Machine Learning Projekt 2

August Tollerup (s204139)
Smilla Due (s204153)

November 2021

1 Introduction

The data set of the world happiness scores, used in the 2016 world happiness report, was found to contain 4 important key features. Namely, "economy", "family", "health" and "freedom". In this project we will reinforce our analysis of the data set by applying 3 machine learning models to analyse to what degree we can train models to predict the outcome of a country yet to be rated. We will apply 3 models of regression and 3 of classification. The regression models should predict the Happiness Score, and the classification should predict the binary Happy attribute. For regression we will apply a Linear Regression-, Neural Network- and a baseline model. For classification we will apply a Logistic Regression-, Neural Network- and a baseline model. Where we prior focused on 4 key attributes we will now take all 7 attributes into account. Furthermore, from prior analysis it was discovered that the happiness score was based on the summation of the other attributes. This leads us to hypothesise that a Linear Regression model will be highly efficient in predicting the happiness score based on the attributes. Therefore, we hope to accomplish a very good linear regression model, that can predict the happiness score from the different attributes.

2 Regression, part a

When working with data in different spaces we need to standardise our data. An example of different spaces could be the shoe size and height of a person. If we want to work with both the shoe size and height in our model, we need to translate/scale these features into the same space. If we do not scale the features there is a high probability of the features developing multicolinearity. This effectively means that we cannot measure the impact each feature has on the model.

2.1 Standardisation

One way of doing this is to standardise each feature. We have chosen to standardise our data with the Standard Deviation Standardisation:

$$\hat{X}_{ij} = \frac{X_{ij} - \mu_j}{\hat{s}_j}, \quad \mu_j = \frac{1}{N}\sum_{i=1}^{N} X_{kj}, \quad \hat{s}_j = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(X_{ij} - \mu_j)^2}$$

We standardise each column by subtracting the mean, μ_X, and dividing by the standard deviation, σ_X, for each column.
In addition to standardisation, if some data is categorical (nominal) we need to convert this into a binary feature. An example of feature transformation from ordinal to binary is one-hot encoding or one out of k coding. For this data set we have not deemed it necessary to do further feature transformation since we do not have any categorical features.

2.2 Linear Regression and Regularisation

The Linear Regression model finds the best fit line for the given input data. This means the model only works for a continuous output value. Furthermore, the model will use a Mean Squared Error loss function, which calculates the mean distance of each training point to the fit model. A simple linear regression model would be a single input and single output value:

$$y = w_1 x + w_0$$

Where w_1 and w_0 are the weights. If a weight is set to 0, it essentially means the model disregards that weights' feature in the prediction of the target value y. In some cases the weights can become larger than necessary and we therefore need to implement the regularisation term λ. We can then optimise our model using cross-validation to get the optimal regularisation term. If we suppose the following:

$$w^* = \left(\hat{X}^\top \hat{X} + \lambda I\right) \setminus \left(\hat{X}^\top \hat{y}\right) \propto \frac{Xy}{X^2 + \lambda}$$

Then we can see that if $\lambda = 0$, there is no effect. But, as we increase $\lambda \to \infty, \mathbf{w}^* \to 0$. Essentially, when we regularise our model we are optimising the bias-variance trade off. In short, we consider variance to be the flexibility of our model, and the bias to be how constant the model is. We chose to find the optimal λ in a range between 0.001 and 1. We did this after testing on a wide range all the way down to $1 \cdot 10^{-7}$, and found that there was little or no difference in the generalisation error when the λ was between $1 \cdot 10^{-7} - 1 \cdot 10^{-3}$. This tendency can also be seen on figure 1, where the curve flattens out around $1 \cdot 10^{-2}$. Since out data is perfectly

linear, a lower regularisation factor will not give us a higher mean squared error, and we therefore do not have a graph where the generalisation error first drop and then increases. Instead we have a graph, where the mean squared error is at its lowest around $1 \cdot 10^{-7}$ when the regularisation error is below $1 \cdot 10^{-2}$ and then increases when the regularisation error also does so. One can therefore discuss if it even makes much sense to implement regularisation to a problem like ours, where the attributes explain all of the happiness score that we want to predict[1].

