Introduction to Machine Learning and Data Mining

Course 02450

DANMARKS TEKNISKE UNIVERSITET

ML - Project 2

Hongjin Chen (s232289), Bozhi Lyu (s232251), Jialu C. Christiansen (s194175)

16. november 2023

DTU

Problem Statement

The objective of this project is to examine the impact of weather variables, including temperature, relative humidity, etc., on forest fires in Algeria's Bejaia and Sidi Bel-abbes regions. We will analyze the provided weather data to understand the relationships between these variables and fire incidents. Furthermore, our goal is to develop a predictive model that quantitatively assesses the risk of fire based on the available weather data.

Contributions

	Regression	Classification	Discussion	Exam quest.
s232289	30%	30%	40%	33.33%
s194175	40%	30%	30%	33.33%
s232251	30%	40%	30%	33.33%

Indhold

1 Regression

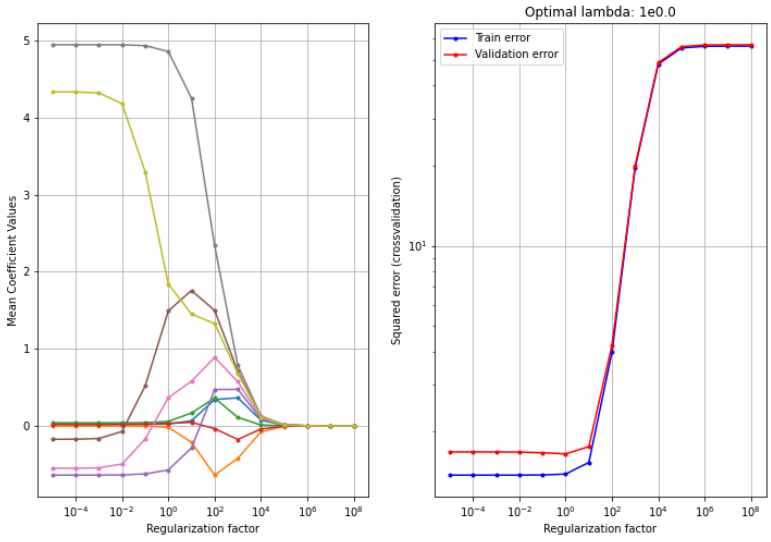
1.1 Part A - Dataset and Task Description

The data set [1] consists of 244 instances from two regions of Algeria: the Bejaia region and the Sidi Bel-abbes region. Each region contributes 122 instances to the data set. The data cover a period from June 2012 to September 2012 and comprises 11 attributes and 1 output attribute(fire/not fire).

We will use linear regression model to fit our data set with the first nine attributes: Temperature(Temp), Relative Humidity(RH), Wind speed(Ws), Rain, Fine Fuel Moisture Code(FFMC), Duff Moisture Code(DMC), Drought Code(DC), Initial Spread Index(ISI) and Buildup Spread Index(BUI) to predict the variable Fire Weather Index(FWI). Because all of the features are ratio features, and their respective orders of magnitude vary greatly, the data matrix X is normalized to better represent the relative relationship between each feature.

1.2 Part A - Linear regression models

We introduced a regularization parameter λ to regression models and estimated the generalization error for different λ in a 10-fold cross-validation to choose the optimal one.

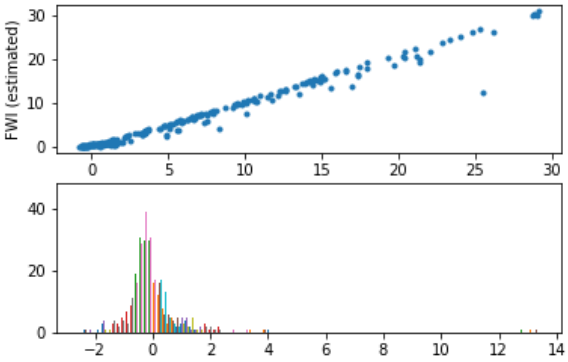


Figur 1: Regularization factor

The left figure 1 indicates the Mean Coefficient Values against different Regularization factor λ. As the regularization factor increases, the coefficients are penalized more, causing them to shrink. At extremely high regularization values (towards the right), the coefficients are almost zero, indicating the model is highly regularized, potentially to the point of being too simplistic.

