Danmarks Tekniske Universitet

(02450) Introduction to Machine Learning and Data Mining

Project 2

Silvia Rosvang Andersen (s214702)

Signe Djernis Olsen (s206759)

Sofie Kodal Larsen (s214699)

ID \ Section	1	2	3	4	5
s214702	20%	40%	30%	45%	33.33%
s206759	40%	40%	30%	25%	33.33%
s214699	40%	20%	40%	30%	33.33%

Table 1: Responsibility for report

April 18th 2023

Contents

1 Introduction

This project will use weather data from Albury in Australia to make weather predictions using machine learning methods.

In the last project, we explored the data and made PCA. In this project, we want to use the data to predict certain attributes. We will predict with both regression and classification. The goal is to make a regression model that is able to predict how many mm of rain are going to fall on the given day. Furthermore, we want to train a classification model to predict if it is going to rain the next day. Through the report, we will compare different models to find the most accurate way to predict.

The regressions models we compare are baseline, linear regression with regularization, and artificial neural network. The classification models we compare are baseline, logistic regression, and classification trees. We compare the models using statistical performance evaluation.

The theory of this report is heavily inspired by the lecture notes[1].

2 Regression

The aim of making the regression is to be able to predict the amount of rain there will fall (Rain fall) on a certain day based on 8 other attributes from the same day. The attributes we predict based on are: WindSpeed, Humidity, Pressure, Temperature, MinTemperature, MaxTemperature, WindGustSpeed and RainTomorrow.

We transform our data set $\mathbf{X} \in \mathbb{R}^{2120\times 9}$ with a standardization setting the mean μ to 0 and the standard deviation σ to 1 for each attribute. The standardization of our data set $\mathbf{X}$ is carried out by

$$X^*_{j,i} = \frac{X_{j,i} - \mu_i}{\sigma_i} \quad , \quad j \in \{1,..,2120\},\ i \in \{1,..,9\}$$

where $\mathbf{X}^*$ is the standardized data set. We want to find a model, that predicts the rainfall from the other attributes. Therefore, we extract the attribute RainFall from $\mathbf{X}^*$ and define it as the response $\mathbf{y}^* \in \mathbb{R}^{2120\times 1}$, which means that $\mathbf{X}^* \in \mathbb{R}^{2120\times 8}$. We continue working with the transformed data set described in table 2.

No.	Attribute description (unit)	Abbrev
x_1	Minimum temperature (°C)	MinTemp
x_2	Maximum temperature (°C)	MaxTemp
x_3	Speed of wind gust (km/h)	WindGustSpeed
x_4	Speed of wind (km/h)	WindSpeed
x_5	Humidity level (percent)	Humidity
x_6	Pressure level (hPa)	Pressure
x_7	Average temperature (°C)	Temp
x_8	Rain or no rain tomorrow (binary)	RainTomorrow
y	Amount of rainfall (mm)	RainFall

Table 2: Attributes for regression model

[1] Introduction to Machine Learning and Data Mining, Lecture notes, Spring 2022, version 1.0, Tue Herlau, Mikkel N. Schmidt and Morten Mørup

We want to find a linear regression model defined as

$$f(\mathbf{x}) = \mathbf{x}^T\boldsymbol{\omega}$$

where the input $\mathbf{x} = [1, x_1, \cdots, x_8]^T$ contains fixed standardized values of all the attributes, $\boldsymbol{\omega} = [\omega_0, \omega_1, \cdots, \omega_8]$ are the model weights and $f(\mathbf{x})$ is the predicted standardized values of rainfall.

2.1 Regularization

From linear regression the cost function is defined as

$$E(\boldsymbol{\omega}) = ||\mathbf{y}^* - \mathbf{X}^*\boldsymbol{\omega}||^2$$

where $\mathbf{X}^*$ and $\mathbf{y}^*$ is the standardized data and $\boldsymbol{\omega}$ is some weight that relates x_i to y_i. We now introduce the regularization factor λ to penalize large weights by

$$E_\lambda(\boldsymbol{\omega}) = ||\mathbf{y}^* - \mathbf{X}^*\boldsymbol{\omega}||^2 + \lambda||\boldsymbol{\omega}||^2, \ \lambda \geq 0.$$

If we solve the derivative of E_λ with respect to $\boldsymbol{\omega}$ equal to zero, we see that

$$\begin{aligned} \frac{dE_\lambda}{d\boldsymbol{\omega}} &= -(\mathbf{X}^*)^T(\mathbf{y}^* - \mathbf{X}^*\boldsymbol{\omega})^2 + 2\boldsymbol{\omega} = 0 \\ \Rightarrow \boldsymbol{\omega}^* &= ((\mathbf{X}^*)^T\mathbf{X}^* + \lambda\mathbf{I})\backslash((\mathbf{X}^*)^T\mathbf{y}) \end{aligned}$$

where $\boldsymbol{\omega}^* \in \mathbb{R}^{9\times1}$ is the regularized weights. The goal is to find the optimal λ.

