.samuelsson94@gmail.com . Supervisor: Chun-Biu Li.

Sammanfattning

För en ljudströmningstjänst vars framgång i stor grad definieras av hur pass väl dess användare väljer att bevara sin prenumeration av tjänsten, är det av uppenbart intresse att vilja prediktera vem som riskerar annullera sin tjänst, i syfte av att kunna vidta proaktiva åtgärder. Därutöver är det viktigt att kunna dra inferens om de bakomliggande processer kring varför användare förblir prenumeranter över tid, således också kunna anpassa tjänsten efter användarnas önskemål. Detta arbete studerar nyttan i att använda "boostade" beslutsträd utifrån ett prediktivt- och statistiskt inferensperspektiv. Resultaten visar enbart en måttlig förbättring i prediktiv stryrka för den mer komplexa modellklassen av boostade beslutsträd, i jämförelse med en logistisk regressionsmodell som referens. Det vidhålls däremot att vidare studier centrerat runt användning av fler variabler skulle kunna tala till fördel för de boostade beslutsträden som modellklass.

Foreword and Acknowledgements

I would like to take this opportunity to express my deepest gratitude towards the people who have helped and supported me throughout this conducted work, especially during partly challenging and uncertain times.

Firstly, I would like to thank my two supervisors: Chun-Biu Li from the Mathematical Department of Stockholm University, and Filip Stojanovski from the Premium Product Insights team at Spotify AB. Your continuous guidance and support have been integral for me to finish the project. I would also like to thank my manager Pär Axelsson and the whole Premium Product Insights team who have made me feel like home from the first day I arrived. Without any official supervisor or mentor statuses, I am greatly thankful towards Henrik Eriksson and Patrik Liu Tran for shouldering roles as undeclared mentors in different aspects. Moreover, I would like to express my appreciation towards Amir Rahnama and my brother Daniel Samuelsson for their insightful comments and proof reading. I am also highly grateful for the love and support I have received from friends, family and my girlfriend Julia.

Lastly, I would like to acknowledge all the excellent professors, lecturers and teacher assistants at the department from who I have had the honor to learn from during my five years of studies.

Contents

Abbreviations

ALE	Accumulated Local Effects
AU	Arbitrary Unit
AUC	Area Under Receiver Operating Characteristic (ROC) Curve
CART	Classification and Regression Trees
CDF	Cumulative Distribution Function
FN	False Negatives
FNR	False Negative Rate
FP	False Positives
FPR	False Positive Rate
ICE	Individual Conditional Expectation
MLE	Maximum Likelihood Estimate
PD	Partial Dependence
PDP	Partial Dependence Plot
ROC	Receiver Operating Characteristic
TN	True Negatives
TP	True Positives
TNR	True Negative Rate
TPR	True Positive Rate
WHO	World Health Organization

Chapter 1

Introduction

For companies operating in the digital service domain, growing the business in terms of number of customers is evidently an important objective of the business operations, regardless of whether the customers are people or other businesses. This is especially true for digital services for which a customer or a user is paying for accessibility towards the service by e.g. a monthly payment plan, as maximizing the number of users on plans with similar payment structures would be expected to generate larger financial returns. In order to grow the business in terms of the number of users, the number of newly acquired users must undeniably be greater than the number of churned (lost) users, for some considered time period. Hence, acquiring and retaining users are the two fundamental blocks of driving user growth. Interestingly, the cost of acquisition (of users) is often claimed to be around five times greater than the cost of retention [22], showcasing the importance of retaining existing users. Assuming some degree of rationality of consumers, retention could be seen as a proxy towards user satisfaction of the service. Therefore, the task of retaining users, and sub-consequently driving user growth, can be seen as a proxy task of maximizing user satisfaction. Moreover, in a digital era where immense amount of user data is generated and stored by the tech companies offering these services, quantitative methods can be used to predict retention based on user behavioural data generated from a given user, for which proactive measures can be used to positively influence the latent user satisfaction and thus the retention levels. Due to psychological aspects and nature of habits [10, chap. 1], one can suspect that any targeted proactive measures designed to increase the likelihood of user retention for a given individual, are most effective during the *onboarding* phase, meaning when a user first starts interacting with the service.

1.1 Background

Spotify is a client-server audio streaming service, consumed by millions of users across all regions of the world. As of March 2020, the streaming service was constituted by over 50 million songs and 1 million podcast titles, which were consumed by 130 million subscribers and by 286 million monthly active users [25]. In this context, subscribers refer to users who are on one of the Spotify Premium plans, typically through a paid subscription model, and consequently enjoying a larger set of features on the service. Moreover, users can interact with a subset of the functionality of the Spotify Premium offering through an advertisement-based free version, called Spotify Free, which explains why the number of monthly active users surpasses the number of subscribers (as of March 2020). Throughout this work, *premium* will be used as an adjective to describe something in relationship towards the Spotify Premium entity. For example, premium features refer to the features that are exclusive towards the (premium) subscribers, and hence not available for non-subscribers.

As the number of (premium) subscribers is a vital metric for the overall success of the Spotify business, retention is consequently an important metric as well. Moreover, previous research has shown that different types of users interact with a music streaming service differently [17, 33], and may respond differently towards various service offerings and recommendations that are competing of space and exposure. For the purpose of understanding which premium propositions generate the most value for users, it is hence important to understand how they may vary over different user personas or segments.

In a statistical setting characterised by a high-dimensional feature/covariate space of various types, as one can expect to find in an audio streaming service, it is important for any deployed model to handle these rather complex spaces adequately. Although the mathematical folklore "*No free lunch*" theorem [34] typically applies in most statistical or data-mining applications, meaning that one cannot expect to know beforehand which statistical model will perform best for a specific task, some arguments have been made for various boosting methods for decision trees. Leo Breiman famously called the first boosting algorithm for decision trees, known as AdaBoost [11], as the "*best off-the-shelf classifier in the world*". Moreover, in [15, pp. 350-352] gradient boosted models as generalizations of AdaBoost, are reasoned for as near-optimal "off-the-shelf" methods for predictive learning data-mining applications, while still being fairly interpretable.

1.2 Problem formulation and purpose

The studied problem for this work is both predictive and inferential. This work aims to study the effectiveness of boosted tree models as classifiers for user retention, posed as a binary target. Moreover, to stem as a basis for future recommendations and further experiments, it is of interest to infer the statistical relationships that form the predictive mapping. With the Spotify personas as a qualitative mental model, it is also of interest to study how the predictions vary over different user cohorts, assumed to be characterised by the available quantitative representations in the feature space. Furthermore, the onboarding phase of when a user first interacts with the premium offering should is presumed to be a preferred time of any proactive recommendations or intervening effects. Consequently, this work is specifically concentrated around data generated during that initial time period and its implications towards retention. Lastly, an accent is put towards the premium exclusive features for the inferential work, in order to better tailor the experience around the Spotify Premium entity. However, due to confidentiality, these premium exclusive features/covariates will not be identified explicitly from the other studied covariates within this thesis, but arise as a central aspect for internal studies at Spotify based on the same research methodology presented here.

1.3 Delimitations

Although Spotify as a service is available in many countries spanned over the whole world, this work centers around the analysis of North American (American and Canadian) users. Moreover, for the purpose of detecting trends in how users interact with the features without any prior experience with the Spotify Premium offerings, which excludes e.g. previously churned users and users who have utilized trial offers in the past. Furthermore, with recent launches of new service plans, such as Spotify Premium Duo [27] and Spotify Premium Family [28] which might skew the interpretation of retention as a proxy towards price elasticity or user satisfaction, only users who started their subscriptions through standard non-price deducted individual premium plans with monthly price plans were included for the study. In this context, price deduction covers the Spotify Premium Student plan [29] and any campaigns with either a deducted monthly price, or any campaigns composed such that the user obtains the service a longer period of time than in a normal setting.

Lastly, the study was constrained towards users enrolling the premium plan during the time period 2019-12-01 to 2020-01-12, and sub-consequently their (binary) service plan status 45 days after enrollment.

1.4 Outline

The remaining part of this thesis is outlined as follows: Chapter 2 is devoted fully towards the data collection and exploration part, with the purpose of familiarizing the reader with collected data and the meaning of the underlying processes; Chapter 3 covers relevant theories of statistical learning, standard and boosted tree methods for classification and ways of interpreting them; Chapter 4 holds the conducted results and experiments with accompanying interpretations and discussions; lastly, Chapter 5 covers the main conclusions obtained from the master's thesis project. For ease of reading, some complementary material can be found separately in the appendices and are referenced throughout the report.

Chapter 2

Data

2.1 Data characteristics

In an audio streaming service, or in digital services in general, there is generally a rich amount of information that could serve useful or descriptive towards different user attributes. For this thesis, a smaller subset of all the vast number of variables available will be studied. As some variables stem from the same source, hence have similar meanings, the studied variables will be named accordingly. Throughout this work, the terms *variables*, *features*, *predictors* and *covariates* will be used interchangeably.

For confidential reasons, the exact meaning of each variable cannot be disclosed, whereas instead the following identification schema will be followed: GROUP/UNIT/IDENTIFIER. Group refers to a set of features that should have similar meanings (e.g. demographics, behavioural data, listening preferences etc.), unit refers to some type measurement/quantity and identifier is a unique ID for each feature with the same group and unit. The actual groups can not be disclosed for this work, and similarly some of the unit values. For disclosed units, the terminology of Arbitrary Units (AU) will be used. In some cases, unit values such as ATTRIBUTE or BOOL will refer to some arbitrary categorical or binary feature levels. Note also that two features with the same (arbitrary) unit, but different groups, may have different interpretations and should hence in some cases not be compared across groups. For example, there are some fundamental similarities between the features with AU1 as measurement unit across groups, whereas the features with AU2 as measurement unit are generally not comparable between groups.

Furthermore, a smaller percentage of observations were left out from the

analysis as they, from a business perspective, were considered displaying anomalistic behaviours. Due to confidentiality, the exact criteria and percentage of observations left out will not be disclosed in this work. After the (fundamental) outlier removals, the study was left with 32 predictors and a binary target corresponding towards whether the user stayed (retained) on a premium plan or not[1], and 18 516 as the number of observations (one observation per user). However, these 18 516 observations were also subject to (uniform) random sampling without replacement, hence being a subset of the total number of users matching the delimitations of the study. Furthermore, the aggregated fraction of retained users can not be disclosed due to confidentiality.

Moreover, some predictors contain missing values, but where "missing" in this context should for fundamental reasons be interpreted as "not applicable", rather than missing as in unobserved. Therefore, the missing values have an intrinsic meaning and are rather to be seen as a foundation for an additional level in the categorical or binary setting. For the continuous setting (for GROUP1/AU1/4 and GROUP1/AU1/5), these missing values were imputed as 0 on fundamental grounds. A brief summary of each predictor can be found as of Table 2.1.

[1]Technically, the definition of retention in this context is relaxed to regard retention on the Spotify Premium (which includes multiple plans) opposed to retention towards the specific premium plan for which conversions were delimited towards. More concretely, users who are observed to subscribe towards another premium plan after the 45 days of study, are still to be considered as retained premium subscribers. On the contrary, users who are not observed to be premium subscribers after 45 days of study, potentially still as active users on the Spotify Free version, are *not* to be considered as retained premium subscribers.

Feature	**Type**	**Range**	**% Missing values**
GROUP1/REACH/1	Binary	$\{0, 1\}$	5.54%
GROUP1/AU1/3	Continuous	$\mathbb{R}^+$	0%
GROUP1/AU1/4	Continuous	$\mathbb{R}^+$	11.46%
GROUP1/AU1/5	Continuous	$\mathbb{R}^+$	12.78%
GROUP1/AU2/1	Ordinal	$\mathbb{N}$	0%
GROUP2/AU1/1	Continuous	$\mathbb{R}^+$	0%
GROUP2/AU1/2	Continuous	$\mathbb{R}^+$	0%
GROUP2/AU1/3	Continuous	$\mathbb{R}^+$	0%
GROUP2/AU1/4	Continuous	$\mathbb{R}^+$	0%
GROUP2/AU1/5	Continuous	$\mathbb{R}^+$	0%
GROUP2/AU1/6	Continuous	$\mathbb{R}^+$	0%
GROUP2/AU1/7	Continuous	$\mathbb{R}^+$	0%
GROUP2/AU1/8	Continuous	$\mathbb{R}^+$	0%
GROUP2/AU1/9	Continuous	$\mathbb{R}^+$	0%
GROUP2/AU1/10	Continuous	$\mathbb{R}^+$	0%
GROUP2/AU2/1	Ordinal	$\mathbb{N}$	0%
GROUP3/AU1/1	Continuous	$\mathbb{R}^+$	0%
GROUP3/AU1/2	Continuous	$\mathbb{R}^+$	0%
GROUP3/AU1/3	Continuous	$\mathbb{R}^+$	0%
GROUP3/AU1/4	Continuous	$\mathbb{R}^+$	0%
GROUP3/AU1/7	Continuous	$\mathbb{R}^+$	0%
GROUP3/AU1/9	Continuous	$\mathbb{R}^+$	0%
GROUP3/AU1/10	Continuous	$\mathbb{R}^+$	0%
GROUP3/AU2/1	Ordinal	$\mathbb{N}$	0%
GROUP3/AU1/12	Continuous	$\mathbb{R}^+$	0%
GROUP4/AU1/1	Continuous	$\mathbb{R}^+$	0%
GROUP5/BOOL/1	Binary	$\{0, 1\}$	0%
GROUP5/ATTRIBUTE/2	Categorical	$\{1, \ldots, 7\}$	0.63%
GROUP6/AU2/1	Ordinal	$\{1, \ldots, 31\}$	0%
GROUP6/AU2/2	Ordinal	$\mathbb{N}$	0%
GROUP6/ATTRIBUTE/1	Binary	$\{0, 1\}$	0%
GROUP6/BOOL/1	Binary	$\{0, 1\}$	0%

Table 2.1: Characteristics of the used features for this study. The type regards different types of features, such as categorical, continuous or binary features. The range regards the outcome space for each underlying random variable. Note for the ordinal case, corresponding towards AU2, that some variables have certain constraints which restrict their ranges.

2.2 Empirical distributions

For the extracted data, two immediate observations are striking. Firstly, many AU1 features have a relatively low *reach*, meaning discovery by a user - mathematically in this context that the AU1 measure is strictly greater than zero. This could suggest a discovery problem, where a considerable fraction of users have not, purposely or not, interacted during their first days with a certain feature offered through the subscription service. Figure 2.1 shows reach for the different features that regard AU1 as unit measure, from which one can note that the reach for certain features are as low as 5.9%. A low reach does not necessarily mean a disinterest in the specific offering, but could might as well signal low visibility or exposure of the feature itself, as various features compete against each other for visibility.

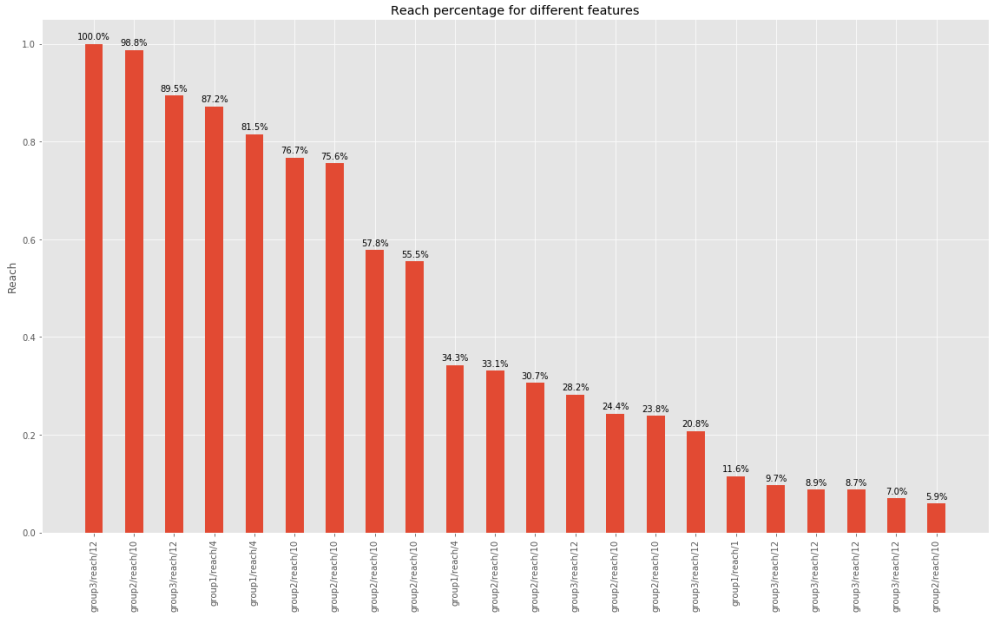

Figure 2.1: Percentage of samples for each feature with AU1 as unit with strictly positive values, i.e. AU1 $>$ 0 for some feature GROUP/AU1/ID.

Secondly, a rapid decay in the empirical distributions for AU1-features was noticed. Figure 2.2 suggests that most features seem to experience a so called *high infant mortality*, meaning high densities for smaller values of AU1, and where the rate of change in probability density decreases for larger AU1 values. One possible explanation on qualitative grounds could be a discovery phase, where newly acquired users are exploring various features and hence discarding some features that does not compete as well as other features for

their interest. The same phenomena seems to hold for the other groups with AU1 as unit, and not just GROUP1 as showcased by Figure 2.2. Similar plots for the other groups can be found as Figure B.1 - Figure B.8 as of Appendix B.

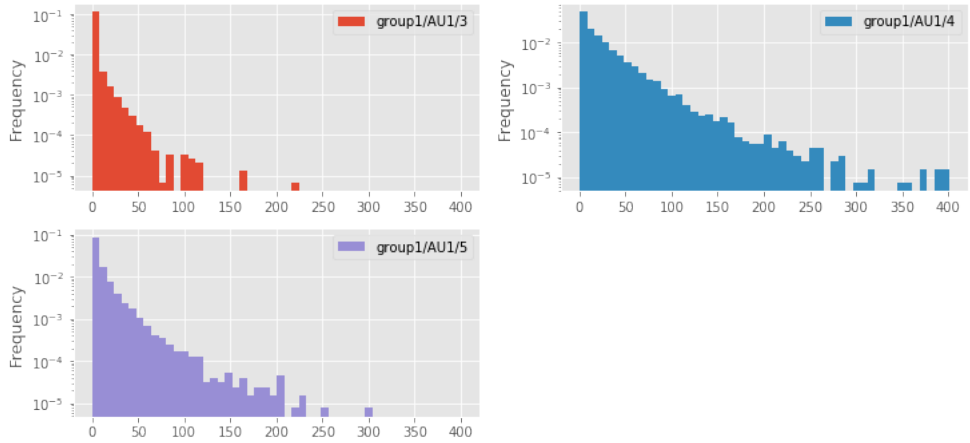

Figure 2.2: Histograms for AU1 features in group 1 on $\log_{10}$-scale for the relative frequency. Note that the AU1 measures have been scaled by some positive factor of $\alpha \ll 1$ to preserve confidentially and make the range smaller.

To reduce the range of the AU1 features and therefore easier discovery of good split candidates for the studied tree methods (see section 3.2), a logarithmic transform was applied prior to any model fitting. Moreover, as many observations are zero-valued for these features, the specific logarithmic function $\log(x+1)$ was used, to ensure a positive and lower-bounded range. Interestingly, Figure 2.3 shows two distinct clusters or mixtures for each empirical logarithmic distribution. The first one as the high-density zero bar, which indirectly captures the reach, and the positive samples captured as a negatively skewed distribution.

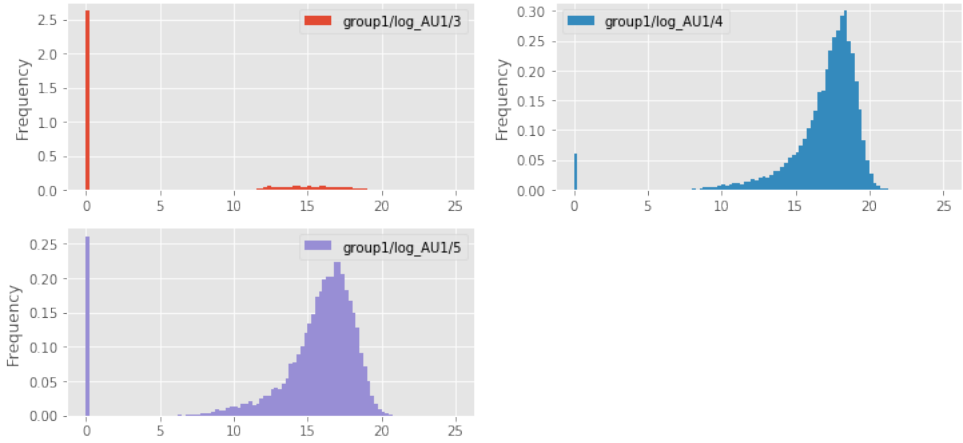

Figure 2.3: Histograms for the AU1 features in group1 (scaled by some factor β) on $\log(x+1)$ scale.

Looking at the pairwise Pearson-Spearman correlations, one can note from Figure 2.4 that most features seem to be non-heavily linearly correlated, with a few exceptions such as the Pearson-Spearman correlation coefficient of 0.8 between GROUP3/LOG_AU1/8 and GROUP6/AU2/1.

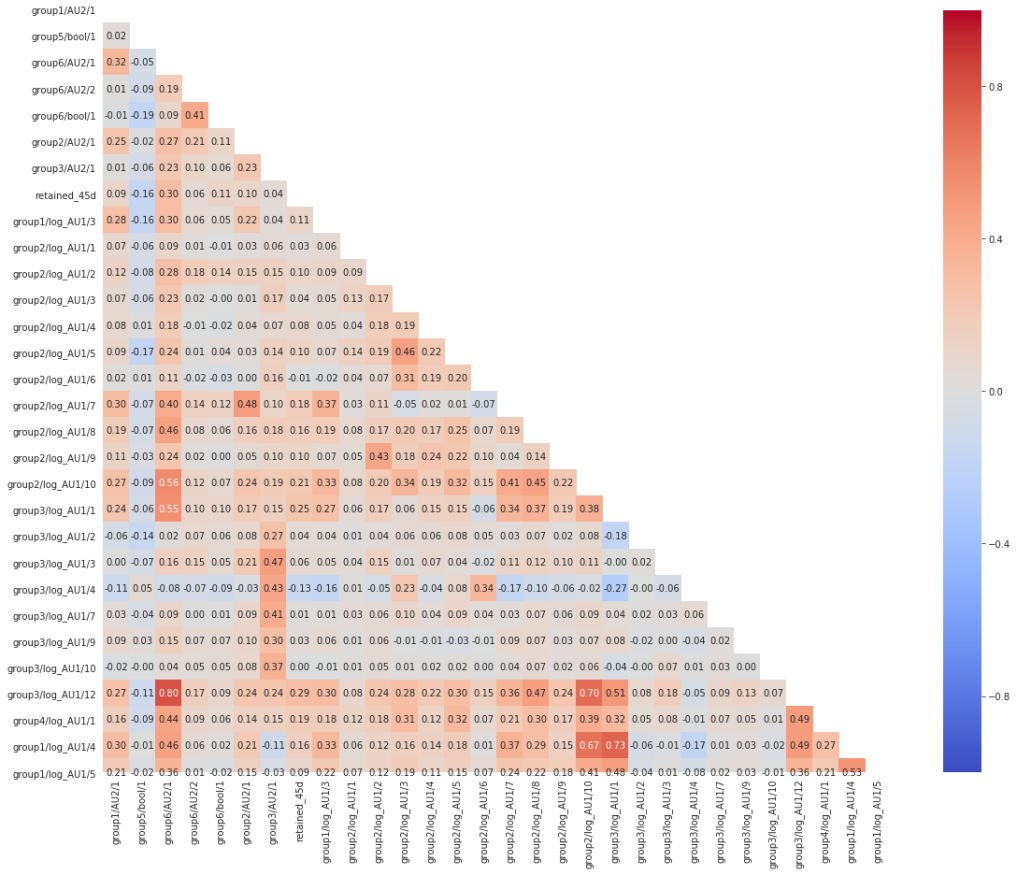

Figure 2.4: (Pearson-Spearman) Correlation matrix for all features, rounded to the nearest two decimals.

2.3 Weibull distribution fitting

The Weibull distribution is one common parametric model in the field of reliability studies and survival analysis [2, p. 209], which deals with questions that regard the expected life-cycle of a product or individual, e.g. when in time a machine is expected to break down. In a customer retention problem setting, a similar view can be utilized where each user (customer) can interact with certain features offered through the service in a limited time period. In a service with many features, it would deem natural to assume that different features have different appeal from the user audience, potentially also conditioned on previous exposure and segments of the users.

For a random variable X, with the positive reals as outcome space, and

parameters $\lambda, k > 0$, the Weibull density function $f(x)$ is given as

$$f(x; \lambda, k) = \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}, \tag{2.1}$$

where the k parameter is referred to as the *shape* and λ as the *scale*. The former parameter has some interesting implications, firstly that the Weibull distribution collapses into an exponential distribution (with expectation λ) for the special case of $k = 1$. If X is interpreted as a time-to-event random variable, then the failure rate is said to be subject to a high infant mortality for $k < 1$ [19], meaning that the failure rate decreases over longer measures of engagement (measured by AU1). Hence, a feature engagement distribution, where each sample corresponds to a "failure", the magnitude of k describes the level of infant mortality for the range of $k \in (0, 1)$, assuming a Weibull model.

To assess the fit of a Weibull model, a popular graphical tool in the field of reliability engineering and survival analysis [18], is the so called Weibull plot, which utilizes the linearity of the model parameters on a log-log scale. This can be seen algebraically by manipulating the cumulative density function $F(x)$

$$\begin{aligned} F(x) &= 1 - e^{-\left(\frac{x}{\lambda}\right)^k} \Leftrightarrow \\ \log(1 - F(x)) &= -\left(\frac{x}{\lambda}\right)^k \Leftrightarrow \\ \log\left(\log(1 - F(x))\right) &= -k \log\left(\frac{x}{\lambda}\right) \Leftrightarrow \\ \log\left(\log(1 - F(x))\right) &= -k \log x + k \log \lambda\,, \end{aligned} \tag{2.2}$$

and hence if a random variable $X \sim \text{Weibull}(\lambda, k)$, then $\log\left(\log(1 - F(x))\right)$ should be linear in $\log(x)$ for realizations x of X. Figure 2.5 shows an example of the visual diagnostics for one of the AU1 features (specifically GROUP1/AU1/3). From the fitted CDF itself, it seems to suggest an adequate fit. However, the log-log CDF and the Weibull plots suggest some an overestimate of the probability density for lower AU1 values. One can compare the bottom left Weibull plot with the one for simulated data from a true Weibull distribution (in this case with the same parameters). Evidently, the Weibull plot which corresponds towards the empirical data is not completely linear, and the Weibull distribution serves at most as a simplified model of the decaying characteristics of the AU1 features. Figure B.9 - Figure B.30 of Appendix B showcases similar behaviours, some with more acceptable fits

and some with worse. Interestingly, as can be seen from Table 2.3, all fitted Weibull distributions for these features has estimates of $k < 1$, indicating the high infant mortality effect mentioned earlier in this section. Under a Weibull model, comparing the k-values for different distribution fits (assuming $k < 1$ for all fits) could indicate the overall appeal of first engaging with a newly discovered feature. One should however not form too strong conclusions around these estimates, or their interpretations, as the distribution fits are acceptable at best, especially for smaller values of AU1.

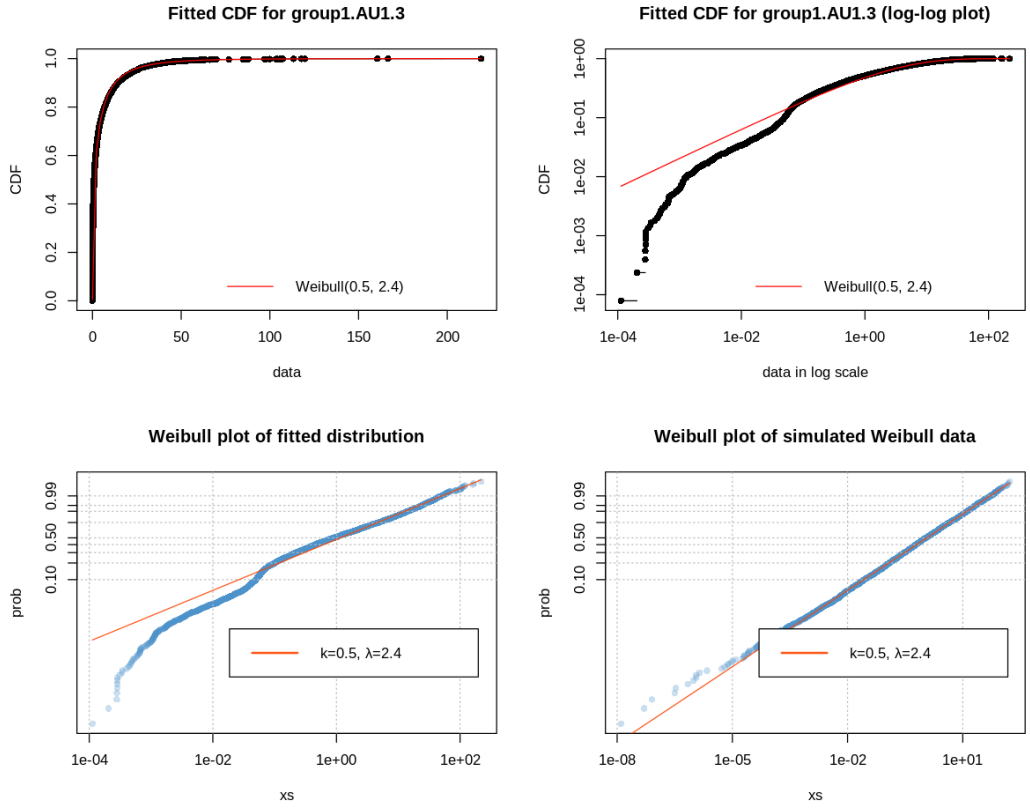

Figure 2.5: Weibull fitting towards the positive values of GROUP1/AU1/3. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale in order to spot deviations easier. The bottom panels show the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

Feature	k_{MLE}	**Wald 95% CI**
GROUP1/AU1/3	0.499	$[0.489, 0.509]$
GROUP1/AU1/4	0.700	$[0.691, 0.708]$
GROUP1/AU1/5	0.605	$[0.597, 0.612]$
GROUP2/AU1/1	0.498	$[0.475, 0.521$
GROUP2/AU1/2	0.503	$[0.491, 0.514]$
GROUP2/AU1/3	0.523	$[0.517, 0.530]$
GROUP2/AU1/4	0.475	$[0.465, 0.486]$
GROUP2/AU1/5	0.496	$[0.488, 0.503]$
GROUP2/AU1/6	0.497	$[0.487, 0.507]$
GROUP2/AU1/7	0.633	$[0.623, 0.642]$
GROUP2/AU1/8	0.522	$[0.515, 0.529]$
GROUP2/AU1/9	0.512	$[0.502, 0.522]$
GROUP2/AU1/10	0.723	$[0.715, 0.731]$
GROUP3/AU1/1	0.809	$[0.800, 0.819]$
GROUP3/AU1/2	0.459	$[0.442, 0.477]$
GROUP3/AU1/3	0.522	$[0.509, 0.535]$
GROUP3/AU1/4	0.548	$[0.537, 0.560]$
GROUP3/AU1/7	0.465	$[0.447, 0.482]$
GROUP3/AU1/9	0.606	$[0.581, 0.632]$
GROUP3/AU1/10	0.413	$[0.399, 0.428]$
GROUP3/AU1/12	0.864	$[0.854, 0.874]$
GROUP4/AU1/1	0.590	$[0.583, 0.597]$

Table 2.2: Maximum Likelihood point estimates for k and corresponding 95% Wald confidence intervals for the parameter estimates.

Chapter 3

Theory

3.1 Supervised statistical learning

The *supervised (statistical) learning problem* is characterised by the presence of an outcome or target variable to assist the learning (fitting) procedure of finding a predictive model or function approximation $f_\theta : x \mapsto y$, where x is a set of features/covariates/predictors and y the targets [15, p. 9]. Moreover, the function f is parameterized by its model parameters θ, to uniquely define f within a class of functions $\mathcal{F}$. For example, a predictive model, say a linear regression model, is uniquely defined within the class of all linear regression models by its assigned weights to each feature.

Remark. *Not all statistical learning problems fall under the supervised learning category. For instance, in the unsupervised learning problem, where no target measures are available during training, the task is rather to infer clusters or similarities between different data points [15, chap. 14].*

Typically, supervised learning tasks are divided into two separate categories based on the inherent meaning of the target variable. Problems with the appearance of a numerical or continuous target variable are referred to as regression problems, whereas tasks constituted by a categorical or discrete target variable are referred to as classification problems. This distinction proves useful when selecting an appropriate algorithm or family of functions for a given supervised learning problem. For example, a linear regression model might be adequate for the regression problem of predicting house prices as a function of square meters, but would not be applicable for classifying emails as spam. On the contrary, a logistic regression model would qualify as a strong candidate to consider for the latter task, but not for the former.

