Supervised learning:

Classification and regression

Australian weather

Group members

Anton Mosquera Storgaard - s214611

Jacob Borregaard - s181487

Mathias Correll Damsgaard - s214647

Group 230

Date: 15th november 2022

Academy: Technical University of Denmark - DTU

Subject: Introduction to Machine Learning and Data Mining - 02450

Assignment: Project

Contents

1 Regression

1.1 Part a

As we explained in part one of the project, our wish is to be able to predict the variable **Rainfall** which is a ratio data type which is continuous and non-negative. It has the unit millimeters. To start of with we wanting to be able to predict this variable based on the following attributes: **MinTemp**, **MaxTemp**, **Evaporation**, **Sunshine**, **WindGustSpeed**, **Humidity9am**, **Pressure9am**, **Cloud9am**, **Temp9am**. In the last project we learned that we should log-transform **MaxTemp** and take the root of **Humidity9am** and **Evaporation**. Furthermore we concluded that **WindHustSpeed** should be removed as an attribute based on that it had almost no correlation with the other attributes. Finally we will standardize the data so that each column has mean 0 and standard deviation 1.

When doing regression an easy way to control for overfitting is the use of a regularization parameter λ. We use this parameter and $K = 10$ fold cross-validation to get a figure of the estimated generalization error as a function of λ:

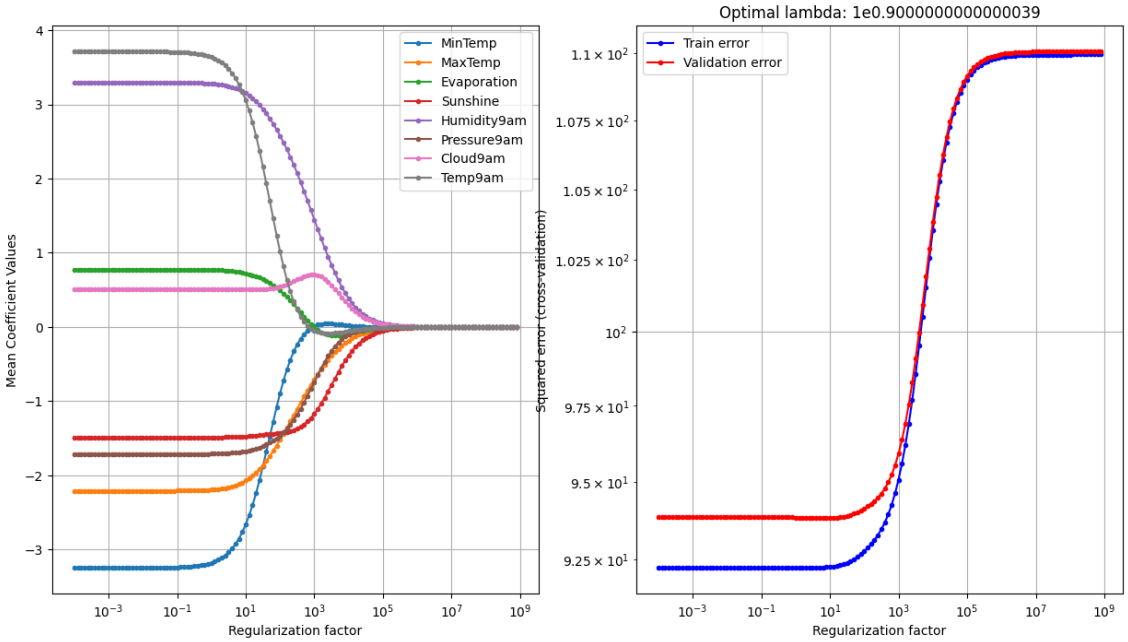

Figure 1: Train and validation error as a function of λ

We see that the regularization does not improve the result of the linear regression. This however is a clear indication that the linear regression model does not over-fit the data. This may be due to the fact that we have a good amount of data. We see that the optimal λ is about $1e^{0.9}$ and after that point the regularization simply worsens the models performance. We have a test error of 59.84 without regularization and a test error of 59.60 with regularization so we see no real improvement. The R squared value is only 0.16 for the training data for both of the models, so linear regression does not fit the data very well.

