COMP4431 Artificial Intelligence

Machine Learning

Raymond Pang
Department of Computing
The Hong Kong Polytechnic University

Quiz Arrangement (held on 6 Mar)

- Quiz is open book and notes
- Scope related to **Lecture 4 and 5**
- Use of mobile phone, Internet and other electronic devices are NOT ALLOWED!!
- Quiz will be held at the end of the lecture session
- Quiz time is **15 mins**
- Lecture will end at 7:45pm, Quiz starts at **7:55pm**

Agenda

- Machine learning
- Basics of Classification
- Decision Tree
- Clustering

Machine Learning

- Always mixed with AI and Deep Learning nowadays
- Definition
  - A branch of artificial intelligence
  - Computational model that can learn from experience in the environment with respect to improve the performance of some tasks
- According to the feedback from environment
  - Unsupervised learning
  - Supervised learning
  - Reinforcement learning

Supervised Learning

- A learning task involves a set of input and a set of desired output.
  - Usually refers as training dataset, which requires intensive labeling of input to give the output
- The set of possible relationships between input and output variables is known as the **model**.
- A model can be numerical functions, symbolic rules, decision trees and artificial neural nets.

Source: ASCE Global Center of Excellence in Computing

Supervised Learning

- The learning algorithm attempts to find the best hypothesis that maps input to output using “feedback”.
- Feedback consists of a set of points (training data) for which values of input and output variables are known.
- Since training data are used, this is supervised learning.

Source: ASCE Global Center of Excellence in Computing

Unsupervised Learning

- In unsupervised learning, output variables are not known.
- Unsupervised learning algorithms identify trends in data and make inferences without knowledge of correct answers.

Source: ASCE Global Center of Excellence in Computing

Reinforcement Learning

- Reinforcement learning is concerned with how software ought to initiate actions in an environment so as to maximize some notion of long-term reward.
- Reinforcement learning algorithms identify ways to maps states of the world to the actions the software ought to take in those states.
- Reinforcement learning may involve learning from mistakes.

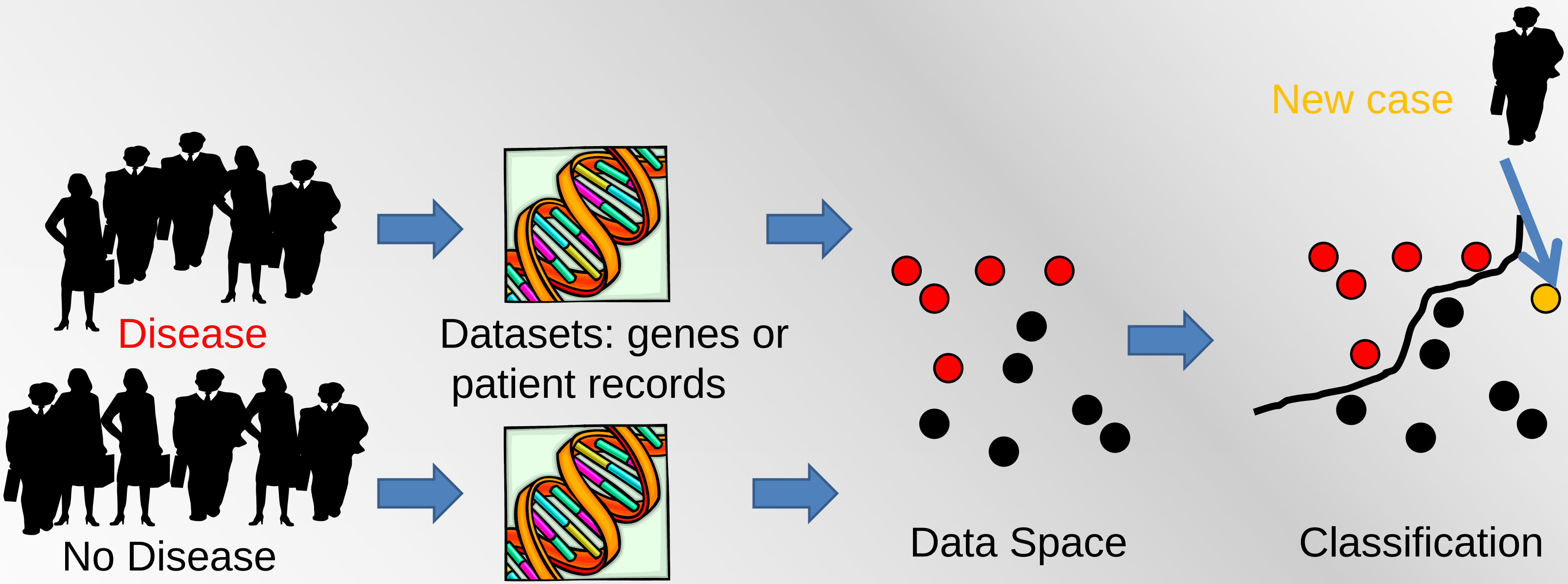
Source: ASCE Global Center of Excellence in Computing

