INTRODUCTION TO MACHINE LEARNING (NPFL054)

A template for Homework #2

Name: Pavel Mikuláš

School year: 3.

- **Provide answers for the exercises.**
- **For each exercise, your answer cannot exceed one sheet of paper.**

1.1 Multiple linear regression

Model summary:

```
(Intercept)    cylinders  displacement   horsepower       weight
  -17.95460     -0.48971      0.02398     -0.01818     -0.00671
acceleration        year    japanese     european
    0.07910     0.77703      2.63000      2.85323
```

Intercept:

Since most values will never be zero(acceleration, year, horsepower, weight, cylinders) this values has no meaning, because it is the expected **mpg** mean when all features are zero.

Cylinders:

This value would imply that the more cylinders a car have, the lower its mpg will be. However it has low statistical significance[Pr(>|t|) > 0.1] so we can't really deduce much from it.

Displacement:

The higher the engine displacement the more **mpg** a car should have.

Horsepower:

The higher the horsepower, the lower the **mpg.** But again this value has low statistical significance.

Weight:

The heavier the car, the lower the **mpg**. This value however small has very high statistical significance[Pr(>|t| < 0.001)] so this statement will likely be true.

Acceleration:

The higher the acceleration, the higher the **mpg**. This might have something to do with the fact, that lighter cars are likely to accelerate faster. However once again this value has very low statistical significance.

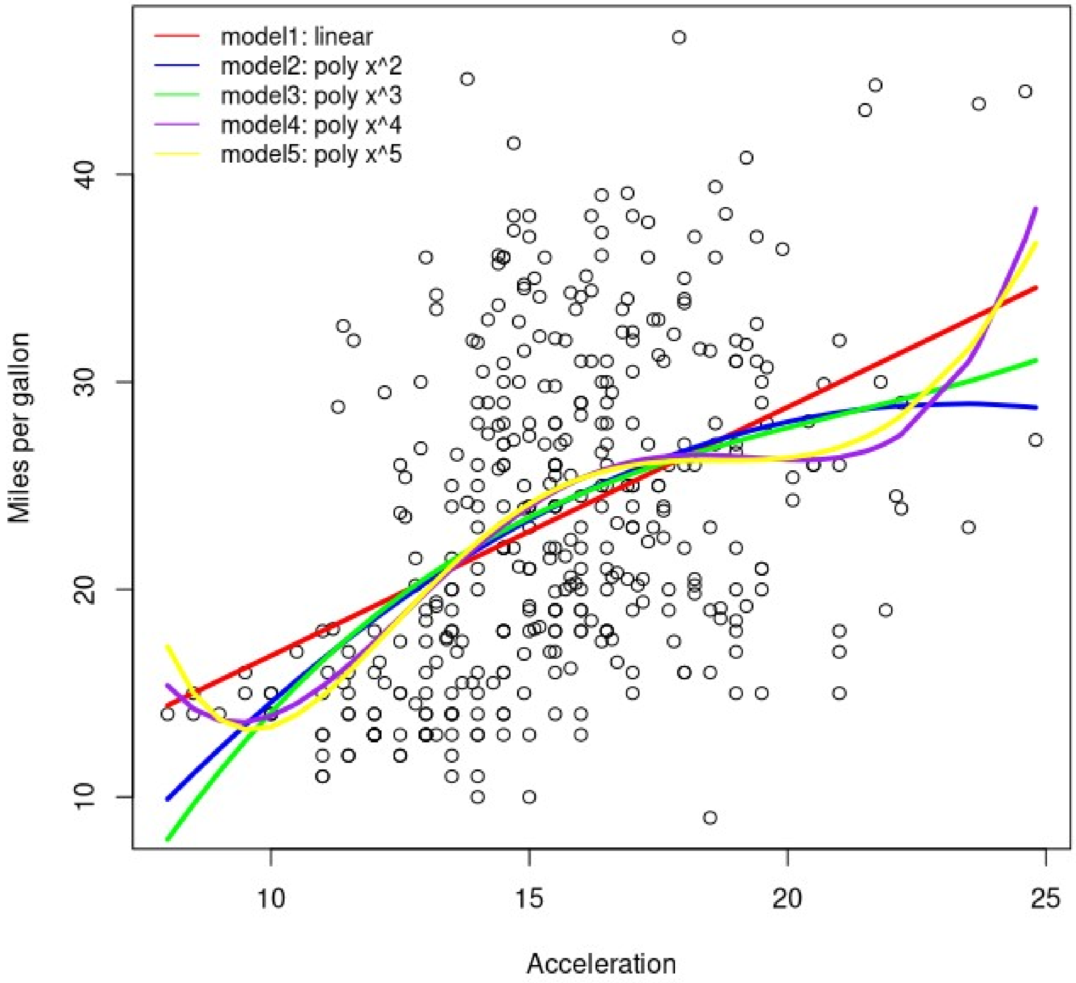
Year:

Newer cars will have higher **mpg.** This value also has high statistical significance. It also makes sense, because the goal at the time the database was created was to lower the **mpg.** And car industry was developing really fast at the time.

