Project 2 - Supervised Learning: Classification and Regression

02450: Introduction to Machine Learning and Data Mining

David Miles-Skov (s204755)
Đurđija Milinković (s204724)
Leonard Theisler (s204688)

April 11, 2024

Contents

1 Contributions . . . **1**

2 Exam questions . . . **1**

2.1 Spring 2019 q13 . . . 1

2.2 Spring 2019 q15 . . . 1

2.3 Spring 2019 q15 . . . 2

2.4 Spring 2019 q20 . . . 2

2.5 Spring 2019 q22 . . . 2

2.6 Spring 2019 q26 . . . 2

3 Regression . . . **3**

3.1 Part a . . . 3

3.1.1 Model Objectives . . . 3

3.1.2 Feature Transformations . . . 3

3.1.3 Linear Regression . . . 4

3.1.4 Ridge Regression . . . 4

3.1.5 Generalization error . . . 4

3.2 Part b . . . 6

3.2.1 Two-level cross validation . . . 6

3.2.2 Statistical Evaluation of Model Performance - Regression . . . 6

4 Classification . . . **8**

4.1 Classification Problem . . . 8

4.2 Classification Model Comparison . . . 8

4.2.1 Statistical Evaluation of Model Performance - Classification . . . 9

4.3 Logistic regression model: training and prediction . . . 10

5 Discussion . . . **10**

References . . . **12**

6 Appendix . . . **13**

6.1 Derivation of Objective Function for Linear Regression . . . 13

6.2 Analysis of Model Performances using Data Set X_2 . . . 14

6.2.1 Statistical Comparison of Different Models for $X2$ . . . 14

6.3 Optimal Regularisation Parameter for Logistic Regression . . . 16

6.4 Activation Functions . . . 16

1 Contributions

Section	Exam Questions	Regression A	Regression B	Classification	Discussion
David Miles-Skov: s204755	33.33%	20%	40%	40%	33.33%
Durdija Milinkovic: s204724	33.33%	40%	20%	40%	33.33%
Leonard Theisler: s204688	33.33%	40%	40%	20%	33.33%

2 Exam questions

2.1 Spring 2019 q13

In order to find which prediction is correct, we will compute the true positive rate and false positive rate for each prediction using different thresholds. We will then check if the values that are found correspond to points on the ROC curve. The values are computed in the table below, and are shown in green if the point is on the curve, and in red if it is not.

Threshold	0.8	0.6	0.5
TPR, FPR, A	1/4, 1/4	1/2, 3/4	
TPR, FPR, B	1/3, 1/4		
TPR, FPR, C	1/4, 1/4	3/4, 1/2	3/4, 1
TPR, FPR, D	1/4, 1/4	3/4, 1/2	1, 3/4

Prediction C is the the only one that has all of the points on the curve and it is therefore the correct answer.

2.2 Spring 2019 q15

We will use the formaulas from the slides to compute the impurity gain using the classification error as impurity measure. There are $33 + 28 + 30 + 29 + 4 + 2 + 3 + 5 + 1 = 135$ total observations, and the split $x_7 = 2$ splits it into two branches with 1 and 134 observations. In order to compute the impurity gain, we need to compute the impurity before splitting as well as after splitting on each branch. Before the split, the impurity is equal to:

$$I = 1 - max(37/135, 31/135, 33/135, 34/135) = 1 - 37/135 = 98/135$$

After the split, the impurity on the first branch (with 1 observation) is:

$$I = 1 - max(0, 1/1, 0, 0) = 1 - 1 = 0$$

On the second branch, the impurity is:

$$I = 1 - max(37/134, 30/134, 33/134, 34/134) = 1 - 37/134 = 97/134$$

The impurity gain can then be computed as:

$$\Delta I = 98/135 - (134/135 * 97/134 + 1/135 * 0) = 1/135 = 0.0074$$

Answer C is correct.

2.3 Spring 2019 q15

In order to find the total number of parameters, we need to find the number of parameters between the input layer and the hidden layer, as well as then umber of parameters between the hidden layer and the output layer. There are 10 hidden units in the hidden layer, 7 input variables and a bias term, making the total number of parameters between these two layers to be $(7+1)*10 = 80$. Then, the hidden layer is connected to the output layer and there are 4 possible outputs (congestion level = 1, 2, 3, or 4). As previously stated, there are 10 hidden units and one bias term, bringing the total number of parameters betwenn those two layers to be $(10+1)*4 = 44$. It is then just a matter of adding the two numbers together to obtain the total number of parameters. This gives:

$$80 + 44 = 124$$

Answer A is correct.

