Assignment 3

Hand in

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

Hints

- If you get stuck somewhere: try discussing your ideas/expectation/problems with your colleagues! This assignment is designed to make you think even harder about your data and your results than the previous assignments and likely is the most complex and extensive one in the course.
- If you run into RAM problems try reducing parallelization (e.g. 3 cores might be maximum possible on some laptops).
- This is the last assignment with lots of subtasks and hints, later assignments are going to be more "goal oriented".

Part 1: some more mtcars

Your task is use the `mtcars` dataset to predict `cyl` (as factor) from `wt` and `drat` only. Some colleague/friend/classmate told you that using a radial/gaussian/RBF Kernel SVM with `sigma=30` and `C=10` will give you perfect results on this task. Your task now is:

- What would your performance expectations be from graphical data analysis?
- Reproduce what your colleague said with training this model on the complete dataset with `trainControl(method='none', ...)`. Evaluate the model using the complete dataset used for training. Which results do you obtain, why might they be problematic, and how do they fit what you saw in graphical data analysis? How does the amount of samples play a role in this?
  - Hint: Kappa as classification metric would be a good choice.
  - State a figure plot and confusion matrix/confusion matrix visualization to support your explanation.
- Now properly evaluate different parametrizations for this model using a parameter grid search with leave one out cross validation. For this part of the assignment, using the complete dataset in cross validation (without having an additional held-back test set) is OK - because we would have too less samples to experiment with otherwise.
  - Hint: don't just copy the parameter grid from the slides: this one would be suboptimal for this task. An exponential parameter grid might be more useful, e.g. `3^(lower:upper)`.
  - State the same plots as before, and additionally a) the TPR and TNR and b) a fitness landscape of Kappa over C and sigma to support your explanation.
  - What is the difference to what your colleague told you, and how is this caused?

Part 2: some more diamonds

Your task is to use the `ggplot2` `diamonds` dataset predict `price` from `x` only. Your are limited to using a regular linear model.

- What you you conclude from graphical data analysis?
- Train a generalized linear model `glm` using 10CV with 20 repeats. For this part of the assigment, not having an additional held-back test set and performing CV on all data is OK. When handing data to the `x` variable of `caret::train`, ensure that you hand it as a data frame (subsetting a data frame to a single variable can lead to it becoming a vector instead of a data frame, which will trigger errors in caret. You can e.g. use `x=data.frame(YOUR_X), ...` to obtain a data frame with 1 column from you vector):
  - Use the model to predict the target variable for all samples.
  - State the distribution of the absolute error (e.g. `summary(abs(predicted-observed))`).
  - Visualize the predicted value over observed value in a scatterplot and add the ideal fit as diagonal line for reference.
  - What do you see/derive from this plot?
- Now train another linear model, but use a "trick" before training/evaluating the model: apply a logarithm to the target variable before.
  - Do graphical feature analysis before creating the model: what is different?
  - Again use 10CV with 20 repeats, then use the model to predict the target variable for all samples.
  - Visualize the predicted values over observed values and add the ideal fit as line for reference to the plots: what is different to the situation before?
  - After predicting values with the model, reverse the logarithm and compute the error of those reversed values to the original target variable (again give a `summary(...)` and visualize it).

- Compare the errors of both approaches and state your thoughts!

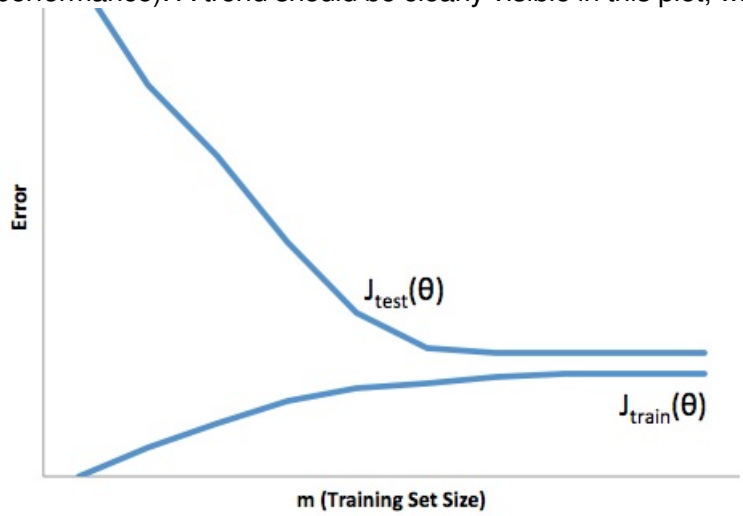
Part 3: some more... segmentation!

