Technical University of Denmark

02450: Introduction to machine learning and data mining

# Project 2: Indian Liver data set, Regression and Classification

| **Authors:** | **Studynumbers** |
|---|---|
| Mads Christian B. Andersen | s173934 |
| Johannes Boe Reiche | s175549 |
| August Semrau Andersen | s183918 |

Mads C. B. Andersen, August S. Andersen, Johannes B. Reiche

November 12, 2019

# Introduction

### Problem at interest

In supervised learning the goal is to learn some mapping from the input space x, to some output space y with the help of learned parameters $\theta$.

$$y = f(x, \theta) + \epsilon$$

In the equation above, $\epsilon$ is a noise term, which is assumed to be normally distributed with mean 0, and variance $\sigma^2$. In this report different regression and classification models will be trained to predict some value of interest. In the case of regression, the goal is to estimate a continuous attribute *direct_bilirubin*, which is a medical measure taken when potential liver patients are examined. In the case of classification, the challenge is to use available attributes to predict a discrete outcome, in our case a binary classification of whether or not individuals are actually affected by a liver disease.

# Learning

### Complexity trade off

To obtain useful regression and classification models it is a necessity that the models can learn structures and patterns from the training data. How well the models learn and thereafter perform depends on several things. One important aspect of models ability to learn and perform is the flexibility/complexity of the model. A too simple model will not be able to pick up patterns and structures from the data, and the result is poor prediction results. On the other hand a too complex model will be able to fit the training data perfectly, but will not be able to generalize well. Meaning that prediction will be bad when the model is presented with new data from the same distribution the training data came from. This is called overfitting, which is naturally not desired. Hence there is a trade off when it comes to the flexibility of a model,and it is therefor crucial for model performance that the flexibility of the model is chosen wisely.

### Methods for handling complexity

The flexibility challenge is for some models solved with regularization methods that prevent overfitting, and for other models the complexity is penalized and accounted for in the cost function. Both methods have the potential to increase general performance, but it is not guaranteed and if the flexibility parameters are chosen poorly, they can hinder learning. Each models flexibility can be adjusted by changing a hyperparameter, also referred to as the complexity controlling parameter, that is unique for the models (In some cases there are several hyperparameters that control the complexity of a model).

In this report the focus will on training different models for both regression and classification using supervised learning, and two-level cross validation to pick the optimal complexity and evaluate the performance of this model. To crown a potential best model, a statistical evaluation will be performed.
For the sake of convenience a description of the attributes are listed in the appendix in table [7].

# Regression

Based on the results of the Principle Component Analysis in the previous project, the attribute chosen for regression is *direct_bilirubin*, as this was the attribute with one of the highest contributions to the PCA with a PC1-direction of $-0.83$. For regression part A and B an adjustment have been made to the data set as it also contains measurements of total bilirubin amount *tot_bilirubin*, which directly correlates with direct bilirubin amount. Therefore regressions will be made both with and without *tot_bilirubin* for linear regression, to properly view the effects of other attributes.

The data matrix $X$ will be standardized ($\mu = 0$ and $\sigma^2 = 1^2$), while the target vector y containing *direct_bilirubin* measurements will remain unchanged. In the last report, one-out-of-K coding was used to convert male/female measurements from string values into usable proxy variables. The same thing is done here, though we will only be using one column instead of two as previously done.

## Model and methods for regression

We will use three different models to make predictions of *direct_bilirubin*. The three models are Linear Regression, Artificial Neural Network (ANN) and a baseline model. Throughout the next sections an analysis and discussion about the three models strength, weakness and performance levels will be made.

## Regularization

As mentioned in the 'Learning'-section it is crucial to the success of the following regression models that we avoid overfitting. In Linear Regression the regularization term $\lambda$ is used to weight the cost function and the complexity penalty (L2 square-loss regularization) as seen beneath.

$$E_\lambda\left(\boldsymbol{w}, w_0\right) = \left\|\boldsymbol{y} - w_0\mathbf{1} - \hat{\boldsymbol{X}}\boldsymbol{w}\right\|^2 + \lambda\|\boldsymbol{w}\|^2, \lambda \geq 0$$

To obtain the best regularization for the linear regression a range of different $\lambda$ values are tested with two-level cross validation, where an estimate of the generalization error is found. The regularization parameter should neither be too small as this will result in high precision, a small bias, but large variance, nor too big resulting in large bias and small variance. It should ideally be somewhere in between. For the linear regression the regularization parameter $\lambda$ is chosen from the range from $10^{-1}$ to $10^4$, and the best one is found by the lowest generalization error found by K = 10 fold cross validation.

The regularization parameter for the linear regression that predicts *direct_bilirubin* (with *tot_bilirubin* removed) is found. A larger version of figure [1] can be seen in 'Appendix' figure [4].

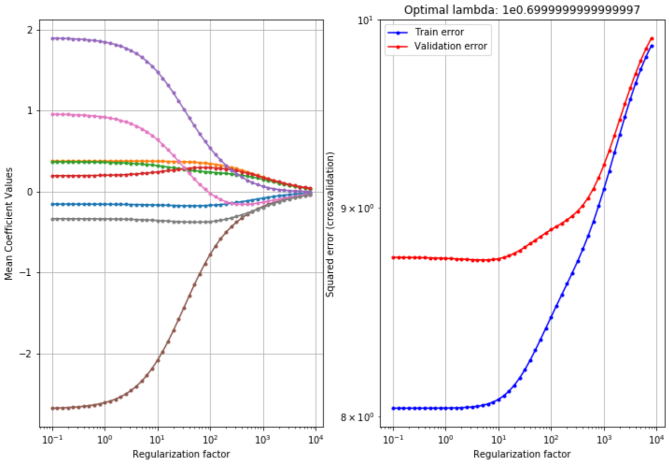


Figure 1: Regularization of linear regression prediction direct_bilirubin, with optimal regularization parameter $\lambda = 1 \cdot 10^{0.69}$ and generalization error 8.737.