The computing of $\boldsymbol{\omega}^*$ has been implemented in Python, and with a 10 fold cross validation where we trained on a $\frac{2120}{10}\times9$ data sets and tested on $\frac{2120}{10}$. Doing this gave us generalization errors for the linear regression without regularization and generalization errors with regularization for the different tested λ. The optimal λ was found as 31.6228. The generalization error for the linear regression goes from 0.8182 to 0.8172, when a regularization factor $\lambda = 31.6228$ is included.

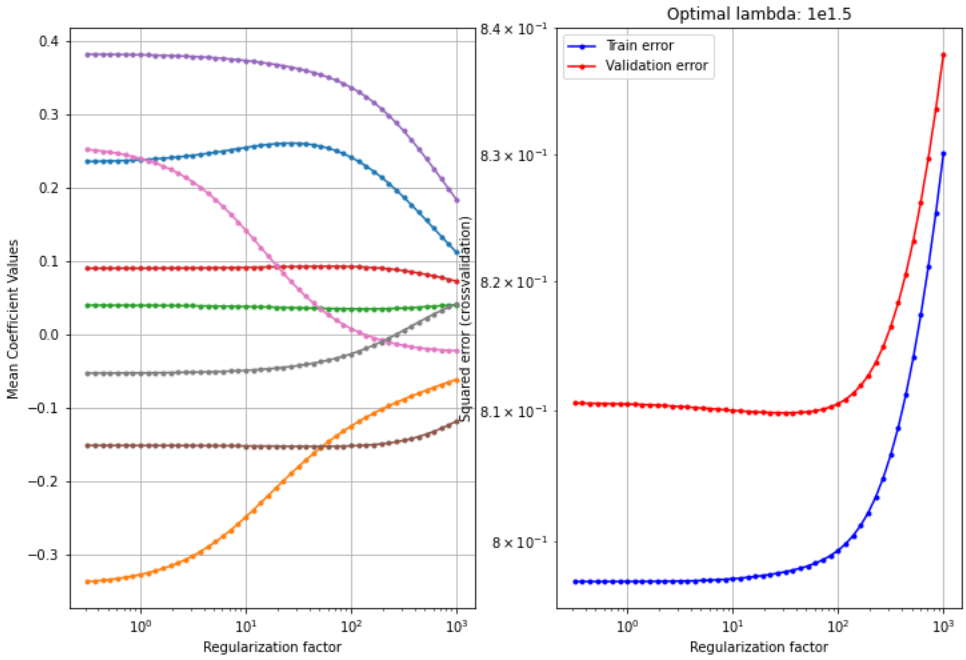

Figure 1: The optimal lambda for ridge regression

Plot 1 shows the train and validation error. The optimal lambda-value will be placed where the validation error is at its minimum. From the plot we can see a small dip on the graph between 10^1 and 10^2, this is where we expect the optimal lambda is placed. The optimal lambda is found to be $10^{1.5} \approx 31.62$.

Using $\lambda = 31.6228$ gives us the regularized linear model

$$f(\mathbf{x}^*) = (\mathbf{x}^*)^T\boldsymbol{\omega}^* = -0.0039 + 0.2552x_1 - 0.2497x_2 + 0.0385x_3 + 0.0918x_4 + 0.3743x_5 - 0.1516x_6 + 0.1441x_7 - 0.0486x_8$$

where $\boldsymbol{\omega}^*$ are the weights and $\mathbf{x}^*$ contains standardized weather information for the prediction day. The output is a standardized prediction of rainfall. We then get the real prediction by transforming the standardized value of rainfall back by

$$y^{est} = f(\mathbf{x}^*) \cdot \sigma_{rainfall} + \mu_{rainfall}.$$

It can be seen Temperature and Humidity have the biggest impact on RainFall, and attributes such as WindSpeed have a lower impact. It makes sense that the wind speed does not effect the amount of rain but humidity and temperature do. As we also saw in project 1, we have a positive correlation between Humidity and RainFall, and therefore the weight for Humidity has a high positive number.

The weights will change every time we run the code because the split of the data would be different.

Using the weights given above, we try to predict the amount of rainfall using randomly selected day with standardized data points x^*.

