Danmarks Tekniske Universitet

DTU

02450 - Introduction to Machine Learning and Data Mining
Project 2

Authors

Name	Contribution
Ian Beissmann	Regression parts A&B
	3.1-3.2
	3.3-3.5

May 7, 2020

Contents

1 REGRESSION part A

The task is to predict the C02 emissions in grams per kilometers of the cars made in 2019, based on the following attributes:

1. Engine size
2. Number of cylinders
3. Fuel type
4. Fuel consumption city (L/100 km)
5. Fuel consumption combined (L/100 km)
6. Fuel consumption highway (L/100 km)
7. Fuel consumption (mpg)
8. CO_2 rating
9. Smog rating

Linear regression with regularization parameter and artificial neural networks were chosen as models to execute the task. Additionally, a simple baseline model is going to be used as well in order to be compared with linear regression model and neural network. In the next sections, working principles of those models are going to be explained. In the end, to compare the models, statistical significance tests are going be provided.

1.1 Data preparation

CO_2 emissions is the variable that is going to be predicted based on all of the other attributes. It is going to be assumed that CO_2 emissions have linear dependence on all the other parameters. In order to execute linear regression with regularization parameter lambda, the data matrix X will be transformed in such way that every column (that corresponds to the certain attribute) has mean equal to 0 and standard deviation equal to 1. Additionally, the matrix Y is going to be transformed so that its mean is equal to 0. Therefore, it will not be necessary to add the column filled with number 1. Also, because the fuel type attribute is nominal and discrete attribute, one-of-K coding is going to be executed on that attribute. Therefore, a number of the total attributes is going to increase from 9 to 12.

1.2 Linear regression with regularization parameter

1.2.1 Regularization parameter λ

The regularization parameter λ is introduced in linear regression models in order to lower the variance of the models. In linear regression model with parameter λ predict rule is going to stay the same:

$$\mathbf{y_i} = w^T \ x_\mathbf{i} \tag{1}$$

What changes with λ is the way weights $\mathbf{w}$ are obtained. Instead of minimizing function

$$\mathbf{E(w)} = ||\mathbf{Xw} - \mathbf{Y}||^2 \tag{2}$$

the following function has to be minimized in dependence of λ

$$\mathbf{E(w)} = ||\mathbf{Xw} - \mathbf{Y}||^2 + \lambda||\mathbf{w}||^2 \tag{3}$$

Therefore, the training formula for obtaining the weights $\mathbf{w}$ is equal to:

$$\mathbf{w}^* = (\mathbf{X_T X} + \lambda \mathbf{I})^{-1} + \mathbf{X^T Y} \tag{4}$$

To predict the output of a new data observation, the model simply uses the equation (1) after the weights $\mathbf{w}$ have been calculated.

1.2.2 Training a model in dependence of λ and results

A model is going to be trained for various values of the parameter λ in order to calculate the generalization error in dependence of λ. In order to be able to estimate the generalization error, the cross validation method with K=10 folds is going to be used. A range of values of λ is going to be from 10^{-6} to 10^6, raising by one order of magnitude.