2.4 Spring 2019 q20

We will start by having a look at rule A. On the decision tree, we can see that when A evaluates to false, it only leads to congestion levels 1 and 2, while it leads to 1, 3, and 4 when it is true. If we have a look at the classification boundary plot, we can see that there is only one value for which this is the case: $b_1 \leq -0.76$ as levels 1 and 2 are to the left of that boundary, and 1, 3, and 4 can be found on the right of that boundary. We therefore know that answers A and C are incorrect. We then have a look at rule B, which seperates between congestion levels 1 (False) and 2 (True). The possibilities for rule B are $b_1 \geq -0.16$ and $b_2 \geq 0.03$. By looking at the classification boundary graph, we can see that the later rule is correct as it seperates between congestion levels 1 and 2 and the other rule does not. As such, **Answer D is correct.**

2.5 Spring 2019 q22

The question tells us that we are running 2 layer cross validation with 4 inner folds, and we can see from the table that there are 5 outer folds. We are also training five artificial neural networks and five logistic regressions. As such, the time it takes to compute the inner layer is four times the sum of the time it takes to test and train five logistic regressions and five artificial neural networks.

$$t_{inner} = 4*(5*25 + 5*9) = 680$$

For the outer layer, the time it takes is the time of the inner layer plus the time it takes to test and train one neural network and one logistic regression. Given that there are five outer folds, the total time is:

$$t_{total} = 5*t_{outer} = 5*(t_inner + 25 + 9) = 3570$$

Answer C is correct.

2.6 Spring 2019 q26

In order to solve this question, we will compute the probability of each answer to belong to class y = 4. This is done by multiplying $\hat{y}_k$ with each weight, and then using the result to compute the softmax output. Doing this for each observation, we obtain:

- Answer A: $P(y=4|\hat{y}) = 3.02 * 10^{-6}$
- Answer B: $P(y=4|\hat{y}) = 0.73$
- Answer C: $P(y=4|\hat{y}) = 1.76 * 10^{-6}$
- Answer D: $P(y=4|\hat{y}) = 4.65 * 10^{-6}$

Answer B is correct as it the only one with a high probability of being equal to 4.

3 Regression

3.1 Part a

3.1.1 Model Objectives

The aims of our regression model is to accurately predict a patients age, given the values of various health markers. In order to investigate the effect of adding more features to our models, we will consider two different datasets:

- X_1: Consists of only continuous attributes.
- X_2: Will make use of all continuous attributes, along with the following categorical and binary attributes:
  1. $ca \in [0,1,2,3]$: Number of major vessels coloured in fluoroscopy.
  2. $num \in [0,1]$: Diagnosis of heart disease.

The number of categorical attributes used in model 2 was heavily limited in order to restrict the addition of new features resulting from *one-hot encoding* and avoid *the curse of dimensionality* [1].

3.1.2 Feature Transformations

Since model 2 will incorporate categorical attributes, one-hot encoding will be performed on ca, resulting in 4 columns containing binary values, where 1 corresponds to the observation belonging to the specific class of $ca \in [0,1,2,3]$.

E.g, an observation x_i with $ca = 2$ will have the following final four columns:

$$x_i = [\cdots, 0, 0, 1, 0]$$

Since we will be incorporating a regularisation parameter, we standardise the columns corresponding to our continuous attributes. Let j correspond to the indices of our continuous attributes, N be the total number of observations and X be our original data set

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

$\hat{X}$ is our new, standardised data set. Note that no further feature transformations are required for our one-hot encoded attributes, as they have the same relative scaling.

3.1.3 Linear Regression

Linear Regression is part of supervised learning that aims to predict a continuous dependent variable y based on a linear transformation of input independent variables x [2]

$$y = \widetilde{\mathbf{x}}^{\top}\mathbf{w} = w_0 + w_1 x_1 + ... + w_M x_M \tag{1}$$

where w_0 is the *offset*, $w_{i\in[1,...,M]}$ are the weights associated with each attribute. When training the linear regression model, we want to find the optimal values for the coefficients w_M. This is done in such a way that the error between the actual and predicted values in the training data is minimized. i.e. as low as possible. It is equivalent to "minimizing the sum-of-squares error function" [2]. A brief derivation is shown in Appendix section 6.1. The minimisation problem is given by

$$\arg\min_{\mathbf{w}} E(\mathbf{w}) = \arg\min_{\mathbf{w}} ||\mathbf{y} - \widetilde{\mathbf{X}}^{\top}\mathbf{w}||^2 \tag{2}$$

Once our linear regression model is done with training and we found the values for the coefficients, we can plug the values of the new dataset into the linear equation above. Now, predictions can be made on a new dataset.