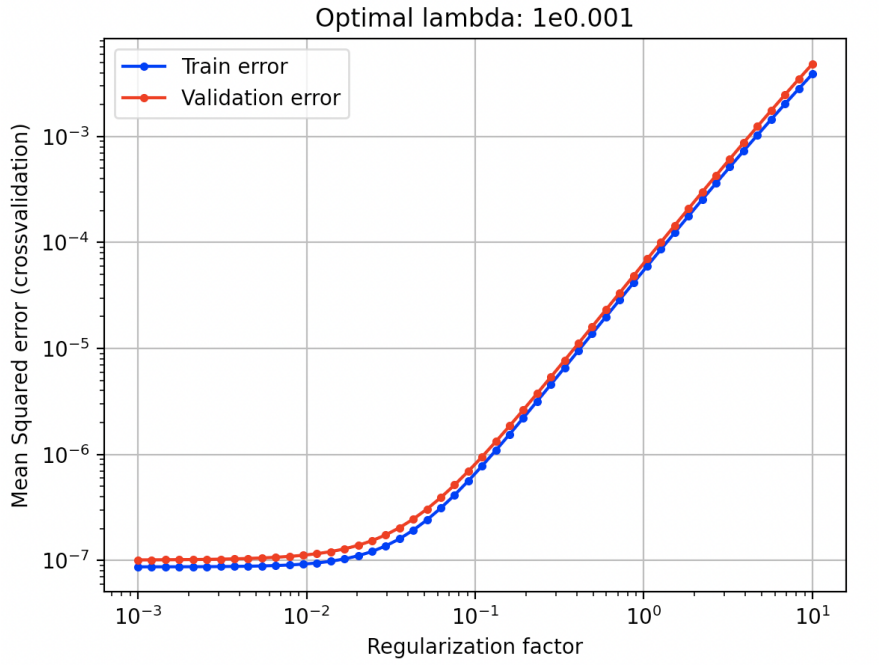

Figure 1: Estimated Generalization Error as a Function of λ

2.3 Sub Conclusion

Using K-fold cross validation with $K = 10$, we have analysed our model and found a generalisation error of $9.296 \cdot 10^{-8}$. In 8 out of the 10 folds we found an optimal $\lambda = 0.001$, which is the same as what we can see on figure 1. This very low generalization error tells us that the linear regression model will perform more very well on new data, which is not very surprising for a data set like ours. With our data, the error is probably mostly a consequence of rounding. The happiness scores are only evaluated at 4 significant digits, hence if we were to round to 4 significant digits for our generalisation error we would get a model with near perfect fit.

3 Regression, part b

It is now evident a Linear Regression model fits our data well, although we need to evaluate it compared to other models. We will compare the three models; The Regularised Linear Regression model, an Artificial Neural Network and a baseline model. We will do this to examine which model is better and if any of the models are significantly better than the other. In this section we will use two-level cross-validation, when we try to answer this, and the error measure is the squared loss per observation.

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

3.1 Baseline model

The baseline model is a linear regression model without features. This means, that the regression will be based on the mean of the training data set, and will predict the happiness scores from this mean value. We expect this to perform poorly, although it is a very simple model from which we can compare the other models. The baseline model is defined as:

$$y_{predict} = \mu_{\text{training data}}$$

[1]The sum of the 7 attributes gives us the happiness score, as explained in our Machine Learning Project 1

Since the baseline model predicts from the mean, the error measure will be the average squared distance between the prediction and the mean. This is essentially the variance of our data, and therefore we expect the error measure to align with the variance, which we found in report 1 to be 1.30.

3.2 Artificial Neural Network

The Artificial Neural Network (ANN), is defined with different numbers of hidden layers. The hidden layers are the layers in the neural network between the input and output layer. The more hidden layers, the more complex the model but also the bigger the chance of over-fitting. In other words, the number of hidden layers can help us control the complexity of our model. We have done a few test-runs of different values of hidden layers, h, and chose to go with the range of 1-10. However, since our data is not very complex, and as Regression, Part a showed, very linear, we expect the ANN to perform well on few hidden layers. The cost function of the ANN is chosen as the Mean Square Error. The chosen non linear activation function is the hyperbolic tangent function. This is an advantage with normally distributed data, since it produces a zero centred output.

