INTRODUCTION TO MACHINE LEARNING
(NPFL054)
A template for Homework #1

Name: Jakub Genči

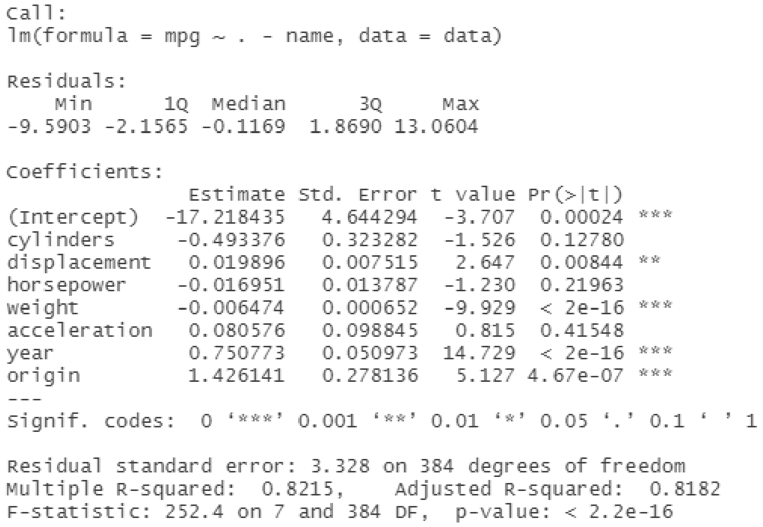
School year: 2021/2022

➓ **Provide answers to the exercises.**

➓ **For each exercise, your answer should not exceed one sheet of paper.**

1.1 Multiple linear regression

After performing the linear regression, we get these results:



