02450 Introduction to machine learning - Report 2

Georgios Kapakoglou(223001), Matija Šipek (222736),Edison Von Matt(223534)

March 28, 2023

Contents

1 Introduction

1.1 Overall problem of interest

For this report we have used a data set for glass identification. It was used by the USA forensic science service to distinguish between seven different types of glass based on their oxide content and refractive index. The data set is multivariate and contains 9 attributes (RI, Na, Mg, Al, Si, K, Ca, Ba, Fe) and the subject area is physical.

2 Regression, Part A

2.1 Variables, feature transformation and normalization

In this part we will try to find out how the refractive index (RI) depends on the different oxide levels described by the attributes (Na, Mg, Al, Si, K, Ca, Ba, Fe) as the RI is continuous and thus suitable for regression. We normalized the data prior to the model implementation. It is to note that we did not apply one-out-of-k encoding as we only wanted to assess the direct relationship between the chemical components and the refractive index, without including the labels of the glass type .

2.2 Regularization Parameter

We chose the same regularization parameter range as in the 8th exercise: *(lambdas - 0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000, 100000, 1e+06, 1e+07, 1e+08)*. We ran some test runs and it always included an initial error drop among the lower lambdas and a quick and a steep error increase after the λ value of 1. Focusing on the test error, the drop in error is usually taking place for the first 3 λ's where afterwards the error increases again after a λ value of 1 and stagnates again after a λ value of 100000 which is also visualized in the 3.2 - however the drop might not be visible but the first five test error values are (1.0998e-06, 1.0998e-06, 1.09978e-06, 1.09969e-06, 1.10138e-06)

Subsequently, we implemented a 10-fold cross-validation using the same test/train fold for each lambda to allow a direct performance comparison across all lambdas and compared the average error values to find the optimal lambda. We used the mean squared error according to the formula: $E_\lambda = ||\hat{y} - \hat{X}w||^2 + \lambda||w||^2$.

By examining Table 1 and 2, it can be observed that both models (regularized linear regression and basic mean estimation) have very similar performance values and that the test error has not decreased as much as we had hoped. However, it is to note that the refractive index value (y) in our data set only ranges from 1.51115 to 1.53393 which results in generally low test error as well as low absolute performance differences among models.

- Training error:	9.59585e-07
- Test error:	1.0998e-06

Table 1: Baseline model (mean estimation) - average mean squared error over 10 folds

- Training error:	9.59623e-07
- Test error:	1.09969e-06

Table 2: Regularized linear regression - average mean squared error over 10 folds (for optimal lambda)