3.1.4 Ridge Regression

Overfitting is a well-known problem in machine learning when we have a model that works well on the trained data, but poorly on a new dataset. I.e, the model "learns" the training data too well, and captures irrelevant patterns leading to high model variance when applied to unseen data. Ridge/L_2 regularization is an attempt to overcome this issue, with the goal being to reduce model variance. It works by applying penalizing terms to the large coefficients, which are controlled by a regularization parameter or constant, λ. We still want to keep the coefficients as small as possible, while minimizing the sum-of-squares error.

The new objective function is written concerning the fact that we don't want the constant term to be affected by the regularization. That can be seen below:

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

The last term, $\lambda||\mathbf{w}||^2$ is known as the regularization term, where λ, as already mentioned, is the regularization parameter that determines how much influence the regularization term has on the overall model. When $\lambda = 0$, it means that it has no influence on the model (similar to the ordinary least-squares cost function (2)) [2].

3.1.5 Generalization error

With this in mind, we introduced a regularization parameter and computed the generalization error for various values of that parameter. The range of regularization strengths is powers of 10 from -5 to 0, to which we have also added numbers between 1 and 100 and powers of 10 from 2 to 4. This was based on a trial run that showed us that lowest validation error was within that range, and more specifically, between 1 and 100. The resulting plots are presented below for dataset $X1$, and illustrate a small drop before the increase in generalization error. The plot for $X2$ can be found in the appendix 2.

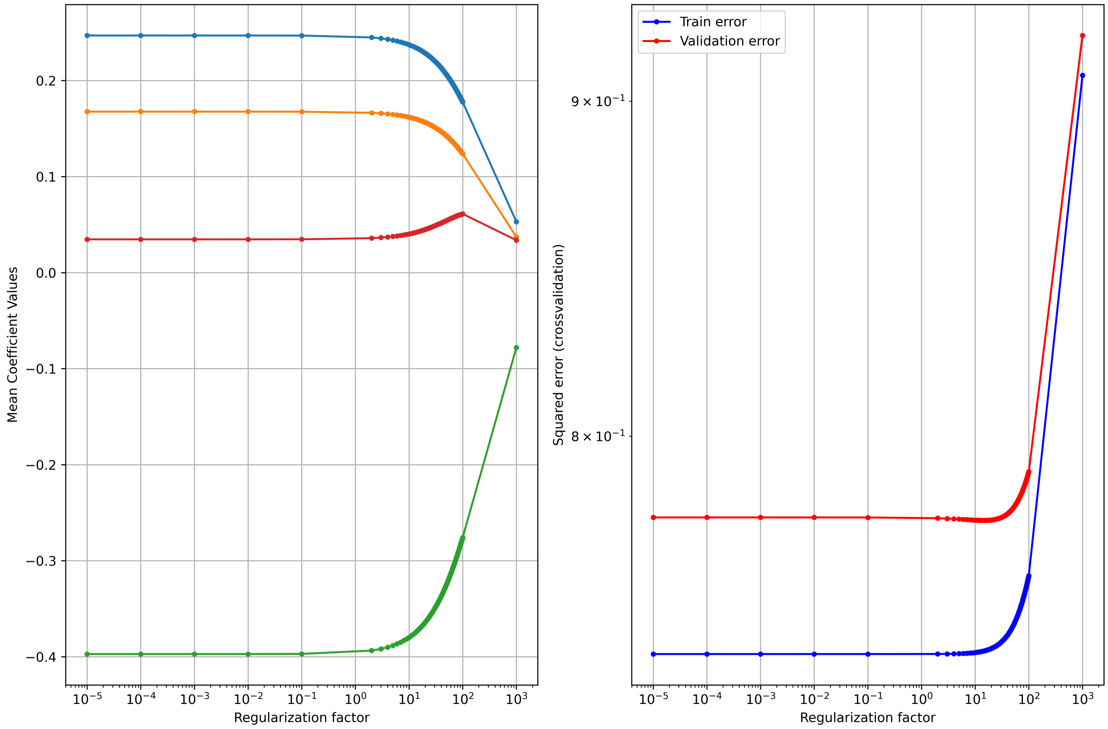


Figure 1: Linear regression with X_1

For both datasets, the optimal lambda tends to be between 15 and 20. X_2 has a slightly lower generalization error than X_1, but they are of a very similar magnitude. As such, we decided to stick with X_1 (no categorical variables) for the rest of this analysis, which is also consistent with the first part of this project.

From the models, we also computed the average weights, after a 10-fold cross validation, to see how they individually influence the output.

	offset	trestbps	chol	thalach	oldpeak	num	ca=0	ca=1	ca=2	ca=3
X_1	0.01	0.24	0.13	-0.33	0.03	/	/	/	/	/
X_2	-0.03	0.26	0.14	-0.33	0.04	-0.1	-0.17	-0.08	0.13	0.01

In our case, the data is first standardized and centralized before the linear regression computed the output. The output is computed by taking a vector of all observations and adding an offset parameter at the beginning, and then multiplying it with the weight vector. The two attributes with the most significant weights are trestbps (resting blood pressure) and thalach (maximum heart rate), with the former having a positive weight, and the latter having a negative one. This seems to be consistent with the problem at hand, as a high blood pressure is more common in older people, and the maximum heart rate tends to decrease with age.