Moreover, an important aspect of supervised statistical learning problems is the distinction between seen and unseen data points. The *training data* refers to data points which have been used to guide the model during the learning procedure, with the objective of generalizing towards unseen data, which also will be elaborated on more mathematical terms later on in this section.

When determining f_θ for a given supervised learning problem, conditioning on the choice of some class of functions $\mathcal{F}$ (the model class), it can be viewed as a minimization problem, where

$$\theta^* = \arg\min_\theta \mathcal{L}(\theta)\,, \tag{3.1}$$

for some *objective function* $\mathcal{L}(\theta)$. The purpose of the objective function is to quantify an associated cost or error for the model parameters θ, and indirectly the predictions $f_\theta(x)$, evaluated against the true targets y. Therefore, different functions in $\mathcal{F}$ can be compared relative each other by their respective losses, hence justifying Equation (3.1).

Intuitively, the objective function should be such that it is minimized when $f_\theta(x)$ and y are *consistent*, and penalized for larger inconsistencies. One realizes however that there are multiple ways of quantifying consistencies, which naturally will affect the solution by minimizing the objective function, as of Equation (3.1). Recalling the linear regression problem, the standard choice of objective function would be the sum of squares, between the predictions and targets, arising from a parametric assumption about $N(0, \sigma^2)$ - normality for the residuals. However, other choices such as the sum of absolute errors, between the predictions and targets, could also be used and would implicitly impose other meanings towards the learning objective.

Generally, the objective function can be seen as the *expected prediction risk* [15, p. 220] over the data distribution P, from which (x, y) are realizations of, and an error-/loss function L that quantifies the divergence/error between the predictions $f_\theta(x)$ and targets y. Hence,

$$\mathcal{L}(\theta) = \mathbb{E}_{(x,y)\sim P}\left[L(f_\theta(x), y)\right]\,, \tag{3.2}$$

which is intractable to minimize directly as P is unknown. Instead, the objective of Equation (3.2) can be minimized empirically by the *empirical predictive risk* $\tilde{\mathcal{L}}(\theta)$, hence

$$\theta^* \approx \arg\min_\theta \tilde{\mathcal{L}}(\theta) = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \left[L(f_\theta(x_i), y_i) \right] , \tag{3.3}$$

where N is the number of samples in the training dataset. If the training dataset can safely be assumed to be representative towards the distribution P, then the approximation as of Equation (3.3) can be expected to be good.

One way of testing the assumption about homogeneity is to holdout some partition of the dataset for the fitting procedure, and later test the generalization capability of the fitted model by assessing the fit on unseen data. This partition is often referred to as the *test data*, and its corresponding empirical predictive risk (evaluated only for samples in the test dataset) is called the *test error* or test prediction error. For symmetry, the empirical predictive risk evaluated for the training dataset will be referred to as the *training error* (or training prediction error). Moreover, as discussed more in detail in the coming subsection, the approximate solution as of Equation (3.3) is also affected by the complexity of the model.

3.1.1 Bias-variance trade-off and model complexity

Generally, when distilling the expected prediction risk, three important terms will appear, namely an *irreducible error*, the bias and the variance. This section is devoted towards explaining the meaning of each of them and how they affect the supervised learning procedure.

Starting with the irreducible error, which arises from stochasticity in the target variable. For instance, consider the task of predicting the number of bicycle rentals by the temperature and rainfall levels. Obviously, the number of bike rentals will inherently depend upon many other exogenous effects (e.g. whether a given day is a weekend or not, local competition etc), that limits the performance of any predictive model for this task. Philosophically, this limitation corresponds to the classical aphorism *"All models are wrong, but some are useful"* by George Box. Mathematically, one can view the irreducible error as the lower bound of the expected prediction risk, i.e.

$$\mathcal{E}_{\text{irr}}(\mathcal{F}) = \inf_{f_\theta \in \mathcal{F}} \mathbb{E}_{(x,y)\sim P} \left[L(f_\theta(x), y) \right] , \tag{3.4}$$

where $\mathcal{E}_{\text{irr}}(\mathcal{F})$ is the irreducible error, conditioned on $\mathcal{F}$ as the set of all models considered. Alternatively, one might view $\mathcal{F}$ as the hypothesis space for some true underlying mapping. Intuitively, the irreducible error, conditioned

on some $\mathcal{F}$, becomes smaller in the presence of (endogenously) strong predictors, as the mapping between the predictors and the target becomes (nearly) deterministic, which consequently yields good generalization capabilities for unseen data. To illustrate the contrary, recap the bike rental problem, with rainfall and temperature as the predictors. If the two predictors do not explain a considerable degree of the number of rentals, the exogenous effects (noise) will be larger, therefore yielding a higher irreducible error.

Moreover, the remaining source of errors can be identified as the *approximation error* and the *estimation error*, also referred interchangeably to as the bias and variance (of predictors) respectively [9]. The first one is the squared difference of the true model average and the expectation of the estimator. While a low bias is obviously desirable, one should also consider the second term of the estimation error, namely the variance of the estimator. The variance corresponds to vast changes in the model parameters for changes in the dataset, suggesting overfitting towards the training data. Therefore, the ideal model would have a low bias and a low variance. In practice however, one typically observes these two terms as functions of the model complexity, where the bias typically decreases for increased model complexity, whereas the opposite effect can be said for the variance [15, p. 37]. Mainly, these two errors arises and changes by the family of models to consider (e.g. linear models), the dataset $(x, y) = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ and potentially the fitting procedure as well (if no closed-form solution exists).

In order to acquire a more mathematical intuition of the relationships between the irreducible error, the bias and the variance, consider a setting with the squared error as the loss function L, yielding an expected prediction risk as (c.f. Equation (3.2))

$$\mathcal{L}(\theta) = \mathbb{E}_{(x,y)\sim P}\left[(f_\theta(x) - y)^2\right] , \tag{3.5}$$

where $\mathbb{E}$ regards the expectation with respect to distribution P, form which (x, y) are realizations from. Hence, $\mathbb{E}$ serves in this context as a shorter notation for $\mathbb{E}_{(x,y)\sim P}$. With the squared error as the loss function, Equation (3.5) can be rewritten as

$$\mathbb{E}\left[(f_\theta(x_0) - y)^2\right] = \operatorname{Var}(\epsilon) + (f(x_0) - \mathbb{E}\left[f_\theta(x_0)\right])^2 + \tag{3.6}$$

$$\operatorname{Var}(f(x_0) - f_\theta(x_0)) , \tag{3.7}$$

evaluated at some point x_0. In this context, $f(x)$ denoted the true underlying model, separated from the measurement error or noise ϵ - assumed to have

a zero-valued expectation, hence $y = f(x) + \epsilon$. Moreover, $f_\theta(x)$ is assumed to be fitted towards a finite set of realizations (x, y) from P (training data), therefore stochastic. The full derivations for Equation (3.6) are available in Appendix A as Lemma (A.1) with accompanying proof. From the error decomposition as of Equation (3.6), it is noticed that the first term corresponds to noise, which is irreducible for any $f_\theta \in \mathcal{F}$ (see also Equation (3.4)). The second term is the squared bias between the expectation of the true mean $f(x_0)$ and the estimate $f_\theta(x_0)$. Lastly, the third term is identified as the variance of the difference between the real model and the fitted one, and depends on how well the regression fit towards the training data generalizes for unseen data.

Consequently, a trade-off between the bias and variance arises when determining the model complexity, which is depicted visually in Figure 3.1. Models with low bias and high variance can be said to overfit towards the training data, i.e. emphasizing too much on local noise apparent in the training data, and hence fail to generalize well for unseen data. On the other hand, models with high bias and low variance underfit, meaning that they fail to capture relevant structure in the feature space, and therefore also fail to generalize well for unseen data.

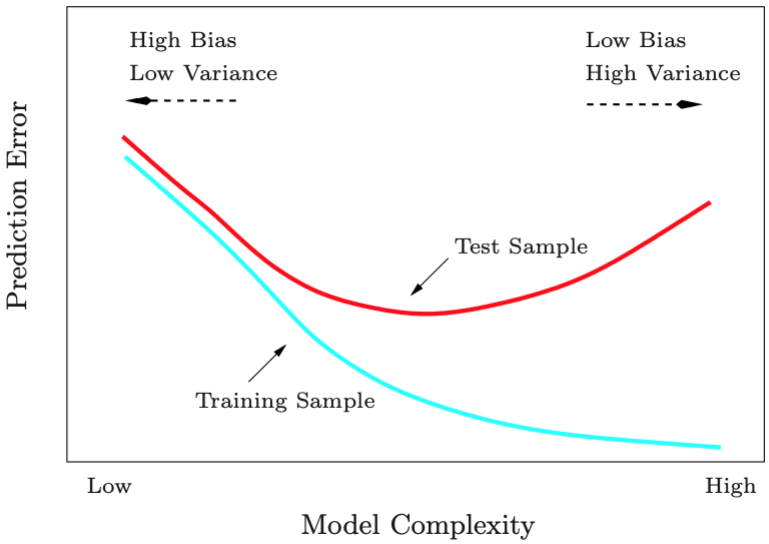

Figure 3.1: Test prediction error (red) and training prediction error (aqua) as a function of model complexity. The test- and training prediction error regard the empirical predictive risk (see Equation (3.3)) evaluated on the test and training dataset respectively. Picture taken from [15, p. 38].

3.1.1.1 Model complexity and regularization

So far as of Section 3.1.1, the existence of a general trade-off between bias and variance, which should delicately be balanced to achieve the best generalization capabilities for a predictive model. However, the term model complexity can mean different things depending on the inherent structure of a model. For example, it might make sense to talk about complexity measures in terms of the number of used features and magnitude of the effect parameters for the case of linear regression. For a specific model structure, the particular parametric assumption could give raise to other useful complexity measures. For instance, for the case of polynomial regression models, the degree of the polynomial could be used as a complexity measure.

One way of combating high model complexities, and consequently high variances, is to add a penalty term towards the loss function for some predefined complexity measure. In other words, the expected empirical risk function would be changed towards (c.f. Equation (3.2)),

$$\mathcal{L}(\theta) = \mathbb{E}_{(x,y)\sim P}\left[E(f_\theta(x), y) + \Omega(\theta)\right] , \tag{3.8}$$

where $\Omega(\theta)$ is some complexity measure of the model parameters θ. Hence, it also follows from Equation (3.8) that the empirical predictive risk obtains as (c.f. Equation (3.3))

$$\tilde{\mathcal{L}}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left[L(f_\theta(x_i), y_i) + \Omega(\theta) \right] . \tag{3.9}$$

Equation (3.9) generalizes other popular regularization techniques such as ridge regression [16] and lasso regression [32]. From a bias-variance standpoint, the introduction of the complexity term introduces a bias, but for the benefit of (generally) reducing the variance [15, p. 224].

Moreover, besides talking about complexity within a class of models, the framework also proves useful to compare different families of models against each other. For instance, simpler models, such as the linear regression model, impose some assumptions on the inherent workings of the model, in the example of linear regression particularly, the features are assumed to interact linearly and in an additive manner. Consequently, as typically the mapping between the features and the target are nonlinear and non-additive in x [15, p. 139], the linear regression will have a high bias compared to more expressive nonlinear models, such as complex mixture models, which might express the true underlying mapping more adequately. On the other hand, as understood from the bias-variance trade-off viewpoint, the former model will have a lower variance than the latter. Hence it appears useful to apply the bias-variance framework not only for model parameter fitting within a class of models, but also to compare families of models against each other.

3.1.1.2 K-fold Cross Validation

Another way of combating overfitting issues, while at the same time being more efficient with using data both for training and validation, is by various cross-validation techniques. The trade-off is mainly the computational aspects of further required fitting procedures, especially for cross-validation methods such as the leave-one-out method. More computationally efficient cross-validation techniques include the K-Fold Cross-Validation [15, sec. 7.10] technique (assuming $K \ll N$). The high-level idea is to split the dataset into K approximately equal sized folds, thereafter taking turns on using each partition as the validation set (kept unseen) while using the remaining $K - 1$ folds for training. The last step is then to average the different estimations, in this case the empirical predictive risk $\tilde{\mathcal{L}}(\theta)$, as

$$CV(\tilde{\mathcal{L}}(\theta)) = \frac{1}{N}\sum_{i=1}^{N}\left[L\left(f_\theta^{(k)}(x_i), y_i\right) + \Omega(\theta)\right], \tag{3.10}$$

where $CV(\tilde{\mathcal{L}}(\theta))$ denotes the (K-fold) cross-validated empirical predictive risk, and $f_\theta^{(k)}$ the fitted model/function after holding the k^{th} ($k \in \{1, \ldots, K\}$) data fold unseen.

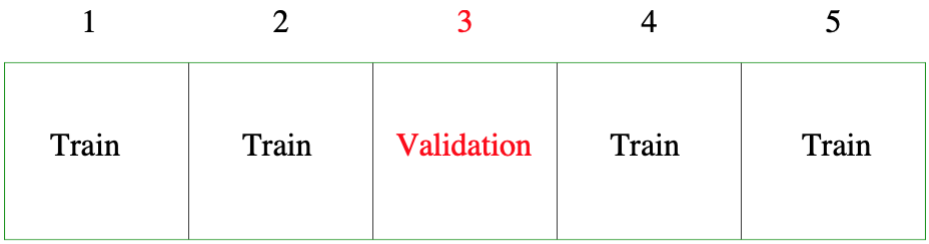

Figure 3.2: Example of a partition of the dataset with 5 folds, where each fold take turns on being the validation set (at this snapshot the third fold). Picture taken from [15, p. 242].

3.1.1.3 Early stopping

Another method to prevent overfitting is to introduce the concept of early stopping as a stopping criterion. As Figure (3.1) suggests that the prediction error on the held-out data reaches a plateau before the corresponding prediction error on the training dataset does. Therefore, a simple strategy to avoid overfitting is to stop the learning process at the lowest point of the test error curve [5, p. 259]. In practice, this can be done by defining a stopping criterion, such as when the prediction error on the test dataset does not decrease over a given number of training iterations. Other similar criterion, e.g. comparing the test error over rolling windows, may also be used [23].

3.1.2 Performance metrics for binary classifiers

Even in the binary domain of classification problems, there are various ways of quantifying the performance of a classifier. The fraction of correctly classified instances, regardless of the target values, of the total number of instances is known simply as the accuracy of a classifier, and can be a reasonable choice of performance metric under the premise that our (two) classes are somewhat

well balanced.

To illustrate the potential pitfalls of the accuracy metric when imbalance of our target variable is present, imagine a financial fraud detection problem with binary outcome, where we for the purpose of illustration could assume that a great majority of the financial transactions are not at fault. One overly simplified way of achieving a great accuracy for our classifier would be to have it, in a deterministic fashion, return negative values (non-fraud) for each instance, independent of any available covariates. Due to the severe class imbalance of the target variable, and by implicitly penalizing type I and type II errors equally, the deterministic classifier would achieve a great accuracy score while at the same time fails miserably at detecting any frauds, which fundamentally makes the classifier useless.

3.1.2.1 Balancing Type I and Type II errors

In the binary classification domain, it is evident that four different scenarios can occur for classification of each sample: true positives, false positives, true negatives and false negatives. Moreover, as only the total number of evaluated samples is fixed, the outcome counts can be summarized by a 2x2 contingency table with a Multinomial sampling strategy [3, p. 25], as depicted in Table 3.1.

	True values	
Predicted values	Positive	Negative
Positive	TP	FP
Negative	FN	TN

Table 3.1: Contingency table of possible binary classification results. TP stands for true positive counts, FP for false positive counts, FN for false negative counts, and TN for true negative counts.

Let x be a feature vector in some feature space X, y its corresponding target, and let $f(x) \in \{1, 2\}$ denote the predicted class of the binary classifier $f(x)$, $Y \in \{1, 2\}$ the true values, where 1 corresponds towards the positive class, and 2 towards the "negative". In order to study the effect of Type I and Type II errors when evaluating a given classifier f, it is of great interest to infer the joint, marginal and conditional probabilities of $f(x)$ and y.

Remark. *As other literature, e.g. [15], tend to use* $\{-1, 1\}$ *as the notation for the binary target. Using the encoding* 2 *as notation for the negative class might not be intuitive, but adds for easier notation and interpretation when using contingency tables by relating towards the row- and column indices.*

Assuming a sample of n_{++} predictions compared against the true target values, has been represented as a contingency similar to Table 3.1, the task remains as inferring the joint probabilities π_{ij}, as well as the marginal probabilities π_{i+} and π_{+j}, for all $i, j \in \{1, 2\}$ (row and columns indexing starts from 0). Maximum likelihood estimates of π_{ij}, π_{i+} and π_{+j} (see Lemma (A.2) of Appendix A and its accompanying proof) obtains as

$$\hat{\pi}_{ij} = \frac{n_{ij}}{n_{++}}\,, \tag{3.11}$$

$$\hat{\pi}_{i+} = \frac{n_{i+}}{n_{++}}\,, \tag{3.12}$$

$$\hat{\pi}_{+j} = \frac{n_{+j}}{n_{++}}\,. \tag{3.13}$$

Lastly, the conditional probabilities $\pi_{i|j} = P(f(x) = i \mid y = j)$, for some $i, j \in \{0, 1\}$, estimate as

$$\hat{\pi}_{i|j} = \frac{\hat{\pi}_{ij}}{\hat{\pi}_{+j}} = \frac{n_{ij}}{n_{+j}}\,. \tag{3.14}$$

As the true underlying rates of true positives (TPR), false positives (FPR), true negatives (TNR) and false positives (FPR), correspond to the different permutations of $\pi_{i|j}$, these rates can be estimated using Equations (3.11)-(3.14) as

$$\widehat{TPR} = \hat{\pi}_{1|1} = \frac{TP}{TP + FN}\,, \tag{3.15}$$

$$\widehat{FPR} = \hat{\pi}_{2|1} = \frac{FP}{TP + FN}\,, \tag{3.16}$$

$$\widehat{TNR} = \hat{\pi}_{2|2} = \frac{TN}{FP + TN}\,, \tag{3.17}$$

$$\widehat{FNR} = \hat{\pi}_{1|2} = \frac{FP}{FP + TN}\,. \tag{3.18}$$

Recalling the posed problem domain of financial fraud detection from Section 3.1.2, the classifier that simply outputs negative values for all financial

transactions, regardless of what information they might carry, will have a perfect true negative rate of 1, but concurrently, assuming the existence of at least one fraud in the evaluation data, have a zero valued true positive rate.

Formally, the output of the classifier, often a "raw"- or pseudo probability of a positive event, can be distinguished from a policy that operates on this raw probability and return the predicted class, hence a function $D : [0, 1] \to \{1, 2\}$. Practically, the space of decision functions is restricted towards indicator functions that thresholds the predicted class probability, e.g.

$$D(x) = \begin{cases} 1 & \text{if } x \geq 0.5, \\ 2 & \text{otherwise,} \end{cases}$$

where x is the predicted probability of belonging to the positive class for a given observation. It appears obvious that by increasing the threshold, $P(f(X) = 1)$ decreases and vice versa. In order to account for this trade-off when evaluating the predictive performance of a given classifier, it is often of interest to examine how the TPR and FPR changes as a function of the threshold for the decision function, which is graphically examined by a *ROC curve* (receiver operating characteristic curve) by plotting the estimated TPR on the y-axis, and the estimated FPR on the x-axis, for varying thresholds. As a baseline, it is easy to see that the true positive rate will equal the false positive rate in the case of a fully random classifier, i.e. $f(X)$ being independent of Y, hence implying that $\hat{\pi}_{1|1} = \hat{\pi}_{1|2}$.

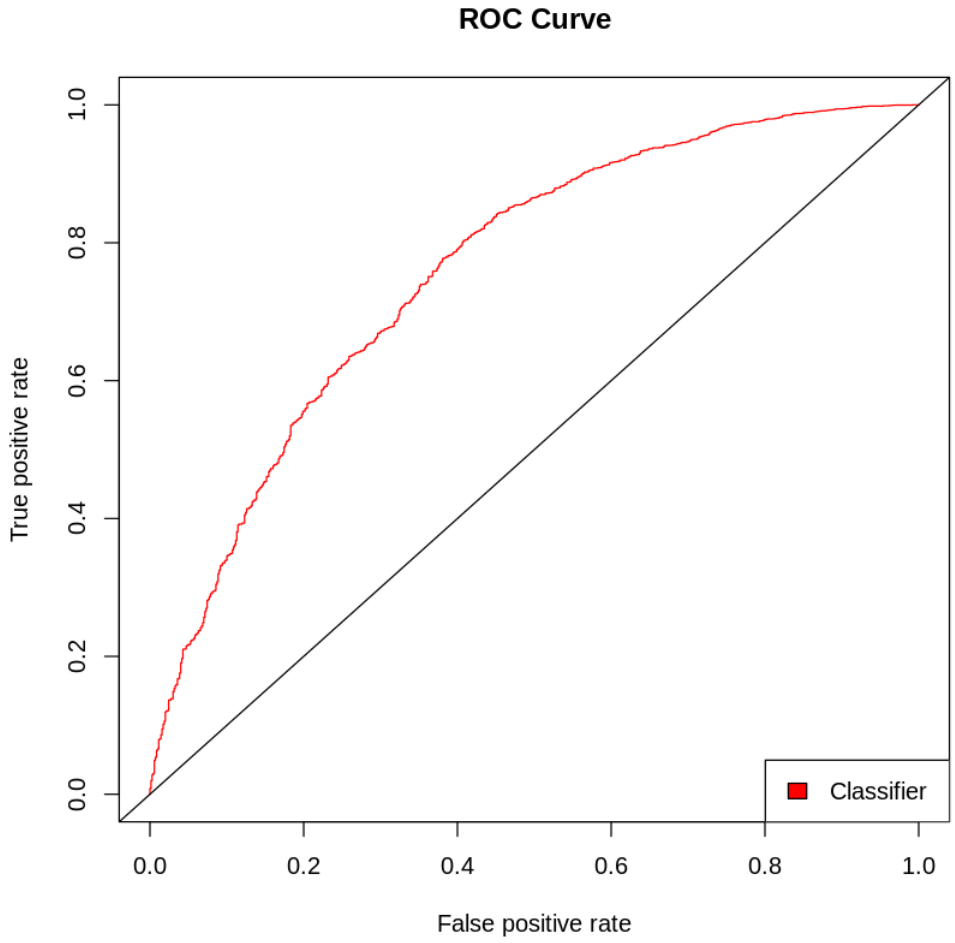

Figure 3.3: Example ROC curve for some binary classifier (red), where the diagonal line corresponds to the ROC curve of a fully random classifier. Higher values for the true positive rate, and lower values for the false positive rate are better.

With the ROC curve in place, it makes also sense to talk about the area under the ROC curve, abbreviated as AUC. Clearly, higher AUC values corresponds to a high (estimated) TPR and a low FPR for most thresholds. Moreover, as the baseline of random guessing would yield an AUC value of 0.5, which yields a lower bound for what can be considered reasonable for any binary classification task.

3.2 Classification and Regression Trees

The core principle of a tree model or decision tree can be described in layman terms - by asking a set of questions with pre-set alternatives in a hierarchical manner, to eventually arrive at an answer/prediction. In computer scientific terms, this process can be thought of as traversing a tree with a question at each node until a leaf (end node) is reached, for which a corresponding prediction is obtained, e.g. as in Figure 3.4.

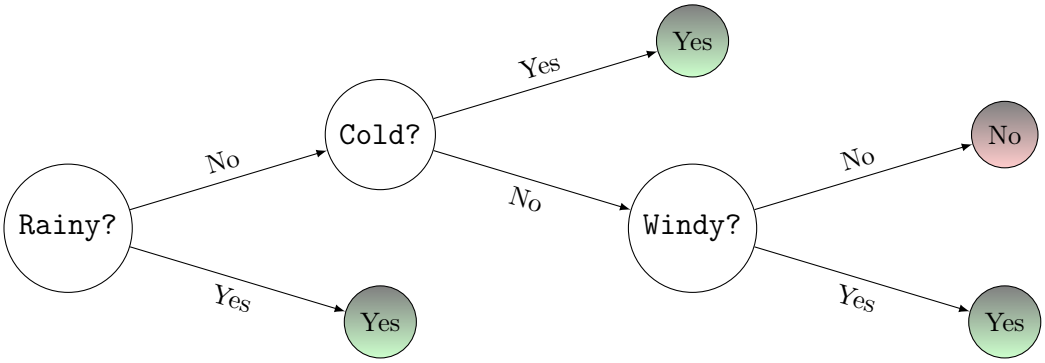

Figure 3.4: Example of a high-level classification tree structure, visualized from left to right, where the task is to predict whether a jacket would be needed given the current whether status. It is noticeable how the intrinsic logic of the classifier arguably follows decision patterns that could resemble how a human would approach answering the same question. Moreover, questions such as *Rainy?*, *Cold?*, and *Windy?*, are subjective and would instead be posed quantitatively from available features by the model, possibly with a similar interpretation.

A popular method for creating such trees, called CART (Classification and Regression Trees), was outlined in [6]. Mathematically, the interpretation of the modelling approach is to divide the feature space into disjoint partitions $R_1, \ldots, R_T$, represented as T leafs of the tree after recursively introducing binary splits in the feature domain. Regardless of whether the prediction problem is a regression or a binary classification problem, for a feature vector x_i in feature space X that the model is assumed to be trained upon, the prediction $f(x_i)$ is given as

$$f(x_i) = \sum_{j=1}^{T} w_j \cdot \mathbb{1}(x_i \in R_j)\,, \tag{3.19}$$

for some constants $w_1, \ldots, w_T$, and where $\mathbb{1}(\cdot)$ denotes the indicator function. The only restriction the binary classification domain imposes is that $w_j \in [0, 1]$ to be interpreted as the probability of a positive event, conditioned on X. Comparing Equation (3.19) with Figure 3.4, it can be noted that the former generalizes the deterministic nature of latter in the binary classification domain, to allow for different policies based on different thresholds, as discussed in Section 3.1.2.1. In other words, Figure 3.4 would be generalized to output raw probabilities of the positive event, in this setting the probability that a jacket will be needed, rather than also imposing a policy of whether to actually bring a jacket.

Remark. *Equation (3.19) can be extended for multi-class classification problems by introducing a second index towards w to denote the conditional probability towards each class, e.g. making w into a matrix, and by constraining that each row-/column vector (depending on how the matrix is composed) equals* 1*.*

3.2.1 Fitting the trees

Although the intrinsic hierarchical binary split (see Figure 3.4) nature of CART models make them highly interpretable fundamentally and mathematically alike, the fitting procedure of determining the optimal tree structure with its corresponding w vector (see Equation (3.19)) is generally computationally infeasible [15, p. 307]. In layman terms, a tree structure refers to which questions to ask at which time by the model and how to utilize that information for predictive purposes. Mathematically it can be thought of as a function $f \in \mathcal{F}$, where $\mathcal{F}$ being the class of CART functions, is determined by its feature partitions $R = \bigcup_{j=1}^{T} R_j$ and weights $w = (w_1, \ldots, w_T)^T$.

To infer the tree structure f, i.e. fitting the tree model, a greedy approach can be utilized. The idea is to start with all data x, to recursively find optimal features and threshold values to split on. Let $\theta_m = (j, t_m)^T$ denote a proposed split for node m (determined by feature partition R_m) and its corresponding feature subset $x^{(m)} = \{x : x \in R_m\}$. Consider now a split t_m for feature $x_j^{(m)}$, where the left and right subsets of x are denoted $R_{\text{left}}(\theta)$ and $R_{\text{right}}(\theta)$, obtains as

$$\begin{aligned} R_{\text{left}}(\theta_m) &= \{x^{(m)} : x_j^{(m)} \leq t_m\}, \\ R_{\text{right}}(\theta_m) &= \{x^{(m)} : x_j^{(m)} > t_m\}. \end{aligned} \tag{3.20}$$

Ideally, a proposed split θ_m should increase the level of homogeneity, or *purity*, within each partition $R_{\text{left}}(\theta_m)$ and $R_{\text{right}}(\theta_m)$ respectively. On the contrary, *impurity* refers to heterogeneity, which can be interpreted as something that should be minimized when comparing different split options. Let $H(x)$ be an impurity function, where a few popular ones will be elaborated upon in Section 3.2.1.1, then a proposed split θ_m at node m can be the sum of impurities in each resulting partition, weighted by the fraction of samples falling into each respective partition. Letting N_m denote the total number of samples available for node m (before the split), n_{left} the number of samples in R_{left}, and n_{right} the number of samples in R_{right}, yields

$$G(R, \theta_m) = \frac{n_{\text{left}}}{N_m} H\left(R_{\text{left}}(\theta_m)\right) + \frac{n_{\text{right}}}{N_m} H\left(R_{\text{right}}(\theta_m)\right) , \tag{3.21}$$

where $G(R, \theta_m)$ can be interpreted as an impurity cost associated for each split proposition θ_m. Therefore, the optimal split θ_m^* for node m will be chosen as the one which minimises the impurity

$$\theta_m^* = \arg\min_{\theta_m} G(R, \theta_m) . \tag{3.22}$$

The above schema can be applied recursively for R_{left} and R_{right} respectively until some stopping criterion is reached, e.g. a maximum depth of the tree or if $N_m = 1$. Heuristically, one can think of the depth of a tree as the maximum numbers of questions/nodes to traverse before arriving at a prediction (leaf node).

Lastly, the weight vector w needs to be inferred. For regression tasks, it is calculated as the mean target value for each partition $R_1, \ldots, R_T$ [15, p. 308], i.e.

$$w_m^* = \frac{1}{N_m} \sum_{x_i \in R_m} y_i . \tag{3.23}$$

For multi-class classification problems, with K outcomes, the probability (bounded leaf weight) for class k at node m, is estimated as the fraction of available observations in partition R_m, hence

$$p_{mk}^* = \frac{1}{N_m} \sum_{x_i \in R_m} \mathbb{1}(y_i = k) . \tag{3.24}$$

In the binary classification domain, where $K = 2$, Equation (3.24) simplifies to

$$w_m^* = p_m^* = \frac{1}{N_m} \sum_{x_i \in R_m} \mathbb{1}(y_i = 1)\,. \tag{3.25}$$

The probability p_m^* from Equation (3.25) is the estimated probability of a positive event at node m, where the estimated probability of a negative event is given as the complement $1 - p_m^*$. Therefore, the same notation of w_m can be used for both regression and binary classification problem, highlighted by the equality between p_m and w_m.

3.2.1.1 Impurity measures

When defining the impurity cost as of Equation (3.21), it was built upon the existence of some impurity measure H. This subsection will touch upon a few common choices for impurity measures for the classification setting.

An intuitive choice may be the misclassification error, defined as

$$H(p_{mk}) = \frac{1}{N_m} \sum_{i=1}^{N_m} \mathbb{1}\left(y_i \neq \arg\max_k p_{mk}\right) = 1 - \max(p_{mk})\,, \tag{3.26}$$

which measures the fraction p of incorrectly classified samples, which intuitively one would like to minimize. However, there are mainly two problems associated with using the misclassification error as impurity measure. The first one is that it is not differentiable, imposing some problems for the minimization problem as of Equation (3.21). The second one being the linear punishment of impurity. To illustrate why that is a problem, consider the example from [15, pp. 309-310], where a balanced two-class training dataset with 400 positive, and 400 negative outcomes, is obtained. Consider two possible splits. The first one creates the leaves (300, 100) and (100, 300), where the first element in the tuples corresponds towards the number of positive samples, and the second element towards the number of negative samples in each leaf. Using the same notation, consider also the split (200, 400) and (200, 0). Computing the misclassification rate for each split yields the rate 0.25 for both splits, where hence the misclassification error as impurity measure would value both splits equally. However, it is often desirable to obtain pure nodes [15, pp. 310], meaning that the second split *should* be evaluated better than the first one.

To combat the two issues mentioned with the misclassification error, two other common ones will be considered, namely the Gini index and the Shannon entropy [24]. As can be noticed from Figure 3.5, both the Gini index and

the entropy measures show a similar concave behaviour, but have different interpretations. The Shannon entropy measure, defined as

$$H(p_{mk}) = -\sum_k p_{mk} \log(p_{mk}) \,, \tag{3.27}$$

stems from the field of information theory and quantifies the level uncertainty or (un)informativeness over a random variable [24], by averaging the informativeness of each possible for a random variable of interest. In the setting of classification trees, it deems obvious that the "questions" (splits) posed towards the feature space should be informative in terms of making a predictive decision. It can be shown that for a random variable with outcome space $\{1, \ldots, K\}$, the entropy achieves its maximum for the case of an uniform random variable (see Lemma (A.3) of Appendix A and its accompanying proof), hence achieves its local maximum at $p_{mk} = 1/K$.

Lastly, the Gini index is defined as

$$H(p_{mk}) = \sum_k p_{mk}(1 - p_{mk}) \,. \tag{3.28}$$

One interpretation of the Gini index is to leverage the stochasticity of the p_{mk} for prediction, and not just truncate that uncertainty by deterministically predicting the class as $\arg\max_k p_{mk}$. Then the training error rate with that policy becomes $\sum_{k \neq k'} p_{mk} p_{mk'}$, which is exactly the Gini index.

