02450: INTRODUCTION TO MACHINE LEARNING AND DATA MINING

Project 2 – Group 14

	Section 1	Section 2	Section 3	Section 4	Exam Questions
Hugh Fulton s236798	20%	20%	60%	33.33%	33.33%
Dylan Myers s237017	20%	60%	20%	33.33%	33.33%
Simon Ritter s237138	60%	20%	20%	33.33%	33.33%

Technical University of Denmark

11.04.2024

Table of Contents

List of Figures

List of Tables

1 REGRESSION PART A

1.1 Introduction

Based on the previous report 1, where the dataset with its attributes has been examined, the goal is now to find machine learning models to predict the concrete compressive strength in this and the following section. There are two main models to be examined. A regularized linear regression model and an artificial neural network. The performance of both models are evaluated and compared to one another.

1.2 Aim

The goal of the regression task is to find a linear regression model that predicts the concrete compressive strength [MPa] based on the eight input attributes. With eight attributes and an off-set weight w_0, the goal is to find nine weights $w_0 - w_8$, which the standardized attributes of a new unseen sample of concrete are multiplied with, to find a prediction of its strength.
There is no need of conducting any feature transformation apart from standardization, as they are all continuous and ratio, as already explained in report 1. It has already been shown that there are no outliers to be excluded from the dataset. 1-of-K coding isn't necessary either, as this is a technique especially used to put nominal attributes (such as a category name) into a binary scheme of 0s and 1s. The standardization is conducted by subtracting the mean of every column and then diving by the standard deviation. The success of standardization is checked by calculating the mean and the standard deviation of the transformed columns, which is equal to 0 and 1 respectively.

1.3 Regularization and Generalization Error

In order to find the right balance between bias and variance of the linear regression model, a regularization parameter λ is introduced to the cost function. This regularization parameter penalizes large weights w_i (excluding the off-set w_0) in order to find the best balance between bias and variance of the model. A suitable range for the 56 λ -values is selected in the range of 10^{-5} and 10^{9}:

```
lambdas = np.power(10.0, np.linspace(-5, 9, num=56))
```

For each of those λ-values, the generalization error is estimated using K=10 fold cross-validation. The results are presented in Figure 1.