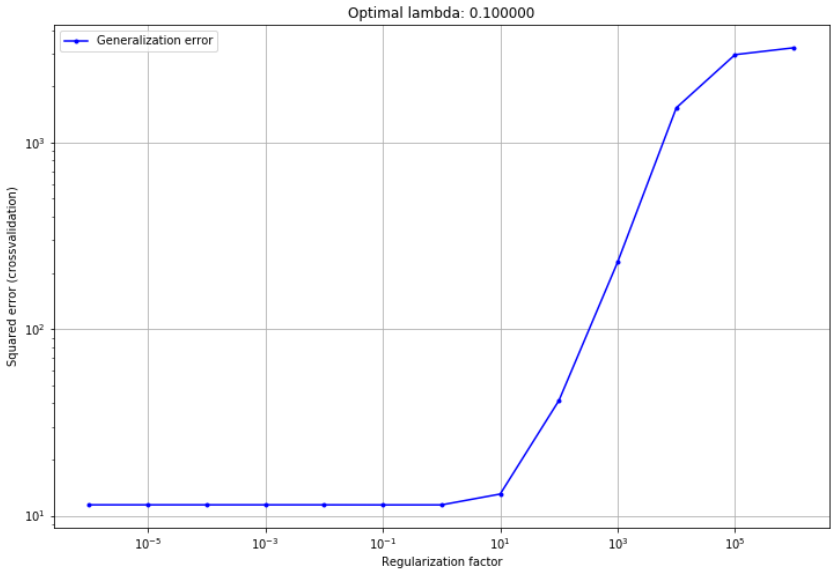

Figure 1: Generalization error in dependence of λ

1.2.3 Parameter observation

Figure 1 shows that the optimal value of λ is 0.1 and the generalization error is E_{gen}=11.366. As model has been trained 10 times (for each of the K folds), the different values for weights were obtained. For one of those folds, the weights were [0.596 0.13 24.27 13.22 21.6 -0.693 -2.94 -0.008 5.17 -19.2 3.05 2.69]. Those weights correspond to attributes in order they were mentioned at the beginning with one exception. Because one-of-K coding

was executed, the classes of a certain fuel type became the last 4 attributes in the code. The biggest influence on CO_2 emissions have attributes that correspond to fuel consumption and if the car runs on ethanol fuel. The result is one could expect as usually the cars that consume more fuel, usually produce more CO_2 because of that. Also, because the weight w_{10} is equal to $w_{10} = -19.2$, it can be concluded that the cars which run on the ethanol fuel produce less CO_2 what is also a result to expect.

2 REGRESSION part B

2.1 Two-level cross validation

Two-level cross validation will be used in order to compare three different models that can be implemented to this data set. The models will be linear regression with regularization parameter that has been introduced in the previous subsection, the artificial neural network (ANN) and the simple baseline, which will simply predict that the output value is equal to the mean of the train set. The artificial neural networks will have only one hidden layer with h different neurons in it, while the complexity parameter for linear regression is going to be parameter λ. The inner fold of the 2-level cross validation serves as a search tool for the optimal value of the complexity parameter. When the optimal value is found, in the outer fold the model with the optimal parameter is going to be trained on the training set defined by the outer cross validation folds.

2.1.1 Training the models and results

The values for the parameter h will be $h = 1, 5, 10, 15, 20, 25, 30$, while the values for the parameter λ will be the same as in the subsection 1.2.1. A bigger numbers of h are taken because the data set consists of a lot of instances (1022) and attributes (12). Because training the neural networks is really time-consuming, the number of folds in both cross validation loops is going to be $K_1 = K_2 = 5$. After choosing the optimal parameter, the optimal models are trained and the corresponding test errors recorded. The results are shown in the following table.

Outer fold	ANN		Linear regression		Baseline
i	h^*	E^{test}	λ^*	E^{test}	E^{test}
1	30	5.66	0.1	14.72	3552
2	25	19.67	0.1	5.52	3503
3	30	6.06	1	8.87	2891
4	30	4.11	1	6.88	3095
5	30	12.05	1	21.24	3263

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

It is clear that the baseline model is not a good model for predicting CO_2 emissions. The ANN and the linear regression models seem to have similar performance but in order to

properly compare those two models, statistical test will have to be evaluated. The optimal values for h^* is 30 almost every time which means that the more complex model neural network is needed in order to predict the CO_2 better. When it comes to the linear regression, the optimal values for λ^* is either 0.1 or 1. In the subsection 1.2.2 the optimal value was 0.1, so for some folds of the 2-level cross-validation, the optimal models are the same as in 1-level cross-validation.

2.2 Statistical evaluation

Statistical evaluation is done to conclude correctly which model is better for this regression problem. Statistical test in this case is **Setup I** where the test is fixed.
Significance level is chosen to be $\alpha = 0.05$.