3.2 Part b

3.2.1 Two-level cross validation

As it can be seen in Table 1, values for three different models are presented using the two-level cross-validation. For each of the $K_1 = 10$ folds i, the optimal value of the number of hidden units h_i^* and regularization parameter λ_i^* are shown after being found after each inner loop. The range of hidden units is 1 to 10, and the range of regularization strengths are the same range as in the previous question. Moreover, the estimated generalization errors E_i^{test} are shown after being evaluated on D_i^{test}. Similarly, baseline test errors E_i^{test} are present, after being evaluated on D_i^{test} as well.

Outer fold (i)	ANN		Linear Regression		Baseline
	h_i^*	E_i^{test}	λ^*	E_i^{test}	E_i^{test}
1	2	0.730	18	0.449	0.645
2	1	1.217	18	0.564	1.053
3	6	1.033	15	0.824	0.827
4	1	1.243	14	0.976	1.364
5	1	1.350	11	0.798	1.127
6	2	0.743	21	0.517	0.736
7	1	1.390	16	1.086	1.174
8	1	1.120	18	0.830	0.960
9	2	1.536	20	0.896	1.406
10	1	1.032	19	0.635	0.789

Table 1: Mean Squared Errors and Optimal Hyperparameters

Looking at the values of the mean squared errors, there seems to not be any pattern in any of the models. For the linear regression, the optimal λ is (in most cases) between 10 and 20, which is to be expected based on part a. Just looking at the error values, it appears to be the case that linear regression is the model that performs the best, while the ANN and baseline models have somewhat comparable performances.

The optimal number of hidden units for the Neural Network is most frequently 1. When this is the case, the Neural Network is simply performing linear regression, with the network weights and bias corresponding to the coefficients and offset in linear regression. It is interesting to note that this occurs, despite the presence of the nonlinear activation function tanh (16) in the input layer.

3.2.2 Statistical Evaluation of Model Performance - Regression

Now, we are interested in seeing if there's a significant performance difference between the fitted ANN, linear regression model and baseline. In each row we represented in this order: ridge regression, ANN, and finally baseline model. Thus, the comparisons are pairwise: ANN vs. linear regression; ANN vs. baseline; and linear regression vs. baseline.

The comparisons were done in accordance with **section 11.3.6** in [2], using the `ttest_twomodels` as part of the `dtuimldmtools` package. Performance was compared using the predicted and test values in the final outer fold. The null hypothesis is that there is no difference in performance between the compared models.

- The p-values are as follows:

	ridge regression	ANN	baseline
ridge regression	/	0.027	0.258
ANN	0.027	/	0.753
baseline	0.258	0.753	/

Table 2: p-values

- The 95% confidence intervals for the three pairwise tests are as follows:

	ridge regression	ANN	baseline
ridge regression	/	(-0.347, -0.023)	(-0.425, 0.118)
ANN	(0.023, 0.347)	/	(-0.172, 0.235)
baseline	(-0.118, 0.425)	(-0.235, 0.172)	/

Table 3: Confidence Intervals

- The mean differences in squared loss $\hat{\theta}$ which compare the mean generalization errors of different pairs of models are shown in the table below:

	ridge regression	ANN	baseline
ridge regression	/	-0.185	-0.153
ANN	0.185	/	0.032
baseline	0.153	-0.032	/

Table 4: $\hat{\theta}$ values

Overall model performances can be summarised as follows (at a 95% confidence level):

$$\text{Ridge Regression} > \text{ANN}$$
$$\text{Ridge Regression} \approx \text{Baseline}$$
$$\text{ANN} \approx \text{Baseline}$$

When comparing the models, we notice that p-value for the comparison between ridge regression and the ANN is less than the significance level $\alpha = 0.05$. That implies that there is a difference in performance between ridge regression and the ANN. Moreover, the CI does not include zero, further supporting our conclusion of significant differences between these models. Thus, we reject the null hypothesis and confirm that ridge regression and the ANN have differing performances. More specifically, we conclude that ridge regression outperforms the ANN, as the CI is entirely below 0 which indicates that the ANN makes predictions with a greater loss than the ridge regression model,

On the other side, the p-value for ANN vs. baseline and ridge regression vs baseline is greater than α, indicating no significant difference in performance between ANN and baseline and ridge regression . Also, the CI includes zero, concluding the same. Thus, we accept the null hypothesis and can say that the baseline model does not have a statistically significant performance difference than the two other models.