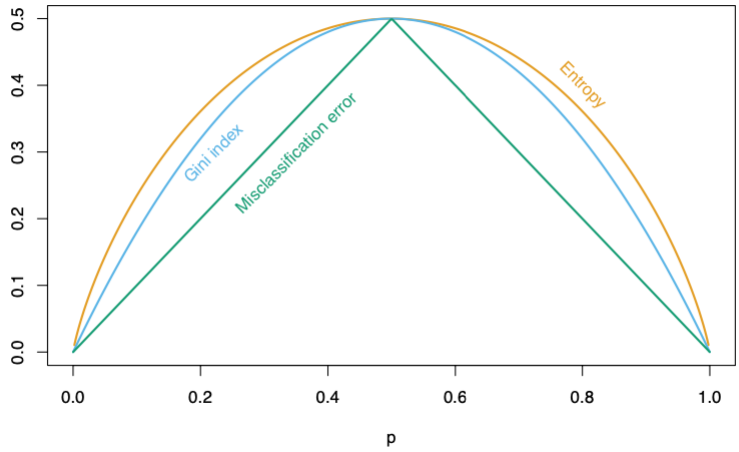

Figure 3.5: Different impurity measures as a function of the fractions of classes in the training data in the binary classification domain. For ease of comparison, the entropy measure has been scaled in order to intersect the point $(0.5, 0.5)$. Picture taken from [15, p. 309].

3.2.2 Overfitting issues

So far, it has been established that classification trees can be useful for solving binary classification problems as they are non-parametric and remain highly interpretable. The trade-off using the bias-variance viewpoint, is that CART models tend to be more prone to overfitting when they grow larger [15, p. 312].

One way of understanding why CART models easily overfit, is to consider the expressiveness of a class of functions, in this case classification trees. For a finite set of data points, $(x_1, y_1), \ldots, (x_n, y_n)$, it is always possible to grow a large enough tree such that $f(x_i) = y_i$ for all data points, in the extreme case by splitting the feature space such that each $x_i, i \in \{1, \ldots, n\}$ falls within a unique feature partition $R_1, \ldots, R_T$ (where $T \geq n$). For continuous features, there exists infinitely many cut-off points (assuming completeness of the real numbers), and hence infinitely many solutions in terms of tree structures. Practically, this is neither useful nor desirable as the model would not be able to generalize well for unseen data. However, the point here is to illustrate that the variance within different classes of functions may vary substantially. Consider instead a linear classifier, which assigns labels $\{-1, 1\}$ for a feature vector x_i based on which side of a hyperplane for the same feature space

x_i falls onto. The intrinsic assumption of linearity and additivity in the feature space for the case of a linear classifier limits the expressiveness of the model class. Geometrically, one can observe that there exists at least one linear classifier that can fully separate the positive labelled samples from the negative ones, when $n \leq 3$. However, as visualized by Figure 3.6, no such linear classifier exists for $n > 3$. Hence, a bias will inevitably be introduced within the class of linear classifiers on the training dataset, similar towards the irreducible error (see Equation (3.4)).

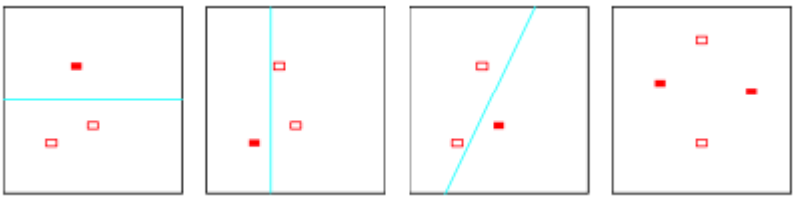

Figure 3.6: For three points, the linear classifiers (with corresponding hyperplane is highlighted in blue) are perfectly able to fully distinguish the labels, but fails for four points. Picture taken from [15, p. 238].

As CART models as model class is more relaxed in terms of assumptions, hence more expressive, it means that they are more prone to overfitting than simpler models, e.g. linear classifiers, when allowed to grow more complex. There are more rigorous arguments around model class expressiveness to be found in the subfield of statistical learning called Vapnik–Chervonenkis dimension theory [15, pp. 237-239].

3.3 Boosting methods for CART models

In social sciences, there exists a theory called *Wisdom of the crowd*, that refers to the idea of obtaining a better decision collectively than relying on any individual expert decision, by different aggregation methods [31]. Many procedures common in the modern society relies on such principles, e.g. democratic voting procedures and trial by jury, but also openly available collective creation processes such as the encyclopedia Wikipedia, or various open-source software projects. More everyday examples, such as in golf by continuously adjusting the position of the ball until the target is hit, can also be said to rely on similar principles. It is evident that in order to obtain a collective knowledge, there must exist a way of aggregating the contribution

of each individual. From a statistical learning perspective, this is exactly what is studied in the subfield ensemble learning - namely to combine individual base learners into a collective one [15, p. 605].

Addressing the first issue of finding a proper aggregation measure, in the field of ensemble learning, some common techniques include bootstrap aggregating, also known as *bagging* [15, p. 282], and different *boosting* techniques. The former works by creating multiple splits from the training data is drawn randomly with replacement and later used to in parallel fit multiple learners that are averaged by majority voting into one "collective" learner. Boosting on the other hand, fits learners sequentially with higher emphasis on training samples that the previous learners struggled to predicts their targets for, in order to ensemble a collective learner. The difference between the two aggregation measures might be explained more intuitively by comparing simple analogies. It may be noted that the examples regarding various voting procedures are easily performed in parallel. On the contrary, contributing towards a Wikipedia page depends on what previously has been written, as well as for the golf shots, which should alter to correct the errors from the previous shots. The former parallelizable voting tasks resemble the bagging aggregation, and the with latter sequential improvements resemble boosting aggregation methods. This section will treat the boosting methods more in detail, specifically in the setting for classification trees (see Section 3.2).

As discussed in subsection 3.2.2, CART models are prone to overfitting when allowed to grow fairly large. The main idea with boosting methods for tree methods is to combine a set of *weak learners* (high bias, low variance), to collectively obtain a low-biased ensemble model [15, p. 337]. To obtain these weak learners from the CART model family, one usually utilizes various regularization methods (see subsection 3.1.1.1), or hard constraints towards the trees to control the variance.

Consider once again the golfing problem. At each shot, the golfer of course does his/her best at calibrating the shot to hit the target. However, it is rare to hit the target at the first shot, therefore the golfer will more likely than not need to sequentially correct the previous shots. Another way to think about the problem is that the golfer tries to reduce the error between the hole and the ball's position after each shot, conditioned on the "knowledge" or position of the previous shots. Hence, major improvements could be expected for the first few shots, where the later ones likely will constitute smaller adjustments. Mathematically, this situation can be interpreted as an additive strategy, where the previous knowledge (golf shots) are saved, and

where the algorithm greedily will seek for the optimal strategy at each point, determined by the empirical predictive risk evaluated at some time step t. Consequently, succeeding learners will focus on samples where the previous ones misclassified [15, p. 338]. Let $\hat{y}_i^{(t)}$ be the prediction with x_i as feature vector for the i^{th} sample at step t. By following an additive strategy [15, pp. 341-342], and letting $\hat{y}_i^{(0)} = 0$, the prediction at step t obtains as

$$\begin{aligned}
\hat{y}_i^{(0)} &= 0\,, \\
\hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i)\,, \\
\hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i)\,, \\
&\dots \\
\hat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)\,,
\end{aligned} \tag{3.29}$$

where $\hat{y}_i^{(t-1)}$ is treated as a known constant when fitting $f_t(x_i)$. Typically, the learners f_k are restricted towards the same model class, i.e. $f_k \in \mathcal{F}\ \forall k \in \{1, \dots, t\}$, for some model class $\mathcal{F}$. For ease of notation, let $\phi(x_i)$ denote the ensemble model $\sum_{k=1}^{t} f_k(x_i)$. To infer its functional components $f_1, \dots, f_t$, the regularized empirical predictive risk (c.f. Equation (3.9))

$$\tilde{\mathcal{L}}(\phi) = \frac{1}{N}\left[\sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{t} \Omega(f_k)\right], \tag{3.30}$$

which will be minimized with respect to ϕ. Minimization of Equation (3.30) is not possible with standard mathematical optimization techniques, as $\tilde{\mathcal{L}}(\phi)$ is parametrized by basis functions $f_1, \dots, f_t$, which in their own are parametrized by their respective function parameters. This imposes a non-Euclidian parameter space which restricts the usage of standard optimization techniques [7]. Instead, the minimization procedure can be performed sequentially/-greedily, i.e. fitting the best f_t at step t, conditioned on what has previously been inferred as of $f_1, \dots, f_{t-1}$ (see Equation (3.29)). Hence, the idea is to fit the optimal tree structure q that maps an observation towards some feature partition $R_j, j \in \{1, \dots, T\}$, which together with the leaf weights w defines a tree function f. Let $\tilde{\mathcal{L}}^{(t)}(\phi)$ denote the regularized empirical predictive risk at step t, conditioned on $f_1, \dots, f_{t-1}$, which is, up to a constant, written as

$$\tilde{\mathcal{L}}^{(t)}(q) \propto \sum_{i=1}^{n} L(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t)\,. \tag{3.31}$$

To obtain Equation (3.31), the fact that $\sum_{k=1}^{t-1} \Omega(f_k)$ is considered known at step t, was used. For the complexity cost in the case of a CART (see Section 3.2) base learner f, it is suggested in [7] to use

$$\Omega(f; \gamma, \lambda) = \gamma T + \frac{1}{2}\lambda ||w||^2 \,, \tag{3.32}$$

where w corresponds towards the leaf weights, and T towards the number of leaves in f. Hence, large trees (i.e. trees with many leaves) will be penalized by the first term of Equation (3.32), and large weight values penalized by the second term. The latter might be more important for regression problems, where each element in w are not bounded within the range of $[0, 1]$, as for classification problems.

Choosing proper parameter values for γ and λ, is tied towards the bias-variance trade-off described in Section 3.1.1. The parameters for the complexity cost term, γ and λ as of Equation (3.32), should be tuned such that the model achieves a good balance between the bias and the variance, where the bias is penalized by the error/loss term, and the variance by the model complexity term. Consider a problem where an user's interest in some topic is predicted as a function of time. Figure (3.7) illustrates how one can think visually about finding a good balance between the error function term $L(f)$ and the model complexity function term $\Omega(f)$, for a fitted regression tree f.

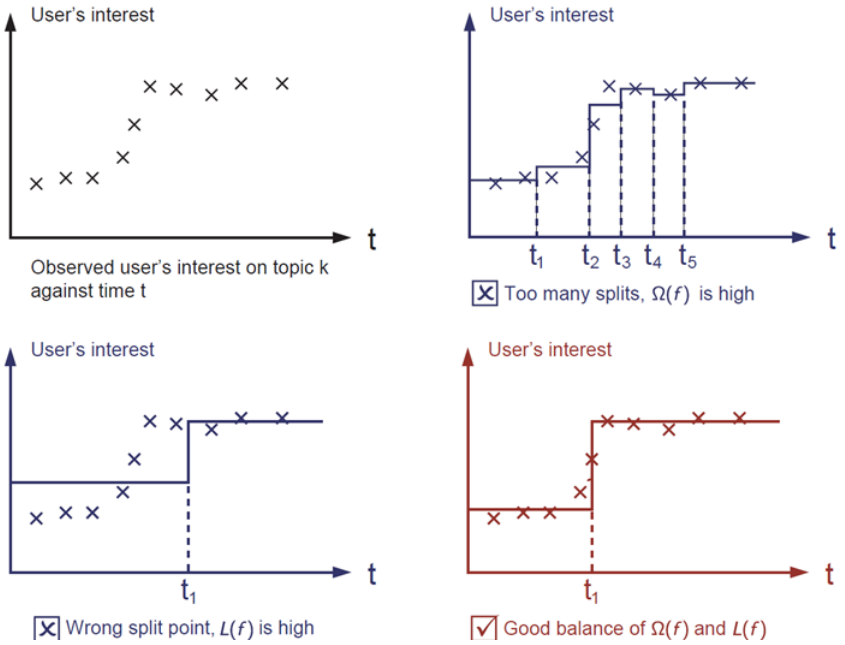

Figure 3.7: Comparison of fitted regression trees for the task of predicting an user's interest on some topic k as a function of time t. The top right panel shows an overfitted solution with too high variance, yielding a high $\Omega(f)$; the bottom left panel showcases an underfitted solution with too high bias, implying a high $L(f)$; lastly, the bottom right panel showcases a good balance between the two terms constituting the minimization objective. Picture taken from [8].

3.3.1 Gradient boosting methods

To optimize $\tilde{\mathcal{L}}^{(t)}(q)$ as of Equation (3.31), for loss functions that are not necessarily available on closed form, the numerical optimization procedure can be generalized by considering the second order Taylor expansion of $L(y_i, \hat{y}_i^{(t-1)} + f_t(x_i))$, which yields that

$$\mathcal{L}^{(t)}(q) \approx \sum_{i=1}^{n} \left[L(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{h_i}{2} f_t^2(x_i) \right] + \Omega(f_t)\,, \tag{3.33}$$

where

$$g_i = \left. \frac{\partial L(y_i, z)}{\partial z} \right|_{z=\hat{y}_i^{(t-1)}},$$

$$h_i = \left. \frac{\partial^2 L(y_i, z)}{\partial z^2} \right|_{z=\hat{y}_i^{(t-1)}},$$

are the first- and second gradients evaluated at $\hat{y}_i^{(t-1)}$, corresponding towards fitting the new tree model towards the pseudo-residuals [15, p. 360] to account for what previous base learners failed to capture.

Remark. *The approximation of Equation (3.33) is technically only true up to a constant, which can be realized by comparing Equation (3.30) with Equation (3.31).*

It is noticed from Equation (3.33) that prior knowledge at step t, i.e. what is captured by $\hat{y}_i^{(t-1)}$, is assumed to be known, hence $L(y_i, \hat{y}_i^{(t-1)})$ can be considered as constants for all $i = 1, \ldots, n$. Therefore, it holds that

$$\mathcal{L}^{(t)}(q) \propto \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{h_i}{2} f_t^2(x_i) \right] + \Omega(f_t)\,. \tag{3.34}$$

Moreover, Equation (3.34) has an important implication, namely that any loss function L can be used in the gradient boosting optimization framework, given that L is of differentiability class C^2 (its second-order derivative is continuous).

Using the model complexity cost term from Equation (3.32) and inserting it into Equation (3.34), and also utilizing the model definition of CART (see Equation (3.19)), yields that

$$\mathcal{L}^{(t)}(q) \propto \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{h_i}{2} f_t^2(x_i) \right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2\,, \tag{3.35}$$

$$= \sum_{j=1}^{T} \left[\left(\sum_{x_i \in R_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{x_i \in R_j} h_i + \lambda \right) w_j^2 \right] + \gamma T\,. \tag{3.36}$$

As $\sum_{x_i \in R_j} g_i$ and $\sum_{x_i \in R_j} h_i + \lambda$ are constants with respect to w_j, differentiation of Equation (3.36) implies that the optimal weight at leaf j, $\hat{w}_j$, obtains as

$$\hat{w}_j = -\frac{\sum_{x_i \in R_j} g_i}{\sum_{x_i \in R_j} h_i + \lambda}\,, \tag{3.37}$$

by standard optimization techniques from differential calculus. Consequently, substitution of w_j with $\hat{w}_j$ in Equation (3.36), yields that the best reduction of the loss at step t, for a tree structure q, is

$$\mathcal{L}^{(t)}(q) \propto -\frac{1}{2}\sum_{j=1}^{T} \frac{\left(\sum_{x_i \in R_j} g_i\right)^2}{\sum_{x_i \in R_j} h_i + \lambda} + \gamma T\,. \tag{3.38}$$

As highlighted in Section 3.2.1, it is infeasible to compare all the different tree structures $q \in \mathcal{Q}$ in a brute force manner. However, using once again a greedy optimization procedure and hence using a single leaf/node as the starting point, it is then possible to evaluate a suggested split by considering the loss reduction before and after imposing the split. Let R be the feature partition at the starting leaf for which the suggested split stems from, then by letting $R = R_{\text{left}} \cup R_{\text{right}}$, i.e. cutting R into two disjoint partitions R_L (left partition) and I_R (right partition). Then the reduction of the objective (empirical predictive risk), as of Equation (3.38), obtains as

$$\mathcal{L}_{\text{split}} = \frac{1}{2}\left[\frac{\left(\sum_{x_i \in R_{\text{left}}} g_i\right)^2}{\sum_{x_i \in R_{\text{left}}} h_i + \lambda} + \frac{\left(\sum_{x_i \in R_{\text{right}}} g_i\right)^2}{\sum_{x_i \in R_{\text{right}}} h_i + \lambda} - \frac{\left(\sum_{x_i \in R} g_i\right)^2}{\sum_{x_i \in R} h_i + \lambda}\right] - \gamma\,, \tag{3.39}$$

which allows for comparisons of multiple suggested splits at each leaf. By greedily finding the best strategy as of Equation (3.39), until reaching a stopping criterion (e.g. the predefined maximum depth for the tree), an optimal tree structure $\hat{q}$ can be learned, which corresponds to the fitted tree model $\hat{f}_t(x)$.

Lastly, recapping the additive model in Equation (3.29) at step t, the ensemble model obtains as

$$\phi(x_i) = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + \hat{f}_t(x_i)\,. \tag{3.40}$$

In the binary classification setting, the prediction for x_i would be obtained as [12]

$$\text{sign}\left(\phi(x_i)\right)\,, \tag{3.41}$$

and hence is *not* to interpreted as pseudo-probabilities, as what was mentioned for the case of subsection 3.2.

3.3.1.1 Shrinkage

Recall the example of sequentially hitting golf shots in order to eventually hit the hole. In some situations, such as when hitting the ball from difficult terrains of the pitch, it might be more fruitful to purposely aim for parts of the pitch which allow for an easier next shot, rather than trying to hit the hole directly, possibly causing an even worse situation for the next shot. Moreover, if the golfer also was liberated from the scoring based on the number of hits needed to hit the hole, the incentive of these "safe plays" would obviously be more evident, assuming there would still exist an associated cost of hitting the ball e.g. in the water or out in the woods. A similar approach can be translated into the setting of gradient boosting methods for tree models.

Another regularization technique specific for the gradient boosting trees, besides the general model complexity cost term as of Equation (3.8), is the introduction of a shrinkage factor. Intuitively, it's a factor $\eta \in (0, 1)$ that suppresses the influence of a newly fitted individual tree towards the ensemble, hence allowing future fitted trees to have a greater impact towards the ensemble model. Modifying Equation (3.40), the prediction of x_i after step t becomes

$$\hat{y}_i^{(t)} = \phi(x_i) = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + \eta \hat{f}_t(x_i)\,, \tag{3.42}$$

with $\text{sign}(\cdot)$ as decision function in the binary classification domain (see Equation (3.41)). Empirically, more aggressive shrinkage ($\eta < 0.1$) has shown to reduce the risk of overfitting [13, 15, p. 365], but requires more iterations in the boosting schema as a trade-off due to the slower convergence. The shrinkage factor hence has similar interpretations and implications as for a *learning rate*, central for some other gradient based sequential supervised learning procedures [5, pp. 143-144].

Remark. *Technically, the range restriction of $\eta \in (0, 1)$ could be relaxed to include larger values where $\eta \geq 1$, but which would defeat the whole purpose and interpretation of a shrinkage factor, and hence will be excluded for this work.*

3.3.2 Loss functions

For a binary classification problem, with targets $y \in \{-1, 1\}$, the classification margin $y_i f(x_i)$, for some classifier f (ϕ in subsection 3.3.1) and feature vector x_i, plays an important role to study. By the signs of the two factors in the classification margin, it follows that $y_i f(x_i) > 0$ for correctly classified samples, and $y_i f(x_i) < 0$ for misclassified ones. Clearly, any reasonable loss function for the binary classification problem should impose a greater penalty towards negative values of the classification margin then the positive ones, hence reducing the misclassification error. In Figure 3.8, a few common loss functions are compared for the classification margin $yf(x)$, where the respective definitions can be found as of Table 3.2.

Loss function	**Definition**
Misclassification	$\mathbb{1}(\text{sign}(f(x)) \neq y)$
Exponential	$e^{-yf(x)}$
Binomial deviance	$\log\left(1 + e^{-2yf(x)}\right)$
Squared error	$(y - f(x))^2$
Support vector (Hinge)	$\max(1 - yf(x), 0)$

Table 3.2: Different loss functions for binary classification and their corresponding definition.

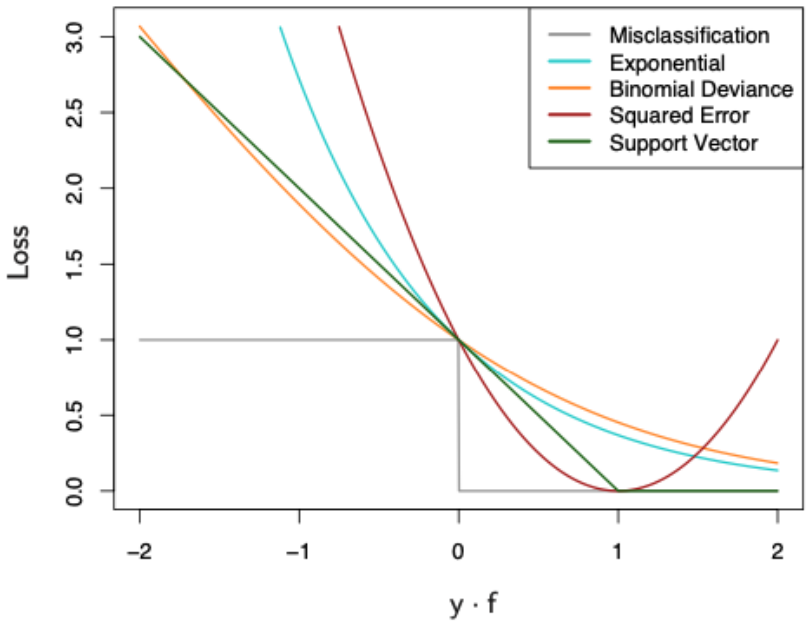

Figure 3.8: Loss functions for a binary classification problem ($y \in \{-1, 1\}$). Picture taken from [15, p. 347].

Starting with the misclassification loss function, there are mainly two problems that disqualifies it as an adequate loss function. Firstly, the discontinuity in $yf(x) = 0$ causes it to be non-differentiable in the same point, and therefore problematic for gradient based optimization methods. Secondly, the misclassification loss function penalizes all negative margins equally, regardless of the proximity to the decision boundary, and does not penalize high uncertainties for correctly classified samples. Moreover, as the squared error increase quadratically for $yf(x) > 1$, it does not constitute as a viable option for classification problems.

Other alternatives, such as the exponential and the binomial deviance loss, can be seen as monotone continuous approximations of the misclassification loss function [15, p. 347]. Both of these imposes greater incentives for reducing the magnitude of negative classification margins rather than increasing the magnitude for the positive ones. From Figure 3.8, one can see that the exponential loss function puts an exponentially increasing influence towards lower values of the classification margin, consequently allowing outliers with

low classification margin to be highly influential, and hence problematic in noisy classification settings [15, p. 348]. The binomial deviance on the other hand is more restrictive towards letting such extreme observations determine too much of the fitting procedure, instead emphasizing more equally over all samples.

In subsection 3.3.1, it was shown that the presented gradient boosting method framework requires that L is of differentiability class C^2. For the case of the exponential and binomial deviance loss, it is easy to verify that they are indeed of differentiability class C^2 due to their exponential nature. For the Support vector loss however, this is not the case as the first-order derivative discontinuous at $yf(x) = 1$. Computationally, smoothed approximations [35] can be utilized in software implementations [7] to enjoy similar properties as of the Support vector loss.

3.4 Interpretation methods

In the field of statistics, the concepts of inference and prediction have two distinguished meanings and purposes. For the case of inference, one is interested in understanding the generative process of the studied data. Unlike descriptive statistics, some distributional assumptions are imposed, hence forming some sort of statistical model to learn from. Pure prediction problems regard the usage of statistical models to predict the outcomes of unseen data points. In many real-world situations however, these two tasks are of mutual interest [20, sec. 2.1]. Surely, a predictive model in its predictions should be accurate by some measure, but ideally also arrive at the right prediction for the right reasons and serve. Interpretation methods are e.g. necessary to evaluate a level of fairness within the model (that no clear biases against certain groups appear either explicitly or implicitly) and to evaluate the causality of a model's mapping on qualitative grounds. In a customer retention problem, this means being able to both predict who will stay on the plan and not; to understand why that is the case to serve for future recommendations, either globally for all customers, or locally for some segment of users; and to avoid any causally or ethically spurious actions/interventions perform sub-consequently.

When talking about interpretable methods, there are two important distinctions to make - intrinsic model interpretability and model-agnostic interpretability [20, sec. 2.2]. The first one regards model-specific, such as inferring feature effects based on the learned model parameters (e.g. in linear- and

logistic regression), and the latter regards general methods to infer the inner workings about a model when treating it, partly or fully, as a "black-box". This section covers one model intrinsic interpretation method for boosted tree models that stems from the highly interpretable CART models as base learners, but also a few model-agnostic methods which are useful when challenges on the intrinsic model interpretability are imposed by boosting methods.

3.4.1 Relative importance for tree models

In many situations, a good starting point when analyzing a large set of features/predictors, is to infer which ones had the highest contributions towards the learning task. Generally, in data-mining applications with a large feature space, a smaller subset of these can be expected to have a substantial impact on the target variable [15, p. 367]. For a single CART tree model T, one can obtain a measure of the relative importance for some feature x_j as

$$\mathcal{I}_j^2(T) = \sum_{t=1}^{J-1} \hat{\tau}_t^2 \mathbb{1}(v(t) = j)\,, \tag{3.43}$$

where $J-1$ is the number of internal (non-child) nodes. Hence, for every node t, a selected feature $x_{v(t)}$ forms two sub-regions by a binary split, and thereafter fits two constants (weights) for each towards the target variable. The feature of choice is the one that maximizes the estimated improvement, denoted $\hat{\tau}_t^2$, in squared error risk over the entire region. Lastly, the global importance measure $\mathcal{I}_j^2(T)$ sums for a given tree T the improvements $\{\hat{\tau}_1^2, \ldots, \hat{\tau}_{J-1}^2\}$ for a feature x_j, conditioned on whether it was chosen as the "top" (the one that maximized the improvement) feature at each internal node $\{1, \ldots, J-1\}$ [15, p. 368].

For an additive model of such trees, e.g. a boosted tree model, the importance measures can be averaged over all M base learners, i.e.

$$\mathcal{I}_j^2 = \frac{1}{M} \sum_{m=1}^{M} \mathcal{I}_j^2(T_m)\,, \tag{3.44}$$

which is argued for in [15, p. 368] being a more stable measurement of feature relative importance than the one from Equation (3.43), operating on one single high-variance learner. Since the values $\mathcal{I}_j^2$ and $\mathcal{I}_j^2(T)$ are to be interpreted relatively, it is a common practice to normalize them towards a range of e.g. $[0, 100]$ or $[0, 1]$ for ease of comparison features in between.

3.4.2 Partial Dependence Plots

Only considering the relative importance omits a large amount of information regarding how a subset of features affect the predictions of the model. In general, one is not only interested in targeting important features, but also how a fitted model has learned the mapping between a subset of features and the target. One way of obtaining that knowledge is by construction of a Partial Dependence Plot (PDP) [15, pp. 369-370].

Let $\mathcal{S}$ be a set of integers, serving as a subset for the feature indices $\{1, \ldots, p\}$ (for a p-dimensional feature space), such that $\mathcal{S} \subset \{1, \ldots, p\}$. Furthermore, let $\mathcal{C}$ be the complement of $\mathcal{S}$ with respect to $\{1, \ldots, p\}$, i.e. $\mathcal{C} = \{1, \ldots, p\} \setminus \mathcal{S} \Leftrightarrow \{1, \ldots, p\} = \mathcal{S} \cup \mathcal{C}$, then the partial dependence of $f(x)$ on $x_{\mathcal{S}}$ is

$$f_{x_{\mathcal{S}}}(x_{\mathcal{S}}) = \mathbb{E}_{x_{\mathcal{C}}} \left[f(x_{\mathcal{S}}, x_{\mathcal{C}}) \right] . \tag{3.45}$$

The partial dependence measure provides a helpful understanding of how $x_{\mathcal{S}}$ influences the function f, under the assumption that $x_{\mathcal{S}}$ does note interact strongly with $x_{\mathcal{C}}$, hence justifying the averaging. Using Monte-Carlo methods, one can obtain an estimate of Equation (3.45) as

$$\hat{f}_{x_{\mathcal{S}}}(x_{\mathcal{S}}) = \frac{1}{n} \sum_{i=1}^{n} f\left(x_{\mathcal{S}}, x_{\mathcal{C}}^{(i)} \right) , \tag{3.46}$$

where $x_{\mathcal{C}}^{(i)}$ is the i^{th} observation in the training dataset that jointly is in $x_{\mathcal{C}}$. Limited by the human visual perception and computational feasibility, one is typically interested in restricting the cardinality (feature dimensions) of $x_{\mathcal{S}}$ to be less than or equal to three. As an example, Figure 3.9 illustrates two one-dimensional PDPs (two different features) for a fitted cervical cancer classification model.

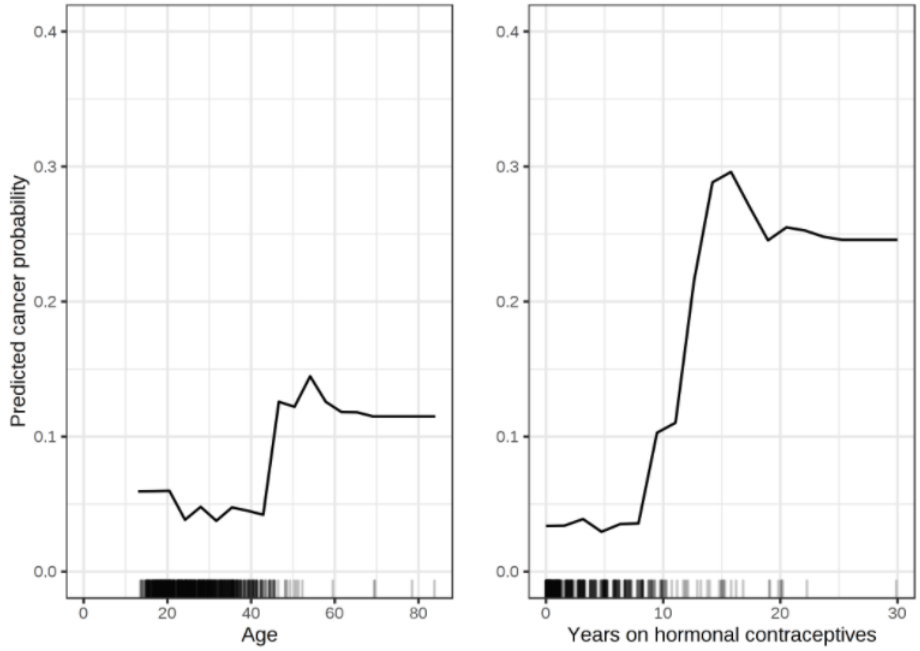

Figure 3.9: Partial dependence plot of cervical cancer probability as a function of age (left) and years on hormonal contraceptives (right) marginally. The marks on the x-axes correspond to the density of the data distribution. Picture taken from [20, sec. 5.1].

3.4.2.1 ICE Plots

One problem with the averaging effect of the partial dependence plot is that disagreeing individual instances are not visualized. One way of capturing this variance of feature effects is to plot each instance of $f(x_S, x_C^{(i)})$ (c.f. Equation (3.46)), which is exactly what is done in an Individual Conditional Expectation (ICE) plot [14] by disaggregating the averages of a PDP. Hence, the inverse also holds, i.e. that a PDP averages the individual lines of an ICE plot [20, sec. 5.2].

One problem with the ICE plots is that different plotted instances may vary simply because they start from different initial values of the features [20, sec. 5.2]. A relatively simple fix is to center the instances around some anchor point x_a (typically at the lower end of the feature space), and plot the difference in predictions to this point, by defining the centered ICE curves as

$$\hat{f}_{\text{cent}}^{(i)} = \hat{f}^{(i)} - \mathbf{1}_{|x_{\mathcal{C}}|}\hat{f}(x_a, x_{\mathcal{C}}^{(i)})\,, \tag{3.47}$$

where $\mathbf{1}_{|x_{\mathcal{C}}|}$ denotes a vector of 1's with dimensionality $|x_{\mathcal{C}}|$ (feature dimensions of the complementary set). Therefore, the interpretation of the y-axis for a centered ICE plot can be seen as the change in the predicted probability of a positive event, in comparison against some baseline obtained in x_a.