Looking into table 2 and comparing more complex models linear regression and ANN with baseline, it is determined that the test is significant, and that both models linear regression and ANN are much better than the baseline. This is determined by inspecting that α is much greater than p-value.

Comparing linear regression to ANN gives different results. In this case, $\alpha < p - value$. As the results, we cannot conclude that linear regression is better than ANN.

Confidence intervals in the table explain 2 that there is probability of $1 - \alpha = 0.95$ that the result will fall into this interval.

	p	θ_L	θ_U
linear regression vs baseline	$6.96 \cdot 10^{-71}$	$-35.81 \cdot 10^2$	$-29.18 \cdot 10^2$
linear regression vs ANN	0.33	-11.63	-6.56
ANN vs baseline	$6.71 \cdot 10^{-71}$	$-35.83 \cdot 10^2$	$-37.59 \cdot 10^2$

Table 2: Setup I. Statistical analysis

3 Classification

3.1 Problem Description

For the Classification chapter we have chosen to try and predict the CO_2 rating of a car as described in 1.1 'Data preparation'. In the dataset the each car is given a CO_2 rating between 1 and 10. For our classification problem we wish to predict if a given car has a CO_2 rating of 1-2, 3-4, 5-6 or if it has a CO_2 rating of 7 or higher. See figure 2 below for a visual representation of the distribution of the different classification targets in our dataset

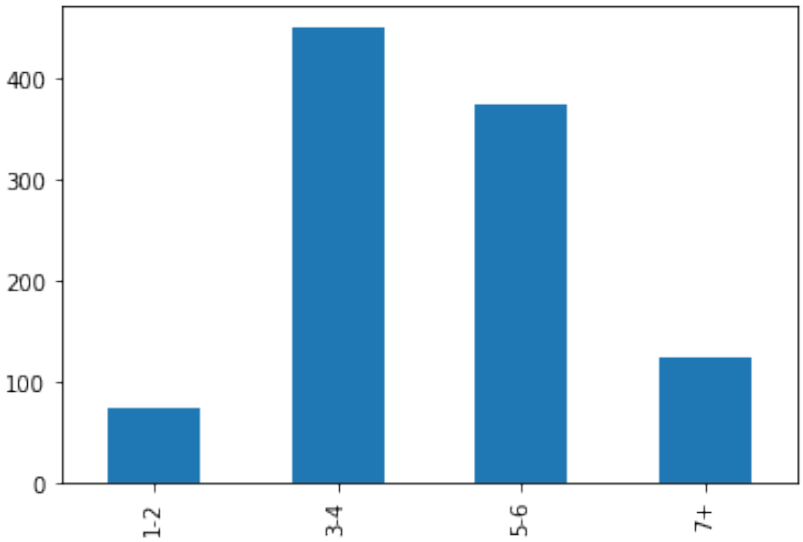

Figure 2: CO_2 rating, data presentation

To solve this multi-class classification problem we have developed a logistic regression model, a k-nearest neighbor model and a baseline model. To compare the different models we calculate the number of missclassified, error score aswell as a report generated using the scikit-learn command metrics.classification_report() which returns the precision score, recall score, f1-score and support score for each individual classification targets aswell as an average.
The precision score indicates the accuracy of positive predictions.

$$Precision = \frac{TruePositive}{TruePositives + FalsePositives}$$

The Recall score indicates the fraction of positives that was correct

$$Recall = \frac{TruePositive}{TruePositive + FalseNegatives}$$

The F1 score is the mean of the precision score and recall score. Because of this, the F1 score can more easily be used to compare to classification models.

$$F1score = 2 * \frac{PrecisionScore * RecallScore}{PrecisionScore + RecallScore}$$

3.2 Classification model comparisons

We will compare logistic regression, method 2, and a baseline. For logistic regression we will once more use λ as a complexity-controlling parameter, and for method 2 a relevant complexity controlling parameter and range of values.We recommend this choice is made based on a trail run, which you do not need to report. Describe which parameter you have chosen and the possible values of the parameters you will examine.

