**Seminar 8**

**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 machine learning models: Start a new Jupyter Notebook and copy the following codes one (line/part) by one, followed by press the keys Shift + Enter.**

# Machine Learning

import warnings

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

# ## An overview of unsupervised learning

import numpy as np

import matplotlib.pyplot as plt

from sklearn import datasets

N\_samples = 2000

dataset\_1 = datasets.make\_circles(n\_samples=N\_samples, noise=0.05, factor=0.3, random\_state=123)[0]

dataset\_1.shape

# Make a large circle containing a smaller circle in 2d.

# A simple toy dataset to visualize clustering and classification algorithms.

# noise: Standard noise is added to the data.

# factor: Scale factor between inner and outer circle in the range (0, 1).

plt.scatter(dataset\_1[:,0], dataset\_1[:,1], edgecolors='white')

plt.show()

from sklearn.cluster import KMeans

km\_1 = KMeans(n\_clusters=2)

labels\_1 = km\_1.fit(dataset\_1).labels\_

labels\_1

km\_1.cluster\_centers\_

plt.scatter(dataset\_1[:,0], dataset\_1[:,1], c=labels\_1, edgecolors='white')

plt.scatter(km\_1.cluster\_centers\_[:,0], km\_1.cluster\_centers\_[:,1], s=200, c=np.unique(labels\_1), edgecolors='black')

plt.show()

# DBSCAN - Density-Based Spatial Clustering of Applications with Noise.

# Finds core samples of high density and expands clusters from them.

# Good for data which contains clusters of similar density.

# eps: The maximum distance between two samples for one to be considered as in the neighborhood of the other.

from sklearn.cluster import DBSCAN

dbs\_1 = DBSCAN(eps=0.4)

labels\_1 = dbs\_1.fit(dataset\_1).labels\_

plt.scatter(dataset\_1[:,0], dataset\_1[:,1], c=labels\_1, edgecolors='white')

plt.show()

1. **Try Case – lending club - 2: Open and run the Jupyter Notebook “Seminar 5.2 Case\_LandingClub-1” in the previous seminar. Click the menu “Cell -> Run All” and wait for it finishes running. In the end of the file, copy the following codes one (line/part) by one, followed by press the keys Shift + Enter.**

# Exploratory Data Analysis (EDA)

## loan\_amnt

loans['loan\_amnt'].describe()

sns.histplot(loans['loan\_amnt'])

loans.groupby('charged\_off')['loan\_amnt'].describe()

## int\_rate

loans['int\_rate'].describe()

sns.histplot(loans['int\_rate'])

loans.groupby('charged\_off')['int\_rate'].describe()

## installment

loans['installment'].describe()

sns.histplot(loans['installment'])

loans.groupby('charged\_off')['installment'].describe()

## emp\_length

loans['emp\_length'].value\_counts().sort\_index()

sns.histplot(loans['emp\_length'])

## annual\_inc

loans['log\_annual\_inc'].describe()

sns.histplot(loans['log\_annual\_inc'])

loans.groupby('charged\_off')['log\_annual\_inc'].describe()

## dti

# Data Dictionary: "A ratio calculated using the borrower’s total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower’s self-reported monthly income."

loans['dti'].describe()

sns.histplot(loans['dti'])

loans.groupby('charged\_off')['dti'].describe()

## revol\_bal

# Data Dictionary: "Total credit revolving balance."

sns.histplot(loans['log\_revol\_bal'])

loans.groupby('charged\_off')['log\_revol\_bal'].describe()