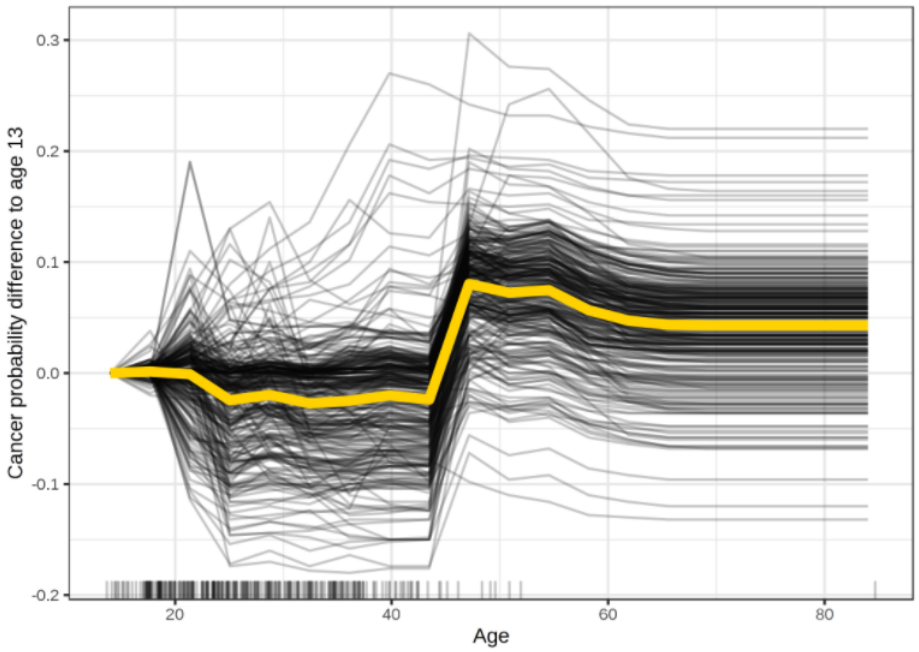

Figure 3.10: Centered ICE plot of cervical cancer probability as a function of age, where each line corresponds to one sample. The yellow line shows the partial dependence, interpreted as the average effect the instances (c.f. Figure 3.9). There is a noticeable increase in predicted cervical cancer probability happening in the age of 40s for most women. One can also note a stagnation of the predicted probabilities after the age of 60, but for which only a few data points are available. The marks on the x-axes correspond towards the density of the data distribution. Picture taken from [20, sec. 5.2].

Naturally, displaying the ICE plots is only fruitful in the one-dimensional setting, as variation is challenging to visualize in higher dimensions.

3.4.3 Feature interactions

In the case of strong feature interaction effects, it can be spurious to interpret the effects of multiple features additively [20, sec. 5.4]. As an illustrative example, consider a restaurant who wants to infer customer satisfaction based on their orders. Assume the side orders of french fries and chocolate sauce both have shown to positively correlate with customer satisfaction. Still, it would most likely not be advisable to recommend either french fries or chocolate sauce as a side regardless of what type of food the customer ordered. Presumably, french fries would constitute a reasonable side order recommendation for a main dish, but not for a dessert, and vice versa for chocolate sauce. Instead, one could model the interaction between the type of food (main dish or dessert) and the side order (french fries or chocolate sauce), to hopefully capture a more reasonable picture.

To later understand about ways of quantifying feature interactions, consider first the relationship between the two-dimensional and the one-dimensional partial dependencies. Let $PD_{jk}(x_j, x_k)$ denote the two-dimensional partial dependence (PD) between features x_j and x_k. Comparing with Equation (3.45), the notation of $PD_{jk}(x_j, x_k)$ would yield that $x_{\mathcal{S}} = \{x_j, x_k\}$ and $x_{\mathcal{C}} = x \setminus \{x_j, x_k\}$. Similarly, let $PD_j(x_j)$ correspond to the one-dimensional (marginal) partial dependence for a single variable x_j, i.e. $x_{\mathcal{S}} = x_j$ and $x_{\mathcal{C}} = x \setminus x_j$. If two features do not interact, an additive interpretation of the partial dependence is possible, i.e.

$$PD_{jk}(x_j, x_k) = PD_j(x_j) + PD_k(x_k)\,. \tag{3.48}$$

Moreover, if a single feature x_j has no interaction with the remaining ones, one can expect to retrieve the one-dimensional partial dependence for x_j as

$$\hat{f}(x) = PD_j(x_j) + PD_{-j}(x_{-j})\,, \tag{3.49}$$

where $PD_{-j}(x_{-j})$ denotes the partial dependence for the remaining $x \setminus x_j$ features. One way of quantifying the the interaction strength between two features x_j and x_k is by computing the fraction of the variance explained by their interaction, and can be estimated by the difference between the two-dimensional partial dependence and their linear one-dimensional counterparts. Hence, if Equation (3.48) holds, the statistic should be zero-valued. One such statistic is the H-statistic [20, sec. 5.4], which, between two features (two-way interaction) x_j and x_k, is defined as

$$H_{jk}^2 = \sum_{i=1}^{n} \left[PD_{jk}(x_j^{(i)}, x_k^{(i)}) - PD_j(x_j^{(i)}) - PD_k(x_k^{(i)}) \right]^2 / \sum_{i=1}^{n} PD_{jk}^2(x_j^{(i)}, x_k^{(i)}) \,. \tag{3.50}$$

Moreover, the total interaction between some feature x_j and the rest of the features $x \setminus x_j$ obtains as

$$H_j^2 = \sum_{i=1}^{n} \left[\hat{f}(x^{(i)}) - PD_j(x_j^{(i)}) - PD_{-j}(x_{-j}^{(i)}) \right]^2 / \sum_{i=1}^{n} \hat{f}^2(x^{(i)}) \,, \tag{3.51}$$

where $\hat{f}(x^{(i)})$ is the prediction for some sample $x^{(i)}$.

Two considerable downsides with the H-statistic for measuring interaction strength is computationally expensive, and also that the integration over feature combinations are based on often naive assumptions about independence between the features [20, sec. 5.4].

Chapter 4

Results and Discussion

4.1 Experiment overview

To find an optimal classifier, 200 random hyperparameter configurations with ranges as of Table 4.1 were generated following a random search strategy [4]. For this random search approach, 75% of the data was used for hyperparameter tuning together with a K-Fold Cross-Validation schema with $K = 5$. Recapping subsection 3.1.1.2 this means that the training set was split into 5 disjoint folds, where each fold was used for validation exactly once. Moreover, an early stopping criterion was imposed of requiring a decrease in the loss every 10 iterations, meaning if no improvement (decrease in loss) has occurred for the validation set, the training procedure stops (see subsection 3.1.1.3 for more details). Thereafter, the best hyperparameter configuration was chosen as the one which (on average for 5 folds) minimized the loss. This random search strategy was independently applied for both the binomial deviance loss and the support vector (hinge) loss in order to compare the performance the two functions in between. For the impurity measure, the Gini index was chosen as it is natively supported by the XGBoost package [7].

Furthermore, to evaluate the generalization capability for the final model defined by the chosen parameter configuration, the AUC (see section 3.1.2) was evaluated for the unseen 25% of the samples. Lastly, as a baseline for predictive benchmarking, a logistic regression model was fitted towards all the available features and evaluated on the heldout test dataset on similar grounds as for the final model.

Hyperparameter	Range
Shrinkage factor	$[0.01, 0.1]$
Maximum tree depth	$\{3, \ldots, 9\}$
Regularization parameter α	$[0.1, 1]$
Regularization parameter λ	$[0.1, 1]$

Table 4.1: Table of hyperparameters and their respective range constituting the hyperparameter search space. The notations of α and λ regard the regularization parameters for the model complexity cost (see Equation (3.32)). The ranges regarding the shrinkage factor and the maximum tree depth followed by heuristics from [15].

4.2 Evaluation

After running the experiments as of subsection 4.1, which is shown in Table 4.2, the best mean AUC values were obtained as 0.7772 and 0.5654 (rounded towards the nearest 4 decimals) for the binomial deviance loss and the hinge loss respectively. Hence, it was found empirically that the binomial deviance loss seemed to yield noticeably better result in terms of AUC than the hinge loss, at least for the explored parameter subspace as of Table 4.1. Therefore, the hyperparameter configuration for the binomial deviance loss setting was chosen as the final model. It is noticeable however that the best choice of the regularization parameter α in the setting of a hinge loss was obtained near the upper boundary in the explored range, or what one can view as the prior over the hyperparameter space, for which a relaxation of these priors could serve as a benefit for the hinge loss.

Loss Function	Mean AUC	Std. error	Wald 95% CI
Binomial deviance	0.7771564	0.007994274	$[0.7615, 0.7928]$
Hinge	0.5653842	0.006823795	$[0.5520, 0.5788]$

Table 4.2: AUC evaluations on test data (heldout data) for the binomial deviance and hinge loss function, based on five fits according to a K-Fold Cross-Validation (with $K = 5$) schema.

Hyperparameter	Binomial deviance loss	Hinge loss
Shrinkage factor	0.02	0.060
Maximum tree depth	4	8
α	0.90	0.99
λ	0.27	0.44

Table 4.3: Best hyperparameters (rounded to the nearest two decimals) for the two respective loss functions.

From this point on, the binomial deviance loss and its corresponding hyperparameter fit will be accepted and referred to as the final/optimal model. Moreover, it is of interest to evaluate the fitting/training procedure of this final model as a function of the number of boosting rounds (the number of base learners constituting the ensemble). Figure B.31 and Figure B.32 of Appendix B show the (logarithmic) loss and AUC respectively for the final model. It shows that the marginal benefit, either in terms of reduction in the loss or increase in AUC, appears to be low slightly before 200 boosting rounds, before eventually stopping by the early stopping criterion after 247 iterations. As performance on the training set seems to be improving, it means that further training iterations would make the model more prone to overfitting if anything. Similar figures for the hinge loss can be found as Figure B.33 and Figure B.34 of Appendix B.

For a predictive benchmark, as mentioned already in section 4.1, a logistic regression model was fitted towards all of the available data. The R summary output, containing among others the log odds parameter effects and corresponding p-values, can be found as of Figure B.35 of Appendix B. Moreover, an AUC value of 0.7422 towards the test dataset was obtained, which suggests that the more complex approach of the boosted tree model only gained around 3.5 percent units relative towards the baseline logistic regression model.

4.3 Model interpretation

With the expressive nature of the boosted tree methods, the suggested interpretation methods as of section 3.4 can be used to identify influential features and how they interact with each other. Studying a larger feature space and how the features interact with each other is however rather computationally expensive, especially for the interaction effects which require, in the worst case, $2n^2$ PDP calls for the two-way H-statistic (between fea-

tures x_j and x_k) and $3n^2$ calls for the overall interaction H-statistic [20, sec. 5.4]. Therefore, this section is limited towards studying the effect and interactions between a smaller subset of features, identified by their (under the final model) relative importance and interaction effects. Figure 4.1 shows the relative importance for the top 20 features under the final model. It is noticed that GROUP6/AU2/1 seems to be the most influential predictor, followed by GROUP3/LOG_AU1/12 and GROUP5/BOOL/1. From the summary of the logistic regression model fit as of Figure B.35, one can note that the null hypotheses of random effects for the above three mentioned features, were rejected on 0.1% significance levels respectively, hence supporting the narrative of their importance.

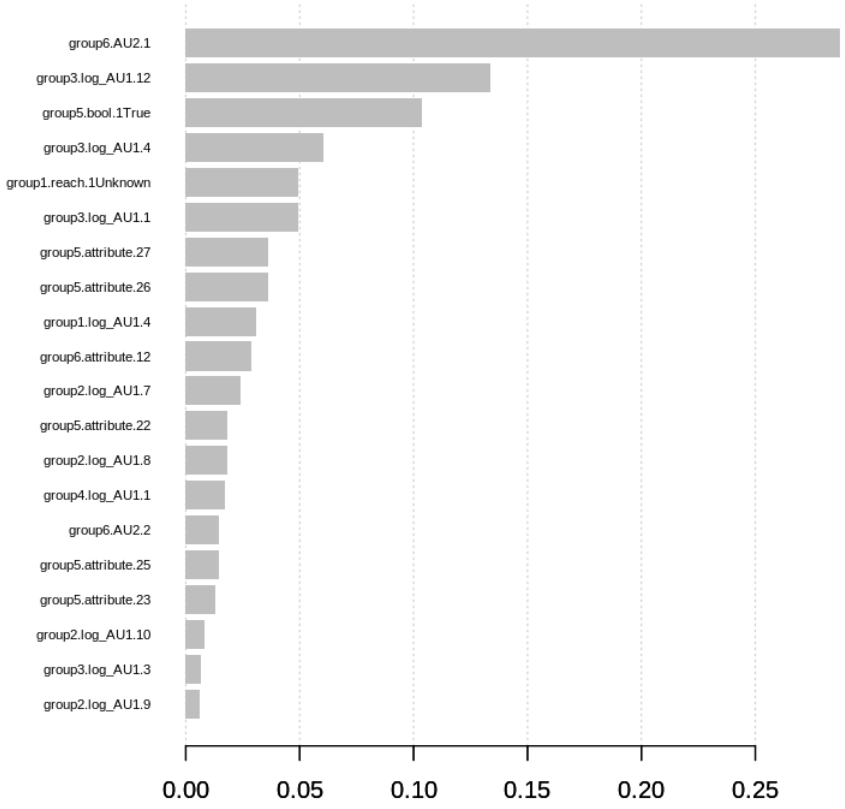

Figure 4.1: Relative importance for the most influential 20 features of the best performing boosted tree model with binomial deviance loss. The relative importance measure is explained more thoroughly for tree models in subsection 3.4.1. It is noticed that the model infers GROUP6/AU2/1 as the single most important predictor. Two other important predictors seem to be GROUP3/LOG_AU1/12 and GROUP5/BOOL/1, being the second and third most important predictors respectively based on the relative importance measure.

For further analysis, the top 6 features based on Figure B.35 will be studied more in depth, and to showcase how one in general can infer the effects of other features of interest. By studying the marginal PDPs for the continuous and ordinal variables out of the above mentioned top 6 features, it is noticed that GROUP6/AU2/1, GROUP3/LOG_AU1/12 and GROUP3/LOG_A1/1 seems

to have a (near) monotonically positive relationship with the target, i.e. increase the likelihood of retaining for larger values of these predictors. On the other hand, the GROUP3/LOG_A1/4 seems to have the opposite effects, decreasing the likelihood of retaining for higher values of GROUP3/LOG_A1/4. Looking at the centered ICE plots, however, it is noticed that there is some variability in terms of the effects towards the targets, especially for GROUP3/LOG_A1/4, but also for GROUP6/AU2/1 and GROUP3/LOG_A1/1. For the case of GROUP3/LOG_AU1/12, it seems to be the case that most instances agree upon the monotonic increase in retention probabilities for larger values of GROUP3/LOG_AU1/12.

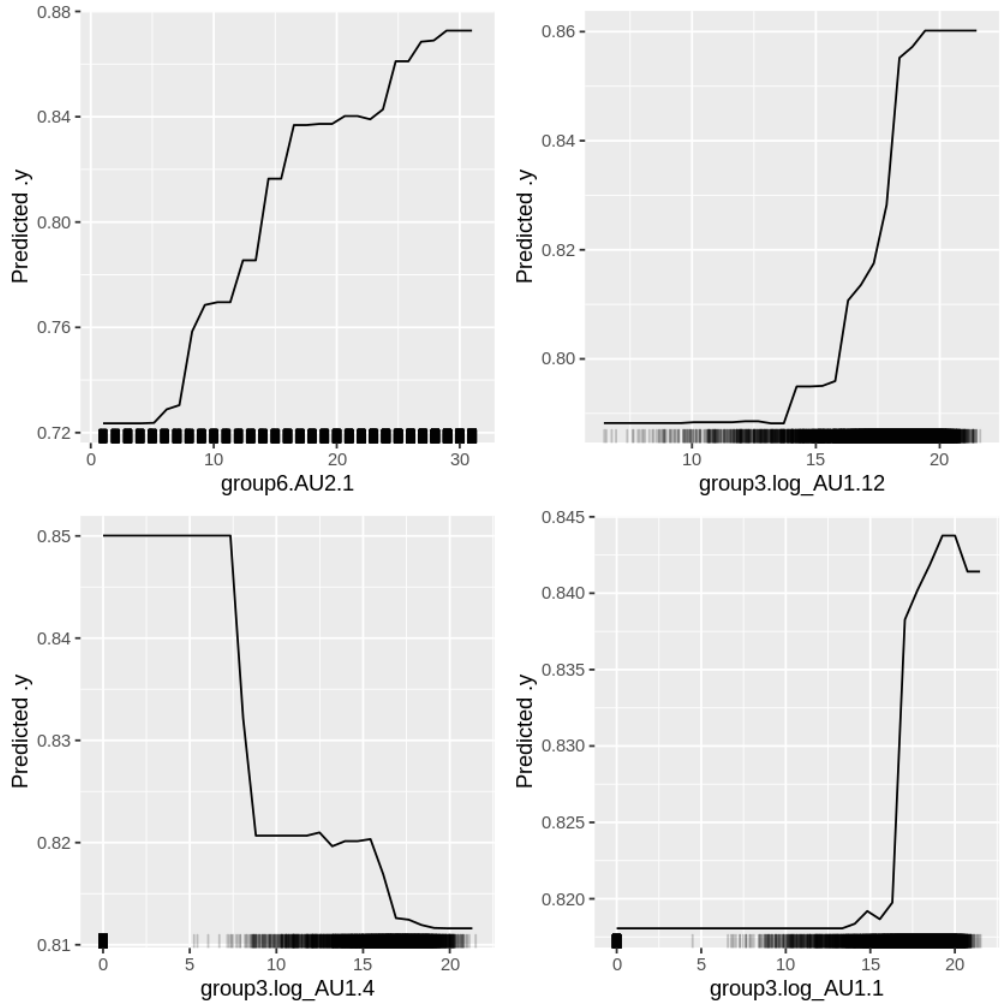

Figure 4.2: Marginal partial dependence plots (see subsection 3.4.2) for the top 4 continuous features in terms of relative importance (see Figure 4.1) under the final model. The partial dependence plots for the categorical features could not be disclosed due to confidentiality. The top left panel shows a monotonic positive near-linear relationship between the predictor GROUP6/AU2/1 and the target. The top right and bottom right panel show a noticeable increase of predictive values for the target for smaller increments of the predictors (GROUP3/LOG_AU1/12 and GROUP3/LOG_AU1/1 respectively) in the regions of LOG_AU1 ≈ 15. Lastly, the bottom left panel shows a negative relationship between the predictor GROUP3/LOG_AU1/4 and the target, meaning lower prediction values in the presence of larger values of GROUP3/LOG_AU1/4.

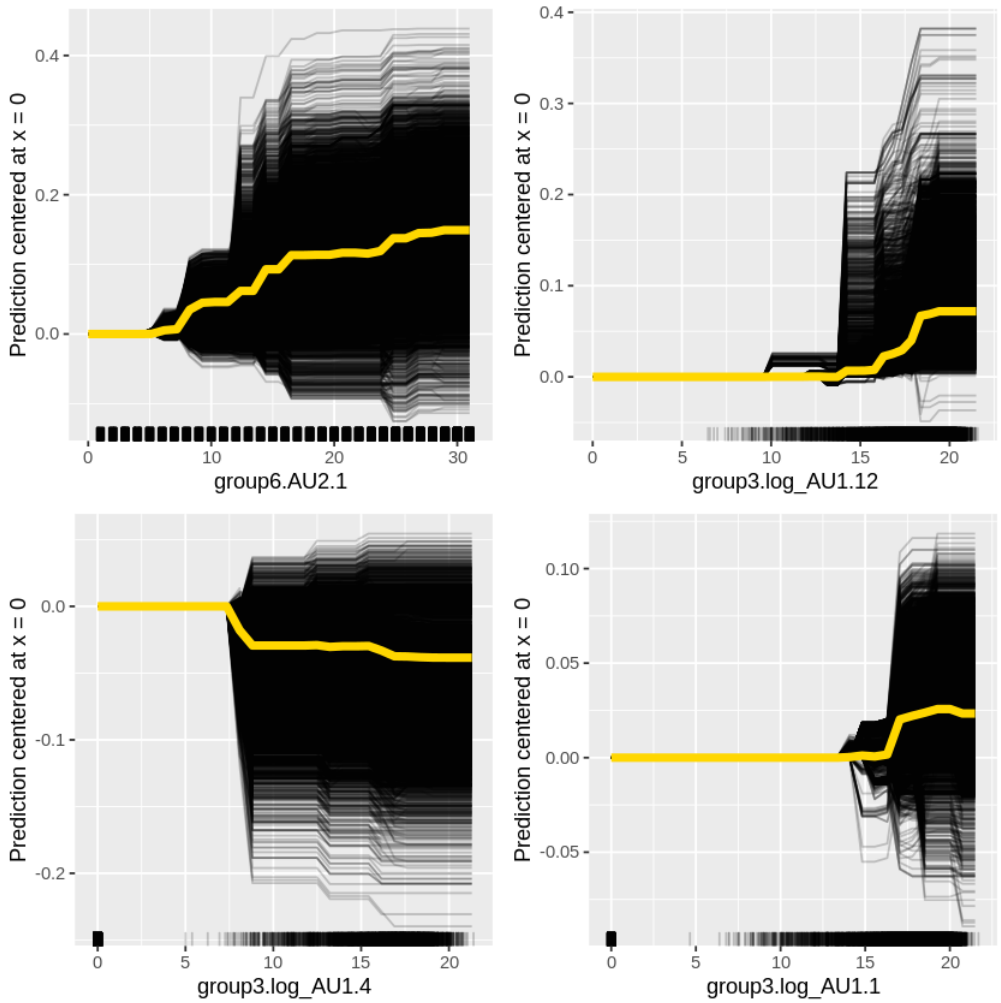

Figure 4.3: Centered ICE plots (see subsection 3.4.2.1) at $x = 0$ for the top 4 continuous features in terms of relative importance (see Figure 4.1) under the final model. The ICE plots for the categorical features could not be disclosed due to confidentiality. Each panel, except for arguably the top right panel, shows by the variability of predictions that some instances have an opposing effect towards the target as what was suggested by the PDP (yellow line). This suggests that the effects for GROUP6/AU2/1, GROUP3/LOG_AU1/4 and GROUP3/LOG_AU1/1 may have opposing interpretations for different cohorts. On the contrary, the top right panel seems to suggest homogeneous view on the positive relationship between GROUP3/LOG_AU1/12 and the target.

Recapping the independence assumption of PDPs (and by definition ICE plots) mentioned in subsection 3.4.2, one can expect some pitfalls with naively

interpreting marginal PDPs without understanding the feature interactions in between. Figure 4.4 shows the overall H-statistic interactions (see Equation 3.51) for each predictor variable. Indeed, the features with the highest relative importance have, generally, considerable interaction strength with other features.

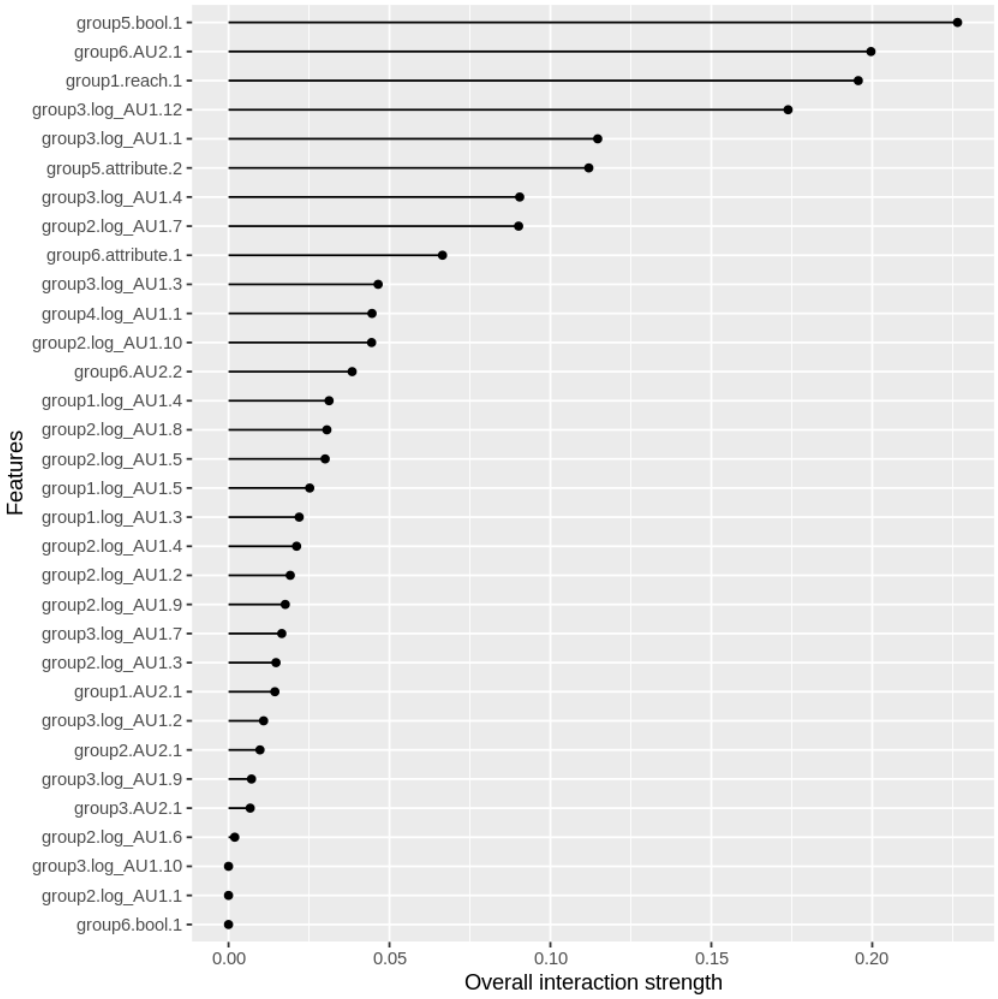

Figure 4.4: Overall feature interactions based on the H-statistic (see Equation 3.51) for the best performing boosted tree model with binomial deviance loss. By comparing the strongest overall interaction effects; GROUP5/BOOL/1, GROUP6/AU2/1 and GROUP1/REACH/1; are suggested as the three leading interaction features.

To identify the strongest two-way interactions in the feature space, two-way

H-statistics were computed over the 5 top categorical or ordinal features (GROUP5/BOOL/1, GROUP6/AU2/1, GROUP1/REACH/1, GROUP5/ATTRIBUTE/2 and GROUP6/ATTRIBUTE/1) against all the other features, which can be found as Figure B.36 - Figure B.40 of Appendix B. The categorical and ordinal features were premiered due to the vast computational expense required for each continuous feature. Thereafter, 2-dimensional partial dependence plots were created for the 4 strongest pairwise interactions for more granular inference.

By examining the two dimensional PDPs for the pairwise strong interaction effects, one can uncover patterns hidden in the marginal counterparts. For example, by the left top panel of Figure 4.5, it is seen that the earlier mentioned decrease effect towards the target for larger values of GROUP3/LOG_AU1/4, only holds for one of the levels of GROUP5/BOOL/1. The existence of such discrepancy was however indicated by the ICE plot as of Figure 4.3, showcasing how the different interpretation methods presented in this thesis can successfully work together. For confidentiality, further such two-dimensional PDPs will not be disclosed for this work.

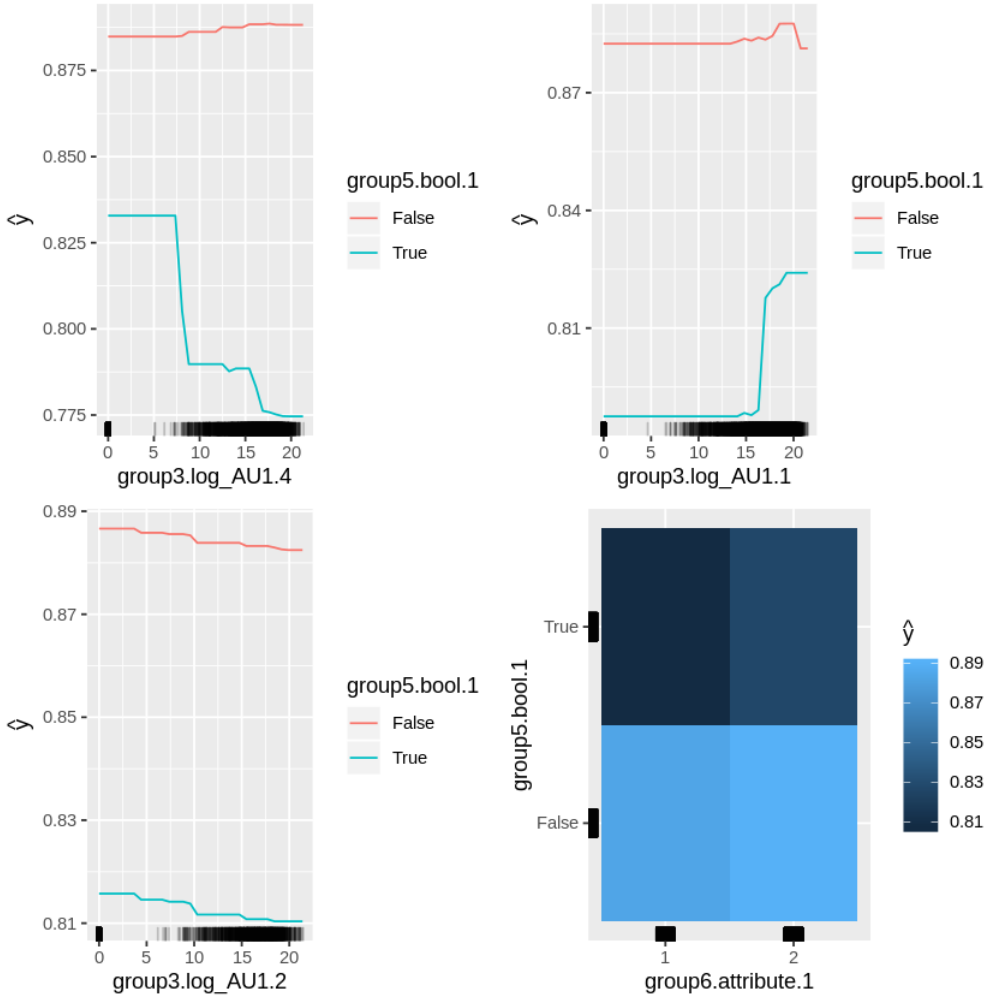

Figure 4.5: Two-dimensional partial dependence plots for GROUP5/BOOL/1 and the 4 variables with largest two-way interaction strength towards GROUP5/BOOL/1. The top left panel shows that the negative relationship between GROUP3/LOG_AU1/4 and the target (see Figure 4.2) seems to only hold for the TRUE level values of GROUP5/BOOL/1. The bottom left and top right panels indicate similar relationships towards the target across groups (levels), but where the FALSE level values of GROUP5/BOOL/1 have larger baseline (initial) predictive values. The bottom right panel shows the two-dimensional PDP for two categorical/binary variables with two levels each, hence four combinations to consider. The combination of FALSE as level value for GROUP5/BOOL/1, and 2 for GROUP6/ATTRIBUTE/1, seems to yield the highest prediction values.

4.4 Future work

As mentioned in section 2.1, only a smaller feature space out of an immense available feature space that can quantitatively describe how various types of music listeners engage with an audio streaming service was explored. Hence, immediate recommendations for future work regard how other groups of features that capture more spectra of these personas affects the modelling. Moreover, a larger feature space could potentially highlight the positive aspects of boosting methods for tree models, by first identifying a smaller subset of the most influential features, and later how they interact with each other. Any results obtained on quantitative grounds with this methodology, including what have been done for this work, should also be assessed on qualitative grounds to ensure sane and meaningful insights.

Moreover, to give the model family of gradient boosting methods some justice, it is worth mentioning that several changes and generalizations can be explored for future work to most likely elevate the performance of the model. This includes more exhaustive hyperparameter space searches, further experiments with early stopping, evaluating more loss functions, and by utilizing other generalizations such as stochastic gradient boosting [13]. The latter works by fitting each base learner towards a subsample of the training data, obtained by sampling some fraction of the training data without replacement. This approach yields faster training, hence allowing for more exhaustive hyperparameter exploration (assuming time is somewhat scarce) and in many cases better predictive results [15, p. 365].

Ideally, one would want to test the interaction effects inferred from this study more thoroughly, in order to reduce the risk of spurious correlations. The same argument applies for any other interactions inferred from further studies utilizing the same research methodology. One way of doing that is through randomized experiments, where users are assigned into "treatment" groups randomly, for the greater purpose of marginalizing out confounding effects. These groups could then be studied by e.g. ANOVA models, that study the variances among and between groups. However, as experiment designs with randomization tend to be expensive and time-consuming to conduct, it would serve as useful prior knowledge to identify good feature interaction candidates from observational data. Hence, further studies centered around a larger feature space could profit from a richer identification ability of strong interactions from observational data, serving as candidates for more expensive randomized studies to validate true effects.