Attributes	MinTemp	MaxTemp	WindGustSpeed	WindSpeed	Humidity	Pressure	Temp	RainTomorrow	**RainFall**
Data points	0.6507	0.0489	0.8318	1.807	-0.8452	-1.362	0.2369	-0.5096	0.2969

Table 3: Data points used to predict RainFall

When calculating the estimate from our example gives a standardized RainFall to 0.2969. To compare the predicted RainFall with the actual RainFall we need to unstandardize the predicted RainFall.

$$y^{est} = 0.2969 \cdot 6.534 \text{ mm} + 2.014 \text{ mm} = 3.953 \text{ mm}.$$

The true RainFall for the day was 0.6 mm. We see that our prediction isn't close to the true value. Our model could be better, but it should be considered that RainFall's range is from 0mm to 104.2mm with large gaps, so an error of 3.4 is maybe okay.

We will now compare the model to two other regression models to see if we can make better predictions.

2.2 Comparing models using the test error

We want to apply and compare three models: baseline model, linear regression with regularization, and Artificial neural network (ANN).

The baseline model (model 1) doesn't include the any features. The prediction from this model is solely based on the mean of $\mathbf{y}$. The baseline model is given by

$$f_{M_1}(\mathbf{x}, \boldsymbol{\omega}) = \frac{1}{N^{train}} \sum_{i \in D^{train}} y_i.$$

The linear regression model (model 2) with regularization, was found in previous section. The generalized form of this model is given by

$$f_{M_2}(\mathbf{x}, \boldsymbol{\omega}) = \mathbf{x}^T \boldsymbol{\omega}^*.$$

The artificial neural network, ANN, (model 3) combines linear models through hidden layers. The general form of an ANN model for each neuron is given by

$$\begin{aligned} f_{M_3}(\mathbf{x}, \boldsymbol{\omega}) &= g_{link}(\omega_n \cdot g_{link}(\omega_{n-1} \cdot \ldots g_{link}(\omega_2 \cdot g_{link}(\mathbf{x}^T \omega_1 + b_1) + b_2) + \ldots) + b_n) \\ &= g_{link}(\mathbf{x}^T \boldsymbol{\omega} + \mathbf{b}) \end{aligned}$$

where g_{link} represents the activation function used in the respective layers and $\mathbf{b}$ are the biases for each layer.

We compare the three models by the test error. Given a model M_s, the test error is found by

$$E_{M_s}^{test} = \frac{1}{N^{test}} \sum_{i \in D^{test}} \left(y_i^{test} - f_{M_s}(x_i, \boldsymbol{\omega}) \right)^2$$

where f_{M_s} is the model fitted to the training data. We will use 2-level cross-validation with $K_1 = K_2 = 10$ fold to compare the models. Since we have 2120 data points with 8 attributes split into 10 for each fold, the test sets are given by $D^{test} \in \mathbb{R}^{212 \times 8}$, and the number of data points in each test set is $N^{test} = |D^{test}| = 212$.

The test error for the three models are calculated and represented in table 4. We train the ANN model for 5 different numbers of hidden units $h_i \in \{1, 2, 3, 5, 10\}$ and train the linear regression model for regularization factors $\lambda \in \{10^k\}$, where k takes 50 evenly spread values between -0.5 and 3. We then test to find the optimal h_i^* and λ_i^* for the ith fold.

Outer fold	ANN		Linear regression		Baseline
i	h_i^*	E_i^{test}	λ_i^*	E_i^{test}	E_i^{test}
1	5	0.8269	19.31	0.8324	0.8830
2	2	0.8724	37.28	1.509	1.823
3	2	0.7773	26.83	0.7674	0.8947
4	2	0.5011	37.28	0.6530	0.9073
5	1	0.6289	5.179	0.7161	0.9822
6	2	1.052	26.83	1.479	1.857
7	1	0.8985	37.82	0.5672	0.7416
8	2	0.5454	37.28	0.5758	0.6559
9	3	0.2552	19.31	0.3043	0.4584
10	5	0.5965	26.83	0.6772	0.7806
Average/mode	2	0.6954	27.34	0.8081	0.9983

Table 4: Comparing three regression models

We see that the λ is fluctuating around the $\lambda = 31$ as was found in the 10-fold cross-validation.

The optimal hidden units are around 2, so an artificial neural network with one layer and 2 units would probably give the best ANN model.

From table 4 it looks like the ANN is the one with the lowest test error, while the baseline is the one with the highest. But to be able to compare the models a statistically evaluate performance between the regression models has to be made.

2.3 Statistically evaluate performance between regression models

We use setup I as described in the lecture notes [2].

