Questions

visualization: box plot
outlier is data point that is less than min val (Q1 − 1.5*IQR) and more than max val (Q3 + 1.5*IQR)

Q1) You want to identify outliers in a dataset with house sale prices in Enschede. What exploratory method would you suggest using for this purpose? Explain what the method does, and how to interpret its output. **(5 points)**

Answer: An outlier is an observation that is numerically distant from the rest of the data. A very straightforward method would be box map (or boxplot). It finds six bins categories to identify the lower and upper outliers. The definition of outliers is a function of a multiple of the inter-quartile range (IQR), the difference between the values for the 75 and 25 percentile. For example, outside 1.5 times the IQR above the upper quartile and below the lower quartile (Q1 - 1.5 * IQR or Q3 + 1.5 * IQR).

Q2) Cluster the following dataset into two groups using the k-means algorithm, considering 2 as the maximum number of iterations, samples 3 and 4 as initial cluster centers and $|x2 - x1| + |y2 - y1|$ as the distance function. Complete the following tables to report your solutions. **(9 points)**

	x	y	Final Clustering labels
1	7	3	1
2	1	6	2
3	9	6	1
4	4	2	2
5	3	9	2

	Initialization	First Iteration	Second Iteration
Cluster Center 1	(9,6)	(9,6)	(8,4.5)
Cluster Center 2	(4,2)	(3.75,5)	(2.6,5.6)

First Iter.

① — cluster 1 : (9−7) + (6−3) = 5
2 : (4−7) + (2−3) = 4 ✓

② — cluster 1 : (9−1) + (6−6) = 8
2 : (4−1) + (2−6) = 7 ✓

③ — cluster 1 ✓

④ — cluster 2 ✓

⑤ — cluster 1 : (9−3) + (6−9) = 9
cluster 2 : (4−3) + (2−9) = 8 ✓

∴ centroid of cluster 1 = (9, 6)
2 = $\left(\frac{7+1+4+3}{4}, \frac{3+6+2+9}{4}\right) = (3.75, 5)$

2nd Iter.

① ~~C1: (9−7)+(6−3)=5~~ ✓
C2: (3.75−7)+(5−3)=5.25

② C1: (9−1)+(6−6)=8
C2: (3.75−1)+(5−6)=3.75 ✓

③ C1: ✓

④ C1: (9−4)+(6−2)=9
C2: (3.75−4)+(5−2)=3.25 ✓

⑤ C1: (9−3)+(6−9)=9
C2: (3.75−3)+(5−9)=4.75 ✓

(this final answ →

centroid of c1: $\left(\frac{9+7}{2}, \frac{6+3}{2}\right) = (8, 4.5)$ *

c2: $\left(\frac{1+4+3}{3}, \frac{6+2+9}{3}\right) = (2.67, 5.67)$ *

Second iteration

	x	y	distance to cluster center 1 (9,6)	distance to cluster center 2 (3.75,5)	labels
1	7	3	5	5.25	1
2	1	6	8	3.75	2
3	9	6	0	6.25	1
4	4	2	9	3.25	2
5	3	9	9	4.75	2

Updating cluster centers:
Cluster Center 1: x = (7+9)/2=8, y = (3+6)/2=4.5
Cluster Center 2: x = (1+4+3)/3=2.6, y = (6+2+9)/3=5.6

- Input layer
  - 4 neurons (4 features)
- Hidden layer
  - can be many as we want
- Output layer
  - 3 neurons (3 classes)

Q3) We have applied an Artificial Neural Network (e.g., Multi-layer Perceptron) to a hypothetical dataset with 4 features (predictor variables) and obtained the following results.

Actual (True) Labels	1	1	1	2	2	2	2	3	3	3
Predicted Labels	1	2	1	1	2	2	2	3	3	1

What would be the possible architecture (i.e., number of layers and neurons in each layer) of the applied algorithm? Motivate and explain your answer. **(5 points)** $\frac{4+3}{2} = 3.5 \approx 4$

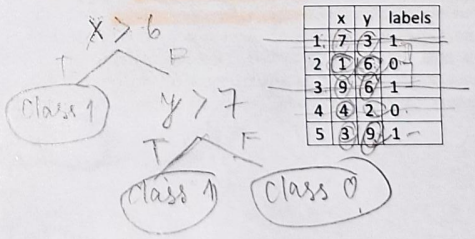
Answer: For the input layer we need one neuron per feature, so 4 neurons. For the hidden layers, there can be any number of layers and neurons. For the output layer the most common approach is to use as many neurons as there are outputs to the classification problem with softmax as the activation function. Based on the actual labels, we have 3 output classes, so 3 neurons in the output layer could be considered. We can determine the hidden layer structure experimentally. A simple structure with one hidden layer is sufficient for most of the problems. For the number of hidden neurons, a rule of thumb is the average of input and output neurons. Here this could be 4 neurons.

