**Seminar 3**

**Put the data files in the same folder as your Jupyter Notebook files if there are data files.**

**We learn coding by imitation. Therefore, we start by copying example codes and run them. Based on the outputs, comments, and the codes, we understand what the codes need and what the codes produce. Then we can modify the codes and apply them to new data for solving new problems.**

1. **Try data pipeline: Start a new Jupyter Notebook and copy the following codes one (line/part) by one, followed by press the keys Shift + Enter.**

# Data Science Pipeline

## Feature Creation

import numpy as np

from sklearn import datasets

from sklearn.model\_selection import train\_test\_split

cali = datasets.fetch\_california\_housing()

cali

print(cali.DESCR)

X = cali['data']

Y = cali['target']

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2,random\_state=123)

X\_train

# Standardize features by removing the mean and scaling to unit variance.

# StandardScaler transfers data to standard normally distributed data: Gaussian with zero mean and unit variance

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

## Dimensionality Reduction

import matplotlib.pyplot as plt

from sklearn import datasets

import numpy as np

iris = datasets.load\_iris()

iris

### PCA

from sklearn.decomposition import PCA

pca\_2c = PCA(n\_components=2)

X\_pca\_2c = pca\_2c.fit\_transform(iris.data)

X\_pca\_2c.shape

plt.scatter(X\_pca\_2c[:,0], X\_pca\_2c[:,1], c=iris.target)

plt.show()

pca\_2c.explained\_variance\_ratio\_.sum()

1. **Try data visualisation: Start a new Jupyter Notebook and copy the following codes one (line/part) by one, followed by press the keys Shift + Enter.**

# Chapter 6: Visualization (and EDA)

## Introducing the Basics of Matplotlib

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

# This is important to show figures in Jupyter Notebook

### Bar graphs

from sklearn.datasets import load\_iris

iris = load\_iris()

iris

print(iris.DESCR)

iris.data

average = np.mean(iris.data, axis=0)

print(average)

range1 = np.arange(4)

range1

plt.title('Vertical bars')

plt.bar(range1,average)

plt.xticks(range1, iris.feature\_names)

plt.show()

plt.title('Horizontal bars')

plt.barh(range1,average)

plt.yticks(range1, iris.feature\_names)

plt.show()

## Introducing Seaborn

import seaborn as sns

import pandas as pd

from sklearn.datasets import load\_iris

iris = load\_iris()

iris\_df = pd.DataFrame(iris.data, columns=iris.feature\_names)

iris\_df['target'] = pd.DataFrame(iris.target)

iris\_df

chart = sns.FacetGrid(iris\_df, col="target")

chart.map(plt.scatter, "sepal length (cm)", "petal length (cm)")

chart = sns.FacetGrid(iris\_df, col="target")

chart.map(sns.histplot, "sepal length (cm)", kde=True, stat="density")

1. **Try case - firm credit - 1: Start a new Jupyter Notebook and copy the following codes one (line/part) by one, followed by press the keys Shift + Enter.**

# Case: Firm Credit

# The following exercises require the attached dataset "WM.csv", which is a sample of the credit status and characteristics of the firms in the UK in 2019 and 2020.

# Load necessary libraries

import warnings

warnings.simplefilter(action='ignore', category=FutureWarning)

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

# This is important to show seaborn figures in Jupyter Notebook

get\_ipython().run\_line\_magic('matplotlib', 'inline')

## Loading datasets

import pandas as pd

data\_all= pd.read\_csv("WM.csv")

data\_all.head()

## EDA

# Show the summary statistics

data\_all.describe()

data\_all.info()

### Show the histogram of 'Creditscore' grouped by 'Creditscoreindicator'

chart = sns.FacetGrid(data\_all, col='Creditscoreindicator')

chart.map(sns.histplot, 'Creditscore')

### Show the scatter plots between 'Creditscore' (Y) and 'ReturnonCapitalEmployed2020' (X) grouped by 'SMEindicator'

chart = sns.FacetGrid(data\_all, col='SMEindicator')

chart.map(plt.scatter, 'ReturnonCapitalEmployed2020', 'Creditscore')