There is a small dip in parameter $\lambda$'s value as seen on the right, but this should be much more significant. The result here shows that introducing regularization to the linear regression is not beneficial as it over-simplifies the model and worsens performance compared to doing nothing.

The regularization parameter is found again, now including *tot_bilirubin*, see figure [2], where a larger version can be found in 'Appendix' [5].

This time there is clearly a $\lambda$ that optimizes the performance a bit, as seen by the dip on the right graph. Hence there is a small advantage to be had from introducing regularization when including *tot_bilirubin*, probably because this attribute introduces more complexity to the model compared to the other attributes.

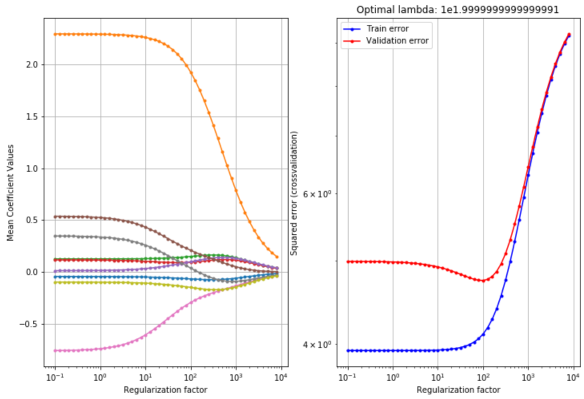


Figure 2: Regularization of linear regression prediction direct_bilirubin including total bilirubin with optimal regularization parameter $\lambda = 1 \cdot 10^{1.99}$ and generalization error 4.745.

The regularization parameter for the ANN is the number of hidden units. By lowering the hidden units, the amount of weights that can be adjusted is reduced, and fewer patterns can be learned. The range of hidden units is [1,10].

### Linear prediction

New data observations are predicted based on the linear regression model. The different attributes now have weights associated with them, and through a linear combination they calculate the target value *direct_bilirubin*:

$$direct\_bilirubin = a_1 \cdot x_1 + a_2 \cdot x_2 + \ldots + a_n \cdot x_n$$

Due to the regularization these weights were slightly different from what a normal linear regression would produce. The weights of the attributes can be seen in table 1.

From these linear regressions we see that for the one including *tot_bilirubin*, this attribute contributes primarily to the regression, which was the reason why we opted to also remove it.

For the linear regression without *tot_bilirubin*, the contribution is more spread out, and while some attributes contribute more than others, etc. *sgpt*, *sgot*, *albumin* and *alkphos*, these don't immediately seem to have a biological explanation as to why they would be drastically more important than the rest.

| Attribute | Weights | Weights w. tot_bil. |
|---|---|---|
| age | -0.17 | -0.06 |
| gender | -0.16 | -0.07 |
| tot_proteins | 0.37 | 0.15 |
| albumin | 0.38 | 0.09 |
| ag_ratio | 0.19 | 0.1 |
| sgpt | 1.69 | 0.22 |
| sgot | -2.38 | -0.31 |
| alkphos | 0.80 | 0.5 |
| is_patient | -0.35 | -0.15 |
| tot_bilirubin | | 1.95 |

Table 1: Table showing weights of regularized attributes predicting Direct Bilirubin.

## Comparison of regression models

In this section we hope to determine how good machine learning models can predict the value of *direct_bilirubin* when *tot_bilirubin* is removed from the dataset. A scenario could be that the attribute *direct_bilirubin* is expensive/time consuming to measure while *tot_bilirubin* is inadequate for any diagnosis of a patient. As mentioned, the three models investigated will be linear regression, ANN and baseline model. After utilizing two-level cross-validation to select the optimal complexity for each model, the three models test errors will be compared and a statistical performance evaluation will be carried out. Lastly, our recommendation for regression problems will be presented.

### Model complexity selection

To investigate and compare which level of complexity that gives the best performance for the selected models, a two-level cross-validation will be used. In each fold we will use $K_1 = K_2 = 10$, and we will test $S = 10$ different hyperparameters in each inner-fold. The range of the complexity hyperparameters is shown below:

| **Model** | **Parameter** | **Range** |
|---|---|---|
| *Lin. Reg.* | $\lambda:$ | $[-2; 8]$ |
| *ANN* | *Units:* | $[1; 10]$ |

The Artificial Neural Network will therefore be a one-layer neural network with the number of hidden units in range 1-10. The error measure used in the two-level CV is the squared loss per observation (1), the squared distance to the actual value of *direct_bilirubin* from the estimated value of *direct_bilirubin* divided by the number of observations in the test dataset $N^{test}$:

$$E = \frac{1}{N^{\text{test}}} \sum_{i=1}^{N^{\text{test}}} (y_i - \hat{y}_i)^2 \tag{1}$$

The first error measure is the validation error, measured for each model $M_s$ tested on the split test set from the inner cross-validation loop. This is used to find the optimal model for this particular split. For each of the tested models we compute a generalization error 2:

$$\hat{E}_s^{gen} = \sum_{j=1}^{K_2} \frac{|D_j^{val}|}{|D_i^{par}|} E_{M_{s,j}}^{val} \tag{2}$$

The optimal model will be the model that had the lowest generalization error, that is $M^*$. $M^*$ is then trained on $D_i^{par}$ to be tested on $D_i^{test}$ where we compute the new estimate of the test error of models trained on a bigger data set in the outer $i$ split. For each outer $i$ split we save the optimal complexity parameter. This gives us a possibility to see how each of the 10 splits optimal model $M^*$ performs and what parameters it had. All of the results is put together in the next section.