Japanese and european:

Those are binary encoding of the country of origin categorical feature. The values suggest that european and japanese cars have way higher **mpg** than american cars. This value also has very high statistical significance. This may explain why the japanese and european cars overtook the american car market.

1.2 Polynomial regression



Degree: 1, R^2: 0.179207050156255

Degree: 2, R^2: 0.193964011032177

Degree: 3, R^2: 0.195508000328511

Degree: 4, R^2: 0.213597901585272

Degree: 5, R^2: 0.214801832251077

2.1 Binary attribute `mpg01` and its entropy

We should use log with base 2, since we are computing entropy of binary data. It also leads to this nice result:

H = -(0*5 * log2(0.5) + 0*5 * log2(0.5)) = 1

That is because both values 1 and 0 are equally probable.

2.3 Trivial classifier accuracy

Trivial classifier accuracy: 44.30%

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

Logistic Regression train error: 9.27%

Accuracy	Precision	Recall	Specificity	F1_Measure
"93.67%"	"93.33%"	"95.45%"	"91.43%"	"94.38%"

Confusion matrix (rows – predicted, columns - true):

	0	1
0	32	2
1	3	42

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-18.638995	6.377691	-2.923	0.003472 **
cylinders	-0.144158	0.472386	-0.305	0.760236
displacement	0.011763	0.014253	0.825	0.409180
horsepower	-0.051460	0.025797	-1.995	0.046069 *
weight	-0.004943	0.001398	-3.536	0.000407 ***
acceleration	0.001715	0.145822	0.012	0.990614
year	0.470023	0.090448	5.197	2.03e-07 ***
japanese	1.629915	0.803351	2.029	0.042469 *
european	0.816048	0.750530	1.087	0.276906

Significance codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

The estimate here has a bit different meaning, it’s the ratio of probabilities that an event will occur given the value of predictor being 1 or 0.

Here I can provide a much more detailed summary of the model prediction than for the linear regression model, beacause I have more space and don’t have to explain every single parameter. But the arguments for my conclusion are the same as for the linear regression.

We can see, that using this algorithm the japanese and european values have much lower statistical significance than for the linear regression model. Acceleration seems almost irrelevat. But weight and year are still very significant.

The high std. errors are likely caused by us not scaling the data to normal distribution beforehand.

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

Threshold(0.5) (rows – predicted, columns – true for confusion matrices)

```
      0  1
  0 32  2
  1  3 42
Accuracy    Precision    Recall       Specificity  F1_Measure
"93.67%"    "93.33%"    "95.45%"    "91.43%"    "94.38%"
```

Threshold(0.1)

```
      0  1
  0 27  0
  1  8 44
Accuracy    Precision     Recall     Specificity  F1_Measure
"89.87%"    "84.62%"     "100%"    "77.14%"    "91.67%"
```

Threshold(0.9)

```
      0  1
  0 35  9
  1  0 35
Accuracy     Precision   Recall         Specificity  F1_Measure
"88.61%"     "100%"     "79.55%"      "100%"     "88.61%"
```

We can see that by decreasing the threshold we increase Recall while dramatically reducing Specificity and Precision. That should be pretty obvious as Recall is defined as TP/(TP + FN) and by lowering the threshold we classify more examples as positive and also less as negative. That means increasing TP and decreasing FN hence the overall quotient will be closer to 1. It goes the other way for Precision and Specificity as we are more likely to classify something as positive we get a higher number of FP and lower number of TN as we'll fail to classify some of them due to the threshold being lower.

When we increase the threshold we are being more strict with calssifying some example as being 1. Hence we'll be getting less FP leading to an increase in Precision and Specificity. But we'll also fail to calssify some examples correctly as 1 by being more strict. So we'll also have more FN and less TP leading to a decrease in Recall.

The choice of threshold depends on what we plan to use the classifier for. We may want to be sure that we only classify examples as 1 when we are really sure or we may not want to forget to classify some. But overall the best of the 3 choices of threshold is 0.5 as It has the best average performance as shown by the F1 meassure.