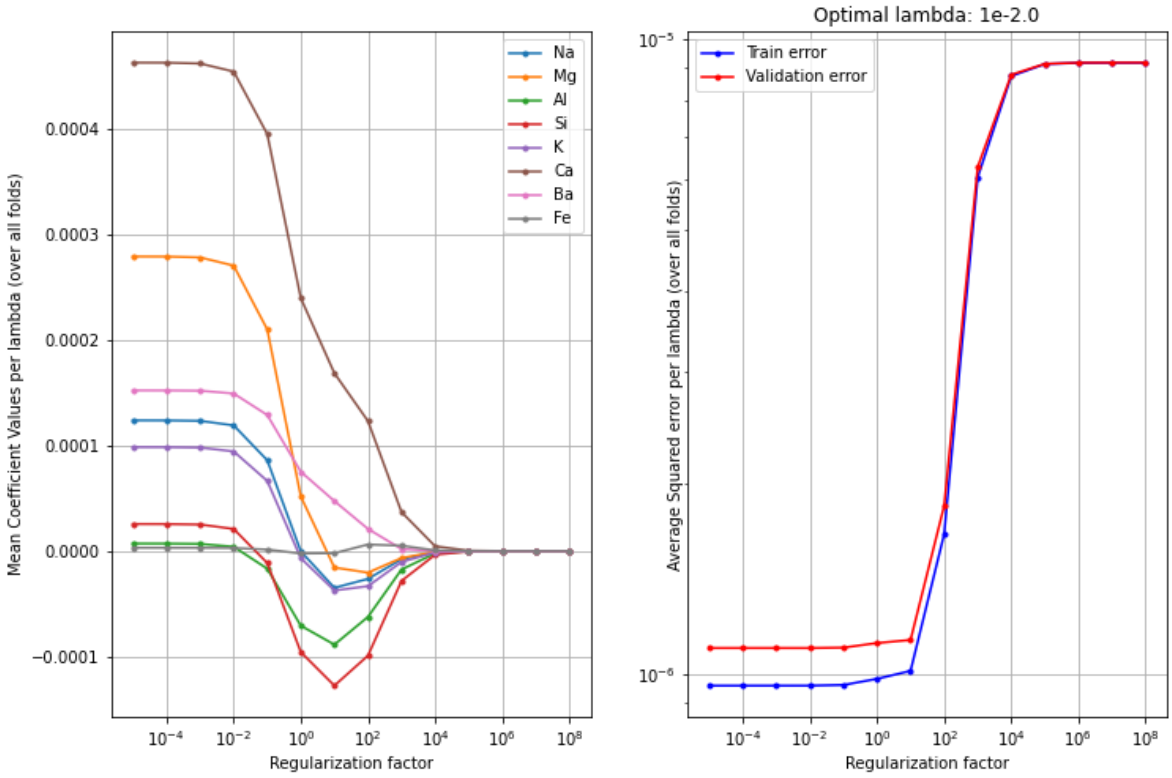

Figure 1: Average Coefficient Weights & Errors across for each λ across all folds

- Na:	0.0001190941
- Mg:	0.0002704004
- Al:	4.1431e-06
- Si:	2.10245e-05
- K:	9.4441e-05
- Ca:	0.0004542197
- Ba:	0.0001492844
- Fe:	2.9285e-06

Table 3: Average weights across all folds for optimal lambda

We reused the figure 3.2 from the 8th exercise and plotted it below for the average values of each lambda across the 10 folds.

From the figure 3.2 on the right side we can see that optimal selected lambda across all 10 cross validation folds is 0.01 and that the mean squared error first decreases slightly and rises sharply after a λ of 10 as noted above.

Regarding the weights, figure 3.2 on the left side, shows the average weights across folds for every lambda while the table 3 below shows the weights for the optimal λ of 0.01 and reveals that the Fe, Al, and Si values were the least relevant in the regression model as the mean coefficient value is the smallest for these three attributes, while the Ca and Mg attributes were accounted in the most as they have the highest mean coefficient values.

3 Regression, part b:

3.1 Implementation and range selection for h and λ

The implementation was done with help of the code snippets from the exercise eight. After a few test runs *(3)*, we decided to use a range of *(n hidden units - 1, 2, 3)* for the number of hidden units, as our algorithm never chose a hidden value number higher than 2 in one of the 30 trial folds and the range reduces the computation time of the following test runs. For the lambda, we chose the same interval as in the task above, as it includes a drop and subsequent increase of the mean squared error value *(lambdas - 0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000, 100000, 1e+06, 1e+07, 1e+08)*.

3.2 Results

In the table below are all of our results from the 2-level-10-fold-cross validation - we used the same training and test splits for all of the three methods (ANN + Regression, Regularized Linear Regression, Baseline (mean estimation)). The errors are measured in the squared loss per observation. Regarding the optimal hidden unit for the artificial neural network, the optimal hidden unit seems to be one across all folds - signaling that a higher hidden unit number would make our model unnecessary complex which does not help in describing/classifying the refractive index based on the chemical components.

Furthermore, for the regularized linear regression, we can see that the inner cross validation selected either a 0.01 or 0.10 as the optimal lambda indicating similar results compared to our part A where the selected optimal lambda was 0.01

Outer fold	ANN and regression		Regularized regression		baseline
i	h_i^*	E_i^{test}	λ_i^*	E_i^{test}	E_i^{test}
1	1	6.3459515e-07	0.01	6.361593e-07	0.000007
2	1	3.790201e-07	0.01	3.888721e-07	0.000008
3	1	7.5366916e-06	0.10	7.111565e-07	0.000006
4	1	1.4829478e-06	0.10	1.626189e-06	0.000016
5	1	6.105169e-07	0.10	3.683012e-07	0.000009
6	1	2.3110777e-06	0.01	1.580579e-06	0.000010
7	1	9.5152706e-07	0.10	9.498334e-07	0.000005
8	1	3.3491426e-06	0.10	3.186348e-06	0.000008
9	1	7.537181e-07	0.01	6.726589e-07	0.000018
10	1	8.41231e-07	0.01	8.851142e-07	0.000005

Table 4: 2-level-10-fold cross validation results - first run

Below in the figures 2 (on the right) and 3 the mean-squared errors of each outer fold is summarized for the two different models. We see that in aggregation the regularized linear regression performs the best, while the artificial neural network is a close competitor but performs slightly worse in some folds, especially in the third fold. Comparing both measures to the baseline in figure 4 we surely see an increase in estimation performance of our two models.