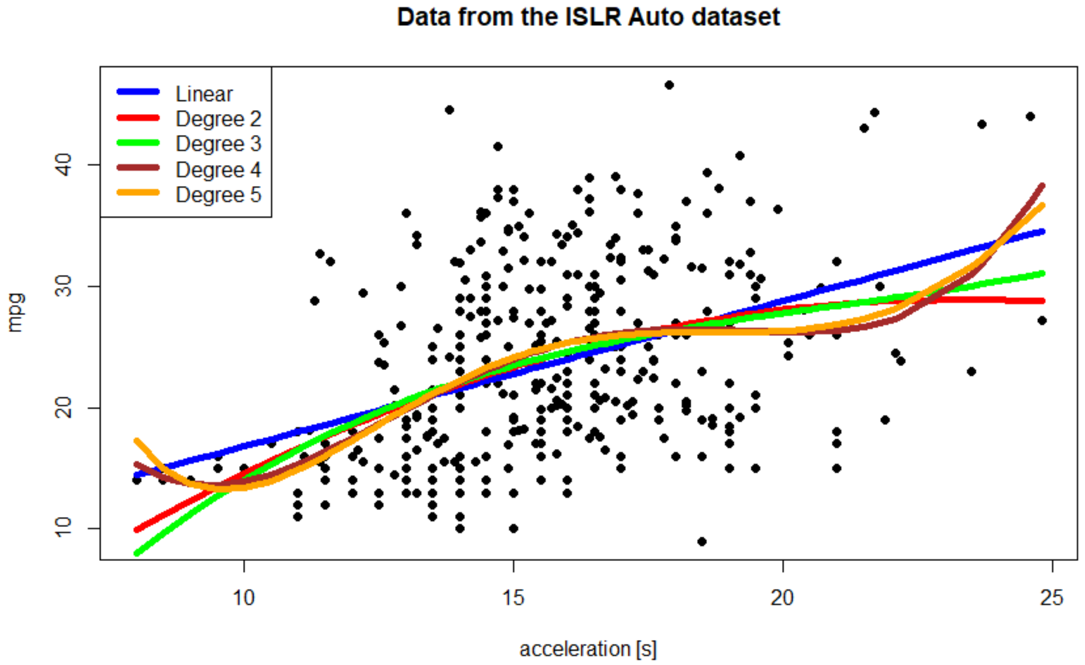
```
Call:
lm(formula = mpg ~ . - name, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.5903 -2.1565 -0.1169  1.8690 13.0604 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
cylinders     -0.493376   0.323282  -1.526  0.12780    
displacement   0.019896   0.007515   2.647  0.00844 ** 
horsepower    -0.016951   0.013787  -1.230  0.21963    
weight        -0.006474   0.000652  -9.929  < 2e-16 ***
acceleration   0.080576   0.098845   0.815  0.41548    
year           0.750773   0.050973  14.729  < 2e-16 ***
origin         1.426141   0.278136   5.127 4.67e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.328 on 384 degrees of freedom
Multiple R-squared:  0.8215,	Adjusted R-squared:  0.8182 
F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

Values of the hypothesis parameters are in the ‘Estimate’ column. All of the parameters used in the regression except ‘origin’ can be easily interpreted as numerical values. Interpretation of the hypothesis parameters is following:

- For two exactly the same cars with different number of cylinders, we predict that *mpg* will differ on average by -0.49 miles with every additional cylinder,
- for two exactly the same cars with different engine displacement, we predict that *mpg* will differ on average by 0.02 miles with every additional inch of the engine displacement,
- for two exactly the same cars with different horsepower, we predict that *mpg* will differ on average by -0.02 miles with every additional unit of horsepower,
- for two exactly the same cars with different weight, we predict that *mpg* will differ on average by -0.01 miles with every additional lbs.,
- for two exactly the same cars with different acceleration, we predict that *mpg* will differ on average by 0.08 miles with every additional second the car needs to accelerate to 60 mph,
- for two exactly the same cars with different year of the model, we predict that *mpg* will differ on average by 0.75 miles for every additional year,
- for two exactly the same cars with different origin, we predict that *mpg* of European cars will be on average 1.42 miles higher than *mpg* of American cars and *mpg* of Japanese cars will be on average 1.42 miles higher than *mpg* of European cars.
- Mathematical meaning of the parameter Θ_0 (-17.22, found in the ‘Intercept’ row) is that the car with value of each parameter equal to zero will have be able to travel -17.22 miles per gallon on average.

1.2 Polynomial regression

Results of the polynomial regression are shown in this picture:



Values of the adjusted R^2 are:

Degree of the model	Adjusted R^2
1	0.18
2	0.19
3	0.19
4	0.21
5	0.2

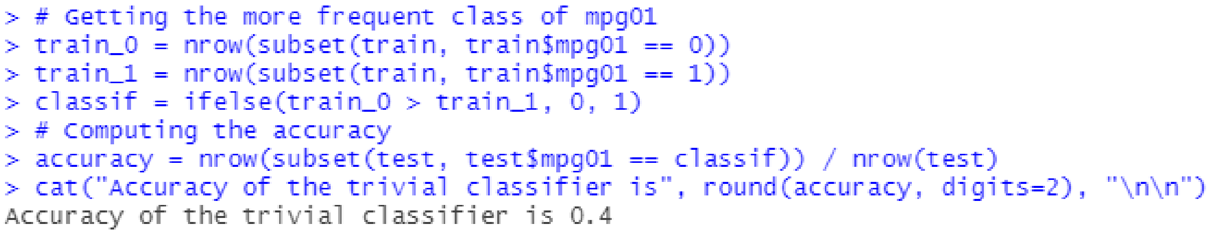
2.1 Binary attribute `mpg01` and its entropy

Creation of the *mpg01* attribute is implemented by using the *ifelse* function. Entropy is then calculated by using the library “entropy”.

Entropy of the *mpg01* attribute is equal to 1. We know that half of the values in *mpg01* is ‘0’ and the other half is ‘1’, because we assign them based on median of *mpg*. Therefore, we have uniform distribution with two possible values, which gives us entropy equal to 1.

2.3 Trivial classifier accuracy

We have a trivial classifier with two classes. This means that we will compute accuracy as a number of samples classified correctly, divided by the number of all samples (in the test set). When we run the code, we will see that accuracy of the classifier on the test set is around 0.4.

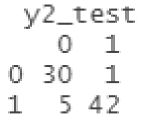