### Regression model comparison

Table [3] seen in 'Results' shows estimated generalization errors of the 3 different regression models, ANN, Linear Regression and Baseline, all performed on two-level cross validation. When comparing the three models (not including the linear regression model with *tot_bilirubin*), it looks like an ANN might be the most precise option, but as we will see next section, this is statistically not the case.
When taking a closer look at the ANN, it is evident that the number of hidden units, which for eight of the 10 folds are 1, shows a discontinuity of what the ANN should find optimal and what is found optimal. Having only one hidden unit in the ANN equates to doing a linear regression, as the input data is treated by weights as in a linear regression, and it therefore seems that there is no advantage in being able to compute more hidden units.

The regularization parameter $\lambda$ found for the linear regression using a two-level cross validation is similar to the one found in section a using one-level cross validation, as is the generalization error. The regularization parameter is found to be very similar, and the graphs showing optimal $\lambda$ also look similar.

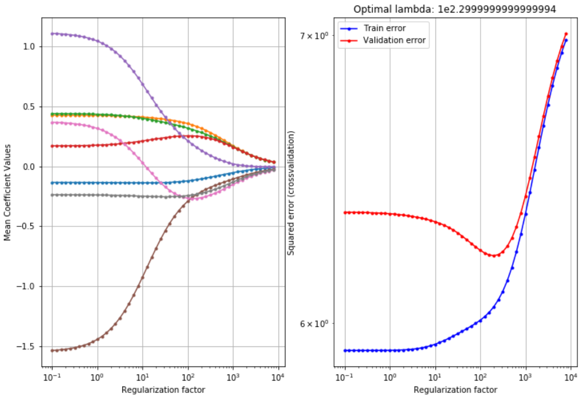


Figure 3: Linear regression (w/o Total Bilirubin) regularization, two-level CV.

That the linear regression is being evaluated on a two-level cross validation should in theory give a higher error rate, as it removes the possibility of creating a good one-level regression by chance. In this case, the is no immediate advantage in doing two-level CV as the linear regression and it's regularization parameter is not able to be optimized through the process of regularization in the first place.

### Statistical performance evaluation of regression models

To evaluate and investigate if there is a significant performance difference between the three tested models, a statistical evaluation is performed. For this report we will consider the setup II where a pairwise comparison will be made. Setup II have been chosen, because we would like to find the model with lowest generalization error for classification on any subset of the Indian population and thereby take the variability from the data set into account.
When calculating the generalization error, we performed one-level cross validation (with $K_1 = 10$) 10 times, where we have chosen the most frequently occurring (best) parameter from table [3] in 'Results' to train the model, and evaluate the performance. Then we compute the average of these 100 trials to get a more statistical sound estimate of the performance. (All the models have had the same seeds in the 100 different runs). The results can be viewed in the table [4] in 'Results'.

The table reveals that there is no statistical evidence pointing toward that one model performs significantly better than the others. The two models that seems to differ the most is ANN and Baseline. That could be down to the fact that ANN performs unstable as table [3] shows compared to the quite stable Baseline model. Besides fold 2, 3 and especially 6, the ANN seems to perform well from table [3] and have potential to perform even better. The fact that none of the models seems to significantly outperform the baseline model is disappointing. Never the less, our recommendation would be to go with the ANN and increase the complexity by extending the range of hidden units to etc. 30 and add one or more layers to the ANN.

# Classification

The purpose of training a classification model in our case is to be able to make prediction on new patients, and tell based on medical measurements whether or not they have a liver disease.

### Model and methods for classification

In this section, four different approaches of classifying liver patients will be investigated, and further compared to each other and a baseline model. The four approaches to this binary classification problem are logistic regression (Log. Reg.), artificial neural networks (ANN), K-nearest neighbors (KNN) and decision trees (DT). After obtaining results for each of the models, a statistical evaluation will be carried out.

### Complexity controlling parameter

As mentioned in the *Learning*-section, for each of the four approaches a two-level cross validation process is performed to find the best complexity and evaluate how well the general performance of the model is. Below, a table is shown with the model and its complexity parameter and in which range the parameters were tested.

| **Model** | **Parameter** | **Range** |
|---|---:|---|
| *Log. Reg.* | $\frac{1}{\lambda}$ : | $[0.01; 1.01]$ |
| *ANN,v1* | *Units:* | $[1; 10]$ |
| *ANN,v2* | *Layers:* | $[1; 5]$ |
| *KNN* | *Neighbors:* | $[1; 31]$ |
| *DT* | *Depth:* | $[1; 21]$ |

For logistic regression the complexity controlling parameter $\lambda$ corresponds to the weighing between cost function and a complexity penalty (L2). The L2 measures the squared size of the weights (the result is that small weights are preferred) and the regularization term is $\lambda \sum \theta_i^2$. With the ANN's there are several hyper parameters that controls complexity. We have chosen to test both the number of hidden units with 1 layer (ANN,v1), and another setup where there is 10 hidden units and varying amount of hidden layers (ANN,v2). For KNN, the parameter describes the amount of neighbors taken into account when classifying. The decision trees parameter is the maximum amount of consecutive rules in a tree, also known as the depth of the tree.