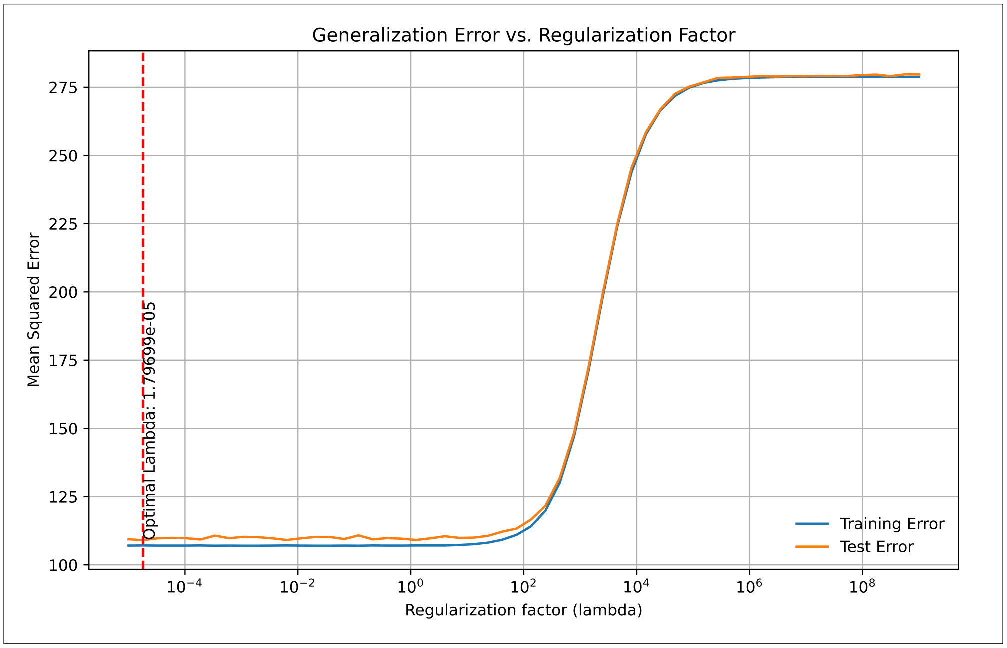


Figure 1: Generalization error as a function of the regularization factor

It is directly apparent that there is no local minimum in the plot of the test error, as it stays approximately constant for values of λ between 10^{-5} and 10^{1}. With the optimal value of $\lambda=1.80*10^{-5} \approx 0$, the effect of regularization for a regression model for the concrete data set is negligible. This means that the errors of a regularized linear regression model are similar to a linear regression model with feature selection. Repeating this algorithm a few times leads to different results for the optimal λ, as the splitting of the entire data set into a training and test set is subject to random selection when using the *.split()* function of sklearn on python. The results, however, only change marginally.

1.4 Output and effects

For the optimal λ, the weights $w_0 - w_8$ are found that minimize the cost function of the model. They are presented in Figure 2. The off-set w_0 has a value 35.82 and corresponds to the expected value (the mean) of the compressive strength of all the data. The attribute cement is weighted the most, a high cement content leads to a high concrete compressive strength. Similar, but not to such a high extent, is the effect of blast furnace slag and fly ash, the substances that are used to replace cement. The age of the concrete also is weighted considerably with 7.21. These results seem plausible, as the named components are known for their beneficial effect on the strength (especially cement) and as the concrete develops its strength over time, that is with a higher age it consolidates more and becomes stronger. The only attribute with a negative weight is water, which demonstrates the inverse proportionality between the water content and the compressive strength, which is widely known and was a priori expected.

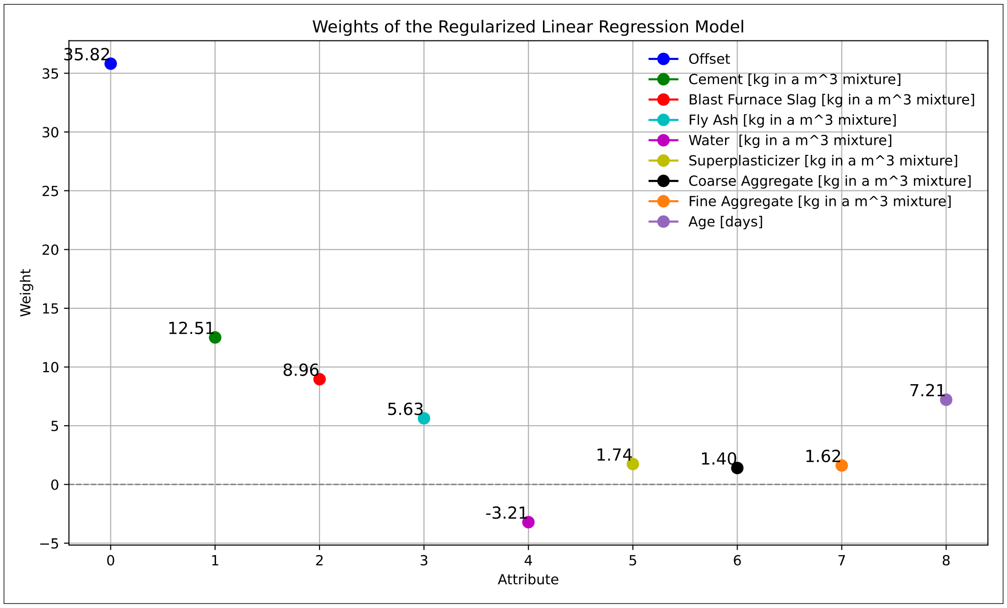


Figure 2: Weights of the regularized linear regression model

The regression model can now be used to predict a value of the concrete compressive strength using the following procedure.