3.3 Two-level cross validation

To train and test our models, we have used two-layer cross validation with each $K_1 = K_2 = 10$. When using only one layer cross validation there occurs a problem, namely that the generalisation error is the best possible generalisation error, which is not representative for new data.
By implementing two-layer cross validation, the data is firstly, partitioned K-times into test and train sets. Secondly, each test and train sets are then partitioned into K test and train sets. We use the second level partitioning to optimise our hyper-parameters; hidden-layers (h) and regularisation constant (λ). To to compare and evaluate each model respectively each model is trained on the same cross-validation partition. The optimal hyper-parameters and the generalisation error for each model are displayed in table 1

Outer Fold	**ANN**		**Linear regression**		**Baseline Model**
i	h_i^*	E_{i,h^*}^{test}	λ_i*	E_{i,λ^*}^{test}	$E_{i,base}^{test}$
1.0	2.0	0.867	0.001	0.000	1.090
2.0	1.0	1.530	0.001	0.000	1.401
3.0	2.0	0.694	0.012	0.000	1.141
4.0	2.0	1.106	0.001	0.000	1.587
5.0	2.0	1.106	0.001	0.000	1.347
6.0	2.0	0.915	0.012	0.000	1.629
7.0	2.0	1.233	0.001	0.000	1.268
8.0	2.0	0.683	0.001	0.000	1.271
9.0	2.0	0.835	0.002	0.000	1.633
10.0	2.0	0.855	0.012	0.000	0.859

Table 1: Results of the Two-level Cross Validation for Regression

In table 1 we see that in 9/10 of the outer folds, it is estimated that two hidden layers would be optimal for the ANN, and that in 6/10 cases $\lambda = 0.001$ would be optimal for the Linear Regression model. This is the same optimal regularisation constant that we found in Regression, part a, which is as expected. We also see in table 2, that we get a marginally higher generalisation error of the linear regression, which can probably be dismissed as a either a consequence of the "random" partition indexing when using k-fold functions or due to the fact that we with one-level cross validation get "the best possible generalisation error", whereas with two-level cross validation, we get a more "realistic" and therefore higher generalisation error. However, the difference is of a magnitude which could depend on the computers accuracy when calculating. When looking at the generalisation errors of the different outer folds one thing clearly stands out - for all 10 folds the generalisation error of the Linear regression model is 0.000. This is a clear indication that the Linear Regression model will be able to predict new data very well. Overall it seems that the generalisation error of the ANN is slightly lower than our baseline model, which also can be seen in table 2.

	ANN	**Linear Regression**	**Baseline Model**
Generalisation Error	0.982	$1.000 * 10^{-7}$	1.323

Table 2: Generalisation Errors of the Regression Models

The baseline model gives us a generalization error of 1.323, which aligns very well with our hypothesis, that it would be close to the variance of 1.30.

3.4 Statistical Evaluation of the Regression Models

We wish to statistically evaluate if there is a performance difference between the three models. To do so, we will evaluate the models pairwise. We have chosen to do so via a paired t-test. The method is described in 11.3.4[2]. We have calculated the confidence interval at a 95 % confidence interval. The results from the t-test can be seen in the table 3 below.

	Lower conf. int.	**Upper conf. int.**	**p-value**
ANN - Baseline Model	0.146	0.590	0.005
Linear Regression - Baseline Model	1.118	1.524	$1.332 * 10^{-7}$
ANN - Linear Regression	0.764	1.141	$1.177 * 10^{-6}$

Table 3: Results of the t-test (Method I)

The confidence intervals listed in the table tells us in what interval the difference between the generalisation errors of the two functions will be. For instance, we now know with statistical significance that the ANN will have a generalisation error that is between 0.146 and 0.590 lower than the baseline model. The p-values tells us, that when you compare the models pairwise they will not perform equally good.
In all of the cases, the p-value is lower than the statistical significance level of $\alpha = 0.05$. We can therefore with statistical evidence reject the null-hypothesis that the models will perform equally well.
In section 3.3, we saw that the generalisation error of the ANN was 0.982 which is lower than the generalisation error of our baseline model, which is at 1.323. However, now we know that there is also strong statistical evidence that the ANN generally is a better regression model than the baseline model. This is very convenient, since it would be a very poor ANN regression model, if it performed equally to a linear regression based on the mean value of the happiness score, instead of the attributes present in the data.
With the significantly low p-values that occur when you compare the Linear Regression model with the other two models, we have as strong evidence, as it gets to reject the hypothesis, that the models perform equally good. The generalisation error of the linear regression model is approximately at zero, so this strong statistical evidence is not surprising. All in all, we can conclude, that the linear regression model by far is the best model of the three, which aligns with our hypothesis that the linear regression model will be highly efficient in predicting the happiness scores.