On the same note regarding the proposed interpretation methods, the Partial Dependence Plot (PDP) is, as mentioned in subsection 3.4.2, built upon a somewhat naive assumption about independence between the studied feature subset $x_{\mathcal{S}}$ and its complement $x_{\mathcal{C}}$. Although interaction strength statistics, such as the H-statistic (see subsection 3.4.3), can provide some insights towards this assumption, the independence assumption might impose some biases towards the interpretation. Another method discussed in [20, sec. 5.3] is the so called Accumulated Local Effects (ALE) plots. The ALE plots are based on averaging the predictions over a conditional distribution $P(x_{\mathcal{C}} \mid x_{\mathcal{S}})$, to be compared with the marginal distribution $P(x_{\mathcal{S}}, x_{\mathcal{C}})$ which the PDP averages over. Therefore, the ALE plots are unbiased [20, sec. 5.3] and hence do not rely on any independence assumptions between features. On the downside, however, ALE plots do not enjoy similar nice relationships towards the ICE plots, and these two plot types cannot be used together as for the PDP. Moreover, the ALE plots are parametric in the sense of choosing a number of intervals to average the conditional distribution over, which for poor choices of the number of intervals can be challenging to interpret.

Regarding any recommendations stemming from the results of this work, it would be useful to study the response of recommendations. In other words, one cannot presumably force the user into engaging with e.g. a non-utilized service feature. Therefore, to better understand which actions should be taken based on the result of this work, one would need to study how the users respond towards recommendations or incentives change their behaviours on the service. Considering that different service features compete in exposure towards the user, an important latent variable would presumably be whether the user actively ignores this feature or whether it is undiscovered. Future work could also stem around a recommendation framework based on the (near) Weibull fits for the AU1 measures, highlighted in section 2.3. The high-level idea would be to premiere offerings which increases the probability of high engagement (high AU1 values) and simultaneously increases the probability of retention, globally or on some cohort level.

If the target variable could be relaxed in time, meaning if the study is not bounded towards a binary 45-day retention target, one could instead utilize methods from survival analysis to study failure times (in this context time of churn/quitting the service) and thereby infer *survival functions*. The survival functions regard the probability for a failure time to exceed some threshold point t in time. Some common methods suitable for further studies, assuming such target relaxation is possible, include the non-parametric Kaplan-Meier estimator [2, p. 70] and Cox proportional hazards regression model [2, p. 34].

Both can be used to infer the survival functions for the whole population as a whole, or for specific cohorts.

Finally, at the time of writing, it is inevitable to mention the recent outbreak of the COVID-19 disease [30] (caused by the SARS-CoV-2 virus) and its potential effects of the study. As it has been shown that the audio streaming behaviours from users have been different during the time after various societal restrictions and lockdowns have been enforced all over the world [26], the timeline of pandemic and studied time period are evidently important to consider, especially in the Northern American region in which the study was conducted. Recapping the time delimitations from section 1.3, and the objective of measuring retention after 45 days, the obtained samples regarded the time period 2019-12-01 to 2020-02-26. The World Health Organization (WHO) labelled COVID-19 as a pandemic at the 11th of March 2020 [1], and shortly after, California declared at the 19th of March the first state-wide order for their residents to stay in their homes [21]. Presumably, the effects, if any, of the pandemic towards the user behaviour within an audio streaming service should be negligible.

Chapter 5

Conclusion

In this study, the theoretical framework of boosting by combining multiple decision trees into an ensemble or "committee" for increased predictive power was presented. For the context of an audio streaming service, where customer retention is of great importance for the business, this work explored whether the boosted tree models could predict customer retention and infer the statistical relationships forming the basis of such predictions. Given that previous qualitative research has shown that different types of users stay on an audio streaming service for different reasons, the study also explored how two-way interactions between features could be utilized for a more granular view of the statistical relationships between the features and the target.

Empirically, it was shown that the boosted tree models performed better in predictive measures than a baseline logistic regression model through an increase in AUC on heldout data by approximately 3.5 percentage points (0.7772 and 0.7422 respectively). Therefore, this study suggests relatively low marginal benefits of the usage of a rather complex class of models of the boosted trees for the explored set of features. However, various presented interpretation methods allowed for the identification of influential features and their relationships towards the target and towards other features.

Lastly, it is worth emphasizing that this work merely scratched the surface of evaluating the performance of boosting tree models for data mining approaches with a massively larger feature space, deduced from an audio streaming service with a vast amount of available data. Hence, further studies centered around larger feature spaces, further generalizations of the boosted tree model and their interpretations could paint a vastly different picture than what this work suggests in terms of finding the high-retaining user, both on predictive and inferential grounds.

References

[1] WHO Director-General's opening remarks at the media briefing on COVID-19 - 11 March 2020. https://www.who.int/dg/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020, 2020. Accessed: 2020-05-05.

[2] O. Aalen, Borgan, and H. Gjessing. *Survival and Event History Analysis: A Process Point of View*. 01 2008. ISBN 0-387-20287-0. doi: 10.1007/978-0-387-68560-1.

[3] A. Agresti. *An Introduction to Categorical Data Analysis*. Wiley Series in Probability and Statistics. Wiley, 2007. ISBN 9780471226185. URL https://books.google.se/books?id=_dJ7mAEACAAJ.

[4] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(10):281–305, 2012. URL http://jmlr.org/papers/v13/bergstra12a.html.

[5] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[6] L. Breiman, J. Friedman, C. Stone, and R. Olshen. *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984. ISBN 9780412048418. URL https://books.google.se/books?id=JwQx-WOmSyQC.

[7] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016. URL http://arxiv.org/abs/1603.02754.

[8] P. H. Cho. Introduction to boosted trees. https://github.com/dmlc/xgboost/blob/master/doc/tutorials/model.rst, 2019. Accessed: 2020-05-02.

[9] P. Domingos. A unifeid bias-variance decomposition and its applications. pages 231–238, 01 2000.

[10] a. Duhigg, Charles. *The power of habit : why we do what we do in life and business*. Random House Trade paperback edition. New York : Random House Trade Paperbacks, 2014., 2014. URL https://search.library.wisc.edu/catalog/9910195246302121.

[11] Y. Freund and R. E. Schapire. A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119 – 139, 1997. ISSN 0022-0000. doi: https://doi.org/10.1006/jcss.1997.1504. URL http://www.sciencedirect.com/science/article/pii/S002200009791504X.

[12] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *Ann. Statist.*, 28(2):337–407, 04 2000. doi: 10.1214/aos/1016218223. URL https://doi.org/10.1214/aos/1016218223.

[13] J. H. Friedman. Stochastic gradient boosting. *Computational Statistics Data Analysis*, 38(4):367–378, 2002. URL https://EconPapers.repec.org/RePEc:eee:csdana:v:38:y:2002:i:4:p:367-378.

[14] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24, 09 2013. doi: 10.1080/10618600.2014.907095.

[15] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[16] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 42(1):80–86, 2000. ISSN 00401706. URL http://www.jstor.org/stable/1271436.

[17] K. Koch. Using service design to create better, faster, stronger designers. https://spotify.design/articles/2019-07-11/using-service-design-to-create-better-faster-stronger-designers/, 2019. Accessed: 2020-05-07.

[18] C.-D. Lai, D. Murthy, and M. Xie. Weibull distributions and their applications. *Springer Handbook of Engineering Statistics*, Chapter 3: 63–78, 02 2006. doi: 10.1007/978-1-84628-288-1_3.

[19] N. Mann, R. Schafer, and N. Singpurwalla. *Methods for statistical analysis of reliability and life data*. Wiley Series in Probability and Statistics - Applied Probability and Statistics Section. Wiley, 1974. ISBN 9780471567370. URL https://books.google.se/books?id=yfdTAAAAMAAJ.

[20] C. Molnar. *Interpretable Machine Learning*. 2019. https://christophm.github.io/interpretable-ml-book/.

[21] C. Norwood. Most states have issued stay-at-home orders, but enforcement varies widely. https://www.pbs.org/newshour/politics/most-states-have-issued-stay-at-home-orders-but-enforcement-varies-widely, 2020. Accessed: 2020-05-06.

[22] P. E. Pfeifer. The optimal ratio of acquisition and retention costs. *Journal of Targeting, Measurement and Analysis for Marketing*, 13 (2):179–188, 2005. doi: 10.1057/palgrave.jt.5740142. URL https://doi.org/10.1057/palgrave.jt.5740142.

[23] L. Prechelt. Early stopping-but when? In G. B. Orr and K.-R. Müller, editors, *Neural Networks: Tricks of the Trade*, volume 1524 of *Lecture Notes in Computer Science*, pages 55–69. Springer, 1996. ISBN 3-540-65311-2. URL http://dblp.uni-trier.de/db/conf/nips/nips1996.html#Prechelt96.

[24] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948. URL http://dblp.uni-trier.de/db/journals/bstj/bstj27.html#Shannon48.

[25] Spotify. Spotify company info. https://newsroom.spotify.com/company-info/, 2020. Accessed: 2020-05-07.

[26] Spotify. How social distancing has shifted Spotify streaming. https://newsroom.spotify.com/2020-03-30/how-social-distancing-has-shifted-spotify-streaming/, 2020. Accessed: 2020-05-05.

[27] Spotify. Spotify Premium Duo. Music for two. https://www.spotify.com/uk/duo/, 2020. Accessed: 2020-05-05.

[28] Spotify. Spotify Premium Family. https://www.spotify.com/uk/family/, 2020. Accessed: 2020-05-05.

[29] Spotify. Save 50% on Spotify Premium for Students. https://www.spotify.com/uk/student/, 2020. Accessed: 2020-05-05.

[30] K. Sternudd. Spotlight on COVID-19. https://ki.se/en/research/spotlight-on-covid-19, 2020. Accessed: 2020-05-05.

[31] J. Surowiecki. *The Wisdom of Crowds*. 01 2005.

[32] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1): 267–288, 1996. ISSN 00359246. URL http://www.jstor.org/stable/2346178.

[33] M. Torres De Souza, O. Hörding, and S. Karol. Spotify the story of spotify personas. https://spotify.design/articles/2019-03-26/the-story-of-spotify-personas/, 2019. Accessed: 2020-05-07.

[34] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[35] Z. Zhang, C. Chen, G. Dai, W.-J. Li, and D.-Y. Yeung. Multi-category large margin classification methods: Hinge losses vs. coherence functions. *Artificial Intelligence*, 215:55 – 78, 2014. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2014.06.002. URL http://www.sciencedirect.com/science/article/pii/S0004370214000794.

Appendices

Appendix A

Calculations

Lemma A.1. *Let $\mathbb{E}$ denote the expectation over some data distribution P, which (x, y) are realizations of. Assuming that the target variable $y = f(x) + \epsilon$, where f is some true model and ϵ noise with zero-valued expected value. Then for a squared error loss function, the expected prediction risk for a fitted model $f_\theta(x)$, evaluated in some point x_0, can be decomposed as*

$$\mathbb{E}\left[(y - f_\theta(x_0))^2\right] = Var(\epsilon) + (f(x_0) - \mathbb{E}\left[f_\theta(x_0)\right])^2 + Var(f(x_0) - f_\theta(x_0)) \ .$$

Proof. Starting with the quadratic expansion of the left-hand side, and by first not condition on $x = x_0$, it follows by the linearity of an expectation that

$$\mathbb{E}\left[(y - f_\theta(x))^2\right] = \mathbb{E}\left[y^2\right] + \mathbb{E}\left[f_\theta(x)^2\right] - 2\mathbb{E}\left[y f_\theta(x)\right] \ .$$

Using the definition of the variance in terms of the second moments, and sub-consequently the relationship $y = f(x) + \epsilon$, one obtains that

$$\begin{aligned}\mathbb{E}\left[(y - f_\theta(x))^2\right] &= \mathrm{Var}\left(y\right) + \mathbb{E}\left[y\right]^2 + \mathrm{Var}\left(f_\theta(x)\right) + \mathbb{E}\left[f_\theta(x)\right]^2 - 2\mathbb{E}\left[y f_\theta(x)\right] \ , \\ &= \mathrm{Var}\left(f(x) + \epsilon\right) + \mathbb{E}\left[f(x) + \epsilon\right]^2 + \mathrm{Var}\left(f_\theta(x)\right) + \mathbb{E}\left[f_\theta(x)\right]^2 \\ &\quad - 2\mathbb{E}\left[(f(x) + \epsilon) f_\theta(x)\right] \ .\end{aligned}$$

Furthermore, utilizing that ϵ is independent of P and that $\mathbb{E}\left[\epsilon\right] = 0$, various well-known properties of the expectation and variance can be exploited, hence

$$\begin{aligned}\mathbb{E}\left[(y - f_\theta(x))^2\right] &= \operatorname{Var}(f(x)) + \operatorname{Var}(\epsilon) + (\mathbb{E}[f(x)] + \cancel{\mathbb{E}[\epsilon]})^2 + \operatorname{Var}(f_\theta(x)) \\ &\quad + \mathbb{E}[f_\theta(x)]^2 - 2\mathbb{E}[f(x)f_\theta(x)] - \cancel{2\mathbb{E}[\epsilon]\mathbb{E}[f_\theta(x)]}, \\ &= \operatorname{Var}(f(x)) + \operatorname{Var}(\epsilon) + \mathbb{E}[f(x)]^2 + \operatorname{Var}(f_\theta(x)) + \mathbb{E}[f_\theta(x)]^2 \\ &\quad - 2\mathbb{E}[f(x)]\,\mathbb{E}[f_\theta(x)] - 2\operatorname{Cov}(f(x), f_\theta(x)) \ ,\end{aligned}$$

where the property $\mathbb{E}[XY] = \mathbb{E}[X]\,\mathbb{E}[Y] + \operatorname{Cov}(X, Y)$ was used for the last equality. Finally, by the summation property of the variance, i.e. that $\operatorname{Var}(X - Y) = \operatorname{Var}(X) + \operatorname{Var}(Y) - \operatorname{Cov}(X, Y)$, it yields that

$$\mathbb{E}\left[(y - f_\theta(x))^2\right] = \operatorname{Var}(\epsilon) + (\mathbb{E}[f(x)] - \mathbb{E}[f_\theta(x)])^2 + \operatorname{Var}(f(x) - f_\theta(x)) \, .$$

Lastly, by imposing the condition of $x = x_0$, it follows that $\mathbb{E}[f(x_0)]$ is a constant, hence

$$\mathbb{E}\left[(y - f_\theta(x_0))^2\right] = \operatorname{Var}(\epsilon) + (f(x_0) - \mathbb{E}[f_\theta(x_0)])^2 + \operatorname{Var}(f(x_0) - f_\theta(x_0)) \ .$$

□

Lemma A.2. *Let $X_1, \ldots, X_m$ be cell counts for a contingency table with m cells. Moreover, let $x_1, \ldots, x_m$ be realizations from $X_1, \ldots, X_m$ such that $\sum_{i=1}^m x_i = n$. Recognized as a multinomial sampling strategy, with joint probability for all cell counts as*

$$P(X_1 = x_1, \ldots, X_m = x_m) = \frac{n!}{x_1! \ldots x_m!} \pi_1^{x_1} \ldots \pi_m^{x_m} \, ,$$

then the Maximum Likelihood Estimate (MLE) of each event probability π_i is $\frac{x_i}{\lambda}$.

Proof. The likelihood obtains as the joint probability function, thus

$$L(\pi) = n! \prod_{i=1}^m \frac{\pi_i^{x_i}}{x_i!} \, ,$$

and hence the log-likelihood as

$$
\begin{aligned}
l(\pi) = \log L(p) &= \log \left(n! \prod_{i=1}^{m} \frac{\pi_i^{x_i}}{x_i!} \right) , \\
&= \log n! + \sum_{i=1}^{m} \log \left(\frac{\pi_i^{x_i}}{x_i!} \right) , \\
&= \log n! + \sum_{i=1}^{m} x_i \log \pi_i - \sum_{i=1}^{m} \log x_i! \, .
\end{aligned}
$$

The Lagrange multiplier method can then be used to introduce the constraint that $\sum_{i=1}^{K} \pi_i = 1$. Let $\mathcal{L}(\pi; \lambda)$ denote the Lagrange multiplier function, then

$$
\mathcal{L}(\pi; \lambda) = \log n! + \sum_{i=1}^{m} x_i \log \pi_i - \sum_{i=1}^{m} \log x_i! + \lambda \left(1 - \sum_{i=1}^{m} \pi_i \right) .
$$

Lastly, by posing the i^{th} derivative to be 0, one obtains

$$
\begin{aligned}
\frac{\partial \mathcal{L}(\pi; \lambda)}{\partial \pi_i} &= 0 \Rightarrow \\
\frac{x_i}{\pi_i} - \lambda &= 0 \Rightarrow \\
\hat{\pi}_i^{\text{MLE}} &= \frac{x_i}{\lambda} \, .
\end{aligned}
$$

□

Lemma A.3. *For a discrete random variable X with outcome space $\{1, \ldots, K\}$, the (discrete) uniform distribution maximizes the (Shannon) entropy.*

Proof. Let $P(X)$ denote the probability density function for X. Furthermore, as maximization of the entropy

$$
H(X) = - \sum_{X} P(X) \log P(X) \, ,
$$

is subject towards the constraint of $\sum_X P(X) = 1$, the Lagrange multiplier method will be used. Let $\mathcal{L}(P(X), \lambda)$ denote the Lagrangian function, then

$$
\mathcal{L}(P(X), \lambda) = - \sum_{X} P(X) \log P(X) + \lambda (1 - \sum_{X} P(X)) \, .
$$

Setting the partial derivatives of the Lagrangian, with respect to $P(X)$ and λ respectively, equal to zero yields

$$\begin{cases} \frac{\partial \mathcal{L}(P(X),\lambda)}{\partial P(X)} & = 0 = -\log P(X) - 1 + \lambda \\ \frac{\partial \mathcal{L}(P(X),\lambda)}{\lambda} & = 0 = 1 - \sum_X P(X) \end{cases} \Rightarrow$$

$$P(X) = e^{(\lambda - 1)} \,,$$

subject to $\sum_X P(X) = 1$. Hence, by the limit definition of e^x, it follows that

$$P(X) = \frac{1}{N+1} \,,$$

which by uniqueness implies that the discrete uniform distribution indeed maximizes the entropy of X. $\square$

Appendix B

Figures

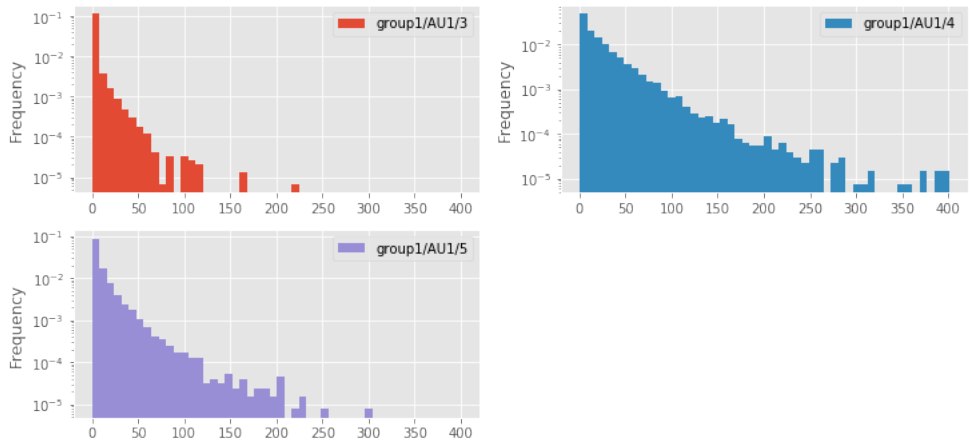

Figure B.1: Histograms for AU1 features in group 1 on $\log_{10}$-scale for the relative frequency. Note that the AU1 measures have been scaled by some positive factor of $\alpha \ll 1$ to preserve confidentially and make the range smaller.

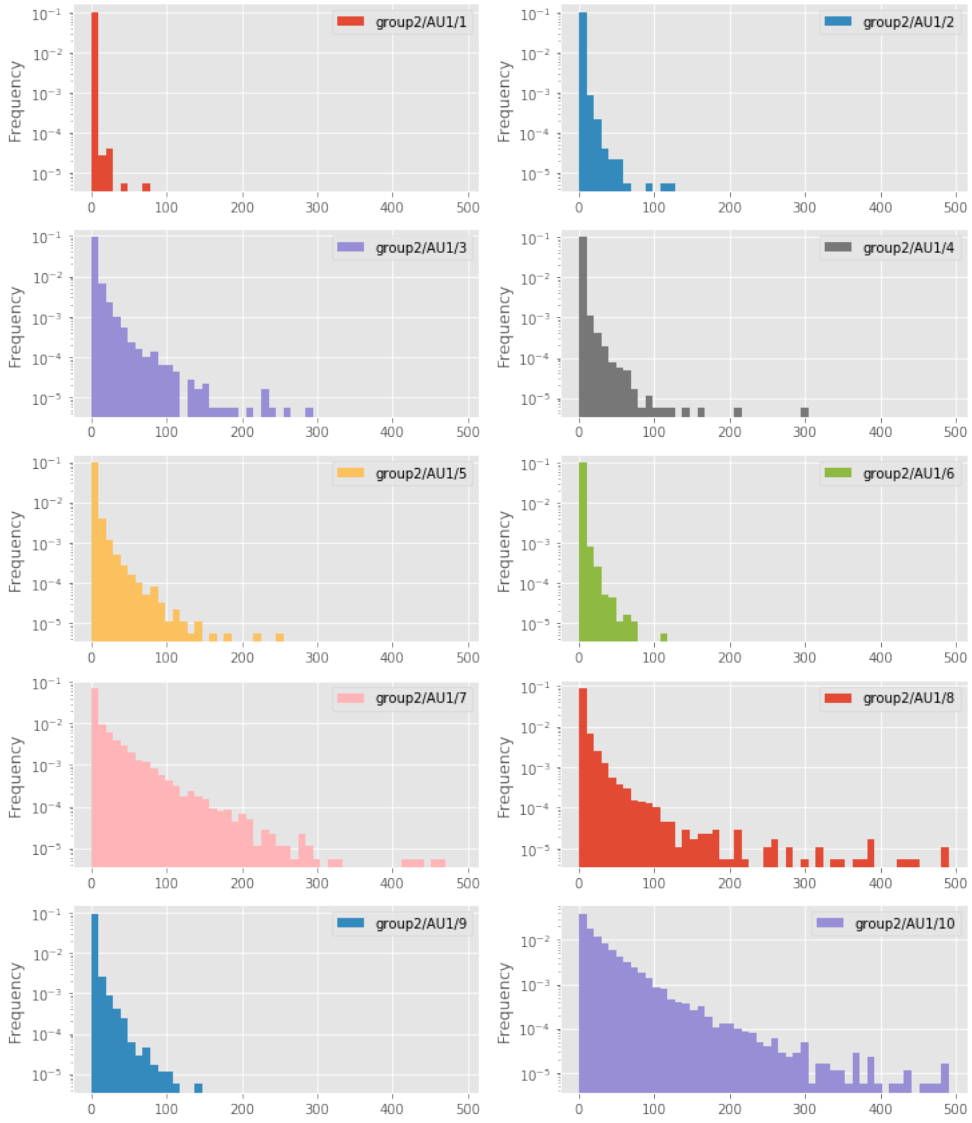

Figure B.2: Histograms for AU1 features in group 2 on $\log_{10}$-scale for the relative frequency. Note that the AU1 measures have been scaled by some positive factor of $\alpha \ll 1$ to preserve confidentially and make the range smaller.

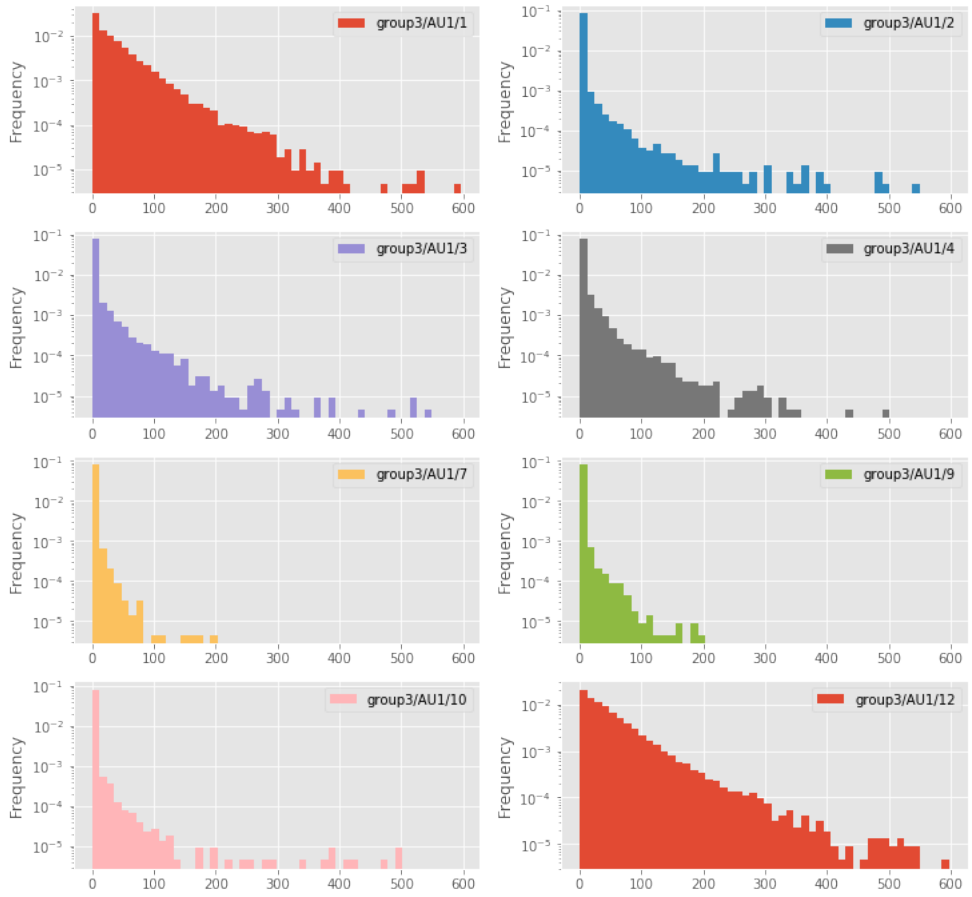

Figure B.3: Histograms for AU1 features in group 3 on $\log_{10}$-scale for the relative frequency. Note that the AU1 measures have been scaled by some positive factor of $\alpha \ll 1$ to preserve confidentially and make the range smaller.

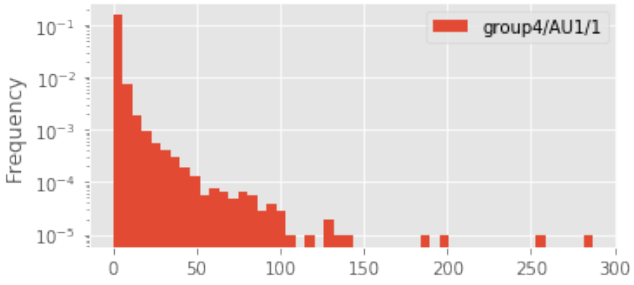

Figure B.4: Histograms for AU1 features in group 4 on $\log_{10}$-scale for the relative frequency. Note that the AU1 measures have been scaled by some positive factor of $\alpha \ll 1$ to preserve confidentially and make the range smaller.

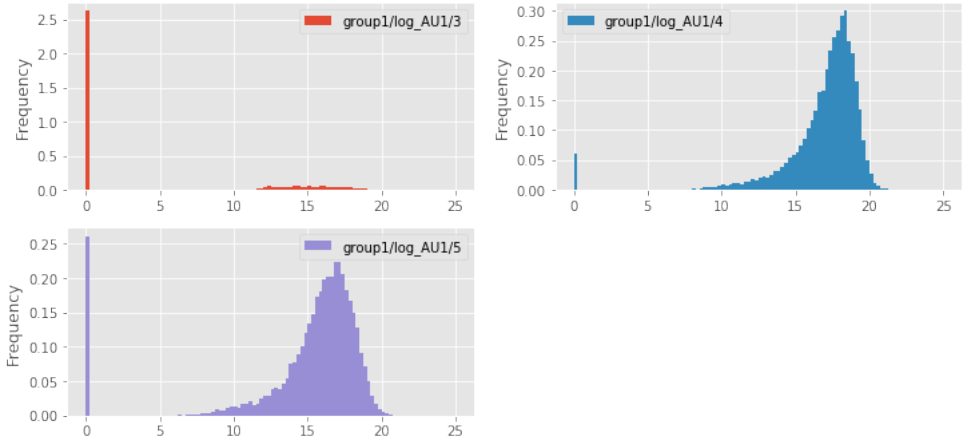

Figure B.5: Histograms for the AU1 features in group1 (scaled by some factor β) on $\log(x+1)$ scale.

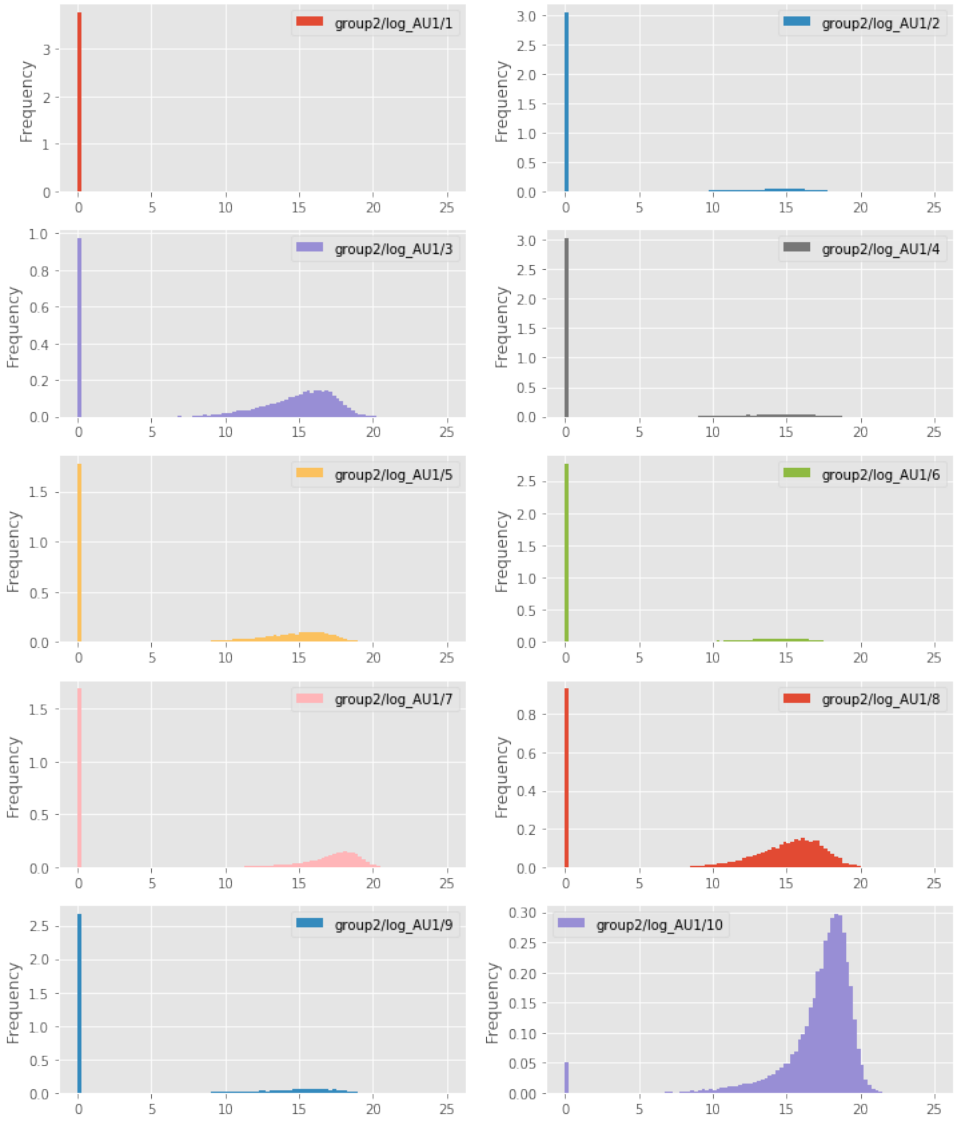

Figure B.6: Histograms for the AU1 features in group2 (scaled by some factor β) on $\log(x+1)$ scale.