We now want to statistically evaluate if there is a significant difference in performance between the regression models. Therefore we define z_i^M as the squared distance between the predicted values y_i^{est} and the true values of y_i for model M:

$$z_i^M = |y_i - y_i^{est}|^2, \quad M \in \{1,2,3\},\ i \in \{1,\cdots,2120\}.$$

The squared difference between the loss of two models A and B is then defined as

$$z_i = |z_i^A - z_i^B|^2, \quad A,B \in \{1,2,3\},\ i \in \{1,\cdots,2120\},\ A \neq B.$$

We then compute the 95% confidence interval $[z_L, z_U]$ by

$$z_L = \mathrm{cdf}_\tau^{-1}\left(\frac{\alpha}{2}|v = n-1\mu = \hat{z}, \sigma = \hat{\sigma}\right)$$
$$z_U = \mathrm{cdf}_\tau^{-1}\left(1-\frac{\alpha}{2}|v = n-1\mu = \hat{z}, \sigma = \hat{\sigma}\right)$$

where $\alpha = 0.05$ for a 95% confidence interval, $v = n-1 = 2120-1 = 2119$ and $\hat{z}$ and $\hat{\sigma}$ is the mean loss and mean standard deviation given by

$$\hat{z} = \frac{1}{n}\sum_i^n z_i, \quad \hat{\sigma} = \frac{1}{n(n-1)}\sum_{i=1}^n (z_i - \hat{z})^2.$$

Finally, we want to calculate the p-value to test the null hypothesis, that two models have the same performance. If $p < 0.05$, we reject the null hypothesis. The p-value is calculated by

$$p = 2\mathrm{cdf}_\tau^{-1}(-|\hat{z}|\ |v = n-1, \mu = 0, \sigma = \hat{\sigma})$$

When comparing the three models, we only look at the models' predictions for the last fold. We make the three comparisons:

1 : ANN and baseline
2 : ANN and linear regression
3 : Baseline and linear regression

Implementing setup I in Python we yield the following results:

$$\hat{\theta}_1 \in [-0.2032, 0.2560] \qquad p_1 = 0.8209$$
$$\hat{\theta}_2 \in [-0.0376, 0.2973] \qquad p_2 = 0.1279$$
$$\hat{\theta}_3 \in [-0.0374, 0.2442] \qquad p_3 = 0.1492.$$

From the confidence interval and the p-value, it can not be said that none of the models are is significantly different in performance from each other. The interval for $\hat{\theta}$ include zero and

[2] Introduction to Machine Learning and Data Mining, Lecture notes, Spring 2022, version 1.0, Tue Herlau, Mikkel N. Schmidt and Morten Mørup

the p-value is above 0.05 for all three comparisons. The null hypothesis can therefore not be rejected, and there is no evidence that any of the models stand out in performance.

We expected ANN to perform significantly better from the other models, since it is a more complex model, but that doesn't seem to be the case.

We will now move away from regression and into classification.

3 Classification

We want to classify whether it is going to rain tomorrow depending on 8 attributes. We notice there is a class imbalanced between dry days and raining days. The imbalance is a factor of about 1:5, and we therefor choose to ignore it, but it is a problem worth considering for further analysis.

This classification will be a binary problem since there are two possibilities for rain tomorrow: yes or no. Therefore we modify table 2 as shown below:

No.	Attribute description (unit)	Abbrev
x_1	Minimum temperature (°C)	MinTemp
x_2	Maximum temperature (°C)	MaxTemp
x_3	Amount of rainfall (mm)	RainFall
x_4	Speed of wind gust (km/h)	WindGustSpeed
x_5	Speed of wind (km/h)	WindSpeed
x_6	Humidity level (percent)	Humidity
x_7	Pressure level (hPa)	Pressure
x_8	Average temperature (°C)	Temp
y	Rain or no rain tomorrow (binary)	RainTomorrow

Table 5: Attributes for classification model

To solve the classification we have decided to compare classification trees, logistic regression and baseline. The stopping criterion for the decision tree is chosen to be the gini impurity measure and max depth. We have chosen our controlling parameter to be the depth of the tree (c). From a trial run, we have chosen to work with the depths, $c \in \{1, \cdots, 15\}$. When working with a complexity parameter over 15 the training error becomes very small, which might seem good at first. This happens because the model overfits and the test error becomes larger, which isn't good for the model. The three models and their belonging values can be seen in table 6.

For classification trees the mode of the depths is 4, which seems like a good depth where it won't be too complex or too simple.

Logistic regression has an optimal λ spread out from 0.0004 to 75 which is a large interval. But from looking at the table it could be said that the optimal λ should be around $20 - 30$.

At first glance, it looks like logistic regression is the best, classification trees are the second best, and baseline is the worst, when looking at the test errors. But we can't compare the models by only looking at the general test error. We have to evaluate the performance statistically.