On the right plot 1 shows that the validation error is slightly drop at $\lambda = 1$ and then increases dramatically. At the optimal lambda value of $\lambda = 1$, where the errors are both at their lowest($E^{\text{train}} = 1.348$,$E^{\text{test}} = 1.696$). So, this is the right optimal lambda value we want to choose. After fitting our linear regression model with the lowest generalization error, we get the weights w^* of our model as followed.

$$w^* = \begin{bmatrix} \text{Temp} & 7.17 \\ \text{RH} & -0.05 \\ \text{Ws} & 0.1 \\ \text{Rain} & 0.01 \\ \text{FFMC} & 0.02 \\ \text{DMC} & -0.79 \\ \text{DC} & 1.56 \\ \text{ISI} & 0.32 \\ \text{BUI} & 5.42 \end{bmatrix}$$



Figur 2: True vs. Predicted value

The weights indicate that Temp and BUI have the most significant positive impact on the output with weights of 7.17 and 5.42, respectively. RH and DMC have negative influences on the output. However, features like Ws, Rain, FFMC, DC, and ISI have positive but have a slight influence on the output. We believed that most of the attributes make sense on the basis of our understanding of our data's actual meaning.

The comparison between predicted FWi and true FWI are shown in Figure 2. The figure above is a scatter plot of true FWI against predicted value, and the bottom one is a histogram of residuals, which are calculated by the difference between the predicted and true values accomplished by our linear regression model. The scatter plot shows a positive linear relationship between the true and predicted values, which indicates that our model

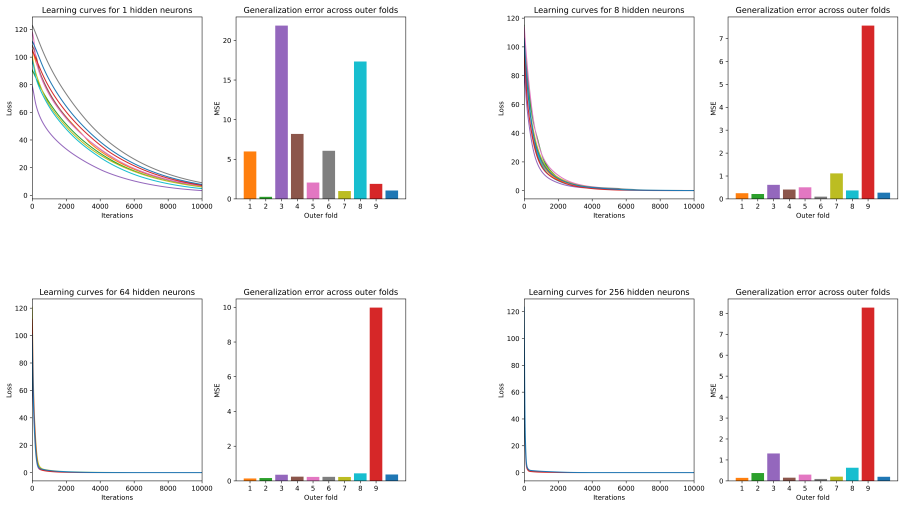
has captured underlying patterns in the data.

The residuals are mostly centered around 0, indicating that, on average, the model neither consistently overpredicts nor underpredicts the FWI. The distribution of residuals appears to be approximately normal, which is a good sign for our model. However, the analysis revealed some discrepancies in our residuals plot2 , especially for higher values of the true FWI and the presence of tails on either side indicates that there are some instances where the model prediction is significantly off from the actual value. This could be due to outliers in the data. As we discussed in the first project, we believed that these outliers should remain in our data (all of our data are collected from real world status such as rain, wind speed, these attributes could be outliers due to some abnormal weather conditions).