1. Given the input $\mathbf{x} = (x_1, x_2, \ldots, x_8)^T$, i.e. the values of all the 8 attributes of an unseen concrete sample, standardization is conducted:

$$\tilde{\boldsymbol{x}} = \frac{(\boldsymbol{x} - \boldsymbol{\mu})}{\boldsymbol{\sigma}} \; with \; \boldsymbol{\mu} = (\mu_1, \; \mu_2, \ldots, \mu_8)^T \; and \; \boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_8)^T$$

2. The compressive strength $\hat{y}$ can be estimated by:

$$\hat{y} = w_0 + \tilde{\boldsymbol{x}}^{\boldsymbol{T}} \boldsymbol{w} \; with \; \boldsymbol{w} = (w_1, w_2, \ldots, w_8)$$

The weights, means and standard deviations are listed in the following table:

i	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
$\mathbf{w_i}$	35.82	12.51	8.96	5.63	-3.21	1.74	1.40	1.62	7.21
$\boldsymbol{\mu_i}$	-	281.17	73.90	54.19	181.57	6.20	972.92	773.58	45.66
$\boldsymbol{\sigma_i}$	-	104.47	86.24	63.97	21.35	5.97	77.71	80.14	63.14

An example is given for the following unseen sample with arbitrary values of its features:

i	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
feature	cement	slag	fly ash	water	superpl.	coarse a.	fine a.	age
$\mathbf{x_i}$	400	100	70	180	5	1000	700	100

Using the described procedure, the compressive strength of this sample is estimated to be $\hat{\boldsymbol{y}}$ = **58.83 MPa.**

2 REGRESSION PART B

2.1 Two-Level Cross-Validation

We began section 2 by fitting an ANN model to the data, using a modified version of the code from Exercise 8.2.6, which fit an ANN model to data with a continuous output. To investigate the complexity controlling parameter, hidden units (h), used K-fold crossvalidation with K = 10, as well as a replicate value of 3 in each fold and max iterations of 10000, for computational simplicity. The investigation revealed that as you increase the complexity controlling parameter, h, to greater than 3, the MSE loss started to increase. This is likely due to our model being too complex for our data and overfitting, which captures noise. Thus, the range of values that we have chosen for the two-level cross-validation analysis is [1,2,3]. As an additional note, it was observed that as the number of iterations increased, the MSE loss decreased, which is what we expected from an ANN model.

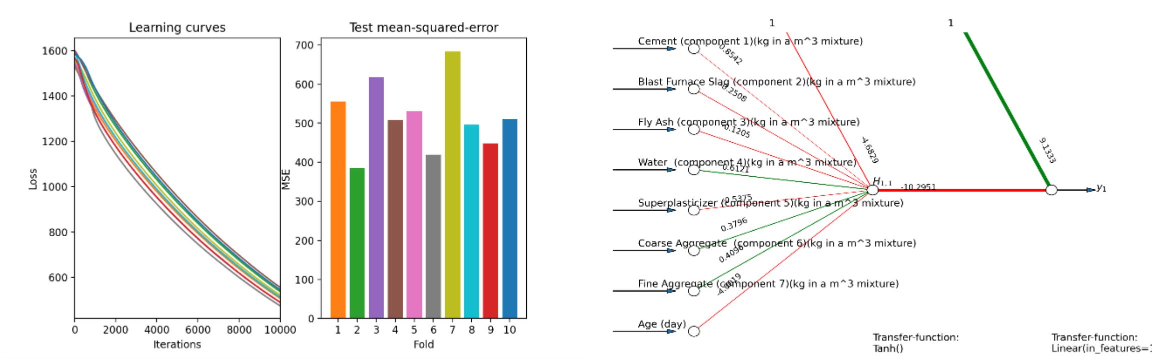


Figure: showing visualization for a k=10 ANN with 1 hidden unit

After the initial investigation we continued with a 2-level cross validation with K1 = K2 = 10 folds. Using this cross validation, the model tested the predetermined hidden unit range inside of each inner fold and found the optimal hidden unit size. The model then trained the ANN with the optimal hidden unit size on the outer fold and computed the test error (per number of observations). The table in section 2.2 contains both values for each K1 = 10 folds. The model was built with code taken from exercises, ChatGPT and a nested loop. The two-level cross validation performed better than the one- level cross validation (for the same K values). Furthermore, same splits are used to train a regression model (from the previous section) and selects a optimal lambda for each K1 = 10 folds. The splits are created using the sklearn.model_selectionKfold function in python.