3.2.1 Logistic regression

	Precision	Recall	F1-score
1-2	1	0.73	0.85
3-4	0.84	0.86	0.85
5-6	0.83	0.85	0.84
7+	1	1	1
Macro Average	0.92	0.86	0.88

Table 3: Logistic Regression Performance score, $\lambda = 0.01$

Our Logistic regression model with complexity-controling parameter $\lambda = 0.01$ performed at the reletavly high F1-score of 0.88. Given the training and test dataset, our logistic regression model was better at predicting CO_2 ratings at 7 or higher, but still performed well at the other ratings. The model has an error rate of 13.7%. For our model we experimented with λ values ranging between 0.001 and 10, but we experienced the best results at $\lambda = 0.01$

3.2.2 Method 2: (k-Nearest neighbor)

	Precision	Recall	F1-score
1-2	1	0.93	0.96
3-4	0.93	0.98	0.95
5-6	0.94	0.92	0.93
7+	0.97	0.92	0.95
Macro Average	0.96	0.94	0.95

Table 4: KNN Performance score, k=2

Using our k-nearest neighbor prediction model, we managed to achieve a respectable F1-score of 0.95 with a k-value of 2. Experimenting with different k-values we can conclude that a k-value of 2 or 3, has the most reliable prediction performance. At $k = 2$ we have an error rate of 4.6%

3.2.3 Baseline

	Precision	Recall	F1-score
1-2	0.00	0.00	0.00
3-4	0.44	1	0.61
5-6	0.00	0.00	0.00
7+	0.00	0.00	0.00
Macro Average	0.11	0.25	0.15

Table 5: Baseline Performance score

Using our baseline model to try and predict the CO_2 rating for a given car, performs at an average f1-score of 0.15 and an error rate of 56.1%

3.3 Two-level cross-validation

Two-level cross validation is used in this task. Compared models are KNN, logistic regression and baseline. Error rate is presented as following:

$$E = \frac{Number\ of\ missclassified\ observations}{N^{test}} \tag{5}$$

Outer fold	KNN		Logistic regression		baseline
i	n_i^*	E_i^{test}	λ_i^*	E_i^{test}	E_i^{test}
1	3	1.46	0.0001	12.19	54.14
2	5	0.48	0.001	9.26	79.51
3	3	0.49	0.01	7.35	65.68
4	2	2.45	0.001	11.27	61.76
5	2	2.94	0.0001	12.74	66.66

Table 6: Classification. Comparison of KNN, Logistic regression and baseline.

The conclusion of results in table 6 is that the best model is KNN, following logistic regression, and baseline model. Better conclusion about the results are explained in Task 4.

Baseline Baseline model is expected to gives substantially bigger error than other two models. The reason is that it identifies every instance as member of the most frequent class. Baseline errors changes from fold to fold. The baseline error would be almost the same in folds if equal percentage of each class is put into every fold. That can be done by using $StratifiedK-fold$ which wasn't implemented in this case.

Generalized error for baseline is 65.55

Logistic regression Logistic regression gives fairly good results. The parameter which values were explored is regularization strength λ. Lambdas that were explored were $(1e^{-4}, 1e^{-3}, , 1e^{-2}, 1e^{-1}, 1.0)$. Higher values that 1.0 only give bad results.

Generalized error for logistic regression is 9.88

KNN This model provides the best results for predicting $CO2rating$ in the explored dataset. Parameter that was explored is number of neighbours. Explored neighbours are $(2, 3, 4, 5, 6)$.

Generalized error for KNN model is 1.56

3.4 Statistical evaluation

Statistical evaluation is done to conclude correctly which model is better for this classification problem. Statistical test in this case is **Setup II** where the test is random.