1.3 Part B - Model Comparison and Performance evaluation

In this section, we conducted a two-level cross-validation on linear regression models, artificial neural networks, and a baseline model in regression tasks, and compare their performance through statistical evaluation.

Before evaluating three different models, a few pre-runs were designed to get a range of parameters. The complexity-control parameter h of ANN ranges (1,2,4,8,16,32,64,128,256) and parameter λ of linear regression ranges (10^{-5},10^{5}). The results are shown in Table 1. By comparing the generalization error of each model, ANN performed better than both linear regression model with optimal hidden units $n = 16$, however, it is strange that we got different optimal $\lambda = 10^{-5}$ in linear regression model from what we got in the previous section.The p-values and PCCs suggest a strong performance of ANN model compared to other two models across all folds. It also makes sense that the number of hidden units increases the speed of convergence of the loss, which can be viewed in the figure below:



Figur 3: Learning rate for 4 different amount of hidden units and their respective generalization errors across outer folds.

Outerfold	**h***	**ann_E**	lambda*	lr_E	baseline	PCC	p-value
1	4	0.090	10^{-2}	0.597	55.791	1.000	0.000
2	16	0.101	10^{-5}	0.131	55.791	1.000	0.000
3	16	0.247	0.1	1.095	55.791	1.000	0.000
4	16	0.109	10^{-5}	0.683	55.791	1.000	0.000
5	16	0.035	10^{-5}	1.252	55.791	1.000	0.000
6	32	0.077	10^{-5}	0.572	55.791	1.000	0.000
7	128	0.075	1.0	0.567	55.791	1.000	0.000
8	8	0.369	10^{-5}	1.082	55.791	1.000	0.000
9	1	1.896	10^{-5}	9.301	51.429	1.000	0.000
10	16	0.167	1.0	1.674	55.791	1.000	0.000

Tabel 1: Two-level cross-validation table used to compare the three models.

Furthermore, the statistical evaluation of three pairwise models is tested using **Setup I**: statistical tests of performance considering the specific training set $\mathcal{D}$ and evaluated by the paired t-tests. In these comparisons, the Null Hypothesis H_0 is: Model $\mathcal{M}_A$ and $\mathcal{M}_B$ have the same performance in regression. After selected form of cross-validation, we computed the loss of two models' generalization error using[1]

$$\hat{z} = \frac{1}{n}\sum_{i=1}^{n} z_i, \text{where } z_i = z_i^A - z_i^B$$

Under $\alpha = 0.05$, p-value and confidence interval are calculated for each of the paired models and the results can be seen in Table 2.

[1] 11.38 eq from 02450 Introduction to Machine Learning and Data Mining Text Book

Paired Model	P-value	Confidence interval
Linear regression - ANN	1.010×10^{-2}	(0.07497, 0.5265)
Linear regression - baseline	1.748×10^{-5}	(-57.52, -23.42)
ANN - baseline	1.657×10^{-5}	(-57.89, -23.65)

Tabel 2: p-value and confidence interval of comparing regression pairwise.

A p-value measures the strength of the evidence against the null hypothesis. Here, the p-value of all three paired models are below the standard $\alpha = 0.05$, suggesting that we need reject H_0 because there is a statistically significant difference in performance between the models compared. The confidence interval between linear regression and ANN is (0.07497, 0.5265) , which means the difference between their performances is likely between these two values and we have 95% confidence that the ANN model has a higher value of the performance than the linear regression model. The confidence intervals between ANN and baseline, linear regression and baseline are (-57.89, -23.65) and (-57.52, -23.42), respectively. The confidence intervals show that both ANN and linear regression both have significant performance differences in the baseline mode, resulting in the same conclusion as in the previous section.