2.2 Table and Comparisons

Table 1 presents a comparison of the three models, the artificial neural network, the linear regression model as well as the baseline model. The values of the test errors were found with a two-level cross-validation with $K_1 = K_2 = 10$, while the same splits into training set and test set were used for all of them. The optimal number of hidden units as well as the optimal lambda value are indicated per each fold in the same table. As already seen before, the values of lambda are near zero (same as for the 1-level cross-validation) and therefore the linear regression model is only to a small extent subject to regularization. The model did not seem to prefer one hidden unit size over the other, as each of the h values in our range [1,2,3] appeared virtually the same number of times. This may perhaps indicate that if we were to continue our analysis, we should increase our range, to try a larger variety of hidden unit types for thoroughness. The mean values of the test errors are the lowest for the ANN, followed by the linear regression model and the baseline. Based on the selected performance indicator, we would choose ANN to model our dataset over a linear regression or baseline model.

Table 1: Comparison of the three models

Outer fold	**ANN**		**Linear Regression**		**Baseline**
i	h_i^*	E_i^{test}	λ_i^*	E_i^{test}	E_i^{test}
1	3	15.30	1.00E-05	96.89	225.29
2	3	14.11	1.00E-05	109.42	280.95
3	1	15.18	1.00E-05	95.80	266.25
4	2	14.66	1.00E-05	98.96	261.11
5	3	15.67	1.00E-05	140.42	300.57
6	2	12.79	1.00E-05	97.48	318.28
7	1	16.39	1.00E-05	105.03	225.38
8	3	14.76	1.00E-05	128.47	304.92
9	2	14.19	1.00E-05	105.48	266.14
10	1	15.15	1.00E-05	116.57	305.43
Mean		14.82		109.45	275.43

2.3 Statistical Evaluation

The results of Table 1 are statistically evaluated using a paired t-test (setup I). The evalutation happens pairwise, so ANN vs. linear regression, ANN vs. baseline and linear regression vs. baseline.

In order to evaluate if there is a significant difference in performance of the two models, the null hypothesis H_0 and the significance level α are defined:

$$H_0: There\ is\ no\ difference\ in\ performance\ of\ the\ two\ compared\ models$$

$$\alpha = 0.05$$

The test statistics are calculated as $z_i = E_{modelA}^{test} - E_{modelB}^{test}$ of the two models A and B that are compared. For this test statistics, a confidence as well as a p-value is found. The results are presented in Table 2.

Table 2: Statistical Evaluation of Model Performance

	ANN vs. Linear Regression	ANN vs. Baseline	Linear Regression vs. Baseline
Confidence Interval	(-107.44, -84.72)	(-289.20, -243.27)	(-190.66, -149.66)
p-value	4.96e-08	4.13e-09	5.76e-08

For all of the three pairwise tests, the p-values are smaller than $\alpha = 5\%$ and therefore the null hypothesis has to be rejected. The conclusion is that the ANN performs better than the linear regression model, and both, the ANN as well as the linear regression model perform better than the baseline.

3 Classification

3.1 Introduction

In this section, we will look at a binary classification problem to distinguish between high and low strength concrete samples within our dataset. We will test for concrete samples which are below 30MPa compressive strength as low strength, and above 30MPa compressive strength as high strength. The goal is to uncover patterns within the dataset that may aid in enhancing concrete production quality and its application.

We start by establishing the baseline using a simple model, progressing to more complex techniques; logistic regression and k-Nearest Neighbors (KNN). This stepwise approach will allow us to assess the performance and effectiveness of the algorithms in accurately performing binary classification on our dataset. Our aim is to find the most efficient model to facilitate informed decisions in the manufacturing of concrete.

3.2 Method Selection

In our analysis, we will select a multiclass classification problem, aiming to categorize concrete samples into distinct quality levels. We evaluate:

- **Baseline Model**: Establishes a benchmark by predicting the training data's most frequent class.
- **Linnear Regression**: Uses regularization (λ) to adjust model complexity, ideal for capturing linear relationships between variables.

For our third comparative method, we chose from these options:

- **Artificial Neural Networks (ANN)**: For complex nonlinear relationships.
- **Classification Trees (CT)**: For straightforward interpretation.

- **k-Nearest Neighbors (KNN)**: Chosen for its simple, assumption-free approach, classifying samples based on the majority class among nearest neighbors.
- **Naïve Bayes (NB)**: Applies when feature independence is a valid assumption.

The selection of **KNN** is due to its direct method and adaptability, particularly suitable for our multiclass dataset, where concrete samples are classified into various quality levels. This method's reliance on the nearest neighbors' majority class, without needing data distribution assumptions, complements our objective to understand and predict concrete quality accurately, crucial for optimizing concrete production processes.