4 Classification

In the classification task we will try to classify a country to the binary category happy/unhappy. As describes in our first report, the category (happy/unhappy) is created by dividing the countries according to their happiness score. The countries with a happiness score above the median of all the happiness scores are classified as happy, whereas the rest is categorised as unhappy. We now wish to implement a good classification model that should be able to predict the happiness scores based on the attributes "Happiness Score", "Economy", "Family", "Health", "Freedom", "Trust", "Generosity" and "Dystopia Residual". The first model will be a Logistical Regression model, the second model an ANN and finally we have a Baseline model, so that we can better compare the models. We will evaluate the models based on two-level cross validation, and we will use the error rate at the error measure.

$$E = \frac{\text{Number of miss-classified observations}}{N_{test}}$$

4.1 Baseline model

The baseline model for the classification task, will categorise all new data to the biggest category of training data set. Since our data set is binary and the categories by definition are equally big, we will assume that the baseline model, will have an accuracy of approximately 50%, and therefore an error rate of around 0.5.

4.2 Logistical Regression

As in the Linear Regression model, from the section 2, we will use λ as the complexity-controlling parameter of the model. With the same procedure as previously, we choose a range of values of λ based on a test-run,

[2]Introduction to Machine Learning and Data Mining, p. 213

and do the two-level cross validation based on this. We have chosen the same range for the Logistic Regression model as for the Linear Regression model, as we expect the range to be broad enough to provide evidence for the optimal lambda.

4.3 Neural Network Classification

For the Neural Network classification we maintain the complexity controlling parameter to be the amount of hidden layers. As in the regression part, we chose to test on a range of 1-10 hidden layers. Finally, we chose the Binary Cross Entropy function as our loss function and the output layer as a sigmoid function, which works well with the data since our target is a binary value.

4.4 Two-level Cross Validation

After running the two-level cross-validation, we get the results that can be seen in table 4.

Outer Fold	**ANN**		**Logistical regression**		**Baseline Model**
i	h_i^*	E_{i,h^*}^{test}	λ_i*	E_{i,λ^*}^{test}	$E_{i,base}^{test}$
1.0	1.0	0.562	0.001	0.000	0.438
2.0	1.0	0.625	0.001	0.000	0.625
3.0	1.0	0.500	0.087	0.000	0.375
4.0	1.0	0.531	0.094	0.000	0.438
5.0	1.0	0.531	0.001	0.000	0.438
6.0	1.0	0.500	0.001	0.062	0.500
7.0	1.0	0.469	0.001	0.000	0.562
8.0	1.0	0.594	0.001	0.000	0.467
9.0	1.0	0.500	0.001	0.000	0.667
10.0	1.0	0.469	0.001	0.067	0.533

Table 4: Results of the Two-level Cross Validation for Classification

In the regression part, we found an optimal number of hidden layers to be 2, however in the classification neural network, we see in 10/10 cases that the optimal hidden layer is 1. This might be an indication that our data set is simple, and therefore is most effectively predicted with a simple model. Additionally, the high Generalisation error could also indicate overfitting of the ANN, where it will have a very low train error but perform poorly on new data. By running the ANN with fewer epochs / iterations we could test this hypothesis, although for this report we will not do so.
The indication that a simple model would perform better is also evident when looking at the logistic regression, where we see, that this performs well. Much like in the regression part, we find the optimal regularisation term λ to be 0.001, which makes sense due to similar arguments as put forward in section 2.2 - Regularisation.

	ANN	**Logistic Regression**	**Baseline Model**
Generalisation Error	0.477	0.013	0.504

Table 5: Generalisation Errors of the Classification Models

The mean generalisation error of the three models can be seen in table 5. Firstly, we notice that the logistic regression, much like the linear regression in the regression part, seem to perform best. This generalisation error tells us that this model would be able to predict on new data whether a country is happy/unhappy with an error rate of only 0.013, corresponding to that it would miss-classify only around 1 % of new data. On the very contrary, the ANN seems to be almost just as bad as the baseline model. The ANN is in other words not much better than flipping a coin, which we did not expect. As mentioned this might be the cause of over-fitting, but with additional test one could have experimented with other activation functions and partitions of the cross validation. For a neural network to really show its advantages, you need to have access to a lot of data, which is not the case with our only 156 countries, which we divide into very small training and test sets, when doing a k=10 fold two-layer cross validation.