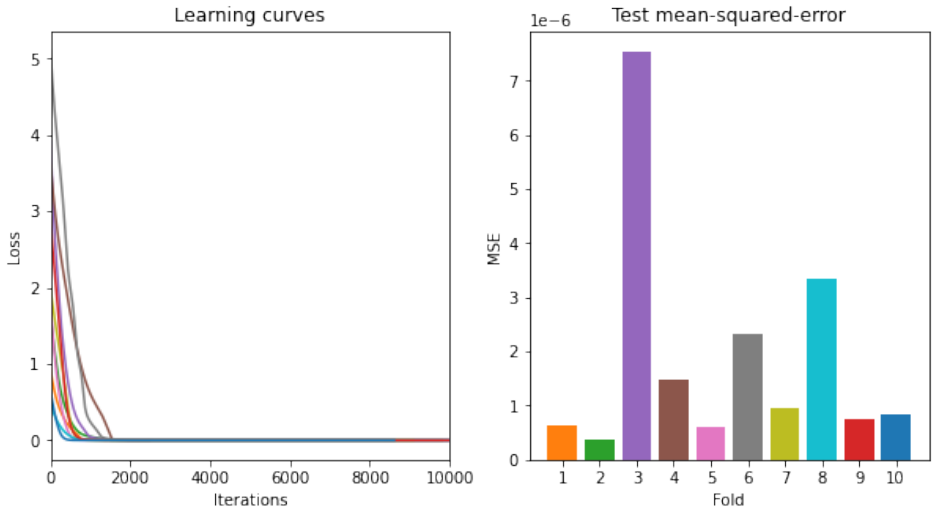

Figure 2: Test mean squared errors across outer folds for the ANN

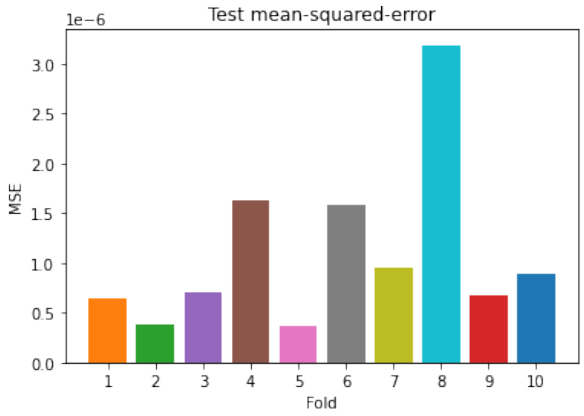

Figure 3: Test mean squared errors across outer folds for the Regularized LR with optimal λ

In the Figures 5 and 6, below the ANN with it's weights for the last fold is visualized and the estimated versus true refractive index value for the model was plotted.

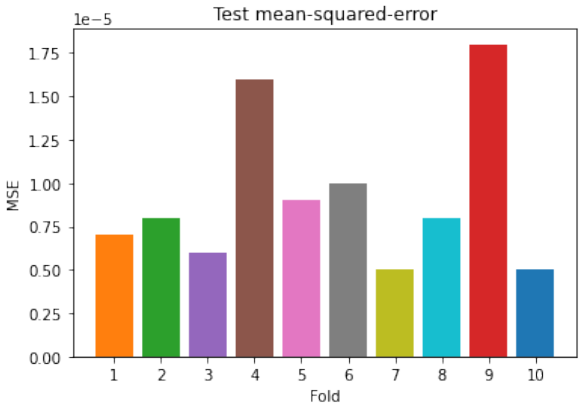

Figure 4: Test mean squared errors across outer folds for the Baseline Model

3.3 Statistical Analysis

For the statistical analysis we decided to use the paired t-test described in Box 11.3.4 in the course book. We ran the 2-level-10-fold-cross-validation two more times to achieve 30 outer-fold results as an $n \ \ 30$ is suggested for this test to assume normal distribution. The results of the additional runs are summarized in the tables below and seem to align with the results of the first run.