### Classification model comparison

To determine the performance of each of the models, the two-level CV algorithm is applied, described in the 'Model selection' section. Setting the random seed to the same number will make sure that the models have been trained and tested on the same splits in the data set. This time we use a different error measure:

$$E = \frac{\textit{Number of mis-classified observations}}{N^{\text{test}}}$$

The results for the performances of each model can be found in table [5] in 'Results'. The main takaway from the results is that it seems that none of the models are able to beat the baseline confidently. The test errors are generally high, around 28 % which is quite high for a classification task. Given that the distribution of *is_patient* vs not *is_patient* is approximately 71 % to 29 %, it is clear that the base line model will achieve a similar test error. Another point is that the ANN seems to drastically improve when several layers with 10 hidden units in each is applied to the task, compared to the more simple one layer model with eight or nine hidden units, see table [5] in 'Results'. The KNN model also seems to benefit from the increased complexity, given that the range of neighbors=K was tested in range (1-31) and either had K=25 or K=28 as the optimal complexity parameter. The main competitor to the baseline model seems to be the KNN model, hence this will be the chosen model for the statistical evaluation in the next section.

### Statistical performance evaluation of classification models

We have chosen to pairwise compare the approaches logistic regression, baseline and the KNN method with setup II. As mentioned in 'Statistical performance evaluation of regression models', Setup II has been chosen to account for the variability of different subsets of the Indian population, which allow a more general conclusion on which model to prefer in the classification task. As with regression, we have chosen the most frequently occurring parameter from table [5] to train the model, and evaluate the performance (with the same one-level cross level procedure as in regression). The obtained generalization errors and their confidence intervals from the 100 trials are found in table [2].

| Method | $E_{gen}$ | Jeffrey CI | Parameter |
|---|---|---|---|
| Log. Reg | 28.03 | [26.88-29.20] | $\frac{1}{\lambda} = 0.91$ |
| KNN | 29.54 | [28.38-30.72] | K = 28 |
| Baseline | 28.49 | [27.34-29.66] | - |

Table 2: Generalization error.

It is seen that the models are not statistically different and have overlapping confidence intervals. The same can be extrapolated from table [6] found in 'Results', where the results from the pairwise model test are found. From table [6] it can be seen that pairwise comparisons all result in p-values in the range of [0.88-0.98], which provide a clear statistical conclusion of no significant difference in classification performance. Hence the null-hypothesis cannot be rejected and neither of the two models compared to the baseline can be said to outperform it.

### Logistic regression

As with the other classification models the goal of logistic regression is to predict whether or not a person is a liver patient, and since this is a binary classification problem, we would like to model the prediction y's density as a Bernoulli variable. If y is modelled as a Bernoulli variable, the output of the linear model will be continuous number which belong to the unit interval [0,1]. In classification task we would like a discrete output, and this can be obtained by re-parametarizing the Bernoulli distribution by applying the sigmoid activation function as the final step of the model.

$$\sigma(z) = \frac{1}{1+e^{-1}}, where \quad z = \tilde{x}_i^T w$$

Our prediction will like the linear regression consist of a linear transformation, and on top of that applying the logistic sigmoid:

$$\hat{y} = \sigma(\tilde{x}_i^T w)$$

Logistic regression can therefor be said to be an extension of linear regression. The same process with inputs being multiplied by weights happens, however we want the logistic regression model to output the probability that the input belongs to a certain class. This is ensured with the activation function that squeezes the output of the regular linear regression model to the correct span. The outputs can be interpreted as probabilities now. This expansion of linear regression means that the output is no longer linearly dependent on the weights, and furthermore the weights that are relevant for classification might not be important in the case of regression. The models differ in what they are trying to predict, and the correlation between the attributes and the output might not be the same.

# Discussion

None of the regression approaches significantly outperform the baseline model for predicting the value of *direct_bilirubin*. In the Linear Regression model that includes *total_bilirubin*, it could very well be the case that the attribute *total_bilirubin* was used to directly predict *direct_bilirubin*. It could be the case that more advanced regression models could do a better regression job. Some more advanced models that could be investigated is ensemble methods, deeper ANNs and perhaps extend the current approaches with other optimization techniques. Therefor it remains to be seen if increased hidden units and layers in the ANN could significantly outperform the baseline model even without *total_bilirubin*. It could also be interesting to see if a machine learning model could perform significantly better on a regression job than the baseline model on other attributes like *tot_protiens* or *ag_ratio*.

Regarding classification, the hopes for obtaining higher accuracy for classification of liver patients than the baseline model is not very optimistic. The high p-values in the pairwise comparisons indicate that the null hypothesis is most likely true, ie. the models perform identically. Test errors for all models are around 28 %, which is quite high, and this fact suggests that the attributes measured on the patients/non-patients isn't very useful to predict if a person is a liver-patient or not. In order to increase the performance on classification on patient vs not-patient, a bigger data set and additional attributes on the patients is needed. A different classification task on the Indian Liver data set could be to classify if a patient is male or female.

### Other work

An example of earlier analysis is done by a mr. Sangeetha James, who focused on regression analysis of the Indian Liver data set. In his analysis, *tot_bilirubin* was also removed, however he focused mainly on the prediction of patient status (disease/no disease).

Another example was done by a group of Indian students, who by focusing on classification obtained results in the range of 60% to 70%, which is similar to our classification results. These students used other algorithms though, including K-star and Naive Bayes.