3.3 Introduction and Model Comparison

To assess the performance of our logistic regression and k-Nearest Neighbors (KNN) models in the context of classifying concrete samples, we must benchmark them against a baseline model that predicts the majority class for all instances.

The baseline serves as a minimal standard, enabling us to quantify the improvements offered by more sophisticated modeling techniques.

Through logistic regression, we implement regularization to optimize model complexity and mitigate overfitting.

The KNN model's effectiveness is evaluated by adjusting the count of nearest neighbors (k). The analysis encompasses parameter optimization via cross-validation, followed by an evaluation of model performance. The objective is to determine the superior method for our classification requirements, providing a basis for advanced predictive modeling.

3.4 Two-Level Cross-Validation for logistic regression

In our classification cross-validation study, K-Nearest Neighbors (KNN), Multinomial Regression, and a Baseline model were evaluated on concrete strength data. KNN, tuning for neighbor count, had the lowest error rate, indicating superior pattern detection. Multinomial Regression, adjusting for regularization, was slightly less accurate. The Baseline model, predictably, underperformed, highlighting the benefit of more advanced models for such prediction tasks. Overall, while KNN led, the final model choice should balance interpretability, computation, and task needs.

Table 3: Comparison of the three models

Outer fold	**KNN**		**Logistic Regression**		**Baseline**
i	k_i^*	E_i^{test}	λ_i^*	E_i^{test}	E_i^{test}
1	3	0.174757	0.359381	0.174757	0.349515
2	1	0.184466	0.0001	0.184466	0.417476
3	1	0.106796	0.0001	0.145631	0.281553

4	3	0.116505	0.0001	0.15534	0.330097
5	1	0.15534	0.0001	0.184466	0.427184
6	3	0.135922	0.359381	0.126214	0.38835
7	5	0.116505	0.046416	0.174757	0.349515
8	1	0.097087	0.0001	0.135922	0.427184
9	1	0.097087	0.359381	0.145631	0.320388
10	1	0.145631	0.046416	0.15534	0.38835
Mean		0.13301		0.158252	0.367961

In our binary classification analysis for concrete strength, K-Nearest Neighbors (KNN) and Logistic Regression both substantially outperformed a simplistic Baseline model, demonstrating their effectiveness in discerning high and low strength concrete samples. KNN, with a mean error rate of 0.13301, slightly surpassed Logistic Regression, which exhibited a mean error rate of 0.158252. The Baseline model's higher mean error rate of 0.367961 underscores the sophistication and predictive accuracy of the KNN and Logistic Regression models. This analysis confirms KNN's slight advantage in accuracy for this dataset.

3.5 Statistical Evaluation of 3 models

Statistical analysis is performed similarly to in section 2. With a the same null hypothesis and level of significance. Mcnemar’s test was used to compare the performance of each pair of model classifiers. The test was conduceted by creating a contingency table which calculated when one model was correct, and the other was not. We then applied Mcnemar’s test to the table to obtain the p-values for our statistical anlaysis. If the p-value is low, then there is there is a statistically significant difference in the two model, if the p-value is high there may be no significant difference.

Table 4: Statistical Evaluation of Model Performance using Mcnemar’s Test

	KNN vs. Logistic Regression	**KNN vs. Baseline**	**Logistic Regression vs. Baseline**
Confidence Interval	(-0.1864 , 0.4303)	(0.0448 , 0.4605)	(0.00193 , 0.4299)
p-value	0.1379	1.097e-06	6.401e-05

- **KNN vs. Logistic Regression**: The p-value is 0.1379, which is above the common significance level of 0.05. This indicates that there is no statistically significant difference in the classification performance between the KNN and Logistic Regression models for this dataset.
- **KNN vs. Baseline**: The p-value is approximately 1.097e-06, which is far below the 0.05 threshold. This suggests that there is a statistically significant difference in performance, with KNN outperforming the Baseline model.
- **Logistic Regression vs. Baseline**: Similarly, the p-value is approximately 6.401e-05, also well below the 0.05 threshold, indicating a statistically significant difference in

performance between Logistic Regression and the Baseline model, with Logistic Regression providing the better performance.