Outer fold	ANN and regression		Regularized regression		baseline
i	h_i^*	E_i^{test}	λ_i^*	E_i^{test}	E_i^{test}
1	1	6.768159e-07	0.001	6.429785e-07	0.000015
2	2	3.580548e-07	0.01	3.601027e-07	0.000007
3	1	1.5808495e-06	0.00001	1.732779e-06	0.000008
4	1	1.95401e-05	0.01	1.091745e-06	0.000003
5	1	5.906533e-07	0.10	5.784822e-07	0.000008
6	1	2.9838116e-06	0.01	2.909440e-06	0.000011
7	1	7.42882e-06	0.01	3.878185e-07	0.000007
8	1	8.650178e-07	0.01	6.367627e-07	0.000006
9	1	2.3076032e-06	0.10	1.330284e-06	0.000006
10	1	2.2116428e-06	0.10	1.424644e-06	0.000021

Table 5: 2-level-10-fold cross validation results - second run

Outer fold	ANN		Regularized regression		baseline
i	h_i^*	E_i^{test}	λ_i^*	E_i^{test}	E_i^{test}
1	1	3.6868303e-07	0.10	3.592076e-07	0.000009
2	1	6.375734e-07	0.01	6.338893e-07	0.000007
3	1	2.8041453e-05	0.10	1.761863e-06	0.000006
4	1	7.6013644e-06	0.10	1.710154e-06	0.000008
5	2	2.0878015e-06	0.01	2.254415e-06	0.000021
6	1	1.979448e-06	0.01	1.385631e-06	0.000011
7	1	3.47984e-07	0.10	3.520981e-07	0.000009
8	1	1.054462e-05	0.01	6.038887e-07	0.000016
9	1	2.822221e-06	0.10	1.221784e-06	0.000003
10	1	4.2839417e-07	0.01	3.714087e-07	0.000003

Table 6: 2-level-10-fold cross validation results - third run

As suggested in the task we compared the models pairwise (ANN vs. baseline; linear regression vs. baseline; ANN vs. linear regression;). To statistically test the performance of the models we calculated the according distributions according to the $z = E_A^{gen} - E_B^{gen}$ formula in the course book and our hypotheses were also chosen according to the course book as below:

H0: Model A and B have the same performance, Z = 0
H1: Model A and B have different performance, Z != 0

3.3.1 A: ANN vs. B: baseline

$\mu : -5.491554e-06$
$\sigma : 2.433226e-12$
Confidence Interval: $[-5.491559e-06, -5.491549e-06]$
$p = 0.001444$, $p \leq 0.05$

The negative mean and the p-value ≤ 0.05, indicates a significant better performance of the ANN compared to the baseline.

3.3.2 A: Regularized linear regression vs. B: baseline

$\mu : -8.141513e-06$
$\sigma : 7.772589e-13$
Confidence Interval: $[-8.141515e-06, -8.141512e-06]$
$p = 3.891725$e-10, $p \leq 0.05$

The negative mean and the p-value ≤ 0.05, indicates a significant better performance of the regularized linear regression compared to the baseline.

3.3.3 A: ANN vs. B: Regularized linear regression

$\mu : 2.649959e-06$
$\sigma : 1.212905e-12$
Confidence Interval: $[2.649957e-06, 2.649962e-06]$
$p = 0.022719$, $p \leq 0.05$

The positive mean and the p-value ≤ 0.05, indicates a significant better performance of the regularized linear regression compared to the ANN.

To summarize, the pairwise statistical test's among the different models clearly show a significant better performance of the regularized linear regression compared to the baseline and the ANN. While the ANN performs significantly better than the baseline.

4 Classification

The classification method can be taken as a natural way to analyze the glass dataset. As it is stated in the description of the dataset, the classification analysis was used to determine if the glass is float processed or not. The implementation was done using code from lesson 7 and lesson 8. The methods compared were *K-nearest neighbor*, *regularized logistic regression*, and *baseline non-regularized logistic regression*.

4.1 Data preparation

For this step, we had different assumptions which create some problems. Firstly, we decided to divide our data set based on window and non-window glass. The window glass class contained building and vehicle windows subdivided into **float processed and not float-processed**. The second class contained containers, tableware, and headlamps, which it takes as non-window glass. However, this division of data creates problems for our methods since the difference was so obvious the models

performed perfectly in a few cases. Thus, we decided to divide our dataset into float-processed and not-float-processed thus deciding to solve a **binary classification** problem. Because of this, we had to remove the non-windows glass rows, so our dataset now has 164 rows. The last 'target' column added is based on this information.

4.2 Results

For the logistic regression part we chosen **lambda** in range [1.e-05, 1.e-04, 1.e-03, 1.e-02, 1.e-01, 1.e+00, 1.e+01, 1.e+02, 1.e+03, 1.e+04]. And, for the KNN method, the range of **K-nearest neighbors** is [1,2,3,4,5,6,7,8,9,10].

