Homework Assignment 1

Data Science and Machine Learning 1 - CEU 2018

Due date: 2018-01-29 8:00 a.m.

General information

The due date of this homework assignment is 2018-01-29 8:00 a.m. You are required to submit two files to Moodle: an `.Rmd` file and the rendered `.html` file with your solutions. Please also include code to the `.html` as well (use the default settings).

Please give short (2-3 sentences) interpretations, explanations to your answers, not only the program code and outputs.

Grading: 13 points can be earned in total (5 each for the first two and 3 for the third). However, 10 points count as 100%. You can earn more than 100%, that is, giving a perfect answer to all questions means 130%. Solving first two and not solving the third is 100% and earning 2 points on the first, 5 on the second and 3 on the third is also 100%.

1. Model selection with a validation set (5 points)

In class we saw the following process: we tuned different models with cross validation on exactly the same folds and we selected the best hyperparameters and also between models based on that procedure. Finally, we can evaluate the model on an independent test set.

Instead, we can choose to cut the data into three parts: a training set that we use to estimate models, a validation set that we use to choose between models and a test set that we only use to assess performance of the chosen model.

Take the real estate dataset used in class and make `log_price` your target variable.

```r
library(data.table)
library(caret)

data <- fread("../../data/king_county_house_prices/kc_house_data.csv")

data[, `:=`(floors = as.numeric(floors), zipcode = factor(zipcode))]
data[, log_price := log(price)]

data[, c("id", "date", "sqft_living15", "sqft_lot15", "price") := NULL]

set.seed(1234)
```

a) Using `createDataPartition`, cut your data into three parts: 50% should be your training data, 25% each your validation and test sets (hint: cut data into two parts, then further cut one part into two).

b) Train three models on the training data via `caret`, without cross validation (`method = "none"`):

- a linear model `lm` with only using `sqft_living` as a predictor (a simple benchmark)
- a linear model `lm` using all available features
- a regression tree (`rpart`) with `cp = 0.0001` (the tune grid should be a dataframe with one column `cp` and one row with value 0.0001)

For `lm` models, the `tuneGrid` argument should not be specified.

```
# fill in the missing details
train_control <- trainControl(method = "none")

simple_linear_fit <- train(...)

linear_fit <- train(...)

rpart_fit <- train(...)
```

c) Compare your models on the validation set and choose the one with the best performance (using RMSE). Use `predict.train` for prediction just like we used `predict` in class.

```
RMSE <- function(x, true_x) sqrt(mean((x - true_x)^2))

simple_linear_rmse <- RMSE(...)
linear_rmse <- RMSE(...)
rpart_rmse <- RMSE(...)
```

d) Evaluate the final model on the test set. Why is it important to have this final set of observations set aside for evaluation? (Hint: think about what we used the validation set for.)

```
final_performance_measure <- RMSE(...)
```

e) Do you think it makes more sense to use this method rather than the one used in class? What can be advantages or disadvantages of one or the other?

2. Predicting developer salaries (5 points)

In this exercise the task is to predict developer salaries using the Stackoverflow Annual Developer Survey 2017. The dataset is downloaded from Kaggle. For simplicity I excluded some columns and prepared some transformations for you.

```
data <- fread("../../data/stackoverflow2017/survey_results_public_selected.csv")

data <- data[!is.na(Salary) & Salary > 0]
data <- data[complete.cases(data)]
data <- data[, Gender := ifelse(Gender == "Male", "Male",
                         ifelse(Gender == "Female", "Female", "Other"))]
large_countries <- data[, .N, by = "Country"][N > 60][["Country"]]
data <- data[, Country := ifelse(Country %in% large_countries, Country, "Other")]
```

a) Describe what the data cleansing steps mean.
b) Using graphs, find at least two interesting features that can contribute to understanding developer salaries.
c) Create a training and a test set assigning 70% to the training set and 30% as the test set.
d) Using `caret` train at least two predictive models to predict **the logarithm** of `Salary` (they can be of the same family but with different hyperparameters or they can be of different families like we used `lm` and `rpart` in the first exercise). Make sure **NOT** to include `Salary` as a predictor variable. Also, just before calling `train`, remember to use `set.seed`.

Then:

- choose the best model based on cross-validation estimation on the training set
- evaluate its performance on the test set

e) Compare the true and predicted values of the test set on a graph. How do you evaluate the model fit based on this graph?

3. Leave-one-out cross validation (3 points)

Leave-one-out cross validation (LOOCV) is a special case of k-fold cross validation where k equals the number of points in the sample.

a) Name a disadvantage of this method compared to using a moderate value (say, 10) for k?
b) Why do you think it can still make sense to compute this measure? In what way can this measure be closer to the “real” performance of the model?

Take the `titanic` dataset.

```r
library(titanic)
library(data.table)

data_train <- data.table(titanic_train)
# recode Survived to factor - needed for binary prediction
data_train[, Survived := factor(ifelse(Survived == 1, "survived", "died"))]
```

c) You can implement LOOCV with `caret` by setting an option in `trainControl`: `method = "loocv"`. and use a simple logit model `glm` for prediction.

- In `caret`, you can use it via `method = "glm"`
- include `classProbs = TRUE` in `trainControl` to let `train` know that you are predicting a binary outcome

Implement both an LOOCV and a 10-fold cross-validation estimation using only `Fare` and `Sex` as predictor features.

d) Compare the accuracy of the model estimated by the two resampling methods via `summary(fitted_model$resample)`. Accuracy is the share of cases predicted correctly.

- How large are the means?
- How do other quantiles look like? Why are quantiles of the accuracy measures of LOOCV so extreme (either 0 or 1)?