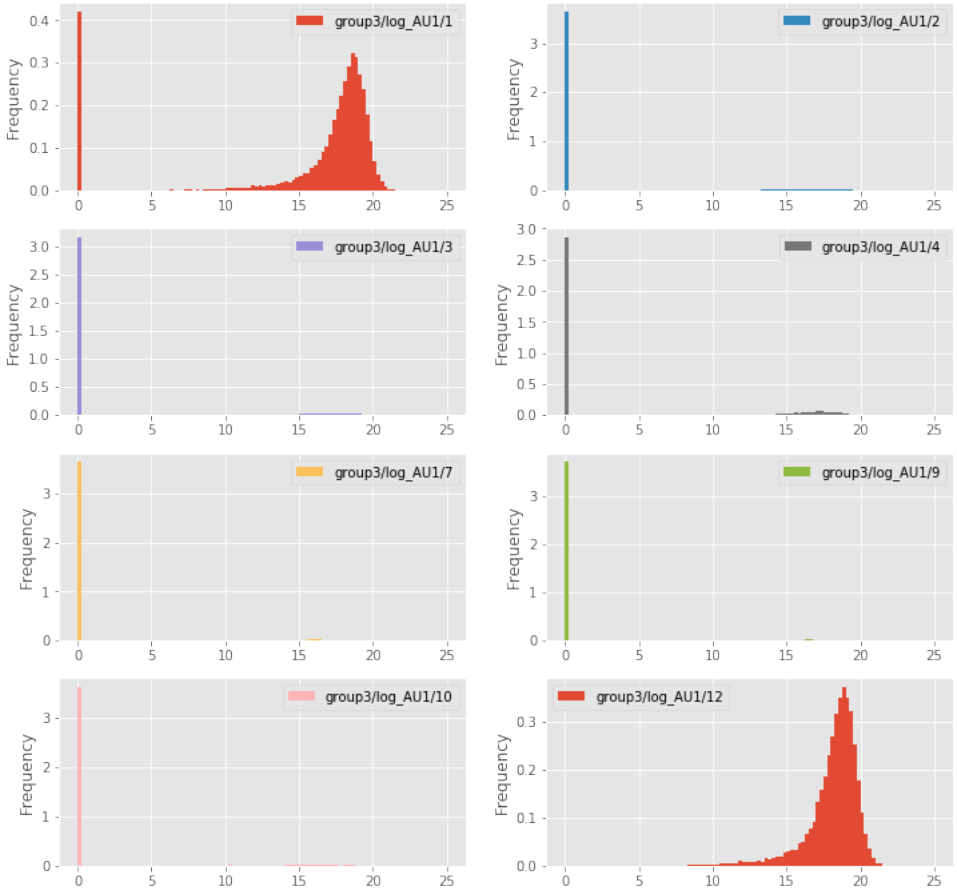

Figure B.7: Histograms for the AU1 features in group3 (scaled by some factor β) on $\log(x+1)$ scale.

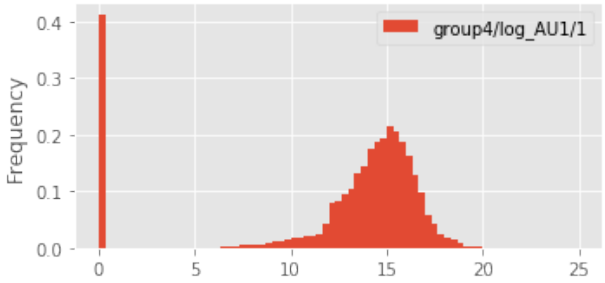

Figure B.8: Histograms for the AU1 features in group4 (scaled by some factor β) on $\log(x+1)$ scale.

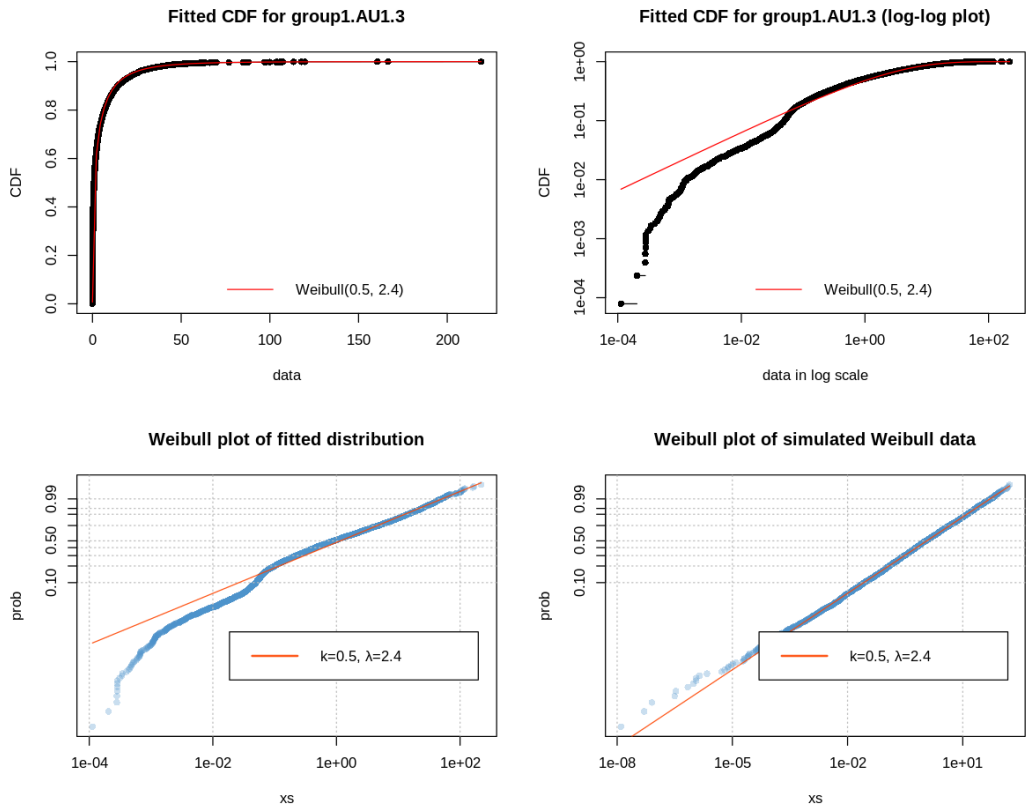

Figure B.9: Weibull fitting towards the positive values of GROUP1/AU1/3. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale for to spot deviations easier. The bottom panels shows the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

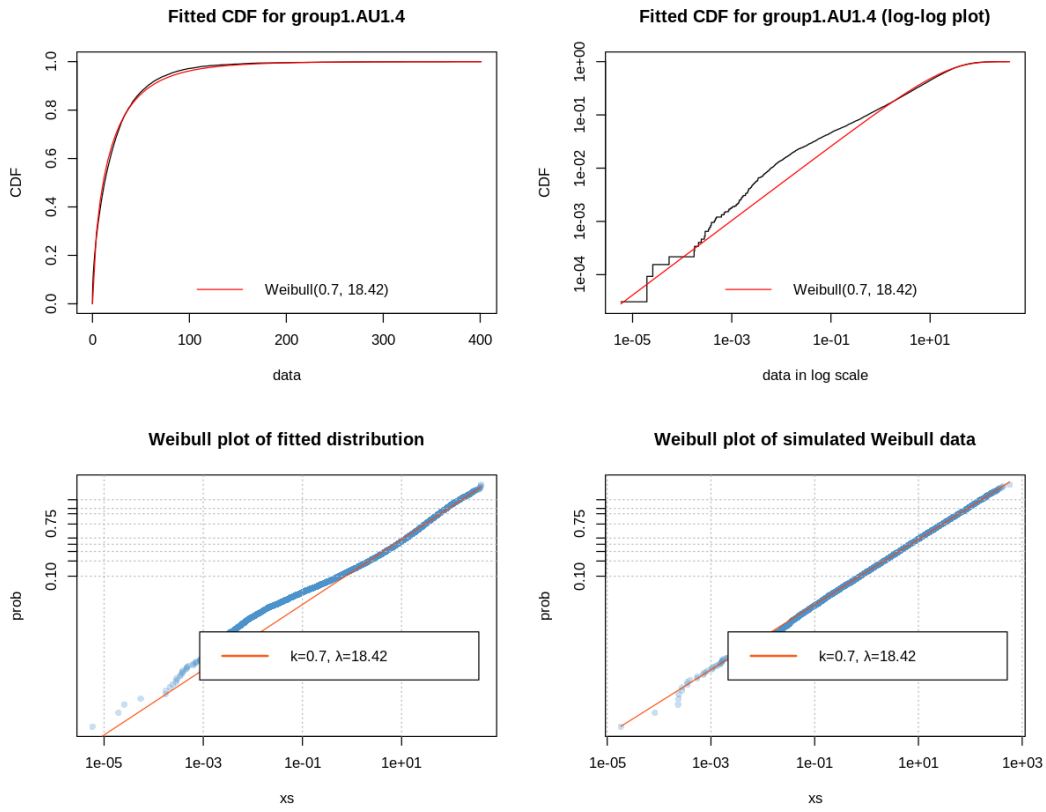

Figure B.10: Weibull fitting towards the positive values of GROUP1/AU1/4. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale for to spot deviations easier. The bottom panels shows the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

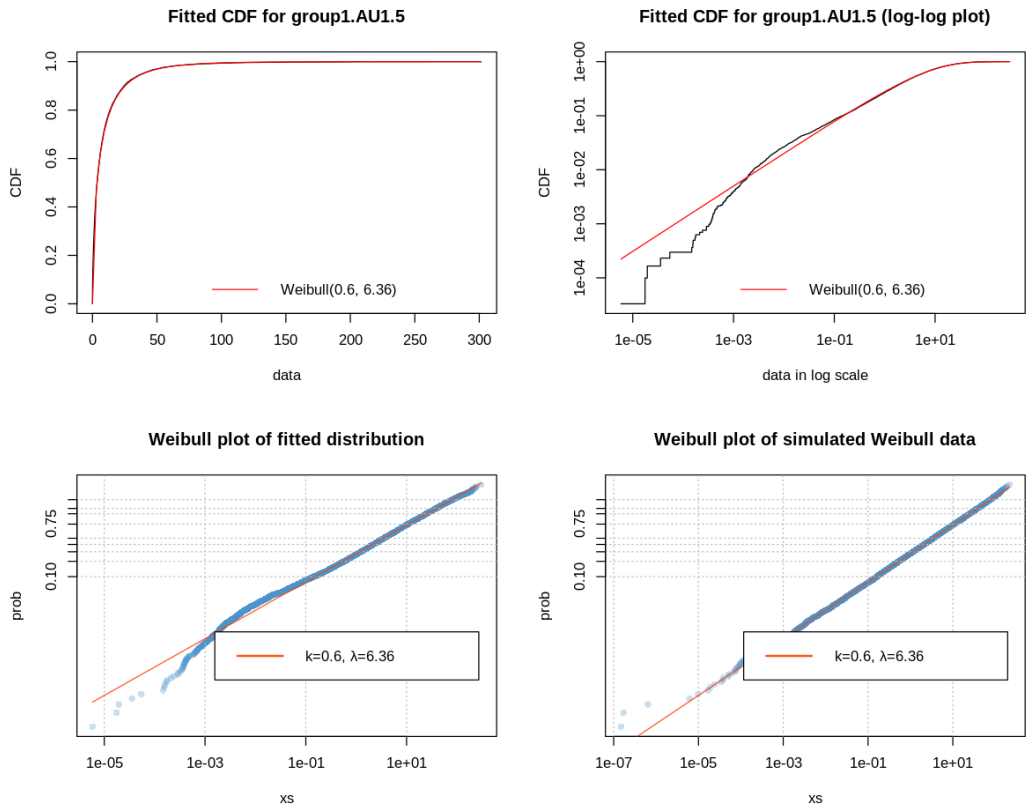

Figure B.11: Weibull fitting towards the positive values of GROUP1/AU1/5. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale for to spot deviations easier. The bottom panels shows the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

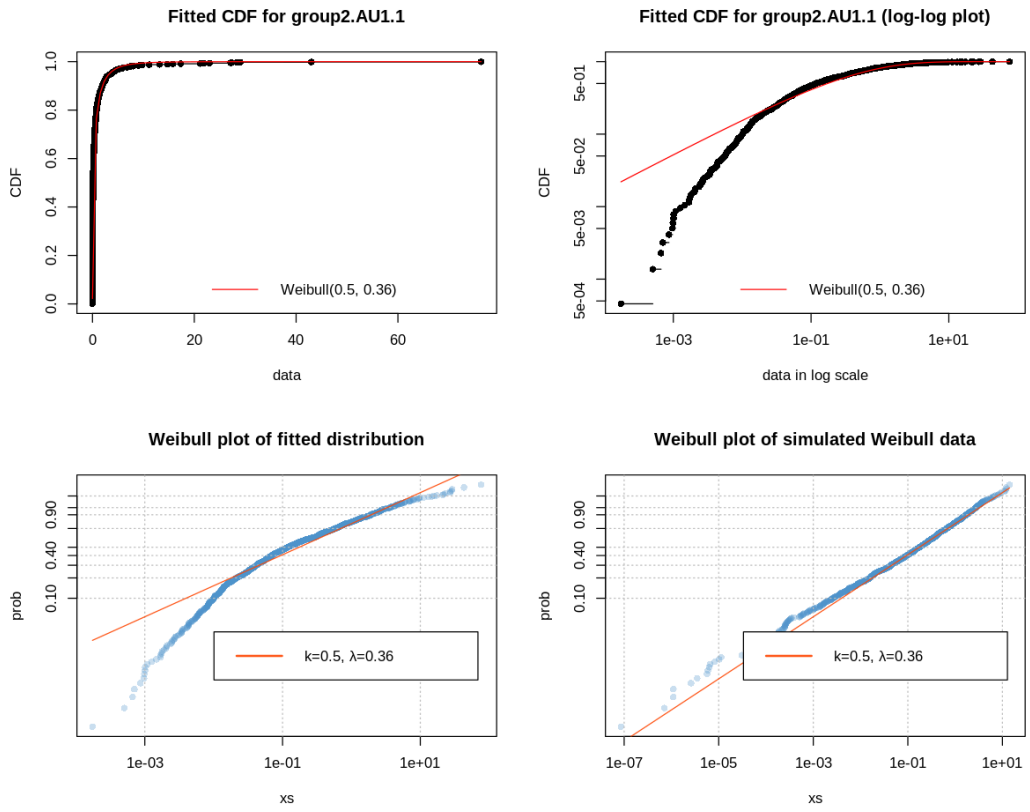

Figure B.12: Weibull fitting towards the positive values of GROUP2/AU1/1. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale for to spot deviations easier. The bottom panels shows the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

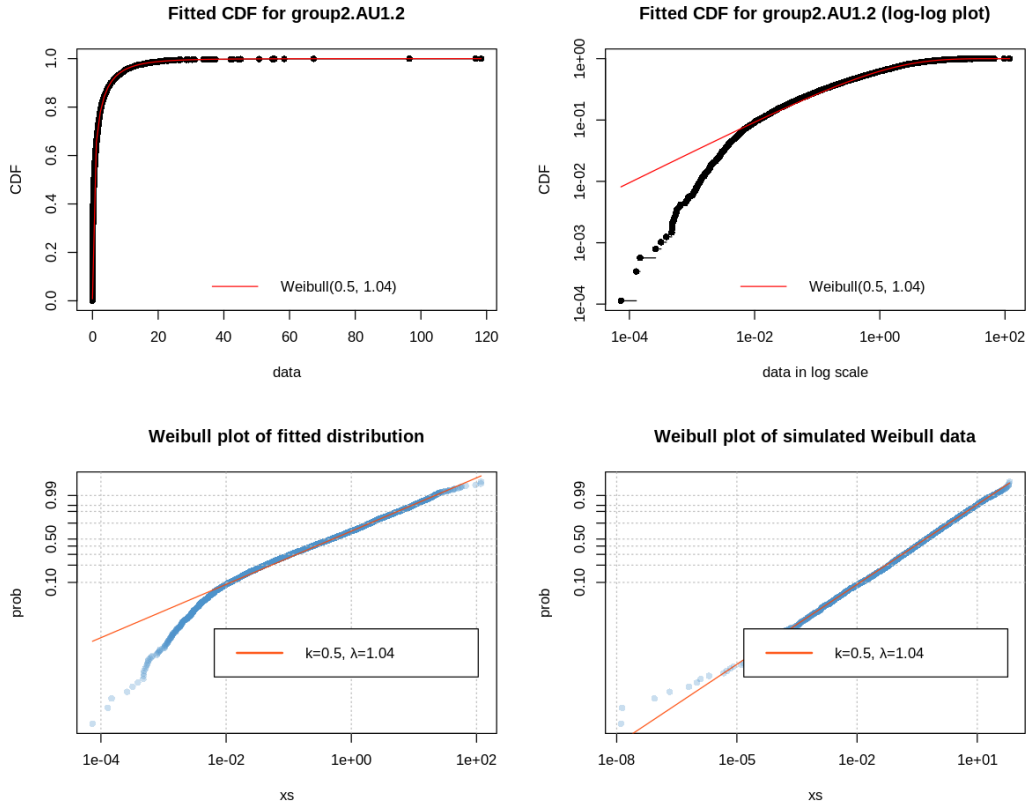

Figure B.13: Weibull fitting towards the positive values of GROUP2/AU1/2. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale for to spot deviations easier. The bottom panels shows the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

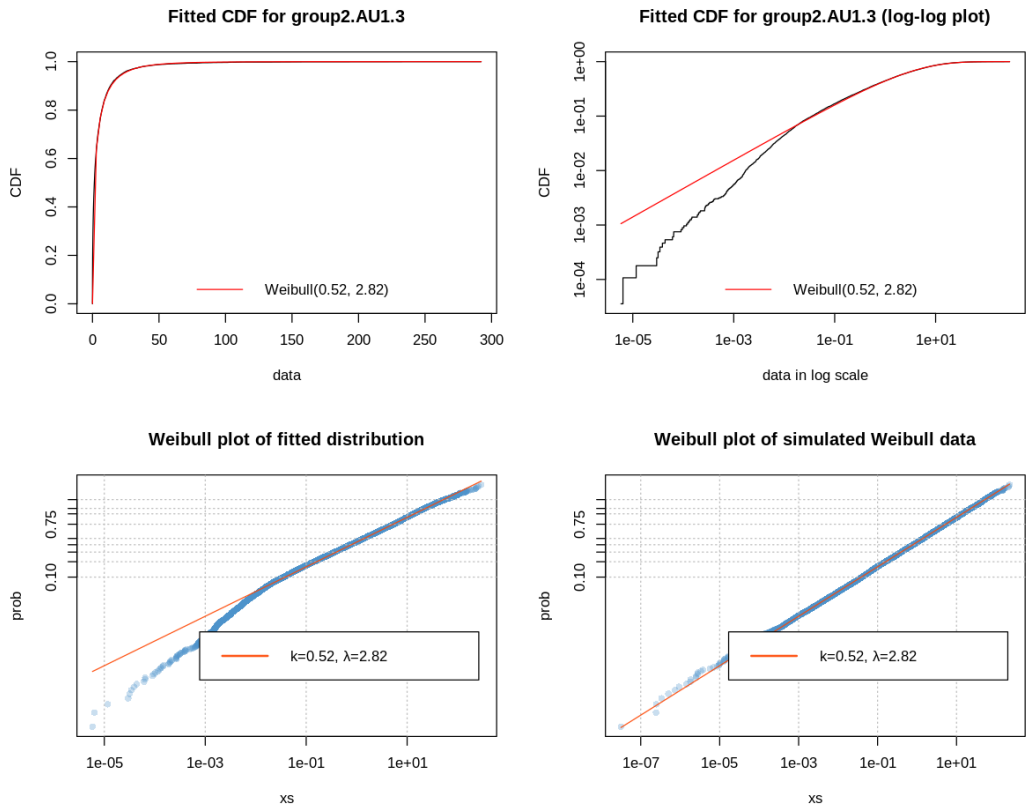

Figure B.14: Weibull fitting towards the positive values of GROUP2/AU1/3. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale for to spot deviations easier. The bottom panels shows the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

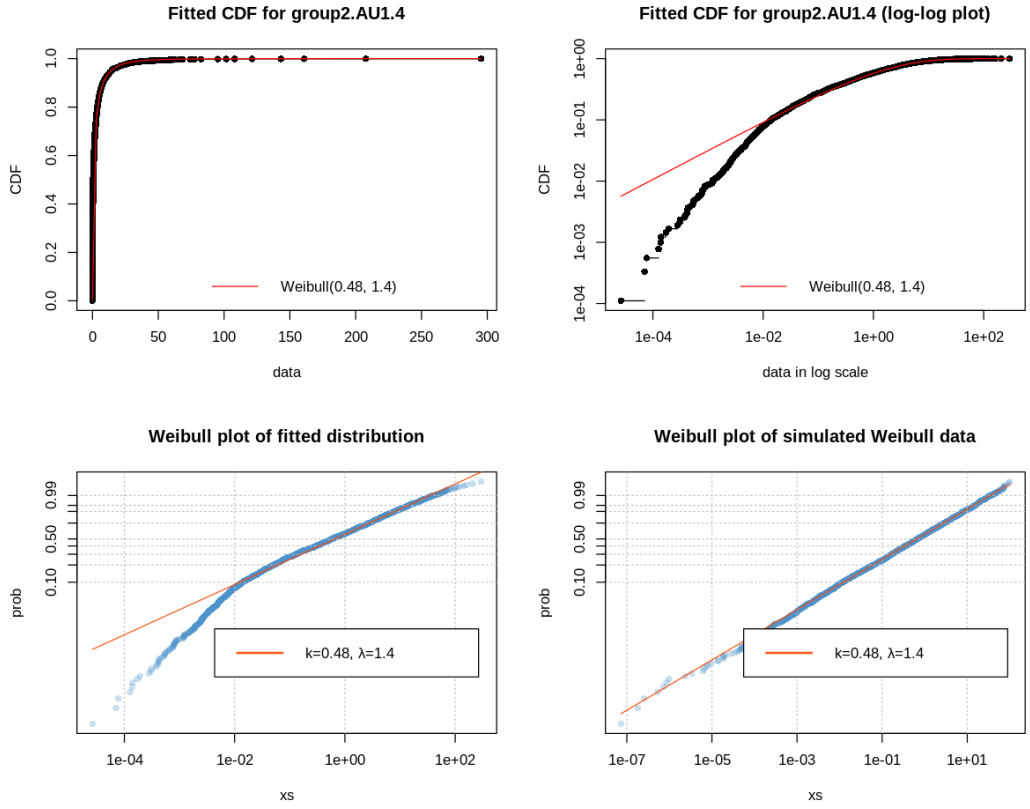

Figure B.15: Weibull fitting towards the positive values of GROUP2/AU1/4. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale for to spot deviations easier. The bottom panels shows the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

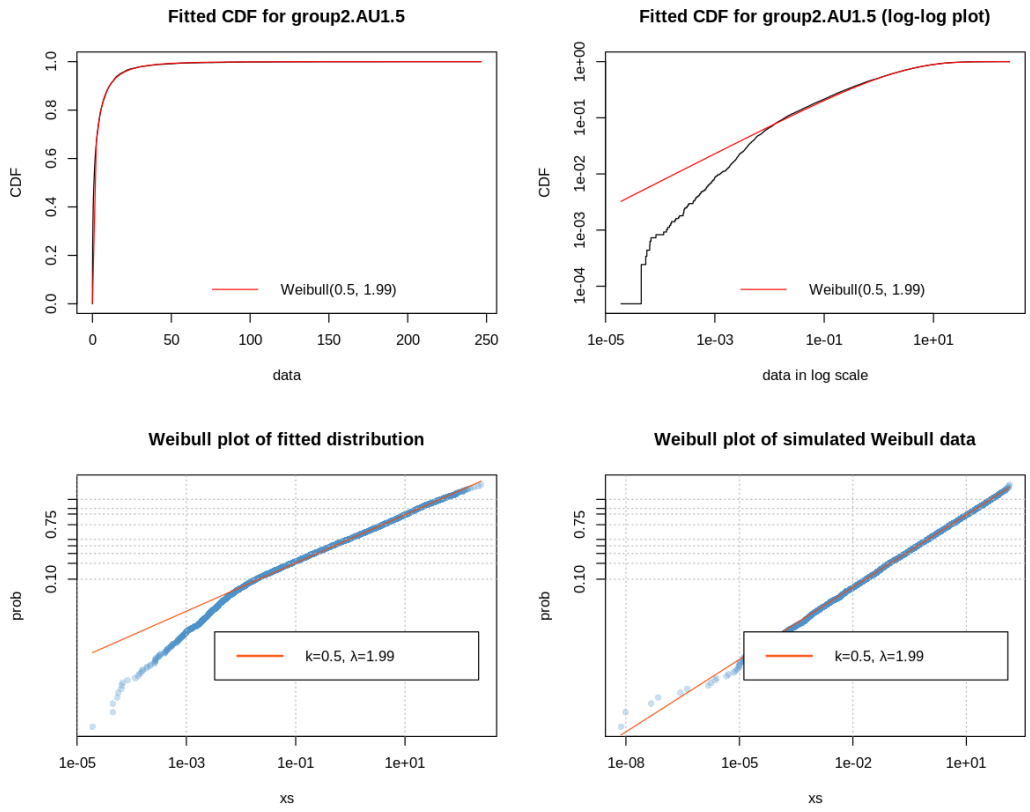

Figure B.16: Weibull fitting towards the positive values of GROUP2/AU1/5. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale for to spot deviations easier. The bottom panels shows the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

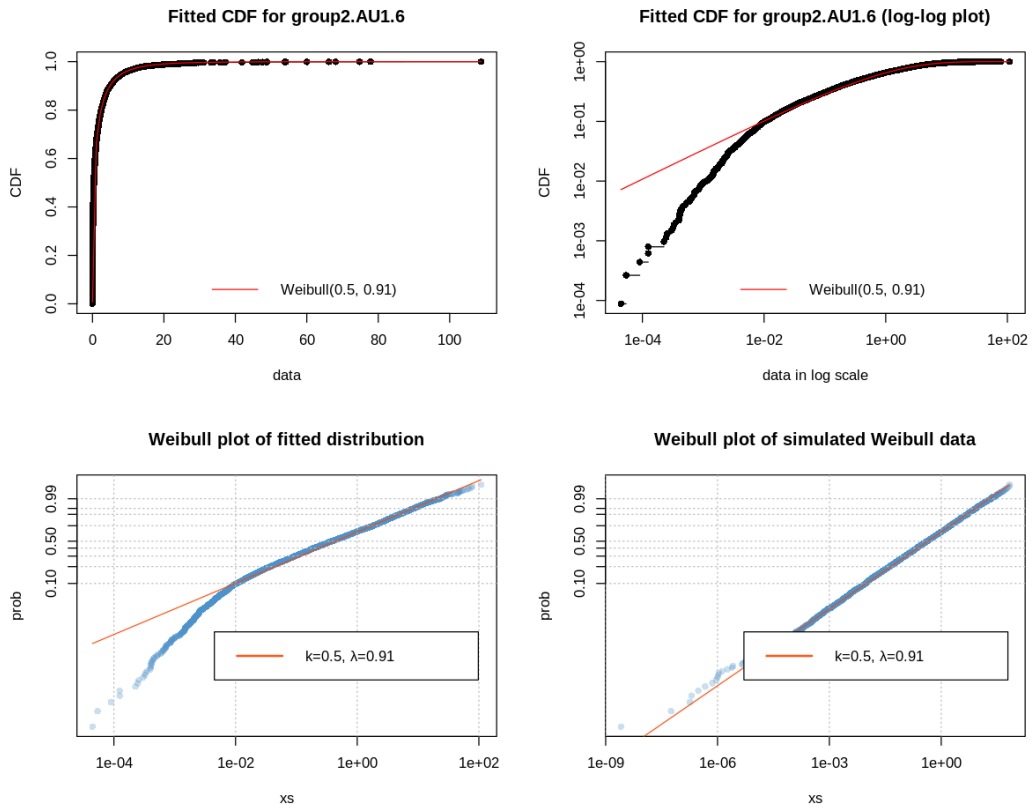

Figure B.17: Weibull fitting towards the positive values of GROUP2/AU1/6. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale for to spot deviations easier. The bottom panels shows the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

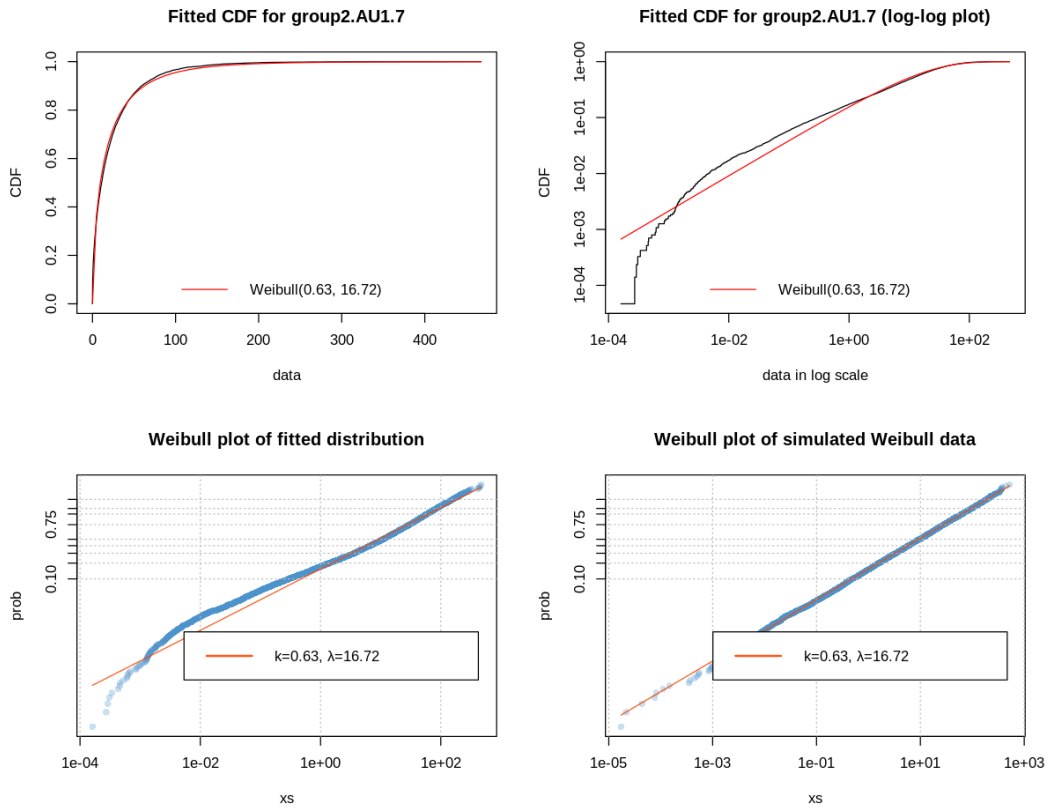

Figure B.18: Weibull fitting towards the positive values of GROUP2/AU1/7. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale for to spot deviations easier. The bottom panels shows the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

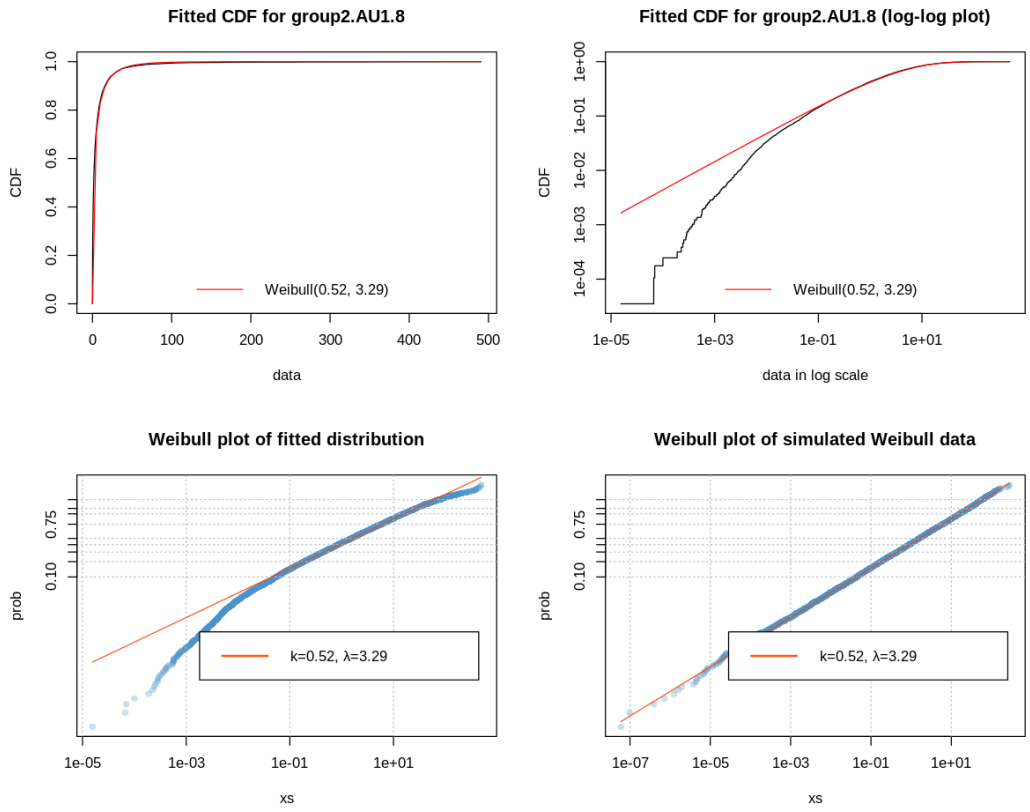

Figure B.19: Weibull fitting towards the positive values of GROUP2/AU1/8. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale for to spot deviations easier. The bottom panels shows the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