What is the overall classification accuracy? **(3 points)**

$\frac{7}{10} = 0.7$

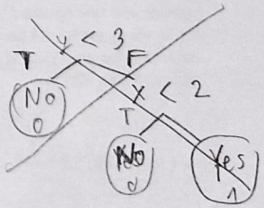
Overall accuracy= (number of correctly classified)/(total number of samples), 7/10=0.7(or 70%)

Q4) Given the dataset below, we build a simple Decision Tree (DT) with depth 2 and having x at the root node.



	x	y	labels
1	7	3	1
2	1	6	0
3	9	6	1
4	4	2	0
5	3	9	1

0 = No
1 = Yes



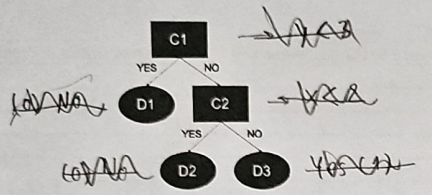
activation function.
help output be in specific range and help model capture non-linear relationship
(otherwise, model only know linear relationship)

advantages
- can use w/both classification & regression problem
- capture non-linear relationship

disadvantage
- complexity → more params to fine not easy to set up
- risk of overfitting

List and explain two advantages and two disadvantages of using Artificial Neural Network (ANN) for this problem. **(4 points)**

Answer: Advantages -Can capture complex patterns -Can model non-linear functions -Can incorporate continuous and discrete inputs/outputs -Can make predictions really fast
Disadvantages -Training data is required -Subjectivity of architecture -Hard to interpret (e.g. feature importance) -Risk of overfitting



What would be the C1, C2, D1, D2 and D3 in figure (a) for this classifier? (8 points)

Answer: For example C1: x>=5.5, D1: class 1, C2: y>=7.5, D2: class 1, D3: class 0

If we classify the test vector [x=2, y=3] with this classifier which class does this vector belong to? (3 points)

[x=2, y=3] will belong to class 0 class 0

Q5) We have a dataset that contains the daily average ground-level fine particle pollution (i.e., PM2.5) measurements collected from 1479 fixed air quality monitoring stations across China during 2014–2016. In this dataset each row denotes the reading from one station at a specific date. Here are its first few rows:

	station_id	station_name	latitude	longtitude	date	PM2.5
1	001001	haidianbeibuxinqu	40.09068	116.173553	1/1/2014	11.3
2	001001	haidianbeibuxinqu	40.09068	116.173553	1/2/2014	12.1
3	001001	haidianbeibuxinqu	40.09068	116.173553	1/3/2014	12.8

Let's imagine, we have also access to spatial datasets that provide several environmental, meteorological, and land-use variables at a fine spatial resolution across China (i.e., 1 km × 1 km).

Given these datasets, can we use a machine learning model to predict daily PM2.5 at 1 km × 1 km grid cells in the entire China? Explain which type of algorithm (i.e. classification or regression) we should use, and what the training data and input and output variables are. Please be as specific as possible. **(7 points)** link dataset by using location.

Answer: Yes, we can use regression algorithms (e.g., Random Forest or Neural Network). The training data can be obtained from the location of ground measurements. We can link the station PM2.5 measurements with the spatial dataset at these locations and build the training dataset. So the environmental, meteorological, and land-use variables would be the predictor variables and PM2.5 the target variable. A rule of thumb to divide the dataset into training and testing could be 80%-20%. The trained model can be used to estimate PM2.5 for locations where we do not have monitoring stations but we have access to the spatial datasets. The success of this methodology requires a good spatial distribution of monitoring stations across China.

- Yes, in this case, it is regression problem.
- training data →.

Q6) In hyperspectral imaging, also termed imaging spectroscopy, the sensor acquires a spectral vector with hundreds or thousands of elements from every pixel in a given scene. It is a powerful technique to identify different materials. Imagine we have a hyperspectral dataset composed of 200 observations (pixels) in which each observation consists of 250 spectral features.

What is the biggest challenge in Machine Learning modeling (e.g., classification and clustering) of this dataset? **(4 points)** many features and less observerations (too)

Answer: This dataset has too many features (i.e., the number of features is more than the number of observations). So we face the Curse of Dimensionality. If we have more features than observations, we have the risk of overfitting in supervised learning, so the performance will drop. Moreover, the observations become harder to cluster. When we have too many features, the observations appear at equal distances from each other. So defining meaningful clusters will be hard.

less observations, data (training data) → overfitting (ML capture too specific pattern)

What could be a possible solution? **(2 points)** → reduce dimensionality (e.g. feature selection / feature extraction, ~~SVM~~, PCA)

A dimensionality reduction technique (i.e., feature extraction or selection) can help us in this situation. For example, PCA is a technique for reducing the dimensionality. It projects the data along the directions where there is the largest variation of data. In this case by reducing the dimension to 10, we can reach a good balance between the number of observations and the number of features: 200/10=20 ???

* too many features
- increase computation time

feature selection vs. feature extraction
- feature selection: select subsets of original features
- feature extraction: create new set of features by transforming the original features → ex. PCA