Recommendations An obvious recommendation would be to obtain more data. The poor performance of the ANN (relative to the baseline and ridge regression) can be partially explained by the relatively small size of the training data (≈ 270 observations when using 10-fold cross-validation). Given a much larger amount of training data, one would expect the neural network to perform better than ridge regression, as more complex/nonlinear relationships within the dataset would be found. However, due the lack of data, we see the ANN frequently tend towards a linear regression model which is inferior to ridge regression (as the number of optimal hidden units is frequently 1).

The use of more attributes (E.g, performing the same statistical analysis with the dataset $X2$) appears to increase the difference in performance between the ANN and ridge regression, but a difference between the models and the baseline is still difficult to show (at the same confidence level of 95%).

4 Classification

4.1 Classification Problem

In this section, we will attempt to build a model that is able to classify patients as healthy or unhealthy based on a number of health metrics. In the original data, the attribute that captures a patient's diagnosis ranges from 0 to 4, with 0 being a perfectly healthy patient with no heart disease. As discussed in the first part of this project, it is more insightful to work with binarized data, where all patients with a diagnosis, no matter how severe, will have 1, and healthy patients will have 0. As such, this is a binary classification problem.

4.2 Classification Model Comparison

Three models for classification: an artificial neural network (with a final *sigmoid* activation function (15)), a ridge logistic regression and a baseline model, that simply finds the most common value in the data and blindly predicts all data points to be equal to that. The range of regularisation parameters to optimise for ridge regression was determined via some preliminary analysis (see Figure **??**) and it can be seen that there is a lot of variation. We suspect that this is due to the relatively small size of the dataset. We proceed with a range of regularisation parameters $\lambda \in \{10^{-5}, 10^{-4}, \cdots, 10^{4}, 10^{5}\}$.

Outer fold (i)	ANN		Logistic Regression		Baseline
	h_i^*	E_i^{test}	λ^*	E_i^{test}	E_i^{test}
1	8	0.333	10^{-5}	0.333	0.333
2	6	0.267	10^{-2}	0.267	0.467
3	9	0.267	10^{-3}	0.267	0.500
4	1	0.233	10^{+1}	0.233	0.567
5	3	0.300	10^{-5}	0.300	0.367
6	9	0.200	10^{+1}	0.200	0.433
7	7	0.133	10^{-3}	0.133	0.500
8	3	0.413	10^{-1}	0.414	0.414
9	5	0.276	10^{-2}	0.276	0.552
10	6	0.380	10^{0}	0.276	0.483

Table 5: Error rates as well as optimal parameters for the three models

4.2.1 Statistical Evaluation of Model Performance - Classification

For our statistical evaluation, we chose to use McNemar's test described in *Box 11.3.2* [2]. Practically, this was implemented using the `mcnemar` function from the `dtuimldmtools` library on the final outer fold in the two-level cross validation for 5. Note the highlighted p-values which are greater than one. This is caused during McNemar's test, where the two classifiers make the exact same predictions according to the test data. The main conclusion is that all classifiers have very similar performances, which is also supported by the confidence intervals (all include 0).

We suspect that this issue is a result of the small final test set (≈ 30 observations) reducing the statistical power of McNemar's test.

	ridge regression	ANN	baseline
ridge regression	/	1	**1.226**
ANN	1	/	1
baseline	**1.226**	1	/

Table 6: p-values

	ridge regression	ANN	baseline
ridge regression	/	(-0.03, 0.096)	(-0.222, 0.222)
ANN	(-0.096, 0.03)	/	(-0.245, 0.18)
baseline	(-0.222, 0.222)	(-0.18, 0.245)	/

Table 7: 95% Confidence Intervals

	ridge regression	ANN	baseline
ridge regression	0	0.033	0
ANN	-0.033	0	-0.033
baseline	0	0.033	0

Table 8: $\hat{\theta}$ values

4.3 Logistic regression model: training and prediction

Logistic regression works in a very similar way to linear regression. However, instead of assuming that the true data is normally distributed about a line, logistic regression assumes a that the target values are Bernoulli distributed. Let $\hat{y}_i = \sigma(\widetilde{\mathbf{x}}_i^\top \mathbf{w})$ where $\sigma(\cdot)$ is the *logistic sigmoid* (15). The probability density of a given observation $y_i = 1$ or 0

$$p(y_i|\mathbf{x}_i, \mathbf{w}) = \text{Bernoulli}(y_i|\hat{y}_i) \tag{3}$$

$$= \hat{y}_i^{y_i}(1 - \hat{y}_i)^{1-y_i} \tag{4}$$

We then proceed with the *maximum likelihood framework*, obtaining the following minimisation problem