Outer fold	Classification trees		Logistic regression		Baseline
i	c_i^*	E_i^{test}	λ_i	E_i^{test}	E_i^{test}
1	4	0.1509	24.421	0.1368	0.2028
2	7	0.1651	3.7276	0.1509	0.2264
3	5	0.1509	24.421	0.1226	0.2217
4	4	0.1509	16.768	0.1368	0.2217
5	2	0.1038	35.565	0.1085	0.1509
6	8	0.1368	0.0133	0.1226	0.2453
7	3	0.1462	75.431	0.1368	0.1934
8	2	0.1840	0.0043	0.1604	0.2358
9	5	0.1085	11.514	0.0991	0.1557
10	4	0.1226	0.0004	0.1038	0.2075
Average/mode	4	0.1420	20.186	0.1278	0.2061

Table 6: Comparing the three classification models

3.1 Statistically evaluate performance between classifiers

When comparing two classifiers, we compare how many y-values have been classified correctly. Given the estimates y_i^{est} and true values y_i with $i \in \{0, \cdots, 2120\}$ for model M we define

$$c_i^M = \begin{cases} 1 & \text{if } y_i^{est} = y_i \\ 0 & \text{if } y_i^{est} \neq y_i. \end{cases}$$

For this we can calculate the confusion matrix. Given the classifiers A and B, the confusion matrix will have the entries

$$n_{11} = \sum_{i=1}^{n} c_i^A c_i^B \qquad \text{Both classifiers are correct}$$

$$n_{12} = \sum_{i=1}^{n} c_i^A (1 - c_i^B) \qquad A \text{ is correct and } B \text{ is wrong}$$

$$n_{21} = \sum_{i=1}^{n} (1 - c_i^A) c_i^B \qquad A \text{ is wrong and } B \text{ is correct}$$

$$n_{22} = \sum_{i=1}^{n} (1 - c_i^A)(1 - c_i^B) \qquad \text{Both classifiers are wrong}$$

and the confusion matrix is then constructed as shown in table 7.

		Classifier B	
		Correctly predicted	Incorrectly predicted
Classifier A	Correctly predicted	n_{11}	n_{12}
	Incorrectly predicted	n_{21}	n_{22}

Table 7: Construction of confusion matrix

Comparing the three classifiers, we construct the confusion matrices shown in table 8, 9 and 10.

		Baseline	
		Correctly predicted	Incorrectly predicted
Logistic regression	Correctly predicted	162	28
	Incorrectly predicted	6	16

Table 8: Confusion matrix with Baseline and Logistic regression

		Classification trees	
		Correctly predicted	Incorrectly predicted
Logistic regression	Correctly predicted	165	25
	Incorrectly predicted	5	17

Table 9: Confusion matrix with Classification trees and Logistic regression

		Baseline	
		Correctly predicted	Incorrectly predicted
Classification trees	Correctly predicted	146	24
	Incorrectly predicted	22	20

Table 10: Confusion matrix with Baseline and Classification trees

From the three constructed confusion matrices we can compare two models to each other. From the three we see that the two models with the highest rate of correctly predicted class are logistic regression and classification trees. It can be seen that logistic regression often predicts correct when the other classification models do not, but it is rarely the other way around.

We will now compare the three classifiers using the McNemar test. The general idea is to calculate the difference in accuracy where the estimated accuracy model M is given by

$$\hat{\theta}_M = \frac{1}{n}\sum_{i=1}^{n} c_i^M.$$

The estimated difference in accuracy of the two classifiers is then given by

$$\hat{\theta} = \hat{\theta}_A - \hat{\theta}_B$$

which can be rewritten to (see appendix 6.1)

$$\hat{\theta} = \frac{n_{12} - n_{21}}{n}.$$

We now calculate the lower and upper bound for the estimated accuracy by

$$\hat{\theta}_L = 2\text{cdf}_B^{-1}(\frac{\alpha}{2}|\alpha = f, \beta = g) - 1, \qquad \hat{\theta}_U = 2\text{cdf}_B^{-1}(1 - \frac{\alpha}{2}|\alpha = f, \beta = g) - 1$$

where

$$f = \frac{\hat{\theta}+1}{2}(Q-1), \quad g = \frac{1-\hat{\theta}}{2}(Q-1), \quad Q = \frac{n^2(n+1)(\hat{\theta}+1)(1-\hat{\theta}))}{n(n_{12}+n_{21}) - (n_{12}-n_{21})^2}.$$

Finally, we want to calculate the p-value to test the null hypothesis, that two classifiers have the same performance. If $p < 0.05$, we reject the null hypothesis. The p-value is calculated by

$$p = 2\text{cdf}_{\text{binom}}(m = \min\{n_{12}, n_{21}\}|\theta = \frac{1}{2}, N = n_{12} + n_{21}).$$

When comparing the three models, we only look at the models' predictions for the last fold. We have three binary vectors $y_i^{est} \in \{0, 1\}\ i\ \in\ \{1, .., 2120\}$. We make the three comparisons:

$$1 : \text{Baseline and logistics regression}$$
$$2 : \text{Classification tress and logistics regression}$$
$$3 : \text{Classification tress and baseline}$$