4.5 Statistical evaluation of the models

To evaluate and compare the models we perform a statistical evaluation. We compare the models pairwise using method 11.4.1[3], and the confidence interval is calculated at 95 %. The difference between the method used for regression and classification lies in method 11.4.1 comparing the generalisation error difference for each fold, which ultimately will result in confidence intervals and p-values based on a more general version of the generalisation error[4]. The differences in the generalisation errors for each fold are displayed in table 6.

Outer Fold	$E^{test}_{i,h^*} - E^{test}_{i,base}$	$E^{test}_{i,h^*} - E^{test}_{i,\lambda^*}$	$E^{test}_{i,base} - E^{test}_{i,\lambda^*}$
1.0	0.125	0.562	0.438
2.0	0.000	0.625	0.625
3.0	0.125	0.500	0.375
4.0	0.094	0.531	0.438
5.0	0.094	0.531	0.438
6.0	0.000	0.438	0.438
7.0	-0.094	0.469	0.562
8.0	0.127	0.594	0.467
9.0	-0.167	0.500	0.667
10.0	-0.065	0.402	0.467

Table 6: Difference in Generalization Error

From table 6, we see that the difference between the errors of the ANN and Baseline lies around zero, which is a clear indication that the models might be equally bad. However, in both of the cases, where the logistic regression is compared with the other models, we clearly see that there is a difference between the models error rates of around 0.5. This is also what we see in the final results of the correlated t-test in table 7, namely that the difference between the error measures of the logistic regression and the two other models at a 95 % confidence interval will be between 0.466-0.564 for the baseline and 0.424-0.559 for the ANN. This is extremely high in a binary classification with 50 % in each class, and is only possible because the logistic regression has an error rate significantly closer to zero while the two others has error rates of around a half (see table 5).

	Lower conf. int.	**Upper conf. int.**	**p-value**
ANN - Baseline Model	-0.052	0.099	0.492
Logistic Regression - Baseline Model	0.466	0.564	$1.980 * 10^{-9}$
ANN - Logistic Regression	0.424	0.559	$4.905 * 10^{-8}$

Table 7: Results of the Correlated t-test (Method II)

From the p-values in table 7 we can conclude, that we have very strong evidence for rejecting the null-hypothesis: the two compared models have the same generalisation error in the two cases with logistic regression. However, we cannot reject the null-hypothesis when we compare the ANN and Baseline model, and we therefore accept H_0. When looking at the confidence interval we also see, that in 95 % of the times $E_{ANN} - E_{baseline} \in [-0.052, 0.099]$ which is in agreement with $H_0 : E_{ANN} - E_{baseline} = 0$. Hence, there is no significant difference between the ANN and the Baseline model.

4.6 Logistical Regression as a concept

Logistic Regression is a predictive analysis. The model works well for binary outputs. It is similar to Linear Regression although it provides the Probabilities of for classifications rather than the best fit line. Whereas linear regression uses MSE or RMSE as a loss function, the Logistic Regression function uses Maximum Likelihood to determine the optimal parameters for the model to maximise the said function. In essence, the Maximum Likelihood Loss-function finds the parameters that makes the observed data most probable. Furthermore, the Logistic Regression model is a transformation of the Linear Regression model by putting the linear model through a sigmoid function. The Logistic Model assumes the data follows a Bernoulli Distribution. When a new datapoint is added the model simply calculates the output of the sigmoid function with a threshold at 0.5 to round either down to class 0 or up to class 1.
We applied the optimal regularisation value and trained a logistic regression model. The weights can be seen in the table below:

[3]Introduction to Machine Learning and Data Mining, p. 216
[4]Introduction to Machine Learning and Data Mining, p. 216

Weights	Logistic Regression	Linear Regression
Economy	13	0.4
Family	11	0.3
Health	9	0.2
Freedom	3	0.1
Trust	4	0.1
Generosity	4	0.1
Dystopia Residual	23	0.5

Table 8: Weights in Logistic and Linear Regression

From above table we see that although the weights are of different magnitude, the same features are weighted higher for both models. This gives us a strong indication that the features, namely: "Economy", "Family" and "Dystopia Residual", are more relevant than the other features for predicting the target value.