First link:
`https://www.kaggle.com/sanjames/liver-patients-a`
`Observation`

Second link:
`https://www.researchgate.net/publication/327838226_Classification_of_Liver_Patient_Dataset_Using_Machine_Learning_Algorithms`

# Results

| Outer fold | ANN Reg. | | Lin Reg. w/o. tot_bilirubin | | Lin Reg. w. tot_bilirubin | | Baseline |
|---|---|---|---|---|---|---|---|
| i | Units | $E_i^{test}$ | $\frac{1}{\lambda}$ | $E_i^{test}$ | $\frac{1}{\lambda}$ | $E_i^{test}$ | $E_i^{test}$ |
| 1 | 5 | 3.63 | -1.0 | 7.79 | 1.89 | 3.91 | 7.72 |
| 2 | 1 | 8.48 | 0.79 | 7.09 | 1.89 | 3.63 | 8.37 |
| 3 | 1 | 9.91 | 0.39 | 7.99 | 0.49 | 5.42 | 8.18 |
| 4 | 1 | 7.03 | -1.0 | 7.56 | 1.89 | 5.23 | 7.16 |
| 5 | 1 | 3.21 | -0.3 | 7.73 | 1.99 | 5.34 | 7.99 |
| 6 | 1 | 14.83 | 0.39 | 7.38 | 1.90 | 5.15 | 8.35 |
| 7 | 1 | 6.78 | 0.09 | 6.95 | 1.89 | 4.95 | 7.78 |
| 8 | 1 | 2.73 | 0.59 | 6.49 | 1.99 | 4.75 | 7.76 |
| 9 | 1 | 4.04 | -1 | 7.28 | 2.09 | 4.88 | 8.08 |
| 10 | 5 | 5.55 | 2.09 | 10.04 | 2.29 | 5.89 | 7.80 |

Table 3: Regression results. (Regularization parameters in log

| | $\mathrm{M}_B$ : *Lin_Reg* | $\mathrm{M}_B$ : *ANN* | $\mathrm{M}_B$ : *Baseline* |
|---|---|---|---|
| $\mathrm{M}_A$ : *Lin_Reg* | | (-1.77 to 3.73)<br>p-value=0.44 | (-1.06 to 0.44)<br>p-value=0.37 |
| $\mathrm{M}_A$ : *ANN* | (-3.73 to 1.77)<br>p-value=0.44 | | (-3.84 to 1.25)<br>p-value=0.28 |
| $\mathrm{M}_A$ : *Baseline* | (-0.44 to 1.06)<br>p-value=0.37 | (-1.25 to 3.84)<br>p-value=0.28 | |

Table 4: Statistical evaluation on regression models

| Outer fold | Baseline | ANN,v1 | | ANN,v2 | | Log. regression | | KNN | | DT | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| i | $E_i^{test}$ | Units | $E_i^{test}$ | Layers | $E_i^{test}$ | $\frac{1}{\lambda}$ | $E_i^{test}$ | K | $E_i^{test}$ | Depth | $E_i^{test}$ |
| 1 | 28.5 | 8 | 55.9 | 2 | 28.8 | 0.91 | 25.4 | 25 | 25.4 | 1 | 28.8 |
| 2 | 28.5 | 4 | 30.5 | 2 | 28.8 | 0.91 | 37.3 | 25 | 25.4 | 1 | 28.8 |
| 3 | 28.5 | 9 | 28.8 | 2 | 28.8 | 0.91 | 27.1 | 28 | 32.2 | 1 | 28.8 |
| 4 | 28.5 | 9 | 30.5 | 2 | 28.8 | 0.91 | 23.7 | 25 | 28.8 | 1 | 28.8 |
| 5 | 28.4 | 9 | 34.5 | 2 | 29.3 | 0.91 | 25.9 | 28 | 27.6 | 1 | 29.3 |
| 6 | 28.5 | 8 | 43.9 | 2 | 28.1 | 0.91 | 29.8 | 28 | 24.6 | 1 | 28.1 |
| 7 | 28.5 | 8 | 38.6 | 3 | 28.1 | 0.91 | 29.8 | 25 | 35.1 | 1 | 28.1 |
| 8 | 28.5 | 8 | 31.6 | 3 | 28.1 | 0.91 | 29.8 | 28 | 28.1 | 1 | 28.1 |
| 9 | 28.5 | 8 | 33.3 | 3 | 28.1 | 0.81 | 29.8 | 28 | 28.1 | 1 | 28.1 |
| 10 | 28.5 | 8 | 42.1 | 3 | 28.1 | 0.91 | 28.1 | 28 | 29.8 | 1 | 28.1 |

Table 5: Classification results

| | $\mathrm{M}_B$ : *Log_Reg* | $\mathrm{M}_B$ : $KNN$ | $\mathrm{M}_B$ : *Baseline* |
|---|---|---|---|
| $\mathrm{M}_A$ : *Log_Reg* | | (-5.14 to 5.46)<br>p-value=0.95 | (-3.49 to 3.85)<br>p-value=0.91 |
| $\mathrm{M}_A$ : $KNN$ | (-5.46 to 5.14)<br>p-value=0.95 | | (-3.17 to 3.21)<br>p-value=0.99 |
| $\mathrm{M}_A$ : *Baseline* | (-3.85 to 3.49)<br>p-value=0.91 | (-3.21 to 3.17)<br>p-value=0.99 | |