Outer fold	K nearest neighbors		Regularized regression		baseline
i	K_i^*	E_i^{test}	λ_i^*	E_i^{test}	E_i^{test}
1	3	0.058824	0.00010	0.470588	0.470588
2	2	0.176471	0.00100	0.294118	0.294118
3	3	0.235294	0.01000	0.352941	0.411765
4	3	0.312500	0.00010	0.562500	0.562500
5	5	0.312500	0.00010	0.750000	0.750000
6	3	0.062500	0.00010	0.437500	0.437500
7	3	0.250000	0.00001	0.500000	0.500000
8	3	0.125000	0.00010	0.375000	0.375000
9	3	0.125000	0.00001	0.375000	0.375000
10	3	0.187500	0.00010	0.500000	0.500000

It can be seen that the KNN method usually performs much better the two other methods, and the nearest neighbor optimized number is in most cases 3. A thing to notice in our result is that for some fold the errors seam 'round' and when rerunning it multiple times the methods performed in a similar way; the reason for this is that we have a relatively small dataset(164 rows), and the

4.3 Statistical Analysis

4.3.1 General error comparison

The generalized error comparison can be seen below. Interestingly, the difference between baseline and non-regularized logistic regression is almost non-existent, as can be seen below it is under .6%. And, when rerunning it the difference between the two was usually in the range of 0.5-3 %. However, the K-nearest neighbors method always outperformed both baseline and logistic regression.

Generalized error: Baseline non-regularized logistic regression **46.18** %
Generalized error: Regularized logistic regression **46.77** %
Generalized error: K-nearest neighbors **18.46** %

4.3.2 Regularized logistic regression vs Baseline non-regularized logistic regression

$\mu : -0.035294$
$\sigma : 0.000861$
Confidence Interval: $[-0.037242, -0.033345]$
$p = 0.259786$, $p > 0.05$

As can be seen from the data summary statistics, regularized logistic regression performs slightly better than the baseline.

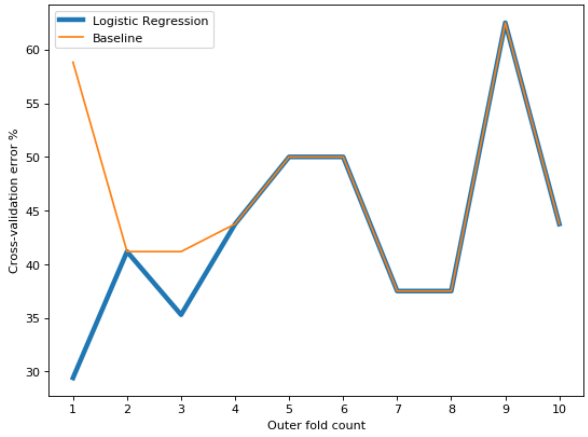

Figure 5: Regularized logistic regression vs baseline non-regularized logistic regression

Figure 7. visually explains how the methods perform. In this iteration from the 4th outer loop onwards, the regularized and non-regularized log regression give similar performance.

4.3.3 K-nearest neighbors vs Baseline non-regularized logistic regression

$\mu : -0.287867$
$\sigma : 0.001401$
Confidence Interval: $[-0.291038, -0.284696]$
$p = 3.035416\text{e-}05$, $p \leq 0.05$

Obviously, the KNN method performs much better than the baseline.

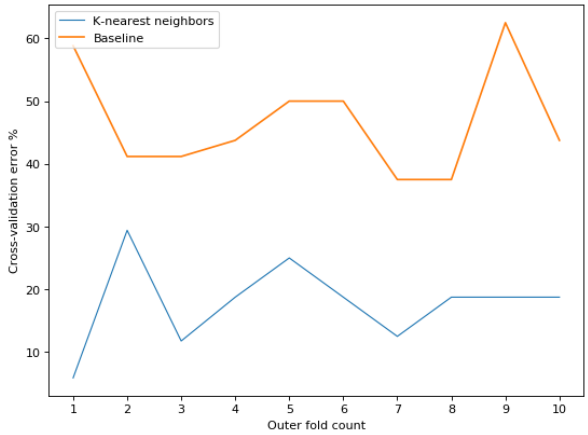

Figure 6: K-nearest neighbors vs baseline non-regularized logistic regression

4.3.4 K-nearest neighbors vs Regularized logistic regression

$\mu : -0.252573$
$\sigma : 0.000676$
Confidence Interval: $[-0.254103, -0.251044]$
$p = 4.555760\text{e-}06$, $p \leq 0.05$

Again, obviously, the KNN method performs much better than the regularized logistic regression since it has a similar performance to the baseline.