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} E(\mathbf{w}) \tag{5}$$

where:

$$E(\mathbf{w}) = -\frac{1}{N} \log \left[\prod_{i=1}^{N} p(y_i|\mathbf{x}_i, \mathbf{w}) \right] = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \right], \quad \hat{y}_i = \sigma \left[\bar{\mathbf{x}}_i^T \mathbf{w} \right] \tag{6}$$

Here the prediction $\hat{y}_i$ is the probability that an observation belongs to the positive class. If $\hat{y}_i > \frac{1}{2}$ the observation predicted as belonging to the positive class, and if $\hat{y}_i \leq \frac{1}{2}$ it is predicted as belonging to the negative class.

age	trestbps	chol	thalach	oldpeak
0.035	0.210	0.156	-0.775	0.779

Table 9: Logistic regression weights

The weights here are different from in the first section. Thalach is still a very important feature for the model and still has a negative weight. Trestbps no longer has a large weight, but oldpeak does, despite having almost zero weight in the regression.

5 Discussion

From both sections of this report, we have gained insight into the data and especially into the performance of different models on this data. In the regression section, we found that the generalization error drops in the range $\lambda \in [0, 100]$, and that $\lambda^* \in [10, 20]$. We also trained three models on this data (ANN, linear regression, and baseline), and found that the linear regression has a comparable performance with the baseline model, but that the ANN performs worse than the linear regression. Moreover, it was found that the ANN frequently tends towards a linear regression model despite its nonlinear activation function in the input layer. Overall our results are quite disappointing regarding model performances, with the size of our dataset leading to some unique issues (see McNemar's test).

This dataset has previously been analyzed using both regression and classification. A number of the findings are discussed in [3], highlighting the performance of different models performing regression and classification. One group of researchers applied regression and classification onto the data using a combination of random forest, decision trees, and hybrid methods and obtained an accuracy of 88.7% when predicting heart disease. Another group of researchers proposed a model using a logistic regression classifier with an accuracy of 93.4% while a third group of researchers applied classification using a K-nearest neighbours approach and achieved 90.8% accuracy. These models perform significantly better than the ones that we developed, and could perhaps be used in a clinical setting. However, it is important to note that all of these models that performed better than ours made use of different techniques, and often a combination of multiple techniques in order to achieve such performance. The classification model using K nearest neighbours used both categorical and continuous attributes, while we only used continuous ones for our classification section.

References

[1] Udacity, "Curse of dimensionality - georgia tech - machine learning," Website, February 2015, retrieved 2024-04-04. [Online]. Available: https://www.udacity.com/course/machine-learning--ud262

[2] M. N. S. Tue Herlau and M. Mørup, *Introduction to Machine Learning and Data Mining*, 2023.

[3] R. R. N. Y. N. B. S. K. D. Khandaker Mohammad Mohi Uddin, "Machine learning-based approach to the diagnosis of cardiovascular vascular disease using a combined dataset," 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666521223000145?ref=cra_js_challenge&fr=RR-1

6 Appendix

6.1 Derivation of Objective Function for Linear Regression

Using the *maximum likelihood framework*, we want to maximise the probability

$$p(y|x, w) \tag{7}$$

Assuming that true observations y are normally distributed about some "true" regression line

$$\mathbf{y} = f(\mathbf{x}, \mathbf{w}) + \varepsilon = \widetilde{\mathbf{x}}^\top \mathbf{w} + \varepsilon \tag{8}$$

As a result of ε being a random, normally distributed variable $\varepsilon = \mathcal{N}(0, \sigma^2)$. Under this assumption, it follows that the probability distribution (7) is also normally distributed, but about the line $\widetilde{\mathbf{x}}^\top \mathbf{w}$

$$p(\mathbf{y}|\mathbf{x}, \mathbf{w}) = \mathcal{N}(y - \widetilde{\mathbf{x}}\mathbf{w}, \sigma^2) \tag{9}$$

Applying the *maximum likelihood framework* in order to find the $\mathbf{w}^*$ which maximises (9) for all observations y_i, given our data $\mathbf{X}$ and weights $\mathbf{w}$

$$\begin{aligned}
\mathbf{w}^* &= \arg\max_{\mathbf{w}} \left[\log p(\mathbf{w}) + \sum_{i=1}^{N} \log p(y_i|\mathbf{x}_i, \mathbf{w}) \right] && (10) \\
&= \arg\max_{\mathbf{w}} \left[\log p(\mathbf{w}) - \frac{N}{2} \log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - \widetilde{\mathbf{x}}_i^\top \mathbf{w})^2 \right] && (11) \\
& && (12)
\end{aligned}$$

$\log p(\mathbf{w}) = 0$, $-\frac{N}{2} \log\left(2\pi\sigma^2\right)$ is constant.