The weights of the attributes are found in Table 1.

Attribute	Offset	MinTemp	MaxTemp	Evap.	Sunshine	Humidity	Pressure	Cloud	Temp
Weight	3.3	-1.27	-2.16	0.44	-1.26	2.85	-1.43	0.33	2.17

Table 1: Weights in the optimal linear regression model

We can use an unseen datapoint to see how the model performs. We have a datapoint in Table 2.

Attribute	MinTemp	MaxTemp	Evap.	Sunshine	Humidity	Pressure	Cloud	Temp	**Rain**
Parameter	0.517	-0.0579	0.370	-1.35	0.598	-1.43	1.23	0.227	**27**

Table 2: New standardized datapoint

Using our model on the datapoint gives the following:

$$0.517\cdot(-1.27)-0.0579\cdot(-2.16)+0.370\cdot 0.44-1.35\cdot(-1.26)+0.598\cdot 2.85-1.43\cdot(-1.43)+1.23\cdot 0.33+3.3 = 8.79$$

So we see that the model predicts $8.79mm$ of rain while the true value for the datapoint is $27mm$. We see that the most important attributes are **Temperature**, **Pressure**, **Sunshine**, and **Humidity** which matches with our prior understanding of what factors have high correlation and impact on the principal components. Specifically low pressure, high humidity and low sunlight gives a higher amount of predicted rain according to the model.

1.2 Part b

We have chosen to use the same range of λ but only with a step size of 1 instead of 0.1 to save some computation time, since we saw in the last part that the increments didn't matter much. For the amount of hidden units in our ANN we did some test runs and saw, that the model would easily overfit with too many hidden units. To again also save some computation time we only tested with three different values of hidden units in the two-level cross-validation being 1, 3 and 5. Since we also saved some computation time on the amount of models to train, we choose to do a two-level cross-validation with $K_1 = K_2 = 10$ folds.

Below is shown a table with the data collected from the two-level cross-validation. It shows the optimal value of λ and hidden units for each outer fold and the test error for the models in that fold.

Outer fold	ANN		RLR		BL
i	h_i^*	E_i^{test}	λ_i^*	E_i^{test}	E_i^{test}
1	1	151.35	10^1	63.98	1009.66
2	1	136.72	10^1	107.38	959.71
3	1	47.14	10^1	46.15	1051.55
4	1	175.23	10^1	61.55	1026.49
5	1	186.15	10^0	123.80	934.37
6	1	116.64	10^0	35.82	1058.23
7	5	178.92	10^0	148.98	910.93
8	1	169.81	10^1	143.17	920.04
9	1	95.01	10^1	78.96	999.78
10	1	50.65	10^1	40.51	1048.76

Table 3: Two-level cross-validation table for three models

We can be see that it is mostly an ANN with only 1 hidden unit selected. We also find that a λ-value of 10^1 is mostly selected as the optimal parameter value, which matches well with the results from **Part a**. Lastly it seems like there is a big difference in the error values for the three models, but to say anything for sure, we will do a **setup II** evaluation, since it fits well with a two-level cross-validation.

Variable	BL vs. RLR	BL vs. ANN	RLR vs. ANN
p-value	$5.81 \cdot 10^{-12}$	$1.04 \cdot 10^{-11}$	$1.94 \cdot 10^{-4}$
CI	$[809.11; 1004.73]$	$[762.07; 960.31]$	$[-83.05; -8.42]$

Table 4: Setup II values

From the statistical setup it can be concluded that the regularized linear regression and artificial neural network is significantly different from the baseline. That can be seen both from the *p-value* being lower than 0.05 and that the confidence interval does not include zero. We can also conclude from the test between the two models that the RLR and ANN is significantly different from each other. We can therefore reject the H_0 hypothesis in all three cases.