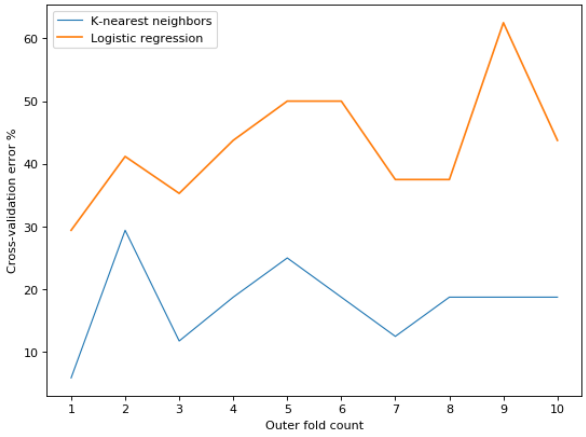

Figure 7: K-nearest neighbors vs baseline non-regularized logistic regression

4.3.5 Logistic regression: Features deemed relevant

Training our model for logistic regression gave the **optimal lambda** at 0.0001

No.	Type	Weight
x_2	RI: Refractive index	3e-05
x_3	Na: Sodium	0.00029
x_4	Mg: Magnesium	0.00028
x_5	Al: Aluminum	-7e-05
x_6	Si: Silicon	0.00122
x_7	K: Potassium	-3e-05
x_8	Ca: Calcium	4e-05
x_9	Ba: Barium	-1e-05
x_{10}	Fe: Iron	-1e-05

Table 7: Attribute information

5 Exam Problems

5.1 Question 1

Answer: C

We solve it by in 3 steps. For each step we used different threshold. And we applied our results to the formulas: $FPR = \frac{FP}{FP+TN}$ and $TPR = \frac{TP}{TP+FN}$

Step1:
We set threshold = 0.8 and we excluded the prediction B as we calculate FPR = 0 and TPR = 0.5 which does not belong to the given graph.

Step2:
We set a threshold=0.5 and we excluded prediction D as we got FPR=0.75 and TPR=1, which is not represented by the curve in the graph.

Step3:
We set a threshold= 0.65 and for prediction A we found that TPR = 0.5 and FPR = 0.75 which does not belong to the given graph, but prediction C gave TPR = 0.75 and FPR = 0.5 which is resembled

by the curve. Thus, the correct answer is C.

5.2 Question 2

Answer: C

We calculate the I(r) by using the formula: $ClassError(v) = 1 - maxp(v|c)$,
where $\Delta = I(r) - \Sigma_{k=1}^{2} \frac{N(v_k)}{N(r)} I(v_k)$

We found that N=135 the total number of our observations.

So,
$I(v_1) = 1 - \frac{37}{134}$, $I(v_2) = 0$
$N(v_1) = 134, N(v_2) = 1, N(r) = 135 \Leftrightarrow I(r) = 1 - \frac{37}{135}$
$\Leftrightarrow \Delta = I(r) - \frac{N(v_1)}{N(r)} I(v_1) - \frac{N(v_2)}{N(r)} I(v_2)$
$\Leftrightarrow$ .... $\Leftrightarrow$ We find a result 0.0074982 and thus we conclude that the correct answer is C

5.3 Question 3

Answer: C
In our case we have only weights.
Given of 7 units and 1 hidden layer with 10 units we have 10 weights for every input, so 70 weights.
Additionally we are given 4 outputs, so we get 40 weights.
In total 70+40=110 and thus answer C is the correct.

5.4 Question 4

Answer: D
On the A split we take the levels (1,2) and (3,1,4), which at fig 4 , for b_1 is at -0.76. Furthermore, on the B split we take only 1,2 which in fig 4, for b_2 is at0.03

5.5 Question 5

Answer: C
By making use of the formula $K_1(k_2L+1)$, where $K_1 = 5$ is the number of outer folds, $K_2 = 4$ for inner folds and $L = 5$ the total number of the parameters we observe that we have to train 105 models.Then, we know that each model has to be tested for both of the two types of models we have, thus we have $105 \times 20 + 105 \times 5 + 105 \times 8 + 105 \times 1 = 3570$

5.6 Question 6

Answer: B
For all given points and for class y=4 we calculate the per-class probabilities. We observe that the probability for B was more than 0.5 and for all the rest was less than 0.00001 and thus B is the correct answer.