Lab 3: Classification with K-means clustering and Hyper-parameter optimization using grid search.

Welcome to this lab, where you will explore one of the most popular machine learning mechanisms: classification. Your task is to implement a K-means clustering classifier and apply it to a new dataset. Additionally, you will do hyper-parameter optimization to find the optimal number of clusters.

In this lab, you will:

- Gain a deeper understanding of the K-means clustering algorithm.
- Implement the classifier from scratch, allowing you to grasp its inner workings.
- Apply the developed K-means clustering classifier on a fresh dataset.
- Observe how the classifier categorizes and clusters data points based on their similarities and patterns.
- Employ grid search, a powerful technique for hyperparameter tuning, to optimize the number of clusters.
- Classify new datapoints.

Load the dataset **Average rent in a rental apartment by year and region. T**his dataset can be found in the canvas assignment page.
Imagine you work in a real state company, and you are performing a study to classify the region's rent by the average income of their population for an upcoming campaign.

In this clustering analysis you will not divide the data in validation and train datasets think about the reason why you are not doing it in this case.

Task 1: Classification with K-means

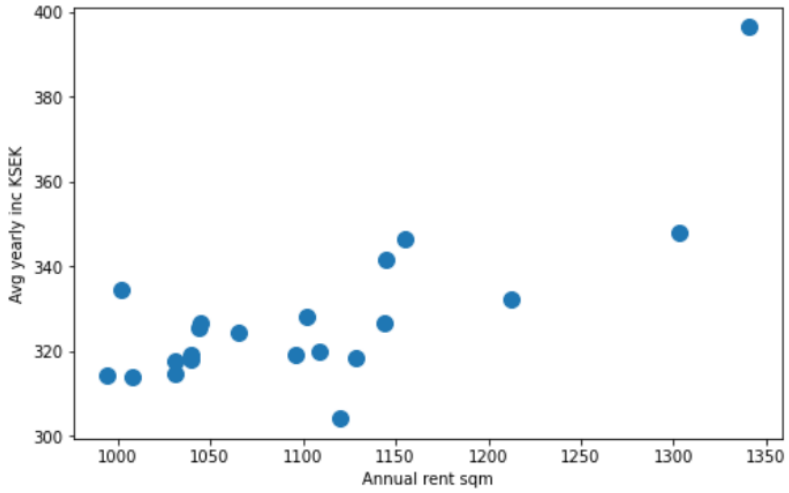
1.1 Load the new dataset found in the canvas page for this Lab: **rent_vs_inc.csv** the same way you did in the last lab. The first few rows of this dataset should look like this:

	year	region	Annual rent sqm	Avg yearly inc KSEK
0	2020	01 Stockholm county	1341	396.428571
1	2020	03 Uppsala county	1303	347.985714
2	2020	04 Södermanland county	1129	318.292857
3	2020	05 Östergötland county	1144	326.757143
4	2020	06 Jönköping county	1044	325.521429

What information can you get by just looking at the table?

1.2 Start by creating a scatter plot for your points.

Scatter plot example:



1.3 As mentioned in the introduction section you will create your K-means function from scratch. For information on how to perform K-means can be found in lecture 11:

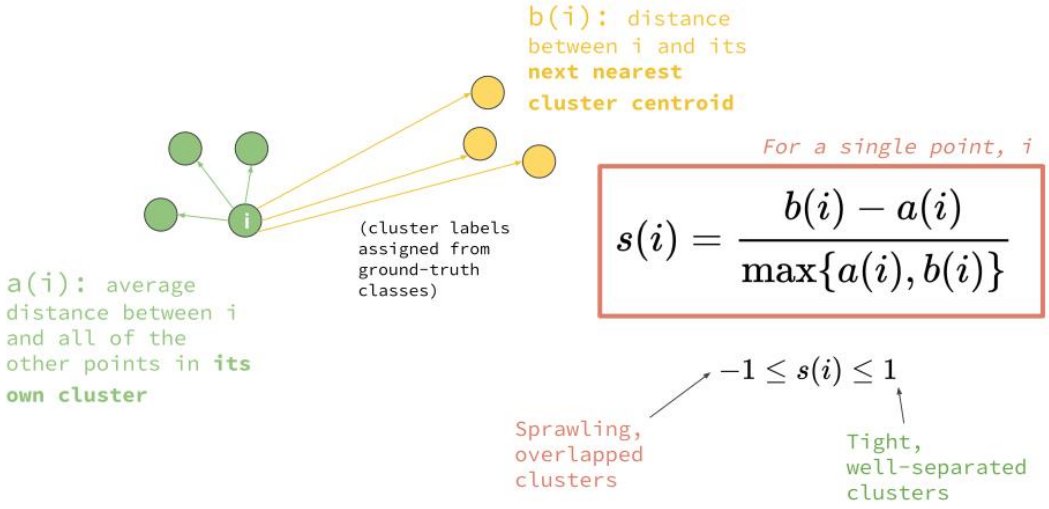
- Initialize the centroids with a starting number of clusters you consider correct, the centroids can be selected as a random point among your sample.
- Find which point belongs to which cluster by finding the closer centroid to every point (Euclidean distance).
- Calculate the mean point among each cluster to obtain the new centroid
- Repeat this process a N (around 10) number of iterations until the mean of the new cluster does not change from the previous iteration.
- Create once again a cluster plot with colors assigned for each cluster. For information on how to do this plot, check: https://pythonguides.com/matplotlib-scatter-plot-color/

Task 2: Hyper-parameter optimization

In this task you will implement grid search to find the optimal number of k-means clusters for this classification task.

2.1 In order to perform the hyper parameter optimization for the number of clusters you will use the silhouette score, for this score create a function that finds the silhouette coefficient for a grid of values between 1 and 10.

- For more information on the silhouette score and how it can be obtained with Scikit-learn, check page 247 of the ML book.



(https://www.platform.ai/post/the-silhouette-loss-function-metric-learning-with-a-cluster-validity-index)

- Where:
  - S(i) is the silhouette coefficient of the data point i.
  - a(i) is the average distance between i and all the other data points in the cluster to which i belong. (Intra-cluster distance)
  - b(i) is the average distance from i to all the points in the closest cluster to which i does not belong. (Inter-cluster distance)
- Create a function a(i) that calculates the average intra-cluster distance from any point i.
- Create a function that calculates the average inter-cluster distance to the closets cluster from datapoint i.
- Create a main function that iterates over all points in the dataset and calculates S(i)
- Perform the grid search obtaining the average S(i) for all points of each cluster value in the grid to get the cluster's silhouette coefficient. (Iterate over the cluster values between 1 to 10
- Graph the cluster's silhouette coefficient for each value in the grid to determine the optimal number of clusters for this exercise.

2.2 Create a scatter plot with the cluster colors for the newly found optimal number of clusters.

2.3 Finally evaluate how your created model predicts new values:

- For this task we will assume we have 3 unnamed regions with the following annual rent and average salary [**`1010, 320.12], [1258, 320], [980, 292.4]`**
- Find which cluster these data points belong to and plot them in the graph.
- Based on the cluster graph do you think your model successfully predicted the cluster for these values?

Optional advanced task

Can you figure out how to make an N-dimensional grid search optimizer, which can handle an arbitrary number of hyperparameters to optimize?