Moreover it can be explained that since we calculated the variables as model subtracted from the baseline, that the RLR and ANN are better than the baseline model, since their values must be lower than the baseline's to make the confidence interval range in positive values. It was then also the ANN subtracted from the RLR, and since the confidence interval is ranges in negative values, the RLR's error is lower and must be the better model. Therefore it is recommended to go with the RLR model and probably with a λ-value of 10^1.

2 Classification

Our classification problem is closely related to our regression problem, with the difference being that instead of trying to predict how much it will rain on a given day in Sydney, we simply want to classify whether it will rain or not. Our approach is approximately the same, by classifying with the same features as in the regression, but classify for the variable **RainToday** instead. This means that the classification is a binary classification problem and we have one-hot encoded it so **RainToday** == "Yes" becomes 1 and **RainToday** == "NO" becomes 0.

2.1 Model 2: Artificial Neural Network

We have chosen an artificial neural network as our Model 2, because we believe that it's a good model for a binary classification problem. Our neural network is composed of an input layer, three hidden layers and an output layer with a logistic sigmoid function as the activation function. We have chosen the amount of hidden units in the three layers as the complexity-controlling parameter of our NN. Where we have the same amount of hidden units in each layer of the network. We will test the network with respectively 1,3 and 5 hidden units in each layer of the NN.

2.2 Logistic regression

The logistic regression is based on the maximum likelihood framework like the linear regression, but differs from the linear regression in the way that we need the output to classify binary rather than giving a continuous output. The classifier is based on the Bernoulli distribution which operates in a [0;1] framework, which is why we need to change it to a logistic model. We achieve that goal through a logistic sigmoid function. which gives the following probability density function:

$$p(y_i|x_i, w) = Bernoulli(y_i|\hat{y_i})$$

where:

$$\hat{y_i} = \sigma(x^T w) \text{ and } \sigma(z) = \frac{1}{1+e^{-z}}$$

The model will then make a prediction based on $\theta = 0.5$ classifying if it will rain or not, given our input features. We are going to use a regularization parameter λ as our complexity-controlling parameter for the logistic regression, just as we did in the linear regression part of the project. However another way we could have tuned our model, would be through the thresholding parameter θ which can change the accuracy of the model and the area under the curve.

We have checked the optimal weights for the logistic regression as well as we did with the linear regression to compare if the same attributes are deemed equally relevant for the two models. The weights can be seen in the following Table 5.

Attribute	Offset	MinTemp	MaxTemp	Evap.	Sunshine	Humidity	Pressure	Cloud	Temp
Weight	0.0	0.17	-1.22	-0.57	-0.03	0.93	-0.30	0.28	0.99

Table 5: Weights in the optimal logistic regression model

What we see is that the most important attributes for prediction are **Temperature**, **Evaporation**, and **Humidity**. Which are a bit different to the linear model, where **Evaporation** wasn't weighted as heavily, but rather **Sunshine** and **Pressure** were more important. This might be due to the difference in the two tasks, as the linear regression tries to predict the amount of rain, while the classifier is simple interested if it rains or not.

2.3 Testing and evaluation

We then did a two-level cross-validation for our three models being the artificial neural network, regularized logistic regression and a baseline which simply predicts the most common class every time.