Implementing the McNemar test in Python we yield the following results:

$$\hat{\theta}_1 \in [0.0517, 0.1556] \qquad p_1 = 0.0002$$
$$\hat{\theta}_2 \in [0.0453, 0.1431] \qquad p_2 = 0.0003$$
$$\hat{\theta}_3 \in [-0.0531, 0.0719] \qquad p_3 = 0.8830.$$

Logistic regression can be concluded to be significantly different from both the baseline classifier and classification trees, because the interval for $\hat{\theta}_1$ and $\hat{\theta}_2$ does not include 0 and the p-value for both is under 0.05. Therefor the null hypothesis can be rejected. If we look at table 6, we can see that the errors E_i^{test} for logistic regression are generally lower, than the general test errors for the other two classifiers. We can therefore conclude with a 95% chance, that logistic regression is significantly better than the other classifiers.

For baseline classifier and classification trees the confidence interval for $\hat{\theta}_3$ does include 0, and has a p-value above 0.05, which means the null hypothesis can not be rejected, and the two classifiers doesn't performance significantly different.

To sum up, logistic regression is significantly better than the baseline classifier and the classification trees. Nothing can be said about the difference in the performance of the baseline classifier and the classification trees.

3.2 Training a logistic regression model

We will now focus on the logistic regression which was the best-performing model for classification. A new model with $\lambda = 25$ is computed, where all the data was trained upon. This yields the following model for the probability for RainFall:

$$f(\mathbf{x}^*) = \sigma(0.5058x_1 - 0.3312x_2 - 0.1090x_3 + 0.6239x_4 - 0.1589x_5 + 1.348x_6 - 0.6500x_7 + 0.003747x_8)$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the sigmoid function. Note that the prediction should be unstandardized with $\mu_{RainTomrrow} = 0.2061$ and $\sigma_{RainTomorrow} = 0.4045$ before using the sigmoid function. The output is the probability of rain the next day, the higher z will be the higher is the probability of rain the next day.

From the model, it is clear to see Humidity (x_7) has the highest relevance for the prediction of rain or no rain the next day. For the regression part, Humidity was also the attribute with the highest relevance for the amount of RainFall the same day.

From project 1, it was also concluded Humidity had the highest coefficient for the first three principal components. When predicting the amount of rain and if it would rain the next day, it makes sense Humidity has the highest impact.

Looking at some of the other attributes, MinTemp, MaxTemp, and Pressure also have high relevance for both classification and regression. The coefficients for them are high in both models. Considering the attribute WindGustSpeed, it is relevant in classification model, but not in the regression model. Both AverageTemperature and WindSpeed does not hold a lot of relevance when one wants to predict the amount of RainFall and predicting if it will rain the day after or not.

We wish to try our model by selecting two random days: one dry day, and one rainy day. We insert the values into our model, to get a prediction.

Attributes	MinTemp	MaxTemp	RainFall	WindGustSpeed	WindSpeed	Humidity	Pressure	Temp	**RainTomorrow**
No rain tomorrow	-0.3437	0.3317	-0.3082	0.8318	0.2893	-1.535	-1.109	0.4446	0.4382
Rain tomorrow	0.7667	-0.2083	-0.3082	-0.8084	-0.2165	0.8225	-0.8138	-0.015	0.7062

Table 11: Data points for two randomly selected days, one with no rain and one with rain.

From the table, we see that the prediction made for the no rain tomorrow data gives an approximate probability of 44% for rain the next day. This is fairly low and means that the model will predict no rain the next day. When looking at the prediction from the rain tomorrow data we see that it has a high probability of rain the next day. The model seems decent, and with these two randomly selected tests, it predicts correctly.

4 Discussion and conclusion

The dataset we used has been analysed before by others, one of them is Maggie Ma[3]. The article compares 3 different classification models. They use logistic regression, decision trees, and random forest to predict whether it is going to rain the next day. From their results, we see the same tendencies as ours. They found that logistic regression was better than decision trees. This is exactly what our statistical evaluation of the different models showed. Furthermore, they found that the best classifier was the random forest classifier.

From the random forest classifier, the paper stated the top 7 most important attributes when determining whether it will rain tomorrow or not. The three most important attributes were Humidity, WindGustSpeed, and Pressure. When looking at our best classifier, the logistic regression, we see that the same 3 attributes hold the most information about the prediction. In our logistic regression classifier, the order of the top 3 is Humidity, Pressure, and then WindGustSpeed.