Table 6: Statistical evaluation on classification models

# 1 Appendix

| **Attribute:** | **Type:** | **Describtion:** |
|---|---|---|
| age | Discrete, Ratio | Age in years |
| gender | Nominal | Gender (Male/Female) |
| tot_bilirubin | Continuous, Ratio | Amount of Direct and Indirect Bilirubin |
| direct_bilirubin | Continuous, Ratio | Amount of Direct Bilirubin |
| tot_proteins | Discrete, Ratio | Total proteins amount in patients blood |
| albumin | Discrete, Ratio | Albumin amount in patients blood |
| ag_ratio | Discrete, Ratio | Ratio between proteins Albumin and Globulin |
| sgpt | Continuous, Ratio | Alamine Aminotransferase amount in patient |
| sgot | Continuous, Ratio | Aspartate Aminotransferase amount in patient |
| alkphos | Continuous, Ratio | Alkaline Phosphotase amount in patient |
| is_patient | Nominal | Liver patient (Yes/No) |

Table 7: Attribute types and descriptions.

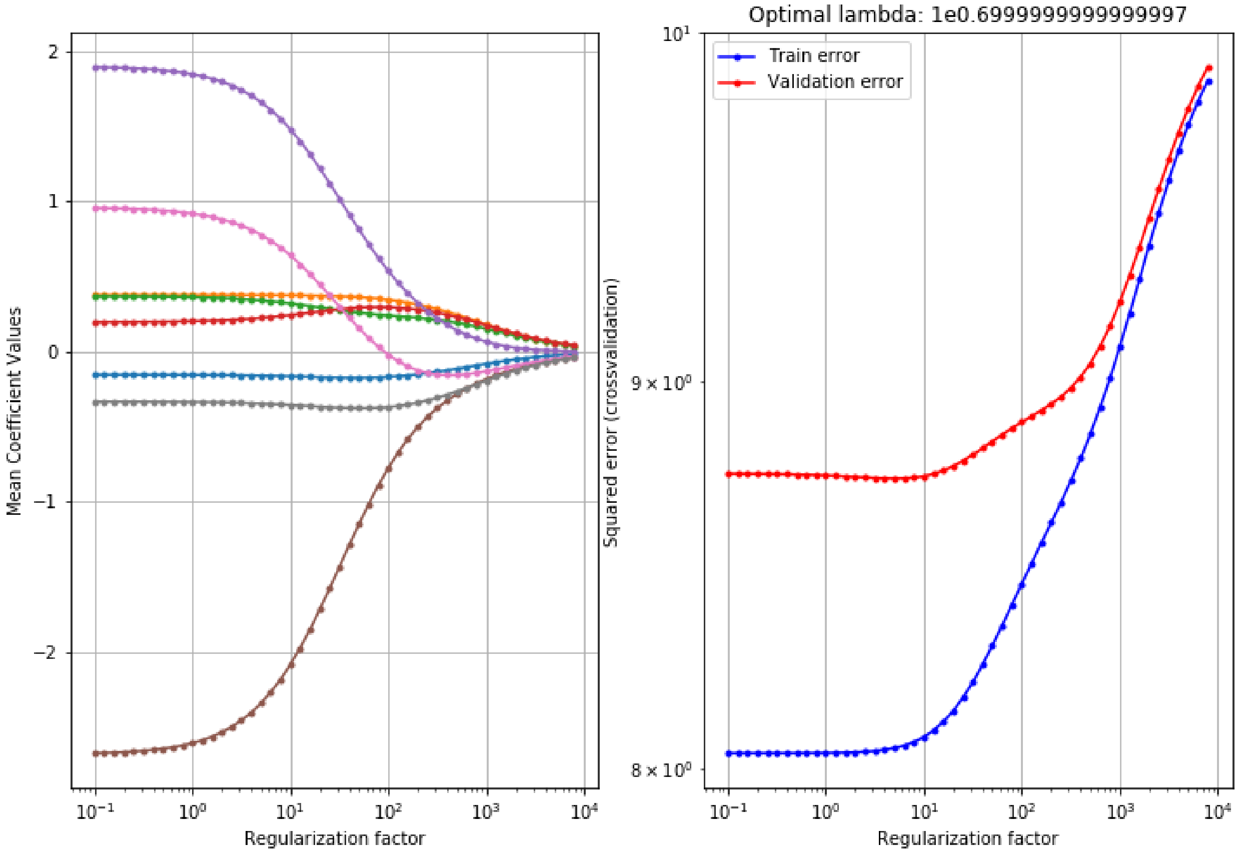


Figure 4: Figure 1 in bigger scale.