Significance level is chosen to be $\alpha = 0.05$.

Looking into table 7 and comparing more complex models KNN and logistic regression with baseline, it is determined that the test is significant, and that both models KNN and logistic regression are much better than the baseline. This is determined by inspecting that α is greater than p-value.

Comparing KNN with logistic regression, gives $\alpha > p - value$. Since significance level is greater than p-value, conclusion is that KNN has a significant difference in error, and that KNN model provides better results.

Confidence intervals in the table 7 explain that there is probability of $1 - \alpha = 0.95$ that the result will fall into this interval.

	p	θ_L	θ_U
knn vs baseline	$3.8913 \cdot 10^{-4}$	-80.25	-47.88
knn vs logistic regression	$5.7130 \cdot 10^{-4}$	-11.63	-6.56
logistic regression vs baseline	$9.2778 \cdot 10^{-4}$	-72.36	-37.59

Table 7: Setup II. Statistical analysis

3.5 Logistic Regression Model prediction

The Logistic regression prediction model tries to solve a classification problem by calculating the probability that an an input belongs to a class given some variables. For a binary classification problem, the goal is to predict if y is positive or negative given x and w .

$$p(y|x, w)$$

This is done by first finding an equation which describes the probability problem stated above, and applying maximum likelihood estmation. Selecting the parameter w^* such that the error function is at a minimum given the training- and test-dataset. Once the probaility equation, parameters and input values are known, the probability of y being positive or negative can be calculated, which ever outcome has the highest probability is the prediction which the model will give.

In the comparison to regression where there goal was to predict CO2 emission, in this task the goal is to predict CO2 rating. Weights consist of 12 values as explained in the regression task while introducing the dataset. Note that the last 3 values are from made from one hot encoding. In this dataset, example of weights is: [1.97093143 -2.74377408 -7.06527144 -3.4927621 -16.43075643 -42.22784233 -26.22203952 0.52925829 1.42837301 -11.61202401 2.20214086 1.85333921]. The most influential attributes for CO2 ratings pre-

diction are fuel consumption and CO2 emission. The most influential attribute of fuel consumption is $Comb(mpg)$

4 Discussion

By using the tools from supervised learning we trained and evaluated three different models for both classification and regression.

4.1 Regression

1. Linear regression with regularization parameter:
The assumption that CO_2 emissions depends linearly on other attributes seem to be valid as the linear regression generated good results. Maybe it is worth the effort to perform a more thorough and quantitative investigation of the data and try to find a better model using some other regression methods, for example the forward selection. This is simple to implement and since linear regression is a fast performing method, also not computationally intensive.

2. Artificial neural networks The ANN model generated similar results to the linear regression model. When taking into consideration the time to train and the time to set up the neural network, the gain in model prediction seems little enough to not be worth the time. On the other hand, introducing an extra hidden layer would maybe be able to create a better model, but it would still be computationally demanding.

3. Baseline As expected, the baseline model did not generate any useful results. The errors of the baseline model were huge and therefore it was not too useful.

4.2 Classification

1. Logistic regression with regularization parameter: Prediction of CO_2 ratings provides fairly good results with this method, but in comparison to KNN it is not good. Further work, and changing parameters can possibly lower the error that this model gives. Lowering the error would improve significance of this model in comparison to KNN.

2. K-nearest neighbour KNN model for predicting CO_2 ratings yields the best results out of this 3 models. With K equal to 2 or 3, this model predicts great results with very low error.

3. Baseline The baseline model is based on the most frequent class. It is intuitive to conclude that baseline gives the worst results, especially since the dataset does have fairly distributed all classes.

4.3 Previous work

When looking through the website where the data set has been obtained [1] we could not find any published papers that analyzed the data with either regression or classification.

References

[1] "Fuel consumption ratings 2019." https://www.kaggle.com/mhendal/2019-fuel-consumption-ratings-20190514.