2 Classification

2.1 Task Description and Model Comparison

In the classification section, the predicted variable is a binary category (fire or not fire) in our dataset, and in the feature transformation step we use "1"for on fire and "0"for not fire. We want to make predictions about fire occurrences according to other attributes (temperature, wind, etc.)

To handle this classification task, We will use a) regularized logistic regression models with the regularization parameter λ as the complexity-controlling parameter; b) K-nearest neighbors models with the number of neighbors k as the complexity-controlling parameter; c) baseline, which will predict every samples in the test-data as belonging to the largest class on the training data.Based on our trial run, we will choose λ in a range of (10^{-5}, 10^5) and k in a range of (1, 10).

2.2 Cross-validation and Generalization Errors Estimation

To select the best model(or best complexity controlling parameters), a two-level cross-validation are applied to both regularized logistic regression models and KNN models. The outer layer consists of a 10-fold cross validation. Within this, the inner cross-validations involve a 10-fold cross validation for regularized logistic regression models and a leave-one-out cross validation for KNN models. As for an error measure, we used the error rate:

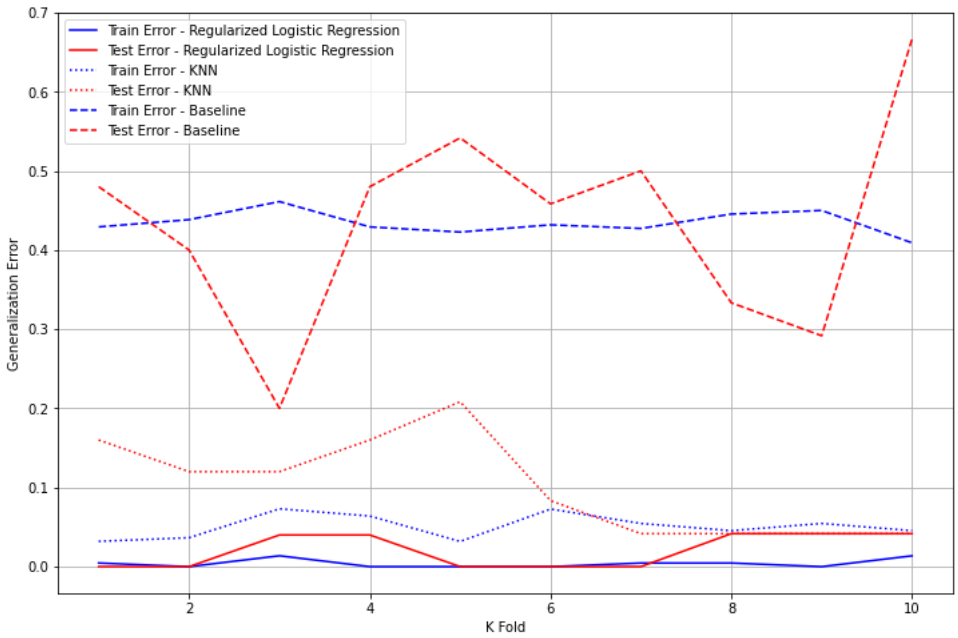
$$E = \frac{\{ \text{ Number of misclassified observations}\}}{N^{\text{test}}}$$

According to Table 3, the generalization errors of logistic regression models are basically lower than that of KNN models. Even in some folds $E^{\text{test}}_{\text{Logistic}} = 0.0$, indicating that the model classified all testing data correctly. The averaged generalization errors on all folds of regularized logistic regression models and KNN models are $E^{\text{test}}_{\text{Logistic}} = 0.02467$ and $E^{\text{test}}_{\text{KNN}} = 0.1018$. The logistic regression model performed better than KNN model in the classification task on this forest fire dataset in general, and both two classifiers are effectively compared to the baseline model with an averaged generalization error value of 0.43516. A comparison of the training errors and testing errors of the three models on