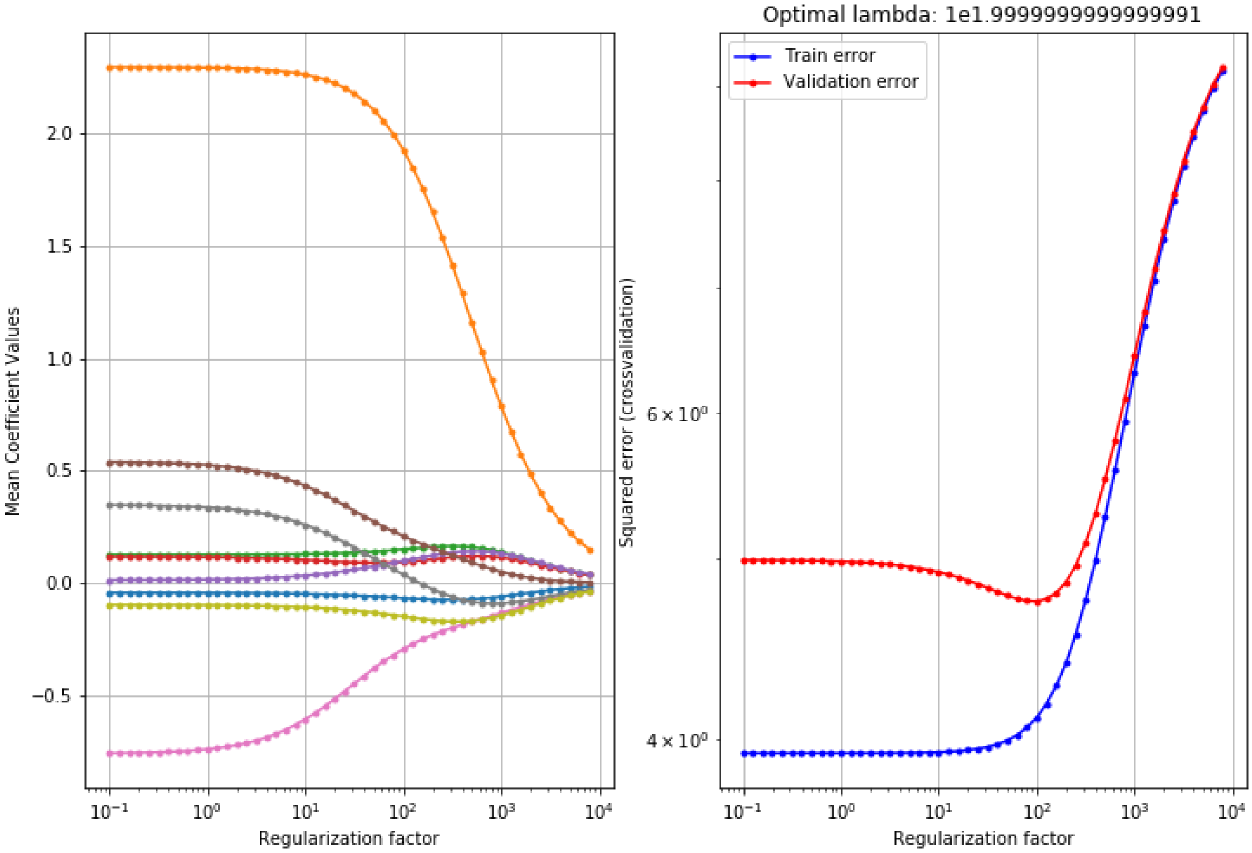


Figure 5: Figure 2 in bigger scale

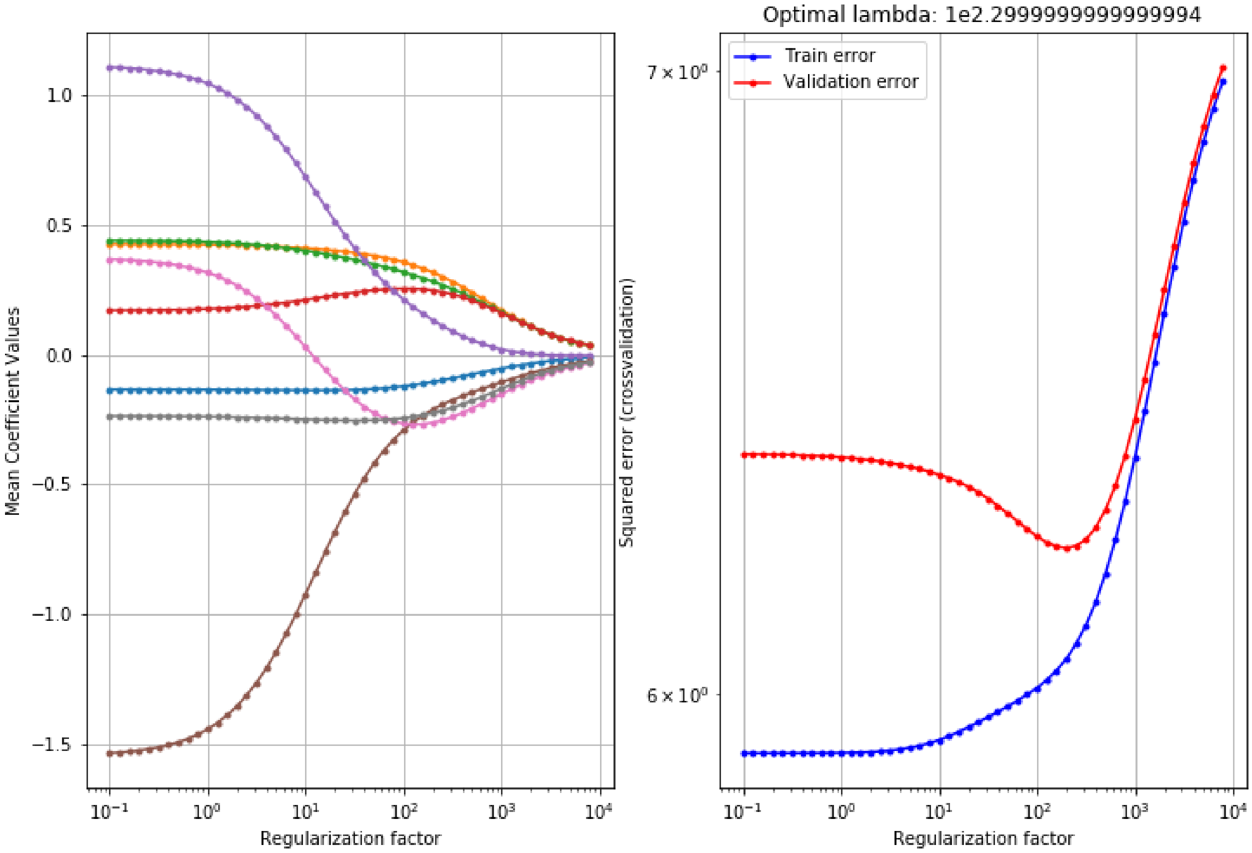
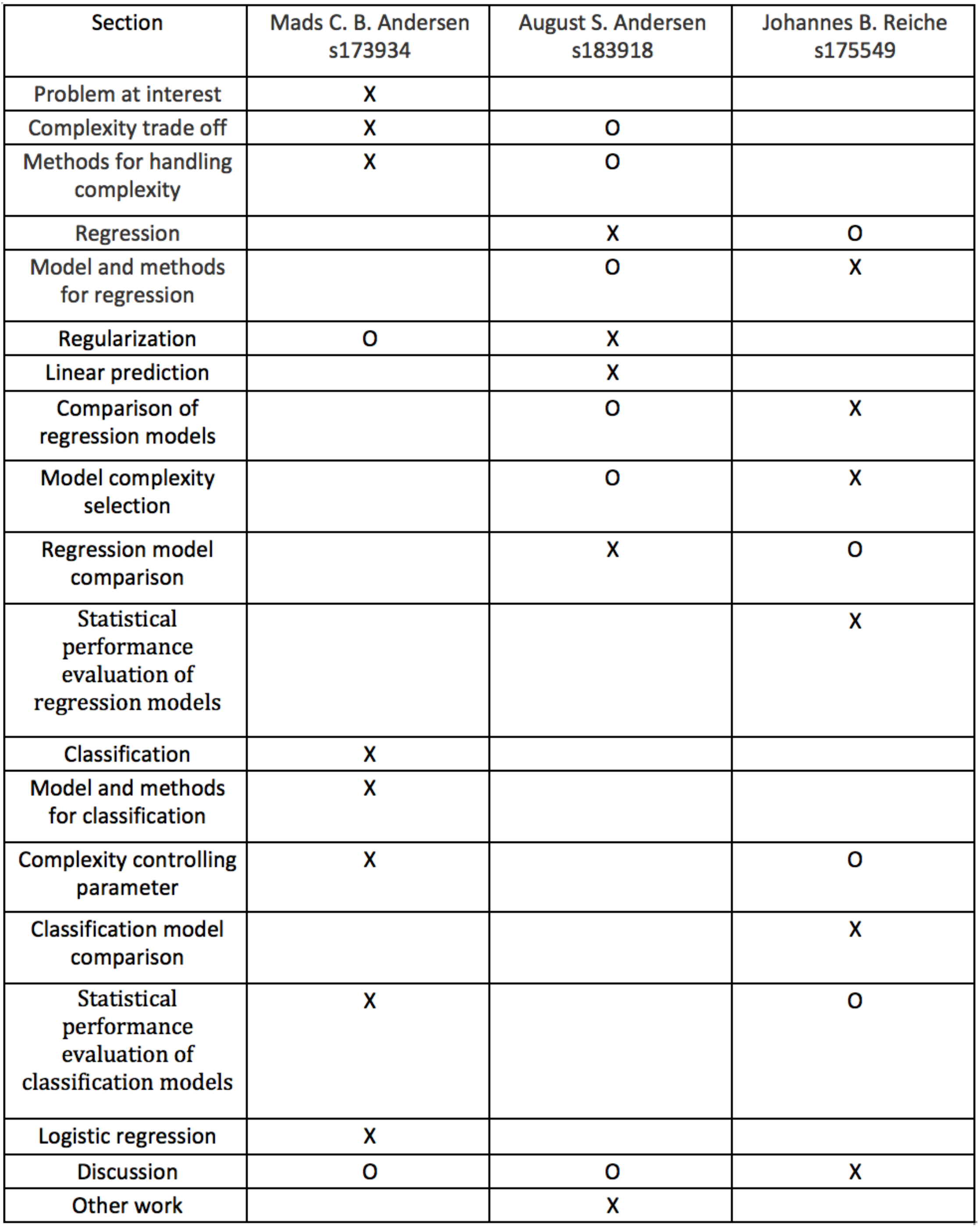


Figure 6: Figure 3 in bigger scale.



| Section | Mads C. B. Andersen<br>s173934 | August S. Andersen<br>s183918 | Johannes B. Reiche<br>s175549 |
|---|---|---|---|
| Problem at interest | X | | |
| Complexity trade off | X | O | |
| Methods for handling complexity | X | O | |
| Regression | | X | O |
| Model and methods for regression | | O | X |
| Regularization | O | X | |
| Linear prediction | | X | |
| Comparison of regression models | | O | X |
| Model complexity selection | | O | X |
| Regression model comparison | | X | O |
| Statistical performance evaluation of regression models | | | X |
| Classification | X | | |
| Model and methods for classification | X | | |
| Complexity controlling parameter | X | | O |
| Classification model comparison | | | X |
| Statistical performance evaluation of classification models | X | | O |
| Logistic regression | X | | |
| Discussion | O | O | X |
| Other work | | X | |

Figure 7: Responsibility table

The person in charge of a section is marked with a 'X' and have been at least 50% responsible for the section. Co-editor(s) have received an 'O'.