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \left[-\frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - \widetilde{\mathbf{x}}_i^\top \mathbf{w})^2 \right]$$

This is equivalent to

$$\begin{aligned}
\mathbf{w}^* &= \arg\min_{\mathbf{w}} E(\mathbf{w}) && (13) \\
&= \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} (y_i - \widetilde{\mathbf{x}}_i^\top \mathbf{w})^2 && (14)
\end{aligned}$$

6.2 Analysis of Model Performances using Data Set X_2

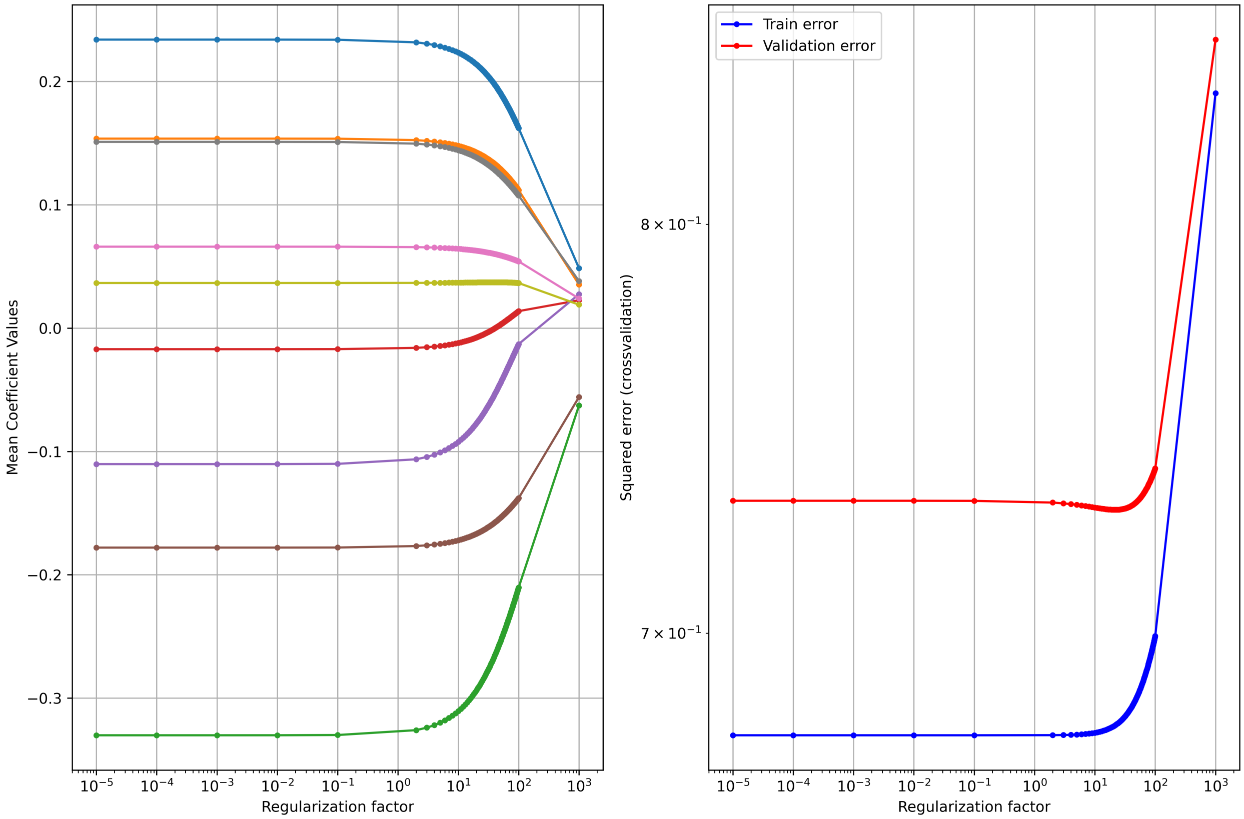


Figure 2: Linear regression with X_2

6.2.1 Statistical Comparison of Different Models for $X2$

- The p-values are as follows:

	ridge regression	ANN	baseline
ridge regression	/	0.00098923	0.07503603
ANN	0.00098923	/	0.16198419
baseline	0.07503603	0.16198419	/

Table 10: p-values

- The 95% confidence intervals for the three pairwise tests are as follows:

	ridge regression	ANN	baseline
ridge regression	/	(-0.643, -0.183)	(-0.452, 0.023)
ANN	(0.183, 0.643)	/	(-0.085, 0.482)
baseline	(-0.023, 0.452)	(-0.482, 0.085)	