```
> # Getting the more frequent class of mpg01
> train_0 = nrow(subset(train, train$mpg01 == 0))
> train_1 = nrow(subset(train, train$mpg01 == 1))
> classif = ifelse(train_0 > train_1, 0, 1)
> # Computing the accuracy
> accuracy = nrow(subset(test, test$mpg01 == classif)) / nrow(test)
> cat("Accuracy of the trivial classifier is", round(accuracy, digits=2), "\n\n")
Accuracy of the trivial classifier is 0.4
```

2.4 Logistic regression – training and test error rate, confusion matrix, Sensitivity, Specificity, interpretation

Part a) Training error rate is computed as 1 – accuracy on the training data set. Training error rate of our model is 0.1.

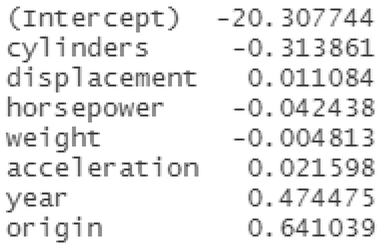
Part b) Confusion matrix for the classification of our test data (true values are in the rows and predictions in the columns):



```
   y2_test
     0  1
0 30  1
1  5 42
```

Test error rate of our model is 0.08, sensitivity is 0.89 and specificity is 0.97.

Part c) We obtain these estimates of the hypothesis parameters:

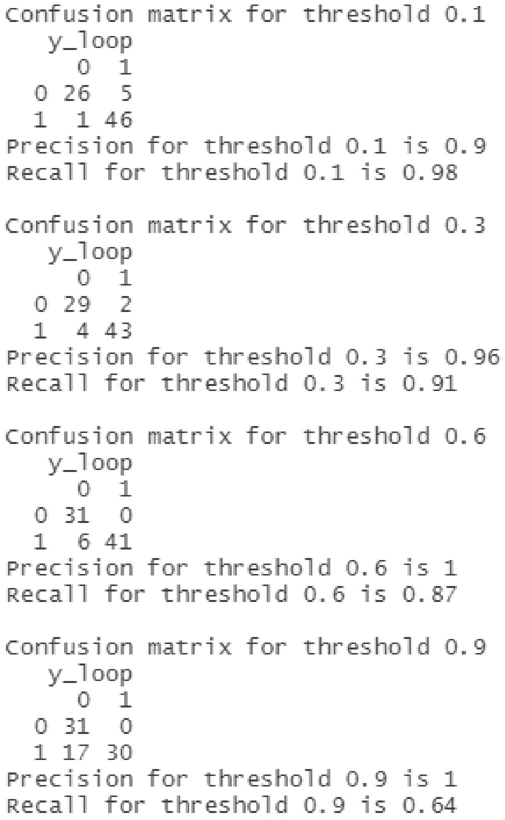


