Assignment 2

Hand in

- As for the previous assignment, hand in your source code and assignment report.

Part 1: lecture script

Go through the `02_model_training.r` script and understand what it achieves. You need this understanding to solve the following tasks. No hand in is required for this part.

Part 2: mtcars classification

Dataset

Use the standard mtcars dataset:

```
data(mtcars)
str(mtcars)
```

Task

Prepare an R script that fulfills the following requirements and answer the included questions (put results into the report that support your answers, like command outputs, figures, tables, etc):

1. We treat the feature `cyl` as the class of samples: imagine we want to predict `cyl` using all the other features. What are your thoughts about the dataset? (E.g.: class balance? Amount of samples? Ratio of samples to features? What do featureplots (pairwise, density, ...) show you? Would it be easy or hard to solve this problem, and why?) State your thoughts in detail!
2. We now want to predict `cyl` using only the features `hp` and `drat`, using a KNN classification model. Review the relation between `hp`, `drat` and `cyl`. What are your expectations now for solving this problem?
3. Train a KNN with k = 5 on the data and enable computing class probabilities in `trainControl()`. Don't use (repeated) split(s) or cross validation, just do a "plain" training (`trainControl(method="none", ...)`) using `Kappa` as classification metric:
    - Hint: ensure that the class variable you handle to `train` and similar functions is a) a factor and b) classes don't have names that start with numbers. Reason: with many ML functions, factor target variables imply classification, while numeric target variables imply regression. Class names starting with numbers are problematic in some cases. Use e.g. `factor(paste0('class', VARIABLE))` to obtain a factor with valid classnames.
    - Optional: you can make training reproducible as well using the `seeds` parameter in `trainControl`.
4. Use the model to predict `cyl` from only `hp` and `drat` and generate a confusionmatrix (`confusionMatrix(predict(model, newData), actualValues)`).
    - What is the resulting TPR, TNR, FNR and FPR? State those as equations and numbers in your report.
    - What is the resulting Kappa?
    - Do you think these results are representative? Why/why not? (Hint: amount of samples!)
5. Now use the same model to predict class probabilities for all samples:
    - Use the predicted class probabilities to create one ROC curve per class and state those along with their AUC.
    - From confusionmatrix and ROC curves: is there one class harder to predict correctly than others?
6. Now train and evaluate two further KNN models with k = 1 and k = 3, and another model using a linear discriminant analysis model `lda`. Compare them using some of the error measures used above. What difference do you observe between models? What do you derive from your results?

Part 3: Diamonds Regression

Dataset

Use the diamonds dataset delivered with the ggplot2 library:

```
library(ggplot2)
data(diamonds)
```

Task

Goal is to predict the diamond `price` from features `carat`, `depth`, `table`, and `x`:

- What are your expectations from graphical data analysis? Are there features that might be useful?
  - Hint: you cannot plot classes as colors anymore, but relations between features and your target variable, e.g. using pairwise plots, might still be visible.
- Use a KNN regression model and try different k, and a `linear model` (`lm`) or `generalized linear model` (`glm`), again with using `trainControl(method="none", ...)`:
  - Use root mean squared error `RMSE` as performance metric.
  - Hint: ensure a numeric target variable is handled to `train`.
- Predict outcomes for your data and compute the RMSE and mean absolute error (MAE):
  - Hint: RMSE = `sqrt(mean((predicted - actual)**2))`
  - Hint: MAE = `mean(abs(predicted - actual))`
- What is the error for these models? To give meaning to this error, state how big the error is compared to the mean/median and standard deviation/mad of the diamond price. What are your thoughts about these results ("best" model)?

Part 4: Problem of Using Same Data for Training and Evaluation

Think about what the problem is with using the same data to train and evaluate a model. KNN with k=1 is an extreme examples for this.

- Explain the problem in your own words.
- What ways can you think of to prevent this problem?