

Data Analytics – Exercises

(Week 09)

In these exercises, you will learn:

- to perform classification analyses using classification trees (CTs).
- to perform classification analyses using random forest (RF) classifiers.

In the data analytics process model, these exercises cover part of the steps “Statistical data analysis and/or Modeling” and “Evaluation & Interpretation” (see figure 1). Results of the exercises must be uploaded as separate files (**no .zip files!**) by each student on Moodle. Details on how to submit the results can be found in the tasks below.

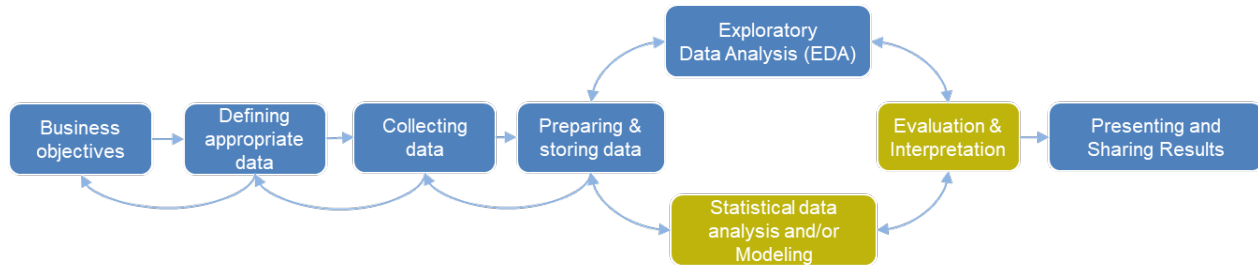


Figure 1: Data analytics process model (see slides of week 01)

Task 1

In this exercise, you will learn to perform classification analyses using a classification tree and a random forest classifier based on the titanic data. The tasks are:

- Run the Jupyter notebook '[classification_analysis_titanic.ipynb](#)' step by step and try to find out, what the Python code does.
- Go to the section 'Classification Tree' -> 'Create train and test samples ...'. Change the parameter `test_size` from 0.20 to 0.50. This will change the proportion of observations (passengers in this case) used for training and testing from 80/20 to 50/50. Compare the accuracy & recall from the classification report of the model based on the 80/20 samples with the one based on the 50/50 samples. In the Jupyter notebook, try to explain the differences (if there are any).
- In the section 'Fit the classification tree model and make predictions' change the `max_depth` parameter and run the Jupyter notebook again. Look at the text representation and graphical output of the tree (you can change the fontsize of the graphic to a smaller value). In the Jupyter notebook, state what you can see.
- In the section 'Random Forest Classifier' -> 'Show feature importance', look at the feature importances in the bar chart. Then go to the section 'Random Forest Classifier' -> 'Create train and test samples ...'. Remove the variables 'Age' and

- 'Sex_male' from the train and test samples and run the Jupyter notebook again. In the Jupyter notebook, state, which feature is now the most important one.
- e) Fit models with/without the variables 'Age' and 'Sex_male' and state how the ROC curve and AUC value change.

To be submitted on Moodle:

- The Jupyter notebook as html-file '[classification_analysis_titanic.html](#)' with the changes and short explanations according to b), c), d) and e)

Task 2

In this exercise, you will perform your own classification analyses using a classification tree and a random forest classifier based on the supermarkets data. In detail, you will create classification models which are able to predict the brand of a supermarket based on municipality-level and other characteristics. Use the Jupyter notebook from task 1 as template to solve the tasks.

- a) Create a new Jupyter notebook '[classification_analysis_supermarkets.ipynb](#)'.
- b) Use the following steps to create a classification tree:
- Load the required Python libraries.
 - Import the supermarkets data 'supermarkets_data_enriched.csv' available on Moodle to a data frame named 'df_supermarkets'. We need the following variables:

	id	bfs_name	bfs_number	lat	lon	brand	pop	pop_dens	frg_pct	emp
0	33126515	Schänis	3315	47.155616	9.037915	SPAR	3876	97.142857	13.054696	1408.0
1	280130028	Schänis	3315	47.155492	9.039666	ALDI	3876	97.142857	13.054696	1408.0

- Remove all missing values from the data frame.
 - Create a subset named 'df_sub' with only 'Migros' and 'Volg' as brands.
- ```
df_sub = df_supermarkets.loc[df_supermarkets['brand'].isin(['Migros', 'Volg'])]
```
- Create train/test samples (X\_train, y\_train, X\_test, y\_test) based on df\_sub.
  - The X\_train and X\_test must contain: lat, lon, pop, pop\_dens, frg\_pct, emp.
  - The y\_train and y\_test must contain the target variable, which is: brand.
  - Fit the classification tree model and make predictions.
  - Print a text representation of the classification tree.
  - Visualize the classification tree.
  - Print the confusion matrix and classification report of the model.
  - Print the ROC curve and AUC of the model.
- c) Use df\_sub from b) and the following steps to create a random forest classifier:
- Create train/test samples (X2\_train, y2\_train, X2\_test, y2\_test).
  - Fit the random forest classifier and make model predictions.
  - Show the confusion matrix and classification report.

- Show the feature importance in a bar chart.
- Print the ROC curve and AUC of the model.

**To be submitted on Moodle:**

- The Jupyter notebook as html-file '[classification\\_analysis\\_supermarkets.html](#)'.