To sum up what we have learned. A regularization parameter $\lambda = 31.62$ yielded a better performance with a lower test error when we wanted to predict the amount of rain there would fall. This linear regression model wasn't significantly better than an artificial neural network regression model and a baseline. To classify whether or not it would rain the next day, the logistic regression model had the best performance, when a statistical evaluation was performed between the three models.
The most important attributes for predicting rain were Humidity, Pressure, and WindGustSpeed. Our results align with previous studies and suggest that our model is effective in predicting rain tomorrow. For further analysis, we could consider the class imbalance between dry and rainy days.

[3]https://ruitingm.medium.com/rain-in-australia-dataset-prediction-with-logistic-regression-decision-tree-and-random-forest-805396ea58fd

5 Exam problems

All the calculations can bee seen in the appendix.

Question 1

The correct answer is **C**.

If we use the threshold $\theta = 0.8$, $\theta = 0.6$ and $\theta = 0.5$ and compute the FPR and TPR, would see that it is C that matches the ROC curve.

Question 2

The correct answer is **C**

The purity gain is computed

$$\Delta = \frac{98}{135} - \left(\frac{1}{135} \cdot 0 + \frac{134}{135} \cdot \frac{97}{134}\right) = 0.00741$$

Question 3

The correct answer is **A**.

The Network contains 124 parameters

$$10 \cdot (7+1) + 4 \cdot (10+1) = 124$$

Question 4

The correct answer is **D**.

In split A option C would not give the correct answer. In split B would it would only hold for options B and D. In split C it is only D that would give the same tree.

Question 5

The correct answer is **C**.

We compute the time taken for the logistics model and the plus it to the neural network time taken.

$$((5 \cdot 8 + 5) \cdot 4 + 8 + 1 \cdot 1) \cdot 5 + 5 \cdot (((5 \cdot 20 + 5 \cdot 5) \cdot 4 + 20) + 5) = 3570$$

Question 6

The correct answer is **B**.

We solve this question by calculating the probability for each observation.

$$\frac{1}{1 + e^{w1 \cdot b} + e^{w2 \cdot b} + e^{w3 \cdot b}}$$

And we get the following answers

$$A : 3.025322996 \cdot 10^{-6} \quad B : 0.7304570365$$
$$C : 8.150541366 \cdot 10^{-9} \quad D : 4.656384486 \cdot 10^{-6}$$

From this, we can see the largest probability is for B.

6 Appendix

6.1 Estimated accuracy

The estimated difference in accuracy of the two classifiers is then given by

$$\hat{\theta} = \hat{\theta}_A - \hat{\theta}_B \tag{1}$$

and the estimated accuracy is calculated by

$$\hat{\theta}_A = \frac{1}{n}\sum_{i=1}^{n} c_i^A, \quad \hat{\theta}_B = \frac{1}{n}\sum_{i=1}^{n} c_i^B. \tag{2}$$

Inserting eq. 2 in eq. 1 and rewriting the expression, we see that

$$\begin{aligned}
\hat{\theta} &= \hat{\theta}_A - \hat{\theta}_B \\
&= \frac{\sum_{i=1}^{n} c_i^A - \sum_{i=1}^{n} c_i^B}{n} \\
&= \frac{\sum_{i=1}^{n} c_i^A - \sum_{i=1}^{n} c_i^B}{n} \\
&= \frac{(\sum_{i=1}^{n} c_i^A - \sum_{i=1}^{n} c_i^A c_i^B) - (\sum_{i=1}^{n} c_i^B - \sum_{i=1}^{n} c_i^A c_i^B)}{n} \\
&= \frac{\sum_{i=1}^{n} (c_i^A - c_i^A c_i^B) - \sum_{i=1}^{n} (c_i^B - c_i^A c_i^B)}{n} \\
&= \frac{\sum_{i=1}^{n} c_i^A (1 - c_i^B) - \sum_{i=1}^{n} c_i^B (1 - c_i^A)}{n} \\
&= \frac{n_{12} - n_{21}}{n}.
\end{aligned}$$

6.2 Exam questions

6.2.1 Question 1

To match ROC we start by setting a threshold, $\theta = 0.8$. A, C and D all have the same values. So we calculate their TPR and FPR the following way

$$FPR = \frac{FP}{FP + TN} = \frac{1}{1+3} = 0.25$$
$$TPR = \frac{TP}{TP + FN} = \frac{1}{1+3} = 0.25$$

This fits with the ROC curve.
We now calculate the same for B

$$FPR = \frac{FP}{FP + TN} = \frac{0}{0+4} = 0$$
$$TPR = \frac{TP}{TP + FN} = \frac{2}{2+2} = 0.5$$

We see that the values for B doesn't fit the ROC and therefore we can exclude B from the next calculations.
We now change theta

$$\theta = 0.6$$

Here C and D have the same values

$$FPR = \frac{FP}{FP+TN} = \frac{2}{2+2} = 0.5$$
$$TPR = \frac{TP}{TP+FN} = \frac{3}{3+1} = 0.75$$