```
(Intercept)  -20.307744
cylinders     -0.313861
displacement   0.011084
horsepower    -0.042438
weight        -0.004813
acceleration   0.021598
year           0.474475
origin         0.641039
```

Mathematically, these estimates give us coefficients for the hyperplane defined by the model (in the direction of each parameter). Empirically, it tells us an average change for log-odds when a unit change of a certain parameter happens while other values stay the same.

2.5 Logistic regression – threshold 0.1, 0.3, 0.6, 0.9, confusion matrix, Precision, Recall, F1-measure, interpretation

Part a) We obtain these values for these thresholds (rows in confusion matrices represent true values, columns represent predicted values):



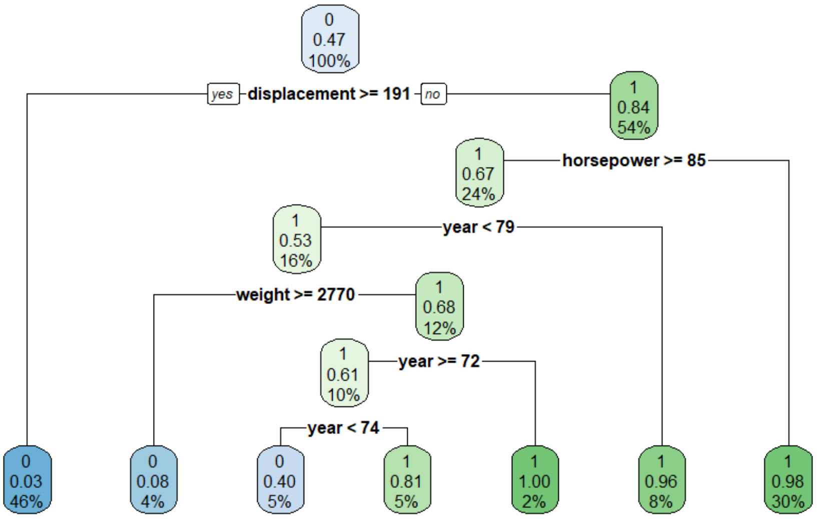
```
Confusion matrix for threshold 0.1
   y_loop
     0  1
  0 26  5
  1  1 46
Precision for threshold 0.1 is 0.9
Recall for threshold 0.1 is 0.98

Confusion matrix for threshold 0.3
   y_loop
     0  1
  0 29  2
  1  4 43
Precision for threshold 0.3 is 0.96
Recall for threshold 0.3 is 0.91

Confusion matrix for threshold 0.6
   y_loop
     0  1
  0 31  0
  1  6 41
Precision for threshold 0.6 is 1
Recall for threshold 0.6 is 0.87

Confusion matrix for threshold 0.9
   y_loop
     0  1
  0 31  0
  1 17 30
Precision for threshold 0.9 is 1
Recall for threshold 0.9 is 0.64
```

Part b) Interpretation for performance measures:

- precision measures relative correctness of the prediction for samples classified as positive,
- recall measures relative correctness of the prediction for samples with positive true value,
- mathematically, F1-measure gives us harmonic mean of precision and recall. Empirically, it’s a measure of accuracy which is closer to lower value from the two than the simple average of precision and recall.

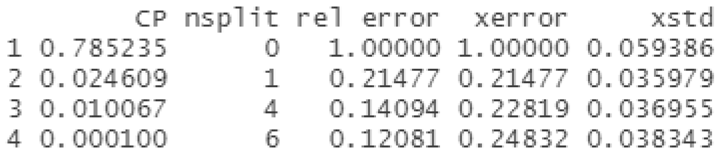
2.6 Decision tree algorithm – training and test error rate, cp parameter

Part a) Plot of the tree (predicted class associated with the leaves is in the first row):



Train error rate of this model is 0.06 and the test error rate is 0.04.

Part b) We start by rebuilding the model with cp = 0.0001. After training the model, we can obtain the information about the tree with the *printcp* function. We obtain these results:



```
        CP nsplit rel error  xerror     xstd
1 0.785235      0   1.00000 1.00000 0.059386
2 0.024609      1   0.21477 0.21477 0.035979
3 0.010067      4   0.14094 0.22819 0.036955
4 0.000100      6   0.12081 0.24832 0.038343
```

Since we want to minimize *xerror*, the best value of cp is 0.024609. We can train a new tree with this value and recompute the error rates.

For the tree with the best cp, we get 0.1 as the train error rate and 0.08 as the test error rate. This means that the accuracy of this model is 0.92 on the test set.