Outer fold	KNN		Logistic regression		baseline
i	x_i^*	E_i^{test}	λ_i^*	E_i^{test}	E_i^{test}
1	3	0.16	10^{-5}	0.0	0.48
2	3	0.12	0.01	0.0	0.4
3	8	0.12	0.1	0.04	0.2
4	8	0.16	10^{-5}	0.04	0.48
5	3	0.2083	10^{-4}	0.0	0.5417
6	10	0.08333	0.01	0.0	0.4583
7	3	0.04167	0.01	0.0	0.5
8	3	0.04167	10^{-5}	0.08333	0.3333
9	3	0.04167	10^{-4}	0.04167	0.2917
10	3	0.04167	10^{-4}	0.08333	0.6667

Tabel 3: model parameter and generalization error after two-level cross-validation

each fold is shown in Fig 4. Detailed statistical evaluation on each model will be addressed in next section.



Figur 4: Generalization error on each fold of three different models in 2-level cross-validation

2.3 Statistical Evaluation on Models

A statistical evaluation was carried on three models pairwise, and we chose the **Setup I** statistical tests of performance considering the specific training set $\mathcal{D}$ and evaluated by McNemar test. In these three comparisons (logistic regression V.S. KNN, logistic regression V.S. baseline, and KNN V.S. baseline), the null hypothesis H_0 is: Model $\mathcal{M}_A$ and $\mathcal{M}_B$ have the same performance in classification.

In the cross-validation section above, we trained and tested both three models in one loop, which means their training and testing data for each fold are from the same dataset splitting and therefore the same. Then we can obtain one-to-one corresponding predictions from three models as well as the true labels.We evaluated the paired-wise models on these predictions then computed the matrix $\mathbf{n}$ [2]. Then, we estimated the performance differences with p-values and confidence intervals[3]. With $\alpha = 0.05$, we got the evaluation performance table of each paired models, as shown in Table 4.

Paired Model	P-value	Confidence interval
(logistic regression V.S. KNN	1.9431×10^{-5}	(0.04740, 0.12460)
logistic regression V.S. baseline	2.7442×10^{-28}	(0.34923, 0.47663)
KNN V.S. baseline	1.1019×10^{-17}	(0.25914, 0.39492)

Tabel 4: p-value and confidence interval of comparing models pairwise on classification.

According to Table 4, when comparing logistic regression classifier and KNN classifier, we got a p-value of 1.9431×10^{-5} less than 0.05, which means the null hypothesis that these two models have the same performance should be rejected; and according to the CI, there is a 95% probability that the logistic regression classifier will be more accurate than the KNN classifier by 0.04740 to 0.1246. The p-values of the two groups of logistic regression classifier and KNN classifier respective comparison with baseline are smaller and close to 0, indicating that the null hypothesis of both two comparisons should be rejected. And according to the CI of the two groups, with a 95% probability, the accuracy of logistic regression and KNN compared to baseline are improved by (0.34923, 0.47663), (0.25914, 0.39492) respectively. In conclude, to rank the performance of these three models: logistic regression $>$ KNN $\gg$ Baseline.

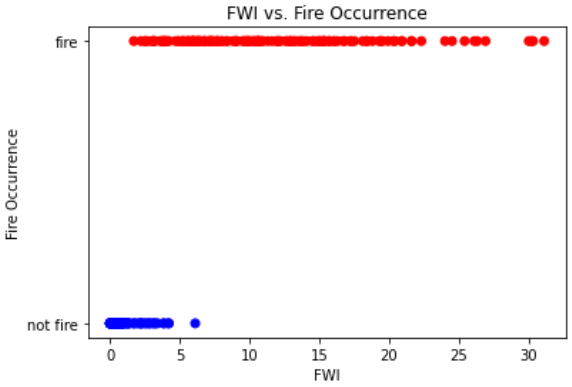
As for how the logistic regression model make a prediction(Question 5), we can compare the procedure with that of linear regression. They have the same steps like model training and eventually get demanding weights and other parameters like regularization factor λ. The difference is the output, which is not the results of a linear combination, but a mapped result through a logistic function. In our dataset, FWI(Fire Weather Index) is the continuous variable to be predicted in linear regression, which indicates the risk of