5 Discussion

We hypothesised that the linear and logistic regression would be very accurate when predicting the happiness score and happy/unhappy category. This is also, what we found in the report, however we had not expected that the linear regression would give a generalisation error of approximately zero. In our first report, we chose to use only 4 attributes to predict from, because then our attributes would not explain all of our data. However, in the feedback of the report, we were encouraged to use more attributes, and we have therefore used all seven, and due to this our attributes explain all of our happiness score[5]. Because of this, the perfect linear regression, actually makes a lot of sense, since our attributes perfectly lineally describes the happiness score, furthermore this also explains the low generalisation error of the logistic regression.
The two ANN's performed surprisingly bad, with very high generalisation errors especially in the classification part. After conducting a correlated t-test, we had to conclude that we accept the null hypothesis; that the ANN and the baseline performed equally bad. As discussed in section 4.4, we would in an additional study have liked to investigate the reasons of this further. We can now only assume what caused the bad results of the ANN's, which we consider to be a mix of the fact that an ANN might be a too complicated model for a so simple and linear data set like ours (especially when using a non-linear activation function), and that with a small data set, and k=10 two-fold cross validation the ANN had to train on very small data sets, which does not justify the true advantages of an ANN. Finally, we also considered this to be a matter of over fitting (see section 4.4). We assume that Naive Bayes, k-nearest neighbour and decision trees all would have been more fit classification models for our problem, which also could have been interesting to analyse further in another report. Creating this report, and using the different models on a real data set has been very rewarding - maybe especially due to the changing results discussed above. To try to understand the big differences in our models performances has given us a broader understanding of both our data set and the machine learning methods, and will from now on be something we can reference to in our further work within machine learning.
The final thing we wish to discuss is how our data set has been analysed before, and how we can relate this to our report.
The data has been thoroughly analysed in the World Happiness Report of 2016. In our first report we summarised many aspects of the report, in particular those that had to do with the weight of the attributes: *"The world happiness report of 2016 found that almost three-quarters of the happiness score can be explained by the six subcategories. 31% of the "share of explained excess happiness over Dystopia" is a matter of wealth, 26% is a matter of the social support from family and friends, 18% is due to healthy life expectancy, 12% is explained via the degree of freedom to make life choices and only 8% and 5% is due to generosity and perceptions of corruption, respectively."* [6]. However, in this report, it is perhaps more interesting to look at some comparable examples of classification and regression used in the happiness report. The things that we found resembled what we have examined in our report the most, are depicted in figure 2.
In the left pane, we see a figure that illustrates how the happiness report have been able to predict the happiness scores from 2013-15 compared to their actual happiness score. In the report, the prediction has been carried out through weighted regression of the predicted attributes[7]. From the figure it seems that the linear regression is rather good to predict the happiness scores, which aligns with what we found in our regression part of the report. However, it doesn't seem to have as high an acurracy as wee accomplished, which is somehow unsurprising since we did the regression on the actual attributes that explains all of the happiness score, and the regression from

[5]The sum of the 7 attributes gives us the happiness score, as explained in our Machine Learning Project 1
[6]Machine Learning Project 1, p. 2
[7]Statistical Appendix, p. 19

Figure 4: Predicted happiness and actual happiness in 2013-15

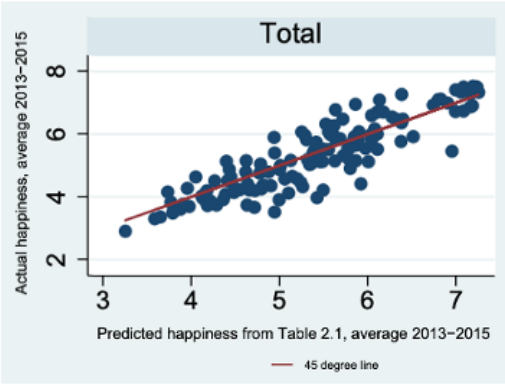

Table 14: Decomposing the happiness difference between the group of top 10 countries/territories and the group of bottom 10 countries/territories in the ranking of happiness scores

	Top 10	Bottom 10	Difference in happiness due to	Share of explained difference due to
Happiness	7.41	3.42		
Logged GDP per capita	10.7	7.37	1.13	.36
Social support	.94	.6	.8	.25
Healthy life expectancy	71.65	53.11	.54	.17
Freedom to make life choices	.93	.63	.32	.1
Generosity	.21	.04	.13	.04
Perceptions of corruption	.36	.74	.22	.07
Total explained difference in happiness			3.14	1
Total difference in happiness			3.99	

Figure 2: Happiness Report 2016, selected figures

the happiness report is carried out on predictions of the attributes, which we from our first report know explains a bit less than three-quarters of the happiness score.