Use features 6 to 20 from `caret::segmentationData` data set:

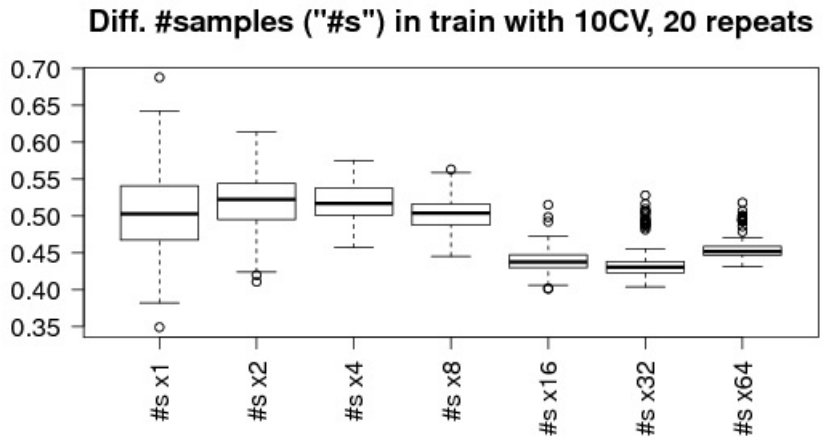
```
library(caret)
data(segmentationData)
str(segmentationData)
```

Task is to predict the `Class` from these features. Split data into a 2% training and 98% test partition with `set.seed(123456)`. Use the training partition with 10CV with 20 repeats to train an LDA (`lda`) model (takes a while to compute). Compute the confusion matrix and Kappa for both the train and test partition. How do train and test errors differ? Keep the model object and training/test error/performance values for later comparison.

- You might encounter warnings like `Warning: In lda.default(x, grouping, ...) : variables are collinear` or `warning: prediction from a rank-deficient fit may be misleading` when training the model. You can ignore these warnings for now.
- Repeat the above for increased partition sizes (always using the same seed for partitioning!). Double the size of the training partition (4%, 8%, ...) until the gap between training and test error closes. Which train/test partition size do you need to close this gap? Keep all created models and metric values for later comparison.
- Visualize the training and test performance for all created models in one plot (using the stored train and test set performance). A trend should be clearly visible in this plot, which might look like this:



- Visualize the CV performance over all models in a second plot (e.g. boxplot). Again, a trend should be clearly visible here, which might look similar to this:



- Given those models, which training and test partition size would you chose for your application scenario, and why? For the chosen model report a) the confusion matrix, TPR, TNR, FPR, and FNR, and b) visualize the ROC curve and state the AUC.
- Think about what it means if your test data set becomes very small: what could be resulting implications?

Part 4: even more segmentation

This is where we combine all previous techniques: data partitioning, parameter grid search, model training+evaluation

and model selection in one classification task. Goal is to obtain the best suited model for a classification task.

Use features 4 to 61 from `caret::segmentationData` data set:

```
library(caret)
data(segmentationData)
str(segmentationData)
```

The task is to predict the `Class` from these features. Perform a 80/20 randomized train/test split, then use 10CV with 5 repeats (20 would be better, but for the sake of runtime, 5 is OK) to train different models (use the AUC `ROC` as metric to be optimized). For models using hyperparameters do an appropriate parameter grid search and visualize their performance over parameter values. You are free to try and experiment with any type of model, but try to understand what their concept is (see hints below). Keep in mind that some models are only applicable to regression (and some only to 2-class-classification, so they might work here, but not with other problems).

- Compare model performances (visualization, e.g. `bwplot(list-of-models)`).
- Compare the CV confusion matrixes, ROC, and AUC of those models. You can e.g. plot the test ROC curve of all models into one figure to allow for an easy comparison.
- Based on those results, state which model you would use in a real life scenario in an application and why. On this model apply the test set and report a) the confusion matrix, and the TPR, TNR, FPR, and FNR, and b) the ROC curve and AUC.
- Hint: these `models(parameters)` should be easy: `glm()`, `lda()`, `knn(k)`, `linearSvm(C)`. Trying other models is appreciated!

Part 5: analyze paper: "Guidelines for Best Practices in Biometrics Research"

The paper "Guidelines for Best Practices in Biometrics Research" by Jain et al. gives guidelines for biometric research. The stated guidelines are useful not only for biometric research, but for many other fields of science and application dealing with data and data based evaluations. Some of the concepts have already been mentioned in the lecture, others are a complement to our class.

- Task: read the paper, understand and shortly summarize each of the guidelines given.