[2] 11.27 eq from Introduction to Machine Learning and Data Mining Text Book

[3] Method 11.3.2: The McNemar test for comparing classifiers, from Introduction to Machine Learning and Data Mining Textbook

fire occurrence. And the fire occurrence is the discrete, binary variable to be predicted in logistic regression. By rights the relevance of the features to the fire occurrence, that is, the weights w, should be similar to that of linear regression models. However, after training a logistic regression model with a $\lambda = 0.01$, which is the most choice of the best folds, we finally got a w^*, completely different from what we got in linear regression.

$$w^* = \begin{bmatrix} \text{Temp} & -2.42134391 \\ \text{RH} & 0.443761779 \\ \text{Ws} & -0.753760885 \\ \text{Rain} & 15.6836707 \\ \text{FFMC} & 53.5646721 \\ \text{DMC} & -12.7520470 \\ \text{DC} & 17.2150875 \\ \text{ISI} & 62.3093739 \\ \text{BUI} & -0.0578763098 \end{bmatrix}$$



Figur 5: FWI distribution and Fire Occurrence

The difference between the weights suggests that logistic regression and linear regression models value different features when making predictions. It makes sense because firstly, FWI correlates with, but is not a determinant of, fire occurrence as shown in the overlapping areas in Fig 5; Then some attributes are correlated, e.g. ISI is relevant to the wind and DC is relevant to temperature and rain; And factors such as the size of the dataset and the choice of regularisation parameter values all have an impact.

3 Discussion and summarization

Our project's primary goal was to apply machine learning techniques to predict the likelihood of forest fires. By employing both regression and classification models, the study aimed to evaluate the most effective methods for understanding and forecasting fire events.

In regression tasks, we used nine attributes to predict the variable Fire Weather Index(FWI), an indicator of the likelihood of a fire. We compared linear regression models, artificial neural network and baseline models in two-level cross-validation. And from out results, artificial neural network performed better than linear regression models and baseline model in detailed tasks. We believed that this is because ANN model can capture more underlying patterns from data. Additionally, it is crucial to select an optimal complexity-controlling parameter that ensures a balance between bias and variances.

The classification task aimed at predicting fire occurrences as a binary outcome (fire or no fire). This part we compared regularized logistic regression models, K-nearest neighbors models (KNN), and baseline models. The logistic regression models showed best performance compared to the KNN and baseline models.

One of the most important things we have learned in project2 was that completing the requirements in code requires not only what we have learned in class, but also more details. For instance, determining whether data should be standardized before or after passing it into the inner cross-validation function, or identifying which portion of the test data in the 2-level cross-validation algorithm should be used for statistical evaluation. These nuanced questions are challenging to uncover solely through lectures, but project2 provided us with an opportunity to address and resolve them, which gave us a deeper understanding of the overall process of machine learning. Meanwhile, our data has not been analyzed previously by any scientist before, which gave us a little challenge to address real word data by machine learning techniques and we believe that both our regression and classification methods achieved our expectations.