In the right pane of figure 2 we see a table, where the attributes impact on the happiness scores for the top 10 and bottom 10 countries are explained. This somewhat resembles what we have carried out in the classification task, where we have looked at the top 50 % (Happy) against the bottom 50 % (Unhappy). If we compare the "share of explained difference due to" with the weights of the logistic regression in table 8 we see the same tendencies, and the same order of the weights (without Dystopia Residuals), with economy being the dominant factor and freedom the least important one. The fact that the perception of freedom in ones country seem to matter so little when measuring the difference between the most happy and most unhappy countries, is actually quite surprising. "Freedom to make life choices" actually explains 12% of the happiness over Dystopia[8], which is higher than for both corruption and generosity. This tells us that there is a difference in weights depending on what problem we analyse (if we have a regression problem were we need to predict the happiness score, or if we have a classification problem, were we have to predict if a country is classified as happy or unhappy). For instance freedom seem to have a lower impact in classification than it does in regression.

Results like these can be very valuable and working with this report has therefore additionally also taught us a lot about the individuality of the perception of happiness and how there seem to be cultural differences between "happy" and "unhappy" countries.

[8]Statistical Appendix - Table 13

6 Responsibility Assignments

We have here a table of the responsibility assignments.

	s204139	s204153
1. Introduction	90%	10%
2. Regression, part a	80%	20%
3. Regression, part b, 3.0-3.2	40%	60%
4. Regression, part b, 3.3	60%	40%
5. Regression, part b, 3.4	40%	60%
6. Classification, 4.1-4.3	90%	10%
7. Classification, 4.4-4.5	10%	90%
8. Classification, 4.6	90%	10%
9. Discussion	10%	90%

7 Exam Questions

- **Exam question 1, Option D:** We see that the curve is "close" to being a straight line which indicates the classifier is random. We therefore expect that the Prediction D is correct.
- **Exam question 3, Option A:** The input is a vector with length 7. It is passed through a single hidden layer with 10 neurons. From that to an output layer with 4 outcomes. The Neural Network will have 7x10 + 10x4 = 110 weights and if we account the biases we get 7x10 + 10x4 + 10 + 4 = 124.
- **Exam question 5, Option C:** $(k_2 \cdot S + 1) \cdot k_1$ = Total number of models we have to train. $k_2 = 4$ and $k_1 = 5$ and $S = 5$. This means that we have a total of models to train of: $(4 \cdot 5 + 1) \cdot 5 = 105$ for the ANN and the Logisic Regression. The time will then be $105 \cdot (20 + 5 + 8 + 1) = 3570$ ms.
- **Exam question 4, Option D:** The reason, that the answer is D, can bee seen when looking at C. If C is true it will be classified as congestion level 4. We can see, that all of congestion level 4 is when $b_1 \geq -0.16$. The only answer, that suggests this i D. Furthermore, we check, that the rest of the classification also aligns with the answers in D, which we conclude that it does.

8 References

- Introduction to Machine Learning and Data Mining, *Tue Herlau, Mikkel N. Schmidt and Morten Mørup*
- World Happiness Report 2016: *https://worldhappiness.report/ed/2016/*
- Statistical Appendix: *https://s3.amazonaws.com/happiness-report/2016/StatisticalAppendixWHR2016.pdf*
- https://stackoverflow.com/questions/67513075/what-is-c-parameter-in-sklearn-logistic-regression
- https://scipy-lectures.org/packages/scikit-learn/auto_examples/plot_linear_model_cv.html
- https://stackoverflow.com/questions/48152674/how-to-check-if-pytorch-is-using-the-gpu
- https://towardsdatascience.com/probability-concepts-explained-maximum-likelihood-estimation-c7b4342fdbb1