3.6 Model Training

The most suitable lamda value which was selected is 0.359381 because this value was calculated to be associated with the lowest mean error rates in the model output shown in table 3.

Our logistic model predicts the probability of an instance belonging to a specific class by using a logistic function on the weighted sum of input features. If this probability is at least 0.5, the instance is classified into the positive class; otherwise, it's classified into the negative class. The weights signal each feature's effect on the prediction, where their direction and size reveal the feature's impact on being in the positive class.
The output of the logistic regression model gives feature coefficients as shown in table **X** below:

Table 5 - Logistic Regression model output

Feature	Coefficient
Cement (component 1)(kg in a m^3 mixture)	-2.6539476600256737
Blast Furnace Slag (component 2)(kg in a m^3 mixture)	-1.5324315419914016
Fly Ash (component 3)(kg in a m^3 mixture)	-0.8575201562174132
Water (component 4)(kg in a m^3 mixture)	0.4678292047783327
Superplasticizer (component 5)(kg in a m^3 mixture)	-0.4283804063413666
Coarse Aggregate (component 6)(kg in a m^3 mixture)	-0.4556299733608221
Fine Aggregate (component 7)(kg in a m^3 mixture)	-0.32617394892769763
Age Category_fresh	4.2470775571518695
Age Category_mature	-2.268611012345804
Age Category_mid-age	0.17284900087808402

The table shows the logistic regression model's coefficients for each feature, indicating their importance and direction of influence on predicting the target variable. The magnitude of these coefficients represents the strength of the influence. Notably, the fresh age category has the most substantial positive impact, suggesting a significant association with the higher class under the logistic regression model's perspective. We can consider whether the same features are influencial in classifying instances into discrete categories. in comparing the two models is that while the direction and relative magnitude of the coefficients may be consistent (indicating similar feature importance), the interpretation is model-specific. In the regression model, coefficients affect the predicted continuous value of compressive strength directly. In the logistic regression model, they affect the log-odds of being in a particular class (e.g., 'High strength' vs. 'Low strength'). As the features deemed significant in the regression model are also significant in the logistic regression model, it would suggest these features are robust predictors across different types of analyses.

4 Discussion

4.1 Learnings From Regression & Classification

The data set consists of eight input attributes, on which different models can be trained to predict the strength of the concrete. Linear regression has shown that some of the attributes (cement, fly ash, slag, age and water) are more indicative of the strength than the others.

Building and tweaking an ANN model from scratch was an interesting and informative experience. The first major takeaway was that the ANN model performs the best on our dataset, when compared to a regression or baseline model. Furthermore, two-level cross validation with K1=K2 = 10 provided a lower test error than one-level cross validation with K1 = 10; however, due to curiosity I tested a K1=5 one-level cross validation and it yielded a **lower** test error. This could be due to variability in our data set, meaning as we split it up into smaller pieces, they become much different to each other. In the future, it would be interesting to investigate K as a complexity controlling factor, instead of h to further expand on this hypothesis.

The classification section of the report reveals that certain components, such as cement content, significantly influence the prediction of concrete strength categories. The logistic regression model identified similar features as impactful, consistent with domain knowledge and the previous regression analysis. Higher cement content, age, and the presence of blast furnace slag and fly ash positively correlate with stronger concrete classifications. Water content shows an inverse relationship, as expected. In comparing our models the logistic regression and KNN models outperformed the baseline, as indicated by the significant p-values from McNemar's test. Between KNN and logistic regression, there was no statistically significant difference, suggesting comparable performance in this context.

4.2 Comparison to Previous Data Analysis

The original study (see references) led to the following two conclusions *(citing):*

1. “The strength model based on the artificial neural network is more accurate than the model based on regression analysis.”
2. “The compressive strength can be calculated using the models built with this methodology. It is convenient and easy to use these models for numerical experiments to review the effects of each variable on the mix proportions. For example, the strength model can be used to study the strength effects of age or water-to-binder ratio.”