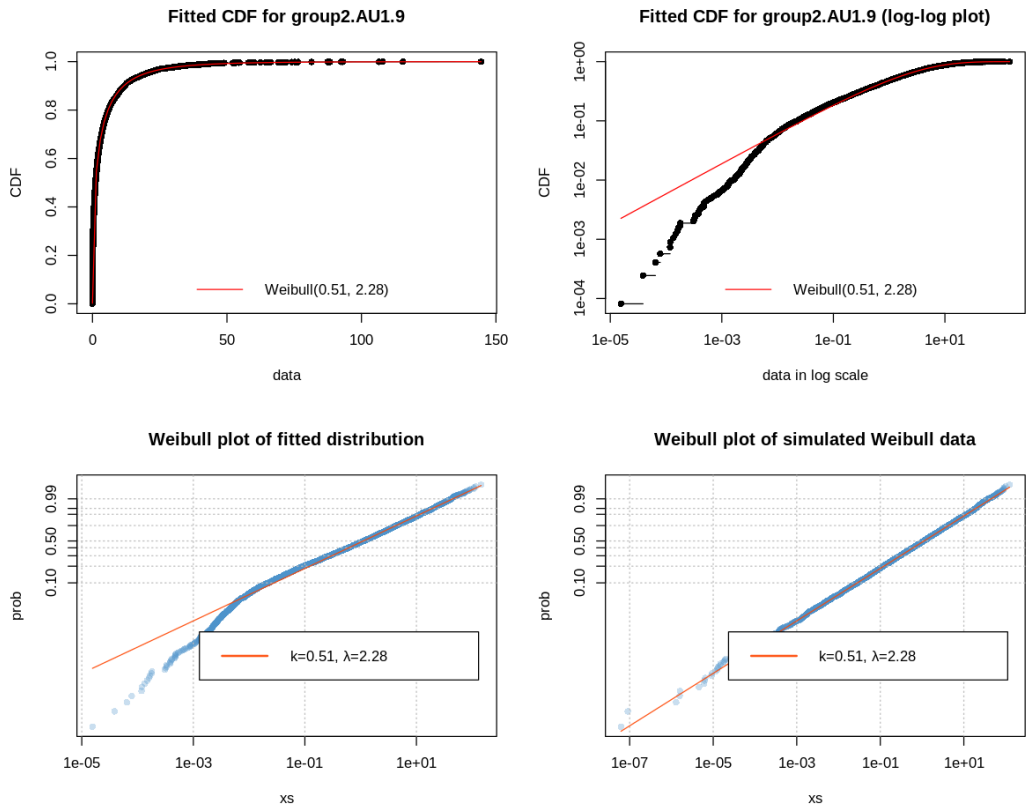

Figure B.20: Weibull fitting towards the positive values of GROUP2/AU1/9. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale for to spot deviations easier. The bottom panels shows the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

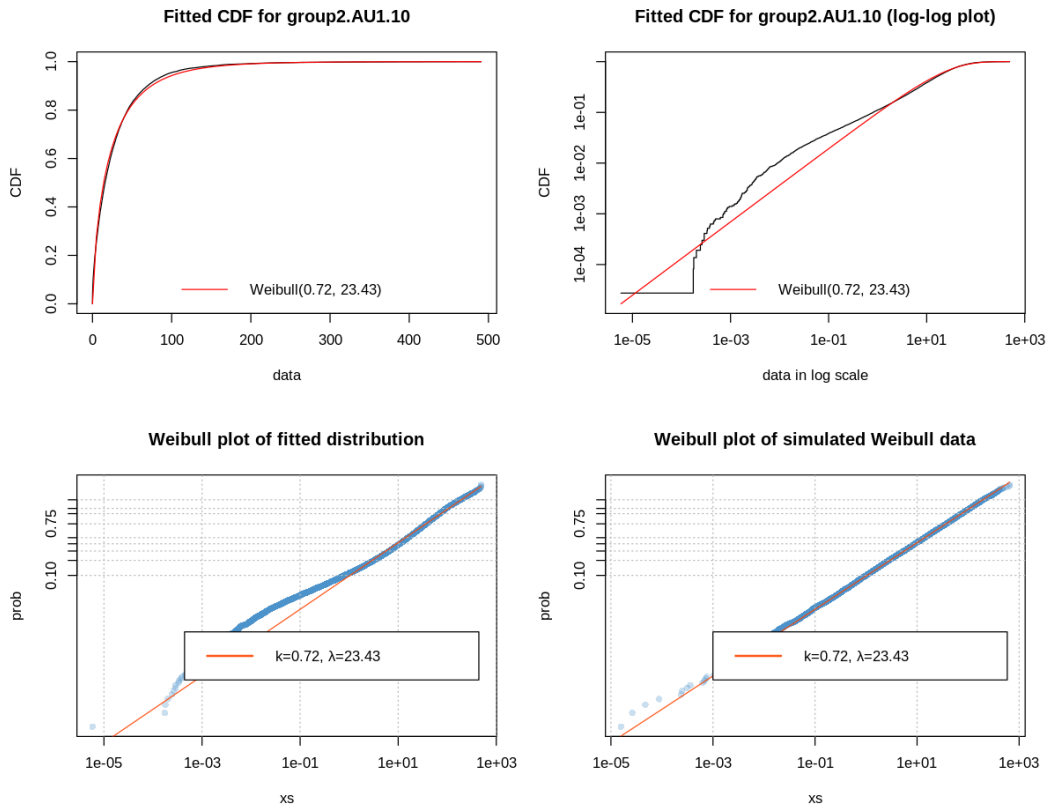

Figure B.21: Weibull fitting towards the positive values of GROUP2/AU1/10. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale for to spot deviations easier. The bottom panels shows the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

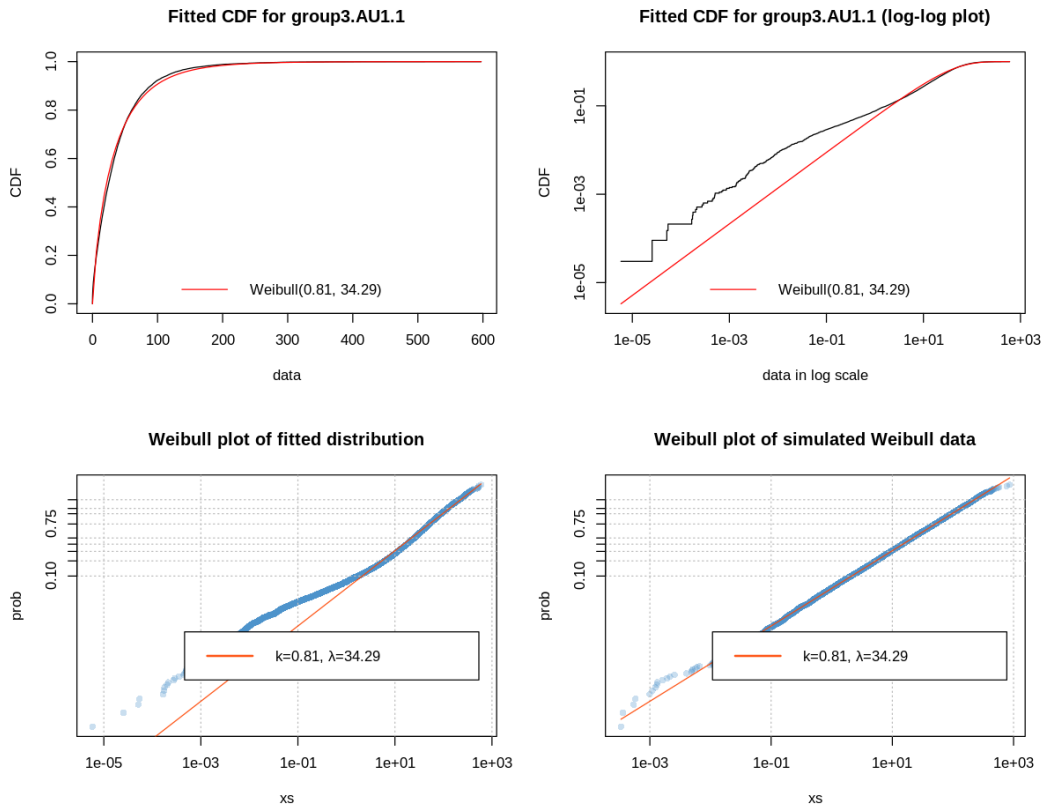

Figure B.22: Weibull fitting towards the positive values of GROUP3/AU1/1. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale for to spot deviations easier. The bottom panels shows the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

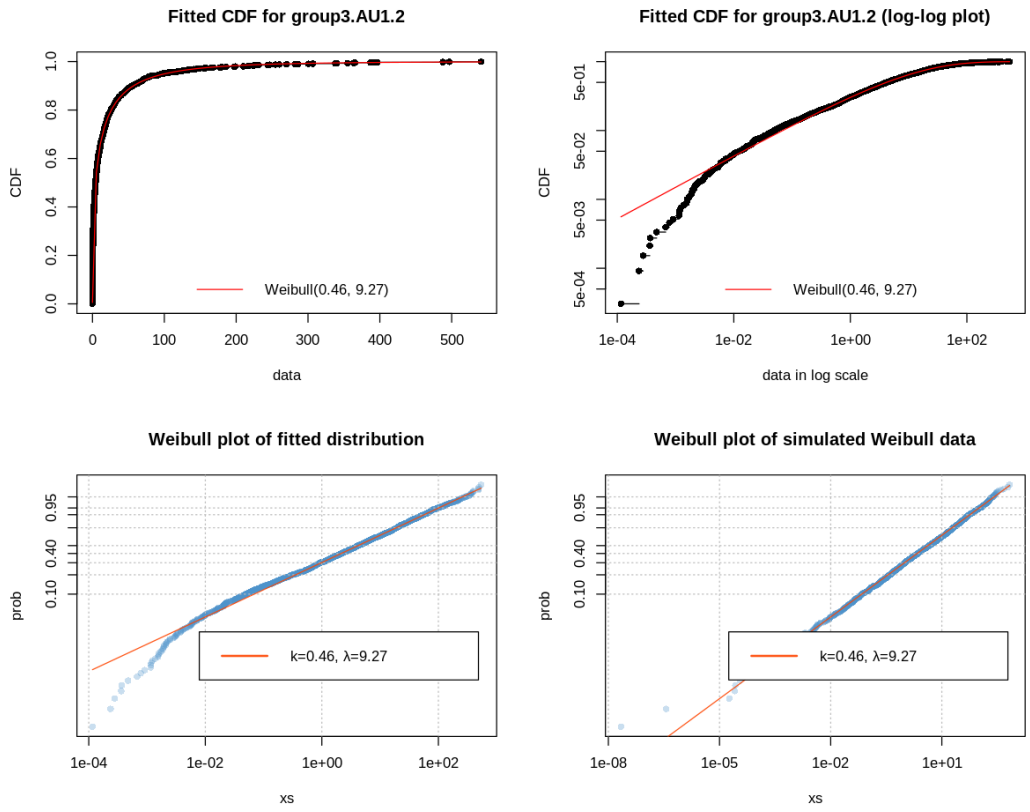

Figure B.23: Weibull fitting towards the positive values of GROUP3/AU1/2. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale for to spot deviations easier. The bottom panels shows the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

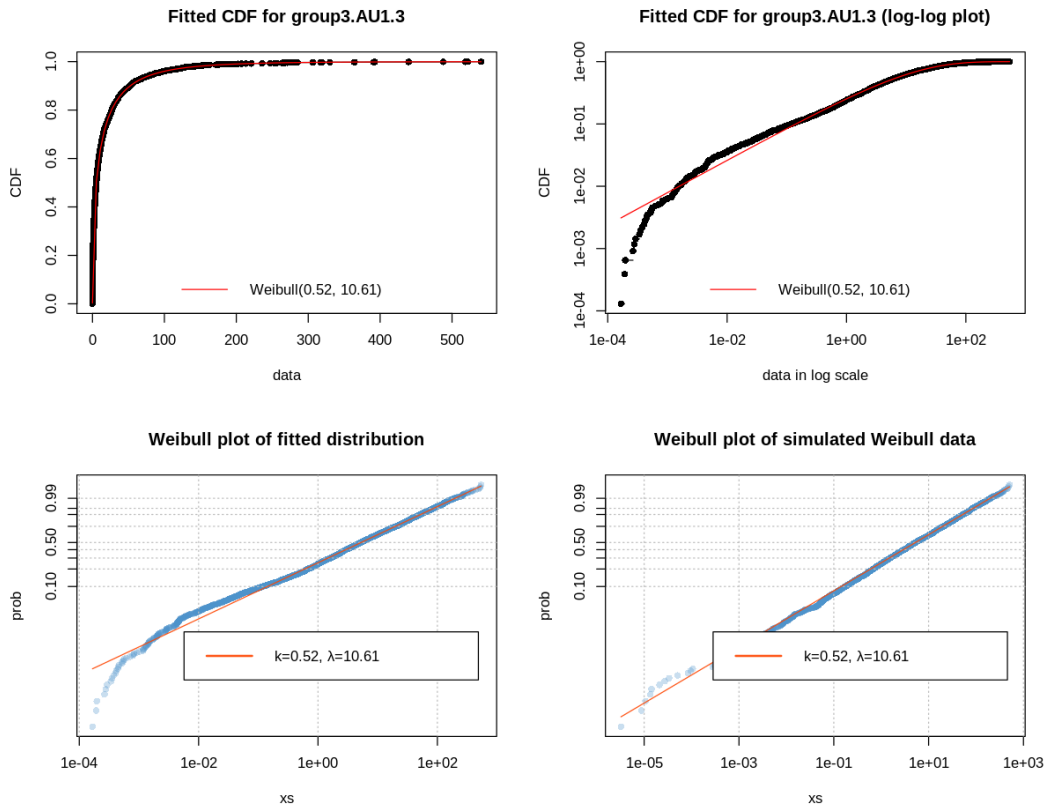

Figure B.24: Weibull fitting towards the positive values of GROUP3/AU1/3. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale for to spot deviations easier. The bottom panels shows the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

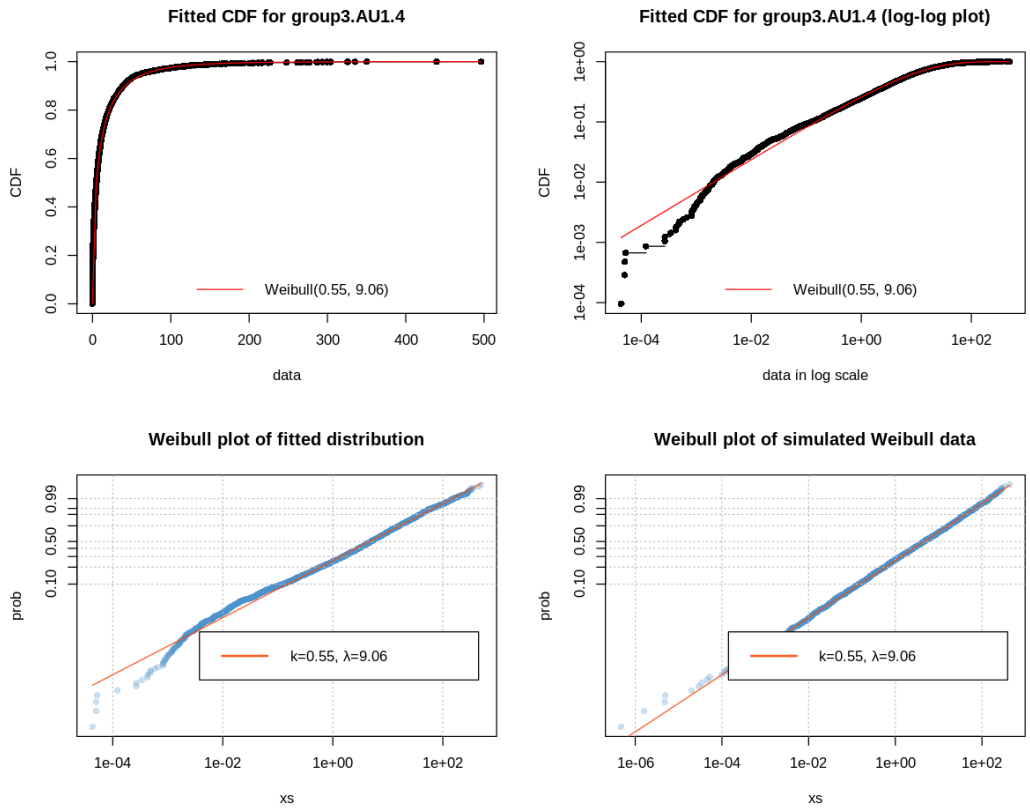

Figure B.25: Weibull fitting towards the positive values of GROUP3/AU1/4. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale for to spot deviations easier. The bottom panels shows the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

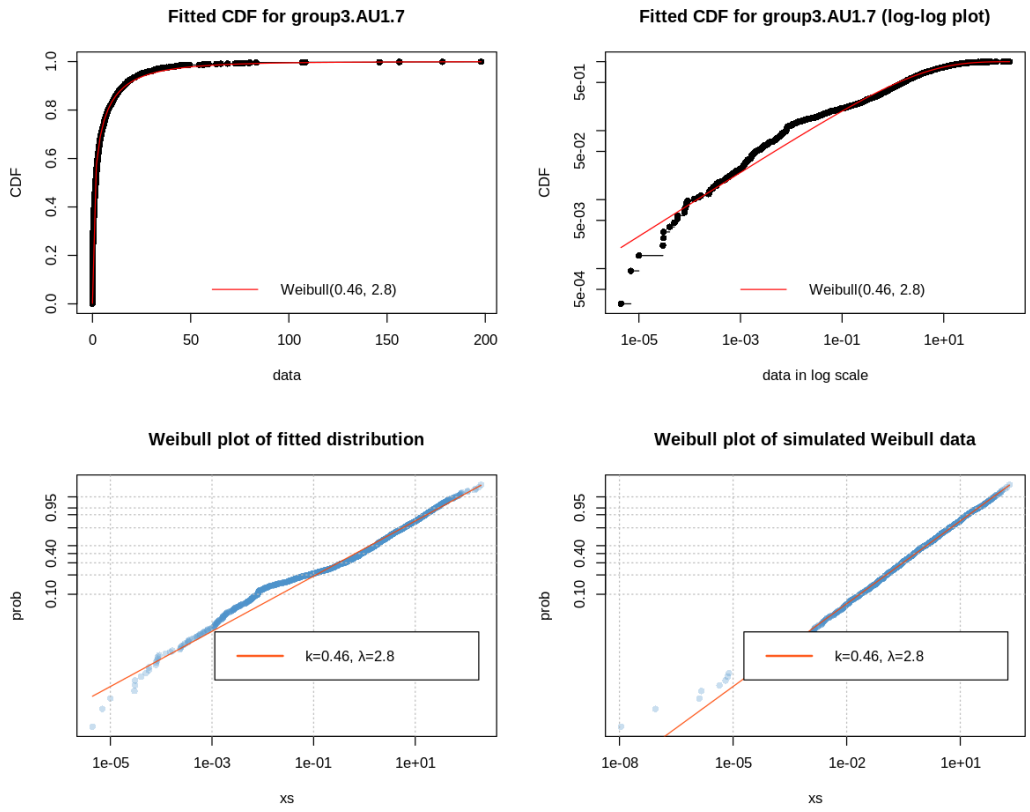

Figure B.26: Weibull fitting towards the positive values of GROUP3/AU1/7. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale for to spot deviations easier. The bottom panels shows the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

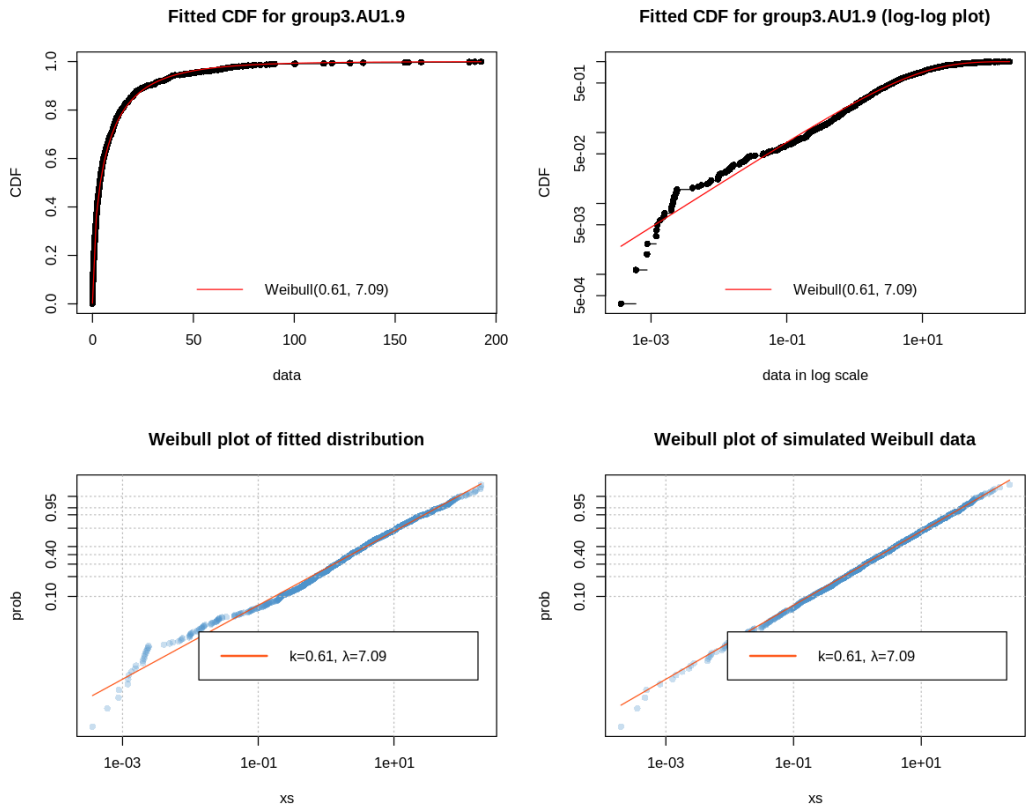

Figure B.27: Weibull fitting towards the positive values of GROUP3/AU1/9. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale for to spot deviations easier. The bottom panels shows the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

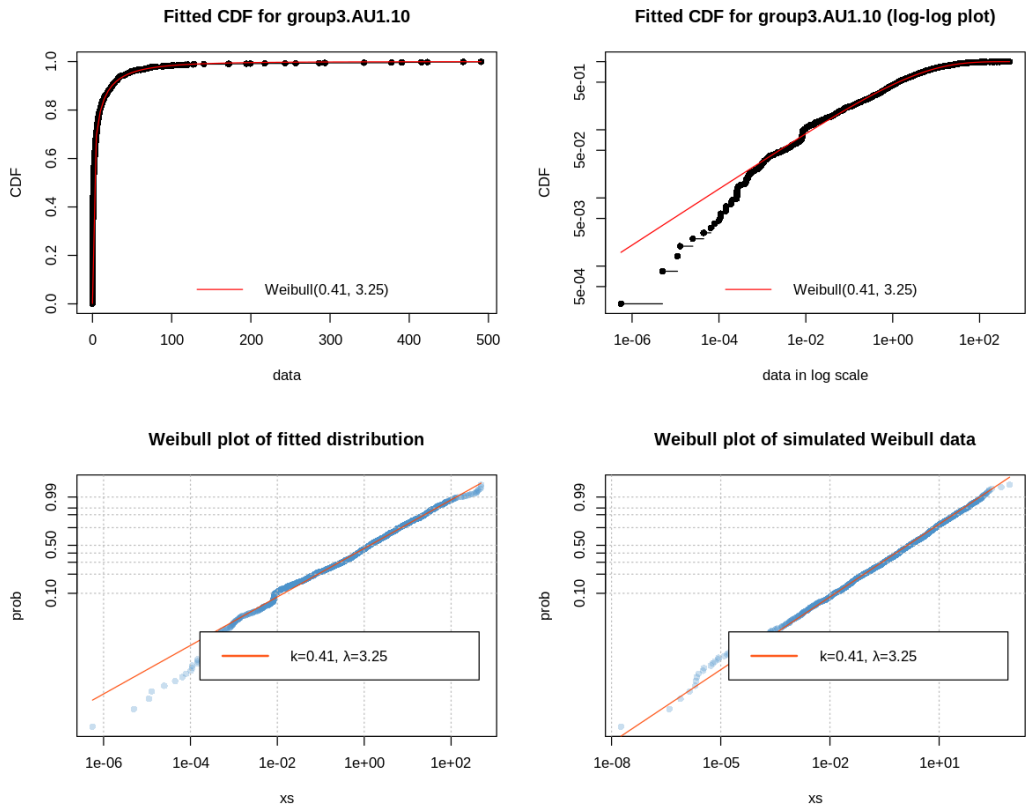

Figure B.28: Weibull fitting towards the positive values of GROUP3/AU1/10. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale for to spot deviations easier. The bottom panels shows the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

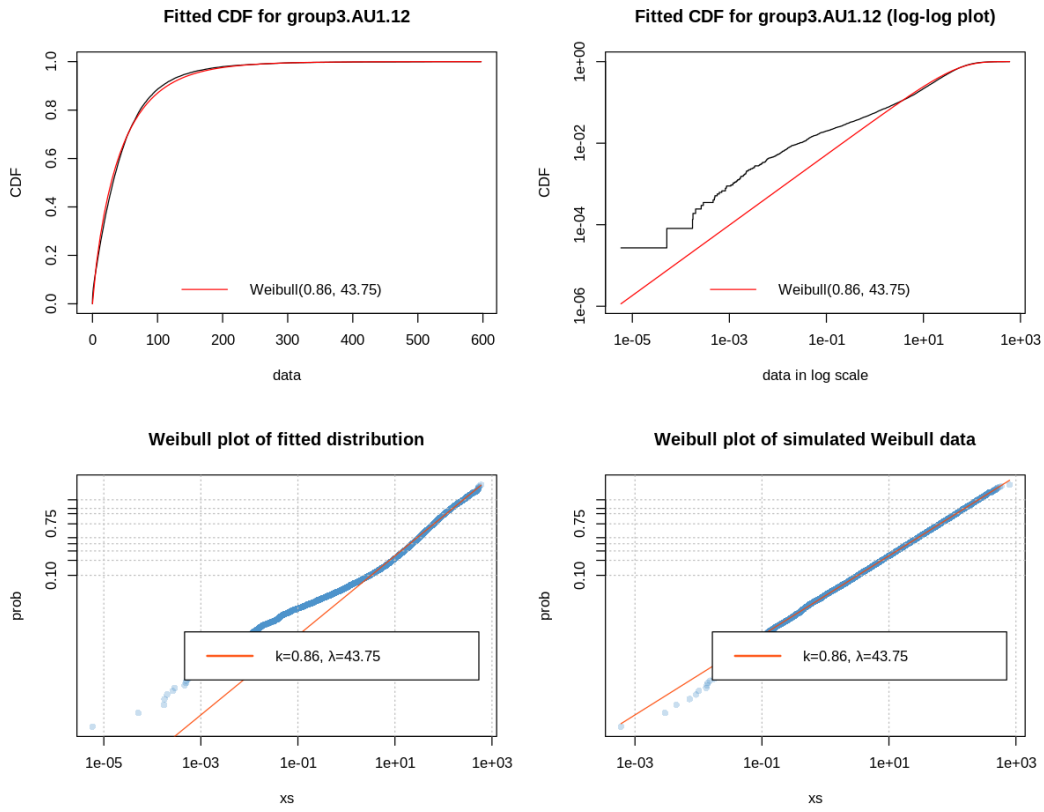

Figure B.29: Weibull fitting towards the positive values of GROUP3/AU1/12. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale for to spot deviations easier. The bottom panels shows the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

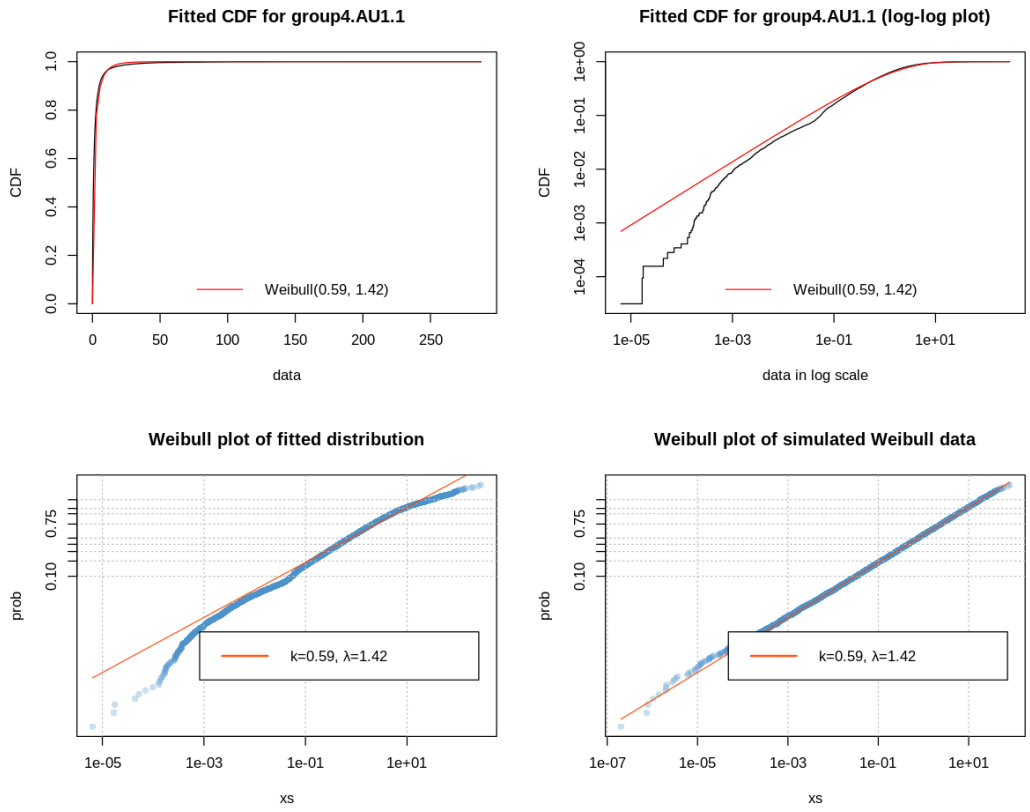

Figure B.30: Weibull fitting towards the positive values of GROUP3/AU1/12. The top left panel shows the Weibull fitted Cumulative Distribution Function (CDF) in red towards the empirical one (black), and the top right panel shows the CDFs on log-log scale for to spot deviations easier. The bottom panels shows the Weibull plot of empirical data (left) and simulated data as reference (right) under the fitted Weibull model.

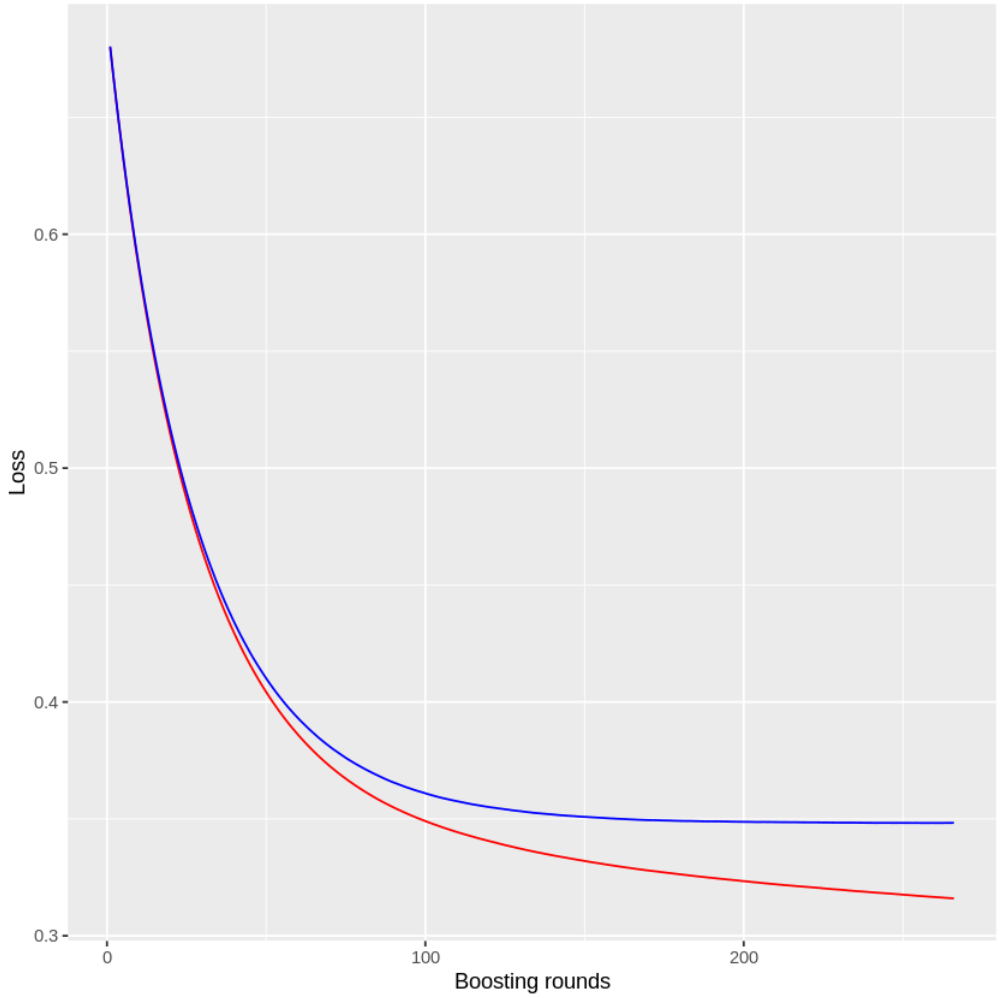

Figure B.31: Loss evaluations for the best boosted tree model with binomial deviance loss over the boosting rounds, shown separately for the training set (red) and the testing set (blue). The loss evaluations are here displayed as the means of the obtained results from K-fold Cross-Validation with $K = 5$. By an early stopping criterion of 10 steps, the training procedure stopped after 247 boosting rounds.