Classification

Basics of Classification

- Trying to predict a ‘class’ (categorical data) given historical training data
- Algorithms use the ‘class’ labels in the training as a teacher
- Classifier then predict the ‘class’ of an unseen sample
  - It is important the classifier can generalize what it learned and
  - apply correctly to unseen samples
- Classifiers can be evaluated by its accuracy
  - False negative and false positive rates are vital

Basics of Classification



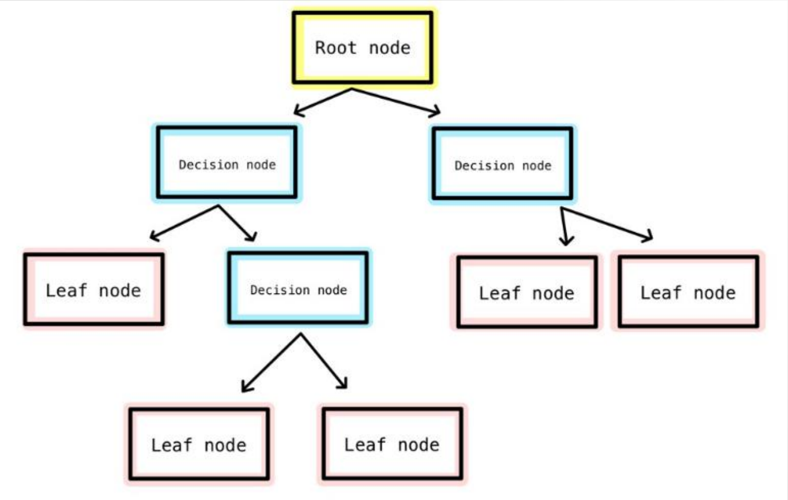
Basics of Classification

- Classification works on the basis we have data labelled with class information
  - Class information usually comes from an expert
  - Usually in medical sciences class information is simply defined as "controls" and "diseased"
  - This is attached to the independent variables and forms a record
  - Hopefully, we have many records in a data set, with equal numbers in each category
- This data is then used to train a classifier
  - Some data is kept back as test/validation data!
  - And provides an accuracy %
- Imbalanced classes cause problems (e.g. in fraud detection we may have few fraudsters but many non-fraud data points)

Decision Tree

Decision Trees

- Decision trees are powerful and popular classifiers while easy to build.
- Decision trees represent rules, which can be understood by humans
  - A flow-chart-like tree structure
  - Internal node denotes a test on an attribute
  - Branch represents an outcome of the test
  - Leaf nodes represent class labels or class distribution



Decision Trees: Illustrated

- You want to estimate an individual's credit risk
- Available knowledge / Attribute
  - Credit history,
  - Debt,
  - Collateral,
  - Income

Decision Trees: Illustrated

- Thus, it is common we have a table collecting all different cases from historical records

NO.	RISK	CREDIT HISTORY	DEBT	COLLATERAL	INCOME
1.	high	bad	high	none	\$0 to \$15k
2.	high	unknown	high	none	\$15 to \$35k
3.	moderate	unknown	low	none	\$15 to \$35k
4.	high	unknown	low	none	\$0 to \$15k
5.	low	unknown	low	none	over \$35k
6.	low	unknown	low	adequate	over \$35k
7.	high	bad	low	none	\$0 to \$15k
8.	moderate	bad	low	adequate	over \$35k
9.	low	good	low	none	over \$35k
10.	low	good	high	adequate	over \$35k
11.	high	good	high	none	\$0 to \$15k
12.	moderate	good	high	none	\$15 to \$35k
13.	low	good	high	none	over \$35k
14.	high	bad	high	none	\$15 to \$35k