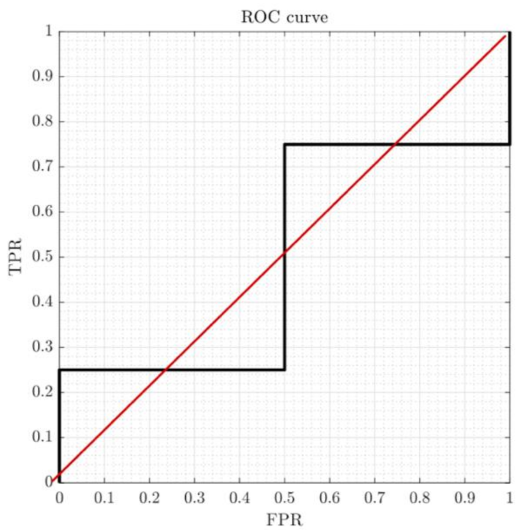
The results found in the first part of this report (linear regression and ANN) led to the same results. Both models can be used to find a prediction of the concrete compressive strength, while the ANN performs better.

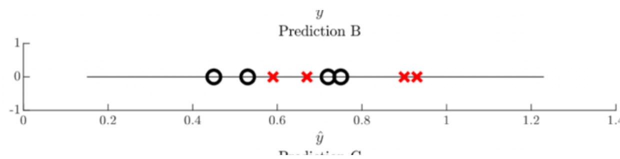
5 Exam Questions

5.1 Question 1: Spring 2019 question 13

- **Answer B is correct.**

Going along the diagonal red line, the data is split into 1,0,1,0. So only B makes sense (4 regions).





5.2 Question 2: Spring 2019 question 15

- **Answer C is correct.**

$$\Delta = I(r) - \frac{N(v_1)}{N} * I(v_1) - \frac{N(v_2)}{N} * I(v_2) = \frac{98}{135} - 0 - \frac{134}{135} * \frac{97}{134} = 0.0074$$

With impurity calculated as ClassError: $I(\quad) = 1 - \max_c p(c|v)$
The root has the split y1:y2:y3:y4 = 37:31:33:34 with N(r) = 135, max(p) = 37/135
The left branch has the split y1:y2:y3:y4 = 0:1:0:0 with N(v1) = 1, max(p) = 1
The right branch has the split y1:y2:y3:y4 = 37:30:33:34 with N(v2) = 134, max(p) = 37/134

5.3 Question 3: Spring 2019 question 18

7*10 input features = 70
10 input bias terms (for each of the 10 hidden units)
4 * 10 output neurons = 40
4 output bias terms (for each of the 4 outputs)
= 124, **therefore Answer A) is correct**

5.4 Question 4: Spring 2019 question 20

By looking at he splits on the classification boundary graph, and comparing these with the decision tree, the only likey answer is option D.

The other optioins are incorrect because:
A: for A:b1>=-0.16 A can be only level 4 which is not it's only possiblilty.
B: for B=b1>=-0.16 the only option for classification would be congestion level 4 which is incorrect.
C: B:b1>=-0.76 means it cannot be congestion level 2 which is incorrect
D: fits the criteria for rule assignments.

Therefore **the correct answer is option D**

5.5 Question 5: Spring 2019 question 22

Time for ANN: = k1 * (k2 * (traintime+ testtime)) *(train+testime)
= 5 * 4 * 25 * 25 = 500 * 25

Time for Reg = k1 * (k2 * (traintime+ testtime)) * (train+testtime)
= 5 * 4 * 9 * 9 = 180 * 9

Total time = 14120 ms

Correct answer D, as it is closest to the computed time

5.6 Question 6: Spring 2019 question 26

Using code to define weight and observation vectors, calculate scores, and then applying the softmax fuinction, probabilities of each observation being part of group 4 were:

Probability of A belonging to class 4: 0.391
Probability of B belonging to class 4: 0.408
Probability of C belonging to class 4: 0.486
Probability of D belonging to class 4: 0.460

Therefore option C is correct

6 References

Dataset: I.-C. Yeh, "Concrete Compressive Strength," UCI Machine Learning Repository, 2007. [Online]. Available: https://doi.org/10.24432/C5PK67. [Accessed: 18-02-2024].

Original Paper: I.-C. Yeh, "Modeling of strength of high-performance concrete using artificial neural networks," Cement and Concrete Research, vol. 28, no. 12, pp. 1797-1808, 1998. [Online]. Available: https://doi.org/10.1016/S0008-8846(98)00165-3. [Accessed: 18-02-2024].