Outer fold	ANN		RLR		BL
i	h_i^*	E_i^{test}	λ_i^*	E_i^{test}	E_i^{test}
1	1	0.169	0.398	0.165	0.237
2	1	0.208	0.1	0.212	0.280
3	1	0.212	1.0	0.191	0.271
4	1	0.182	0.1	0.174	0.254
5	1	0.195	1.0	0.225	0.309
6	1	0.182	1.0	0.174	0.212
7	1	0.169	0.1	0.186	0.284
8	1	0.182	1.0	0.195	0.246
9	1	0.191	0.631	0.2	0.226
10	1	0.187	1.0	0.196	0.264

Table 6: Two-level cross-validation table for three models

Based on the results from Table 6 we get the following results. The ANN seems to perform best with a simpler model with only a single hidden unit in each layer, while the RLR most of the times performs best with a λ-value of 1. This value comes is calculated as $1/\lambda$, since that is the value the logistic regression uses. Both the ANN and the RLR seems to be performing better than the baseline, but it's hard to see at a glance which of the two models that perform the best. In order to determine that we proceeded to do a statistical evaluation of the three models. Here we again followed a **setup II** approach, which gave us the following results:

Variable	BL vs. RLR	BL vs. ANN	RLR vs. ANN
p-value	$1.16 \cdot 10^{-7}$	$5.31 \cdot 10^{-7}$	0.228
CI	$[0.045; 0.088]$	$[0.043; 0.098]$	$[-0.011; 0.018]$

Table 7: Setup II values

From the statistical setup we can see that the regularized logistic regression and artificial neural network is significantly different from the baseline. That can be seen both from the *p-value* being lower than 0.05 and that the confidence interval does not include zero. Further more it can be said that since we calculated the variables as model subtracted from the baseline, that the RLR and ANN are better than the baseline model, since their values must be lower than the baseline's to make the confidence interval range in positive values.

We can also conclude from the test between the two models that the RLR and ANN isn't significantly different from each other, since both the *p-value* is higher than 0.05 and the confidence interval overlap 0. We can therefore reject the H_0 hypothesis, when comparing other models to the baseline, but not between our different models. Therefore it is recommended to not go with the baseline model, but we can't confidently choose between the RLR and the ANN, since we can't say for certain that one model performs better than the other.

Another way in which we could evaluate the performance of our models would be to evaluate them based on their scaled accuracy. An argument to do this type of evaluation is that there is a significant class imbalance between our two classes, rain and no rain, in our data set. The difference comes from the natural fact that it doesn't rain most of the time in Sydney, giving us 1749 of the 2358 observations being no rainfall. An example of the significance of this imbalance, would be that the scaled accuracy would give the baseline a performance that is half as good as it is in our test.

3 Discussion

Through our data analysis we have found evidence that factors as temperature, evaporation, pressure, sunshine and humidity can be used (to a degree) to determine if it will rain or not and about which amount in a given day. But even so, in our classification part, our optimal methods still predicted wrong 20% of the time. This can be explained by the fact that the weather is a fairly complex system and that is why the weather-report still gets it wrong from time to time. In the regression part we saw that the logistic regression outperformed the ANN and in the classification part it was more equal amongst the two methods. It is also a more trivial task to classify whether or not it will rain, than to use regression to predict in what amount. We suspect that our ANN might have been hindered by a lack of size and data which would have allowed it to become much more complex than the more simple logistic regression. For practical purposes we limited our data set to only consider Sydney instead of the whole of Australia and only 3 hidden-layers in the ANN as we did nor have time/GPU-power for a model of a bigger size. These are things which could be researched in future research for improved results.

Other examples of research conducted with this data set is this study about rainfall prediction, which also amongst other methods use logistic regression for classification. Their obtained accuracy was 83.97% which is really close to our result of 80.42% accuracy with logistic regression. The study also used a deep learning classifier which had 4 hidden layers with a maximum of 128 hidden units. This method obtained an accuracy of 84.02% in the classification task compared to our ANN with 81.21% accuracy. Two supposed contributing factors to the minor difference in performance is that they used the whole data set (and not just the Sydney subset), and they used the hold-out method for estimating the error while we used two-fold cross-validation. But nonetheless their results are largely consistent with ours.