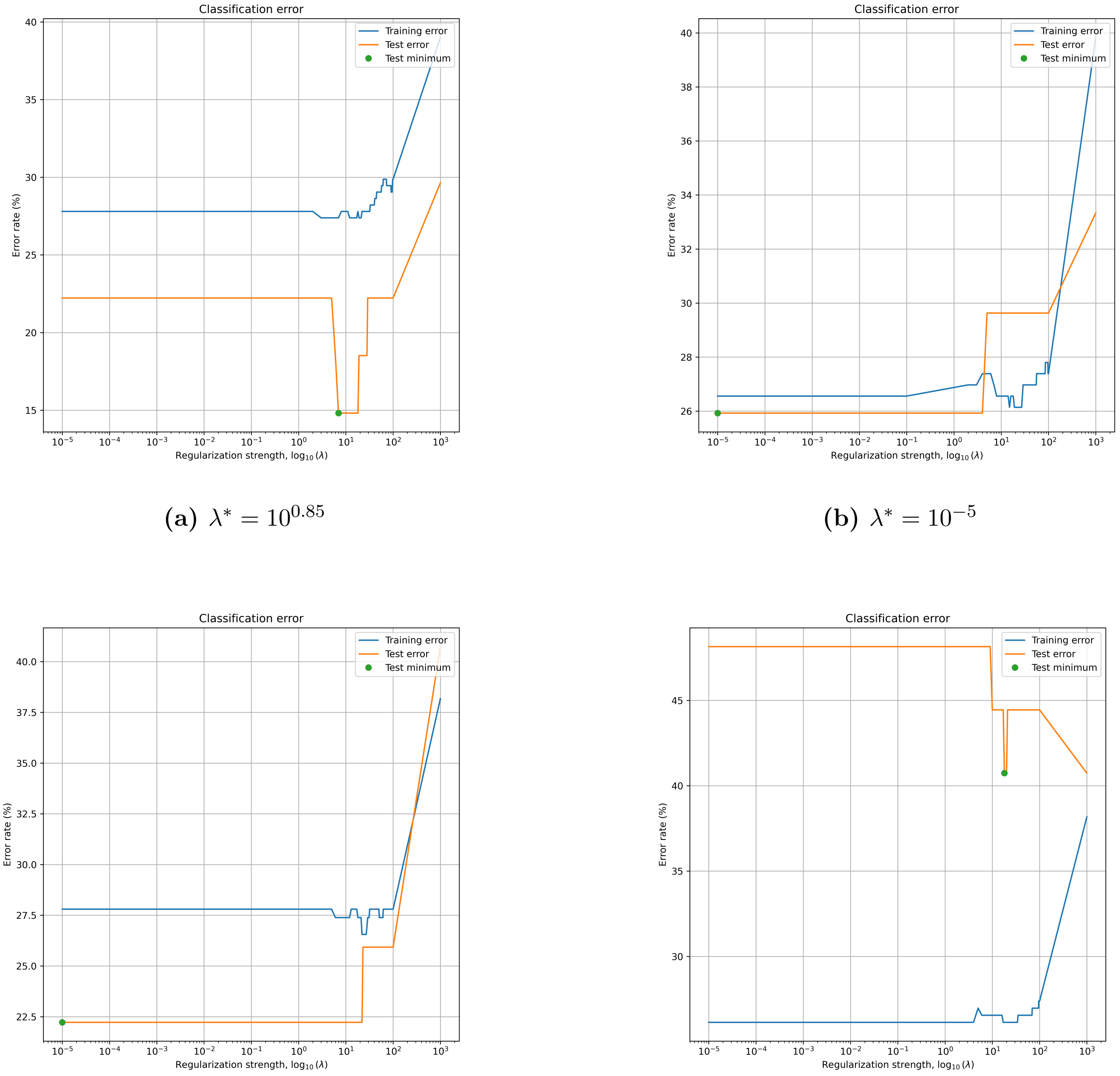
Table 11: Confidence Intervals

- The mean differences in squared loss $\hat{\theta}$ which compare the mean generalization errors of different pairs of models are shown in the table below:

	ridge regression	ANN	baseline
ridge regression	/	-0.185	-0.153
ANN	0.185	/	0.032
baseline	0.153	-0.032	/

Table 12: $\hat{\theta}$ values

6.3 Optimal Regularisation Parameter for Logistic Regression



(a) $\lambda^* = 10^{0.85}$

(b) $\lambda^* = 10^{-5}$

(c) $\lambda^* = 10^{-5}$

(d) $\lambda^* = 10^{1.26}$

6.4 Activation Functions

Sigmoid:

$$S(x) = \frac{1}{1 + e^{-x}} \tag{15}$$

Tanh:

$$\tanh\left(x\right) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{16}$$