4 Appendix

A Exam problems for the project

A.1 Question 1

- We assume that $y = 0$ as negative and $y = 1$ as positive, then we look at all predictions from left to right and started from the very first observation and calculate TP, TN, FP and FN respectively.
- For Prediction A:$TP = 1$,$TN = 4$,$FP = 0$,$FN = 3$, then calculate $FPR = 0$, $TPR = \frac{1}{4}$.
- For Prediction B:$TP = 4$,$TN = 1$,$FP = 3$,$FN = 0$, then calculate $FPR = \frac{3}{4}$, $TPR = 1$. As we can see from the ROC curve, prediction B will not be considered anymore.
- For Prediction C:$TP = 1$,$TN = 4$,$FP = 0$,$FN = 3$, then calculate $FPR = 0$, $TPR = \frac{1}{4}$.
- For Prediction D:$TP = 4$,$TN = 1$,$FP = 3$,$FN = 0$, then calculate $FPR = \frac{3}{4}$, $TPR = 1$. As we can see from the ROC curve, prediction D will not be considered anymore.
- Then we will move one more step from x-axis, calculating as same as above but included two observations.
- For Prediction A:$TP = 2$,$TN = 4$,$FP = 0$,$FN = 2$, then calculate $FPR = 0$, $TPR = \frac{2}{4}$.
- For Prediction C:$TP = 1$,$TN = 3$,$FP = 1$,$FN = 3$, then calculate $FPR = \frac{1}{4}$, $TPR = \frac{1}{4}$.
- According to the ROC curve, we conclude that only prediction C is matched with the plot.
- So the answer is C.

A.2 Question 2

- The total number of the dataset **N**:

$$N = 37 + 31 + 33 + 34 = 135$$

- If we split the data set according to $x_7 = 2$, then the number of each class in left branch should be $y_1 = 0$,$y_2 = 1$,$y_3 = 0$,$y_4 = 0$. For the right:$y_1 = 37$,$y_2 = 30$,$y_3 = 33$,$y_4 = 34$.

- The purity gain Δ for the split is:

$$\Delta = I_r - \frac{N_{v1}}{N_r} I_{v1} - \frac{N_{v2}}{N_r} I_{v2}$$

where $I_r = 1 - \frac{37}{135} = \frac{98}{135}$,$I_{v1} = 1 - 1 = 0$,$I_{v2} = 1 - \frac{37}{134} = \frac{97}{134}$,$N_{v1} = 1$,$N_{v2} = 134$.

So the purity gain Δ:

$$\Delta = \frac{98}{135} - \frac{97}{134} \times \frac{134}{135} \approx 0.0074$$

- So the answer is C.

A.3 Question 3

- There are 7 input attributes(x_1,x_2,...,x_7)and 10 units in the hidden layer. So between input layer and hidden layer, each of the unit, we will have $(7 + 1) \times n_h = 80$ parameters(plus one bias for each unit). Similarly, there will have $(10 + 1) \times 4 = 44$ parameters between hidden layers and output layers. So that is said the total number of the parameters are 124.

- So the answer is A.

A.4 Question 4

- As we can see from figure 3, the decision tree is divided into left and right two parts from split A, where congestion level 1 and 2 on the left and level 1, 3 and 4 on

the right. So we look at figure 4 the classification boundary plot where somewhere between $[-1, -0.5]$ can split the data as described, then we can exclude choice A and C.

- From figure 4, we can see that congestion level 4 is split from level 1 and 3 and it also split from node C according decision tree, based on the vertical line between somewhere near 0.
- So the answer is D.

A.5 Question 5

- Given the two-level cross-validation approach, the outer folds $K_1 = 5$, the total number of neural network models to be *trained* is :

$$K_1(K_2S + 1) = 105$$

,where S = 5 for both models. Since we trained the logistic regression model in same way, the number of models trained is also 105. As every model we use to train is also used for testing, so the number of times we train and test a neural network model is:

$$105 \times (20 + 5) = 2625$$

then the total time we spend to train our models is:

$$105 \times (20 + 5) + 105 \times (8 + 1) = 3570$$

- So the answer is C.