4 Appendix

4.1 Responsibility of each student

Here is shown the contribution for each section for each student in approximate percentage:

Contribution %	Regression A	Regression B	Classification	Discussion	Exam Questions
Anton	50	15	25	35	33
Jacob	15	15	60	25	33
Mathias	35	70	15	20	33

4.2 Exam questions

I Question 1.

Answer is C.

By placing a threshold and iterating through the predictions it becomes clear that there is only one solution. First by placing a threshold at about 0.6 (on the prediction axis) we see that the predictions have the following TPR's and FPR's:

For prediction A and B:

$$TPR = \frac{2}{2+2} = 0.5$$

$$FPR = \frac{3}{3+1} = 0.75$$

While for prediction C and D:

$$TPR = \frac{3}{3+1} = 0.75$$

$$FPR = \frac{2}{2+2} = 0.5$$

We see on the ROC curve on Figure 1 that the only solution is prediction C and D as there is no curve on the point $(0.75, 0.50)$. We can narrow this further down by repeating and selecting a threshold at about 0.75 (on the prediction scale).

Here we obtain the values for D:

$$TPR = \frac{2}{2+2} = 0.5$$

$$FPR = \frac{1}{3+1} = 0.25$$

And for C:

$$TPR = \frac{1}{3+1} = 0.25$$

$$FPR = \frac{2}{2+2} = 0.5$$

We see that the only option is C as there is no ROC curve in the point $(0.25, 0.5)$

II Question 2.

Answer is C

To get the answer we simply use hunts algorithm. The probabilities on the root iare:

$$P(y=1|r) = \frac{37}{135}$$

$$P(y=2|r) = \frac{31}{135}$$

$$P(y=3|r) = \frac{33}{135}$$

$$P(y=4|r) = \frac{34}{135}$$

And for the left branch they are:

$$P(y=1|v1) = \frac{37}{134}$$

$$P(y=2|v1) = \frac{30}{134}$$

$$P(y=3|v3) = \frac{33}{134}$$

$$P(y=4|v1) = \frac{34}{134}$$

And for the right branch they are:

$$P(y=1|v2) = \frac{0}{1} = 0$$

$$P(y=2|v2) = \frac{1}{1} = 1$$

$$P(y=3|v2) = \frac{0}{1} = 0$$

$$P(y=4|v2) = \frac{0}{1} = 0$$

Using the classification error we get the following:

$$I(r) = 1 - \frac{37}{135} = \frac{98}{135}$$

$$I(v1) = 1 - \frac{37}{134} = \frac{97}{134}$$

$$I(v2) = 1 - 1 = 0$$

This gives us an impurity gain of:

$$\Delta = \frac{98}{135} - \frac{134}{135} \cdot \frac{97}{134} - \frac{1}{135} \cdot 0 = 0.0074$$

III Question 3.

The answer is A

The number of trainable parameters can be counted as the number of weights connecting the layers + the biases of the hidden and output layer. We count the number of neurons in the input layer as 7, the hidden layer as 10 and the output layer as 4. This gives us following calculation for the sum of parameters:

$$7 \cdot 10 + 10 \cdot 4 + 10 + 4 = 124$$

IV Question 4.

The answer is D

You can easily see that the correct answer is D, by looking at the two figures and seeing that to get congestion level 4 you need A and C to be true, and those two to give all values of $b_1 \geq -0.16$ which only happens in answer D.

V Question 5.

The answer is C

The combined training and testing time is 34 ms. The training and testing in the inner fold is 4 four times for each of the 5 models. Then the best model is selected and trained and tested again in the outer fold, which there is five of. That gives:

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

VI Question 6.

The answer is B

By calculating the $\hat{y}_k$ for the first 3 classes and then the denominator for all four different b-vectors, it is possible to calculate all the probabilities that a given vector should be classified to a given class. By comparing the four values from softmax only one of the b-vectors has a highest value for class 4 and that is observation $\mathbf{b} = \begin{bmatrix} -0.6 \\ -1.6 \end{bmatrix}$ with a value of 73%.