This fits the ROC curve. We calculate for A

$$FPR = \frac{FP}{FP+TN} = \frac{3}{3+1} = 0.75$$
$$TPR = \frac{TP}{TP+FN} = \frac{2}{2+2} = 0.5$$

This doesn't fit the ROC curve and we exclude A from the further calculations.
We change theta again

$$\theta = 0.5$$

We calculate for C

$$FPR = \frac{FP}{FP+TN} = \frac{4}{4+0} = 1$$
$$TPR = \frac{TP}{TP+FN} = \frac{3}{3+1} = 0.75$$

And we calculate D

$$FPR = \frac{FP}{FP+TN} = \frac{3}{3+1} = 0.75$$
$$TPR = \frac{TP}{TP+FN} = \frac{4}{4+0} = 1$$

From this we can conclude the correct answer must be c, since it matches the ROC curve.

6.2.2 Question 2

To solve this, we use the formula for impurity gain

$$ClassError(v) = 1 - \max_c P(c|v)$$
$$P(c|v) = \frac{\text{number of class c in branch v}}{N(v)}$$

we start off by calculating the total amount of observations

$$N(r) = 33+4+0+28+2+1+30+3+0+29+5+0 = 135$$

Then we can calculate the I(r)

$$I(r) = 1 - \max(\frac{33+4+0}{135}, \frac{28+2+1}{135}, \frac{30+3+0}{135}, \frac{29+5+0}{135}) = \frac{98}{135}$$

The number of observation in the two branches $x_7 = \{0,1\}$ and $x_7 = 2$

$$N(v_1) = 33+28+30+29+4+2+3+5 = 134$$
$$N(v_2) = 0+1+0+0 = 1$$

We now look at the scenario $x_7 = 2$

$$I(x_7 = 2) = I(v_1) = 1 - \max(\frac{0}{N(v_2)}, \frac{1}{N(v_2)}, \frac{0}{N(v_2)}, \frac{0}{N(v_2)}) = 0$$

Then we calculate the scenario $x_7 \neq 2$

$$I(x_7 \neq 1) = I(v_2) = 1 - \max(\frac{33+4}{N(v_1)}, \frac{28+2}{N(v_1)}, \frac{30+3}{N(v_1)}, \frac{29+5}{N(v_1)}) = \frac{8}{11}$$

This is all added up and we get the result

$$\Delta = I_r - \sum_{k=1}^{2} \frac{N(v_k)}{N(r)} I(v_k) = \frac{98}{135} - (\frac{134}{135} \cdot 0 + \frac{1}{135} \cdot \frac{8}{11}) = 0.0074.$$

The correct answer must be **C**

6.2.3 Question 3

We know that there are 7 attributes one hidden layer and 10 hidden units. Besides this we also need to train the bias so we get the equation

$$parameters_{hidden} = (7+1) \cdot 10 = 80.$$

There are 4 possible outcomes and another bias so therefore we add the following

$$parameters_{output} = (10+1) \cdot 4 = 44.$$

And from this we find that we need to train $80 + 44 = 124$ parameters to fit the neural network.

6.2.4 Question 5

I det indre loop skal der trænes og tester i hvert fold med $K_2 = 4$ folds. Det gentages 5 gange, 1 for hver model:

$$inner_{ANN} = (20+5)ms \cdot 5 \cdot 4 \cdot 5 = 2500ms,$$
$$inner_{LOG} = (8+1)ms \cdot 5 \cdot 4 \cdot 5 = 900ms.$$

I det outer fold træner og tester vi én gang i hvert fold for $K_1 = 5$ folds:

$$outer_{ANN} = (20+5)ms \cdot 5 = 125ms,$$
$$outer_{LOG} = (8+1)ms \cdot 5 = 45ms.$$

I alt tager det

$$(2500 + 900 + 125 + 45)ms = 3570ms.$$

6.2.5 Question 6

We solve this question by calculating the probability for each observation. To do this we set k = 4 and use the given formula

$$\frac{1}{1 + \sum_{k'=1}^{3} e^{\hat{y}_{k'}}}.$$

We calculate for each observation

$$\frac{1}{1 + e^{w_1 \cdot b} + e^{w_2 \cdot b} + e^{w_3 \cdot b}}$$

And we get the following answers

$$\begin{aligned} A: &\quad 3.025322996 \cdot 10^{-6} \\ B: &\quad 0.7304570365 \\ C: &\quad 8.150541366 \cdot 10^{-9} \\ D: &\quad 4.656384486 \cdot 10^{-6} \end{aligned}$$

From this we can see the largest probability is for **B**.