A.6 Question 6

- First, we calculate the observation A

$$\hat{y}_1 = \begin{bmatrix} 1 \\ -1.4 \\ 2.6 \end{bmatrix}^T \begin{bmatrix} 1.2 \\ -2.1 \\ 3.2 \end{bmatrix},$$

$$\hat{y}_2 = \begin{bmatrix} 1 \\ -1.4 \\ 2.6 \end{bmatrix}^T \begin{bmatrix} 1.2 \\ -1.7 \\ 2.9 \end{bmatrix},$$

$$\hat{y}_3 = \begin{bmatrix} 1 \\ -1.4 \\ 2.6 \end{bmatrix}^T \begin{bmatrix} 1.3 \\ -1.1 \\ 2.2 \end{bmatrix}.$$

So for observation A, we get:

$$\hat{y}_1 = 12.46$$

$$\hat{y}_2 = 11.12$$

$$\hat{y}_3 = 8.56$$

Then we calculate the per-class probabilities using the softmax transformation:

$$P(y = 1|\hat{y}) = \frac{e^{12.46}}{1 + e^{12.46} + e^{11.12} + e^{8.56}} \approx 0.7800$$

$$P(y = 2|\hat{y}) = \frac{e^{11.12}}{1 + e^{12.46} + e^{11.12} + e^{8.56}} \approx 0.2042$$

$$P(y = 3|\hat{y}) = \frac{e^{8.56}}{1 + e^{12.46} + e^{11.12} + e^{8.56}} \approx 0.01578$$

$$P(y=4|\hat{y}) = \frac{1}{1+e^{12.46}+e^{11.12}+e^{8.56}} \approx 3.025 \times 10^{-6}$$

Since the highest probability is $P(y=1|\hat{y})$, so observation A will not be assigned to class $y=4$.

The same process repeated for observation B, C and D;

For observation B, we get: $\hat{y}_1 = -2.66, \hat{y}_2 = -2.42, \hat{y}_3 = -1.56$

$$P(y=1|\hat{y}) = \frac{e^{-2.66}}{1+e^{-2.66}+e^{-2.42}+e^{-1.56}} \approx 0.0510$$

$$P(y=2|\hat{y}) = \frac{e^{-2.42}}{1+e^{-2.66}+e^{-2.42}+e^{-1.56}} \approx 0.0650$$

$$P(y=3|\hat{y}) = \frac{e^{-1.56}}{1+e^{12.46}+e^{11.12}+e^{8.56}} \approx 0.15350$$

$$P(y=4|\hat{y}) = \frac{1}{1+e^{-2.66}+e^{-2.42}+e^{-1.56}} \approx 0.7304$$

We can see that $P(y=4|\hat{y})$ has the highest probability, so it will be assigned to class $y=4$.

For observation C, we get: $\hat{y}_1 = 12.79, \hat{y}_2 = 12.13, \hat{y}_3 = 9.990$, so the probabilities are: $P(y=1|\hat{y}) \approx 0.6338$, $P(y=2|\hat{y}) \approx 0.3276$, $P(y=3|\hat{y}) \approx 0.03854$, $P(y=4|\hat{y}) \approx 1.767 \times 10^{-6}$, which will be assigned to class $y=1$

For observation D, we get: $\hat{y}_1 = 11.89, \hat{y}_2 = 11.03, \hat{y}_3 = 8.890$, so the probabilities are: $P(y=1|\hat{y}) \approx 0.6789$, $P(y=2|\hat{y}) \approx 0.2873$, $P(y=3|\hat{y}) \approx 0.03380$, $P(y=$

$4|\hat{y}) \approx 4.656 \times 10^{-6}$,which will be assigned to class $y = 1$ as well.

So obviously, only observation B is assigned to class $y = 4$, where $P(y = 4|\hat{y})$ had the highest probability.

- So the answer is B.

B Litteratur

[1] Algerian Forest Fires Dataset - UCI Machine Learning Repository. URL https://archive.ics.uci.edu/dataset/547/algerian+forest+fires+dataset. 3

[2] C. E. Van Wagner. *Development and structure of the Canadian Forest Fire Weather Index System*, volume 35. Canadian Forestry Service, 1987. ISBN 0-662-15198-4.

[3] W.J. de Groot. Interpreting the Canadian Forest Fire Weather Index (FWI) System, 1987.