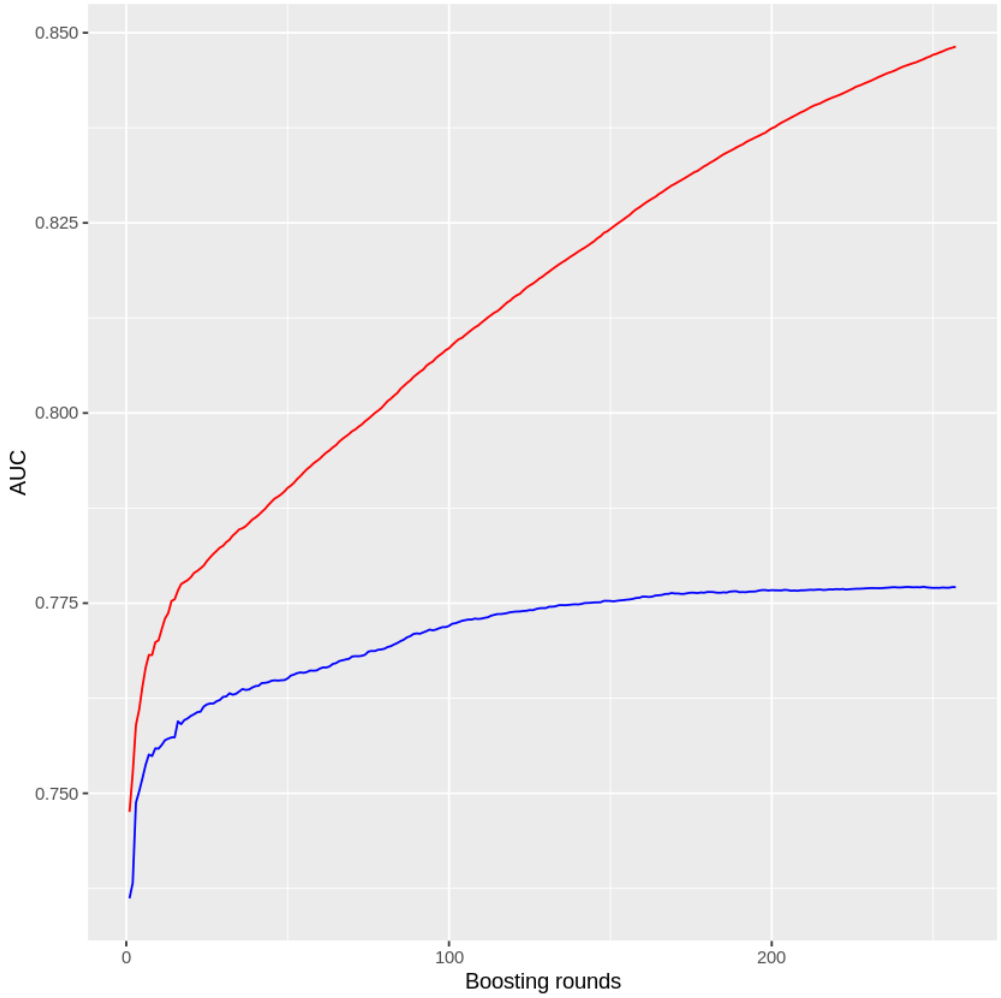

Figure B.32: AUC evaluations for the best boosted tree model with binomial deviance loss over the boosting rounds, shown separately for the training set (red) and the testing set (blue). The AUC values are here displayed as the means of the obtained results from K-fold Cross-Validation with $K = 5$. By an early stopping criterion of 10 steps, the training procedure stopped after 247 boosting rounds.

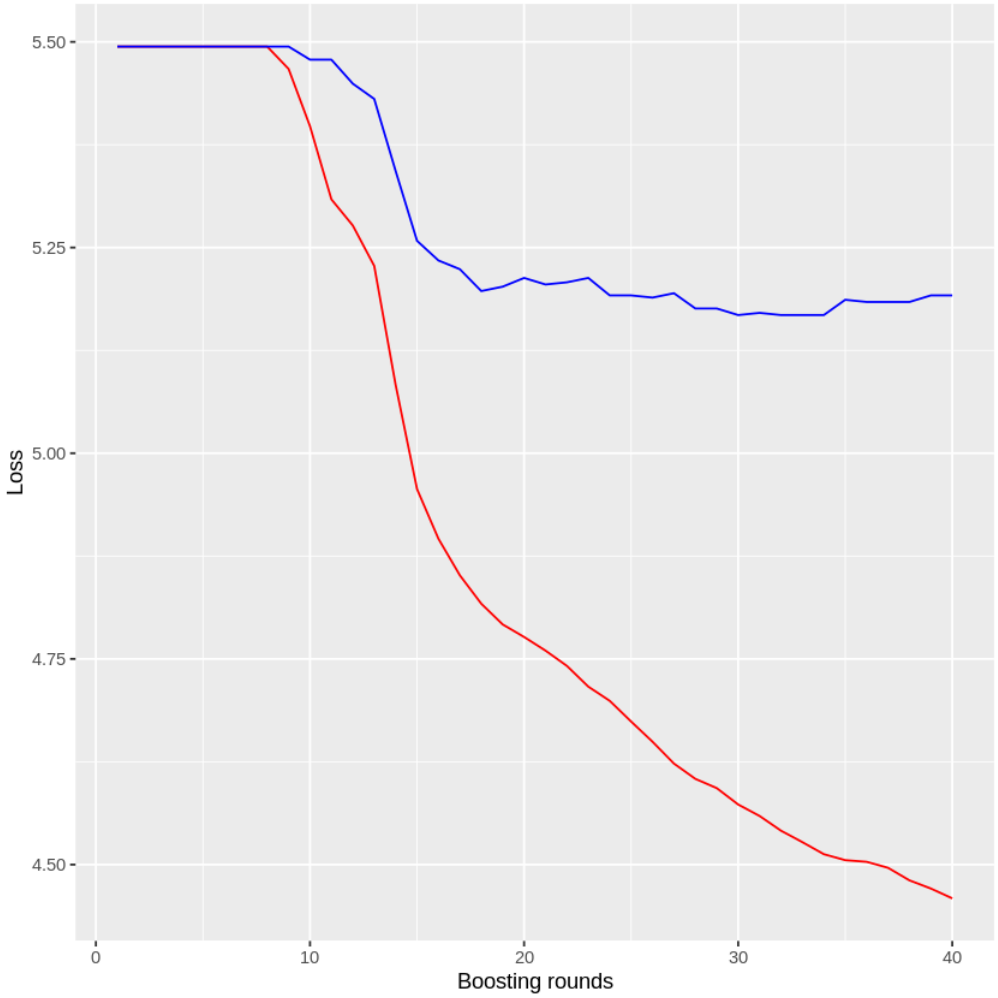

Figure B.33: Loss evaluations for the best boosted tree model with hinge loss over the boosting rounds, shown separately for the training set (red) and the testing set (blue). The loss evaluations are here displayed as the means of the obtained results from K-fold Cross-Validation with $K = 5$. By an early stopping criterion of 10 steps, the training procedure stopped after 30 boosting rounds.

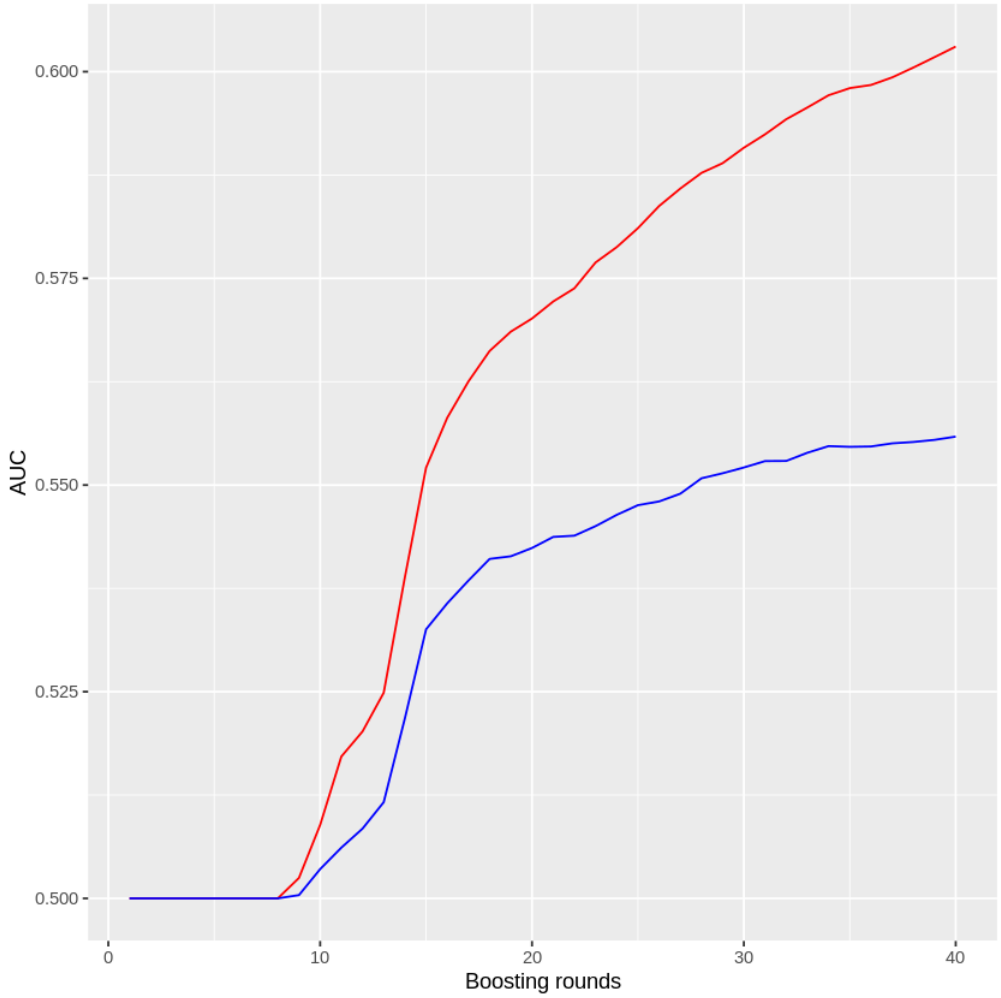

Figure B.34: AUC evaluations for the best boosted tree model with hinge loss over the boosting rounds, shown separately for the training set (red) and the testing set (blue). The AUC values are here displayed as the means of the obtained results from K-fold Cross-Validation with $K = 5$. By an early stopping criterion of 10 steps, the training procedure stopped after 30 boosting rounds.

```
Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)         -1.425e+00  3.925e-01  -3.629 0.000284 ***
group1.AU2.1         2.510e-05  4.625e-05   0.543 0.587377
group1.reach.1False  4.422e-01  1.443e-01   3.065 0.002180 **
group1.reach.1True   2.787e-01  1.708e-01   1.631 0.102831
group5.bool.1True   -7.330e-01  6.763e-02 -10.837  < 2e-16 ***
group5.attribute.22  6.040e-02  1.063e-01   0.568 0.569936
group5.attribute.23  2.297e-01  1.102e-01   2.084 0.037122 *
group5.attribute.24  4.096e-01  1.235e-01   3.317 0.000910 ***
group5.attribute.25  9.893e-01  1.422e-01   6.955 3.52e-12 ***
group5.attribute.26  1.653e+00  1.715e-01   9.638  < 2e-16 ***
group5.attribute.27  2.036e+00  2.381e-01   8.552  < 2e-16 ***
group5.attribute.28  5.915e-01  4.076e-01   1.451 0.146695
group6.AU2.1         7.127e-02  5.081e-03  14.027  < 2e-16 ***
group6.AU2.2        -8.629e-04  2.327e-04  -3.708 0.000209 ***
group6.attribute.12  3.434e-01  1.111e-01   3.092 0.001989 **
group6.bool.1True   -2.193e-03  1.120e-01  -0.020 0.984383
group2.AU2.1         1.759e-02  8.810e-03   1.997 0.045813 *
group3.AU2.1        -5.407e-02  1.340e-01  -0.404 0.686578
group1.log_AU1.3    -9.145e-03  4.651e-03  -1.966 0.049279 *
group2.log_AU1.1     7.723e-03  9.463e-03   0.816 0.414447
group2.log_AU1.2    -1.792e-03  5.691e-03  -0.315 0.752811
group2.log_AU1.3    -1.175e-02  4.959e-03  -2.369 0.017836 *
group2.log_AU1.4     5.773e-03  5.002e-03   1.154 0.248423
group2.log_AU1.5    -6.746e-04  4.268e-03  -0.158 0.874414
group2.log_AU1.6    -3.180e-03  4.684e-03  -0.679 0.497193
group2.log_AU1.7     1.225e-02  4.446e-03   2.755 0.005861 **
group2.log_AU1.8    -7.462e-03  4.631e-03  -1.611 0.107071
group2.log_AU1.9     9.656e-03  4.690e-03   2.059 0.039483 *
group2.log_AU1.10   -1.899e-02  1.560e-02  -1.217 0.223502
group3.log_AU1.1    -2.897e-02  1.628e-02  -1.779 0.075231 .
group3.log_AU1.2    -9.863e-03  1.116e-02  -0.884 0.376747
group3.log_AU1.3    -6.556e-03  9.815e-03  -0.668 0.504123
group3.log_AU1.4    -2.667e-02  9.881e-03  -2.699 0.006947 **
group3.log_AU1.7    -1.108e-02  1.122e-02  -0.988 0.323293
group3.log_AU1.9     1.203e-02  1.166e-02   1.032 0.302294
group3.log_AU1.10   -3.786e-03  1.119e-02  -0.338 0.735170
group3.log_AU1.12    1.071e-01  2.960e-02   3.619 0.000296 ***
group4.log_AU1.1     1.653e-02  5.368e-03   3.079 0.002077 **
group1.log_AU1.4     3.739e-02  1.420e-02   2.633 0.008459 **
group1.log_AU1.5    -5.685e-03  7.408e-03  -0.767 0.442812
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 11698.4  on 13886  degrees of freedom
Residual deviance:  9603.8  on 13847  degrees of freedom
AIC: 9683.8
```

Figure B.35: R summary for the baseline logistic regression model, fitted towards the training data.

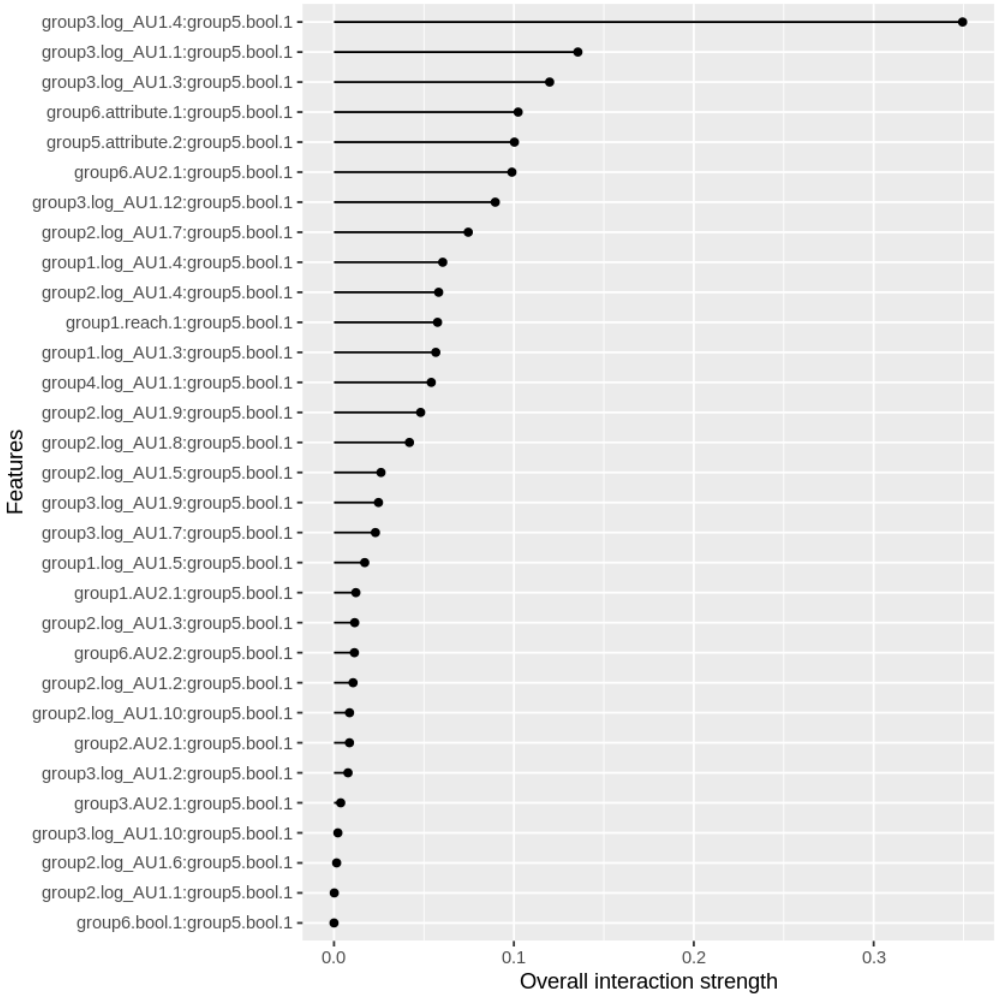

Figure B.36: Pairwise feature interactions for the GROUP5/BOOL/1 feature based on the H-statistic (see Equation 3.50) under the best performing boosted tree model with binomial deviance loss. The two-way interaction between GROUP3/LOG_AU1/4 and GROUP5/BOOL/1 is suggested as the strongest one.

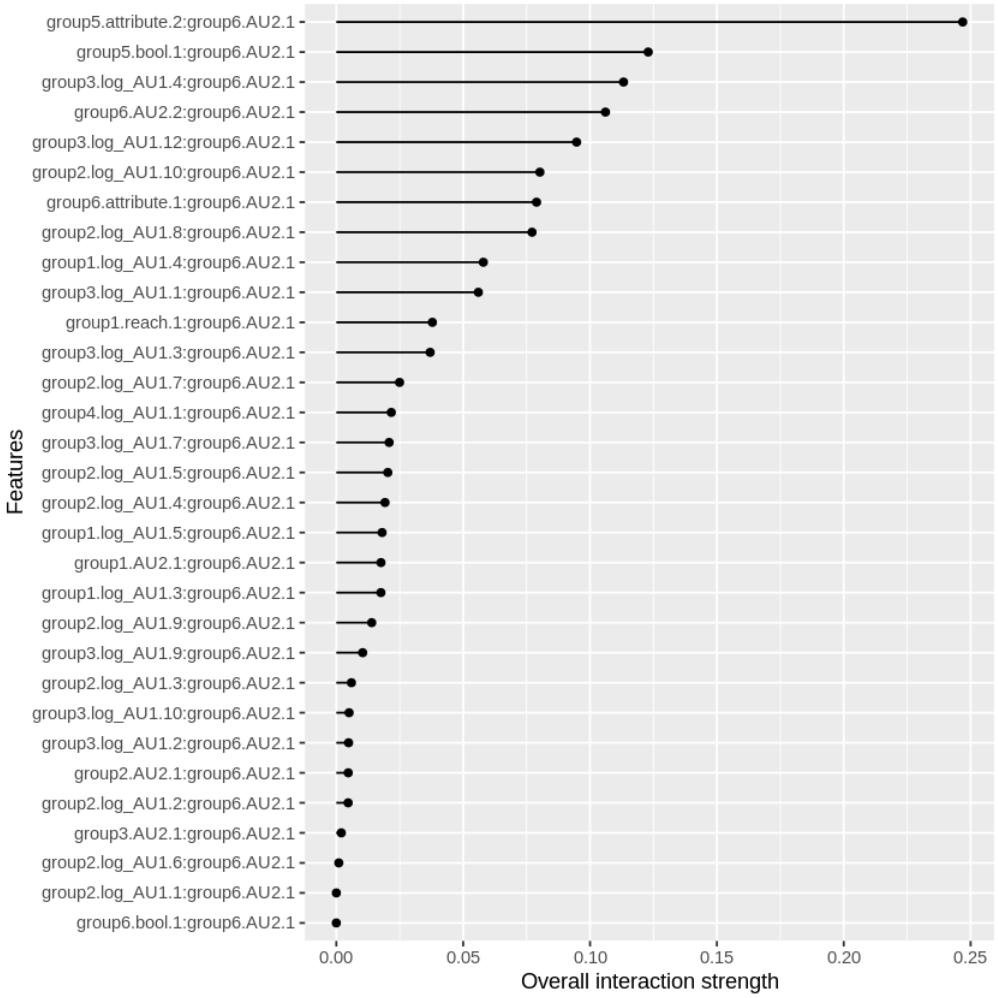

Figure B.37: Pairwise feature interactions for the GROUP6/AU2/1 feature based on the H-statistic (see Equation 3.50) under the best boosted tree model with binomial deviance loss. The two-way interaction between GROUP5/ATTRIBUTE/2 and GROUP6/AU2/1 is suggested as the strongest one.

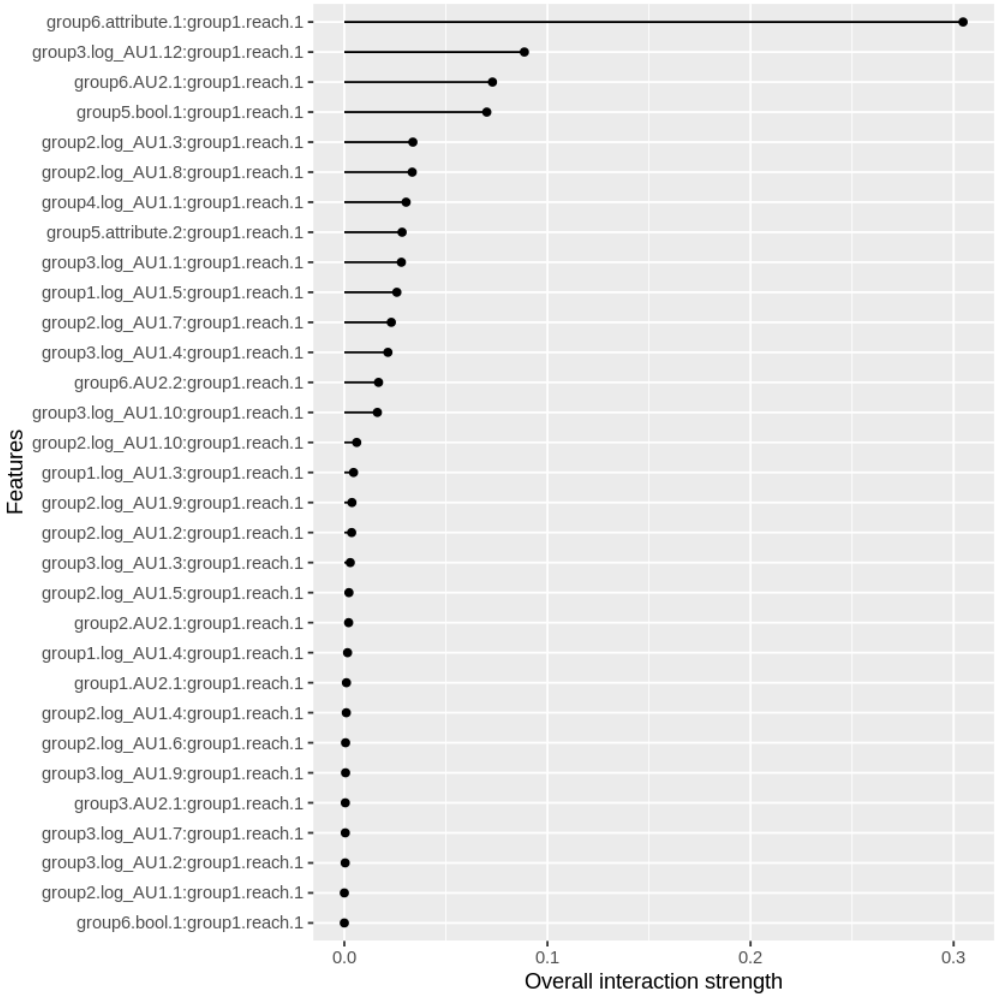

Figure B.38: Pairwise feature interactions for the GROUP1/REACH/1 feature based on the H-statistic (see Equation 3.50) under the best boosted tree model with binomial deviance loss. The two-way interaction between GROUP6/ATTRIBUTE/1 and GROUP1/REACH/1 is suggested as the strongest one.

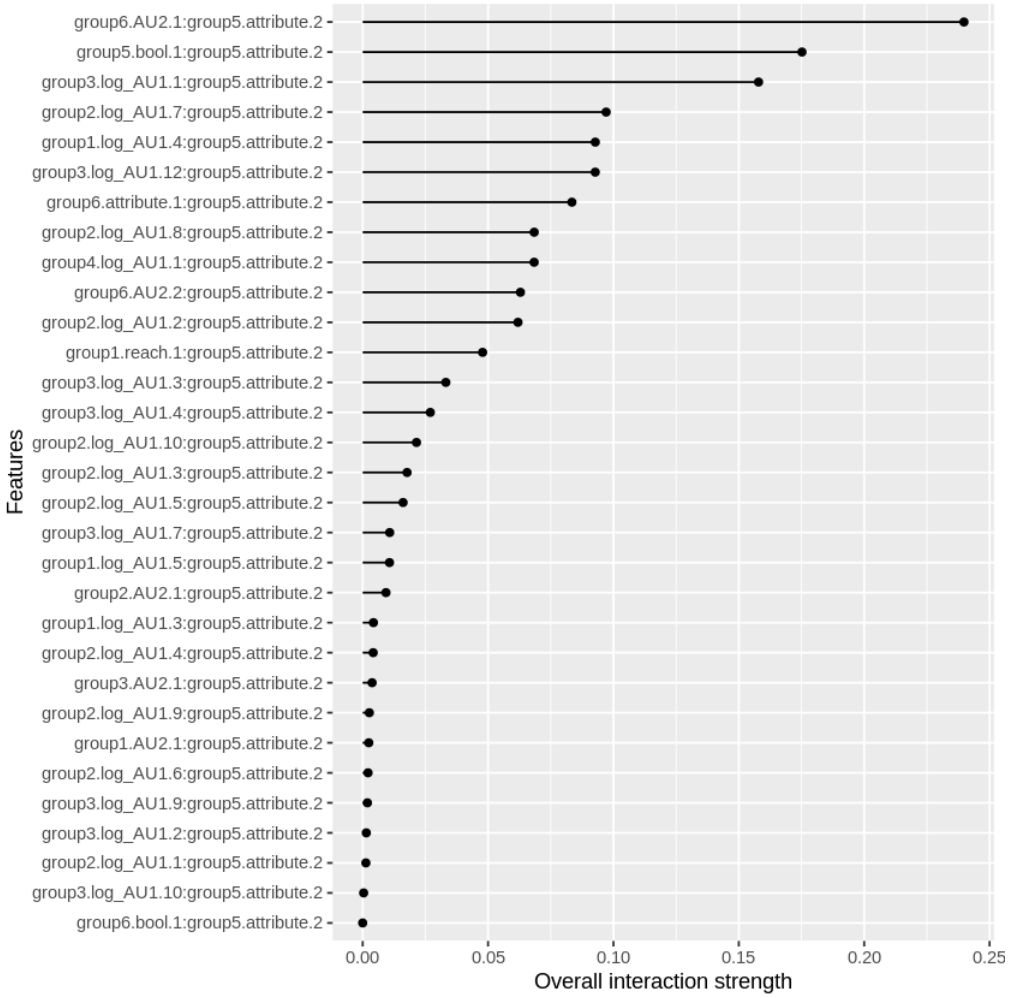

Figure B.39: Pairwise feature interactions for the GROUP5/ATTRIBUTE/2 feature based on the H-statistic (see Equation 3.50) under the best boosted tree model with binomial deviance loss. The two-way interaction between GROUP6/AU2/1 and GROUP5/ATTRIBUTE/2 is suggested as the strongest one. Two other strong two-way interactions (with GROUP5/ATTRIBUTE/2 as the static counterpart) are through the predictors GROUP6/AU2/1 and GROUP5/BOOL/1 respectively.

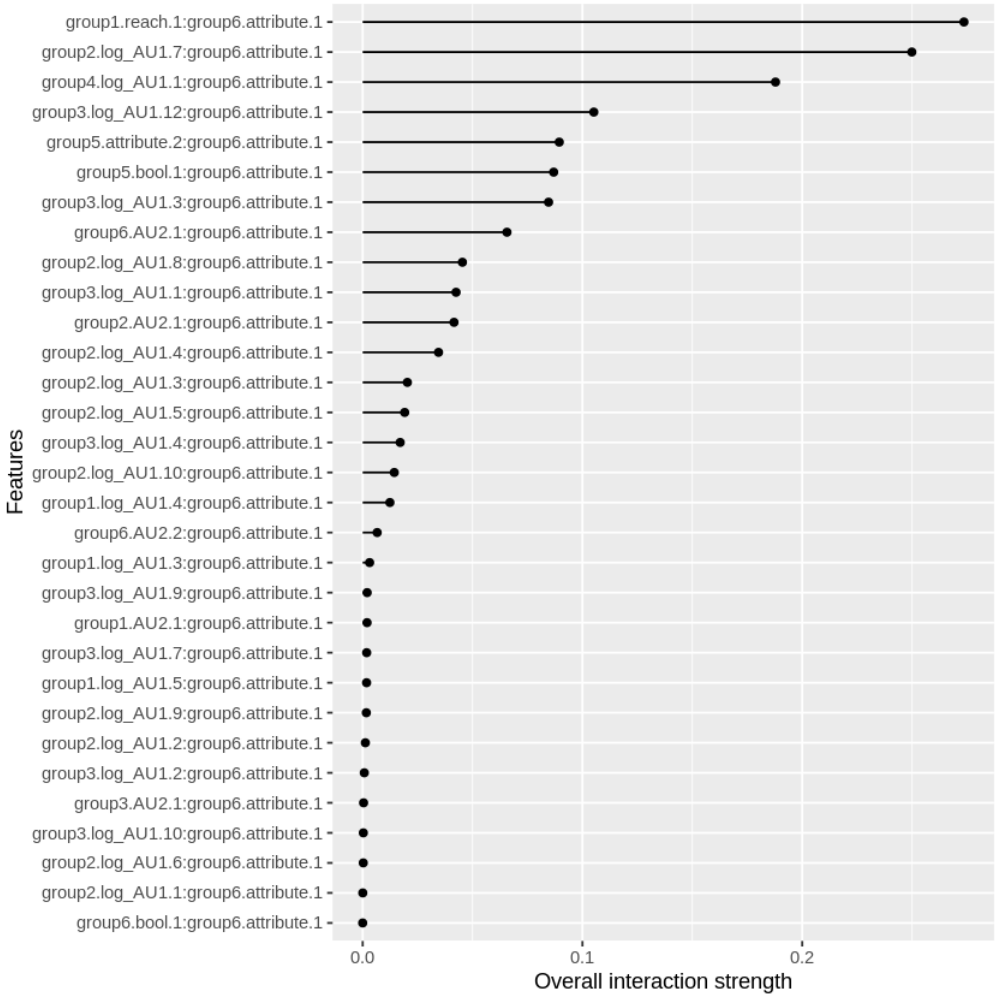

Figure B.40: Pairwise feature interactions for the GROUP6/ATTRIBUTE/1 feature based on the H-statistic (see Equation 3.50) under the best boosted tree model with binomial deviance loss. The two-way interaction between GROUP1/REACH/1 and GROUP6/ATTRIBUTE/1 is suggested as the strongest one, followed closely by the (two-way) interaction between GROUP2/LOG_AU1/7 and GROUP6/ATTRIBUTE/1. The interaction between GROUP4/LOG_AU1/1 and GROUP6/ATTRIBUTE/1 seems also to be a strong interaction effect under the model.