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

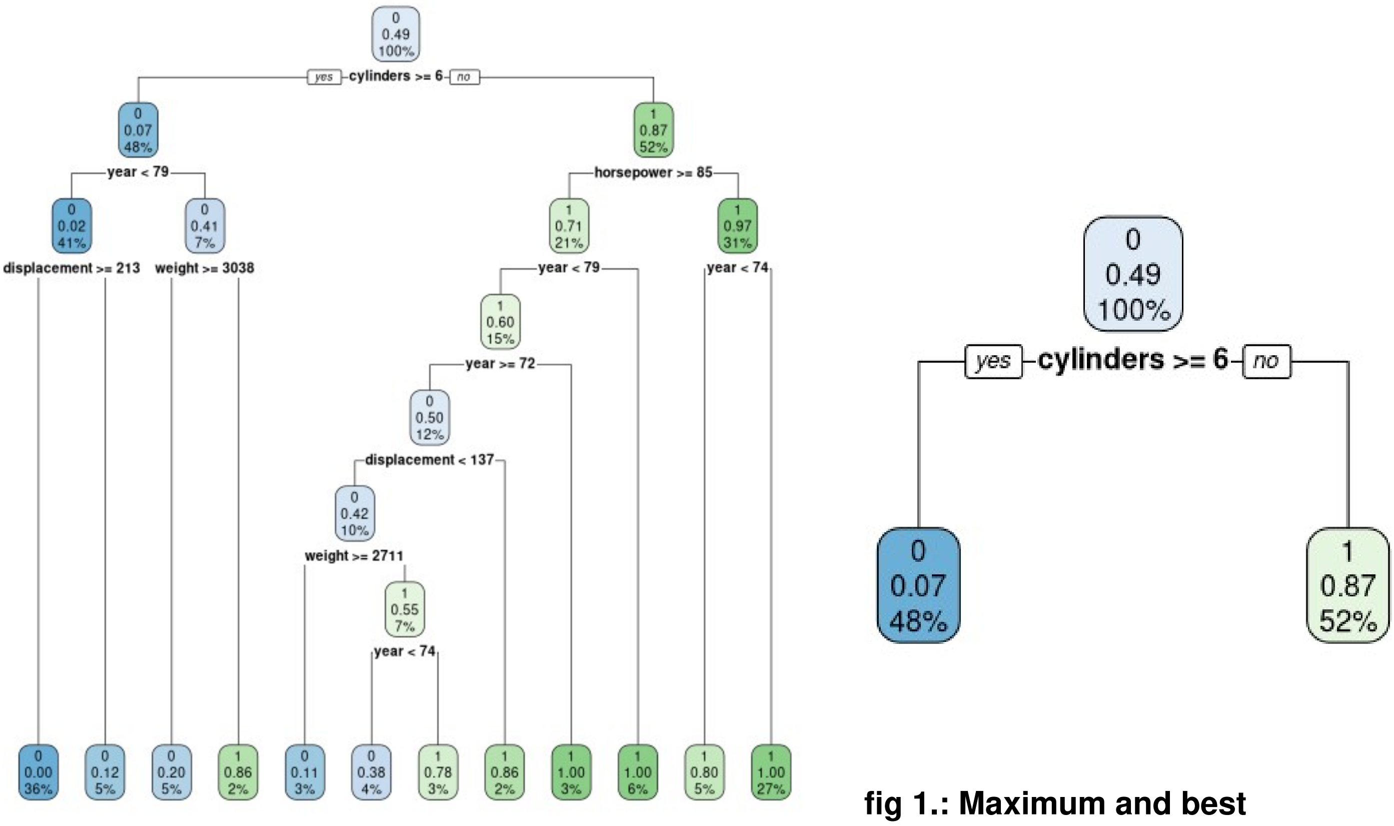


fig 1.: Maximum and best decision trees

Decision Tree algorithm train error: 5.75%

Decision Tree algorithm test error: 5.06%

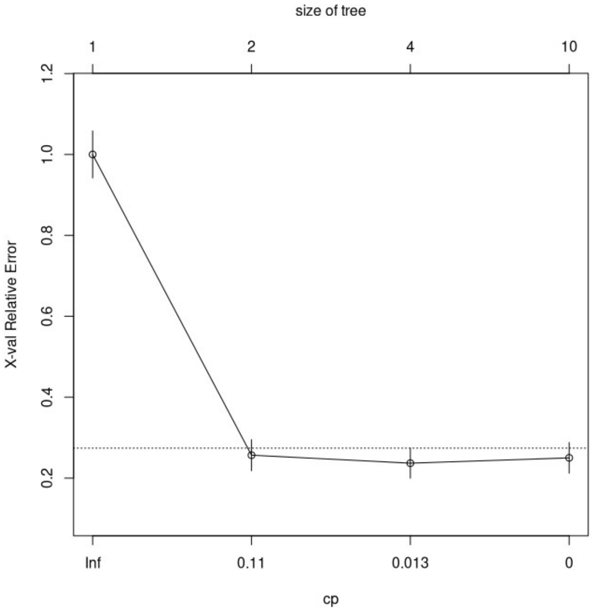
```
        CP    nsplit  rel error    xerror     xstd

1 0.782895     0     1.00000 1.00000 0.058173

2 0.016447     1     0.21711 0.25658 0.038441

3 0.010965     3     0.18421 0.23684 0.037134

4 0.000000     9     0.11842 0.25000 0.038014
```



We can now choose the optimal cp for our decision tree. We are interrested in the CP parameter with the lowest xerror. Because while the relative error decays with higher complexity, we are acutally overfitting if the xerror rises.

We get a best CP of around 0.013, but we have another model with less splits at 0.11 with it's mean below the stderr of the more complex model. So we should choose the simpler one instead. As it can be seen in the CP graph.

Decision Tree algorithm [BEST CP] train error: 10.54%

Decision Tree algorithm [BEST CP] test error: 6.33%

If we crossvalidate, we will get lower error rate for best CP tree. This example is misleading in this way.