https://www.geeksforgeeks.org/decision-tree-introduction-example/

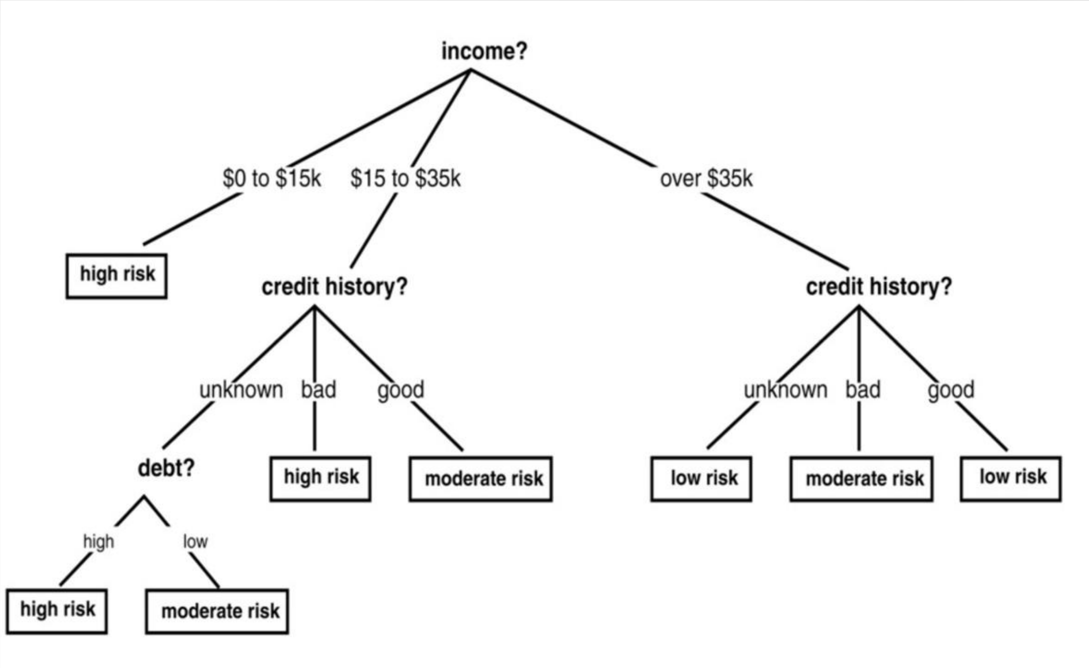
Decision Trees: Illustrated

- For a new comer, who's income is $0-15K, Credit history is bad, but debt is low and have adequate collateral…

Income	Credit History	Debt	Collateral	Risk
$0-15K	bad	low	adequate	???

- A classification problem
- Generalizing the learned rule to new examples

A Sample Tree



(1) Which to start? (root)

(2) Which node to proceed?

(3) When to stop/ come to conclusion?

Decision trees classify instances or examples by starting at the root of the tree and moving through it until a leaf node.

ID3 Heuristic

- ID3 splits attributes based on their ***entropy***.
- Entropy is the measure of disinformation…

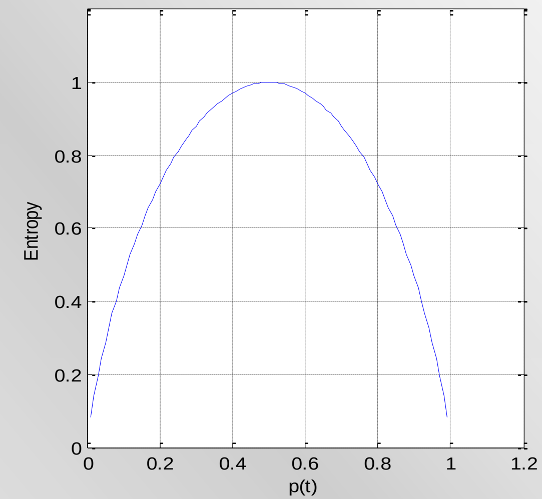
- Selection of an attribute to test at each node - choosing the most useful (Greedy) attribute for classifying examples.

Entropy

- If there is a sequence of M symbols $\{s_1\ s_2\ \dots\ s_M\}$, and the symbols are independent, the entropy H is defined as

$$H=\sum_{1}^{M} P(s_i)\log_2(1/P(s_i)) \quad \text{bits}$$

or $$H=-\sum_{1}^{M} P(s_i)\log_2(P(s_i)) \quad \text{bits}$$



- *$P(s_i)$ is the* probability of s_i
- Which is the lower-bounded to encode the symbols.

Example of Entropy

- Entropy of flipping a fair coin

S = { head, tails}

p(head) = p(tail) = 0.5

$H(coin) = -(1/2 \times \log_2 (1/2) + 1/2 \times \log_2 (1/2))$

$= -(1/2 \times -1 + ½ \times -1) = - (-1/2-1/2)$

$= 1$ bit

Information Gain

High Entropy – High level of Uncertainty

Low Entropy – No Uncertainty.

- The information gain of an attribute a is the expected reduction in entropy caused by partitioning on this attribute

$$Gain(C,a) = Entropy(C) - \sum_{v \in values(S)} \frac{|C_v|}{|C|} Entropy(C_v)$$

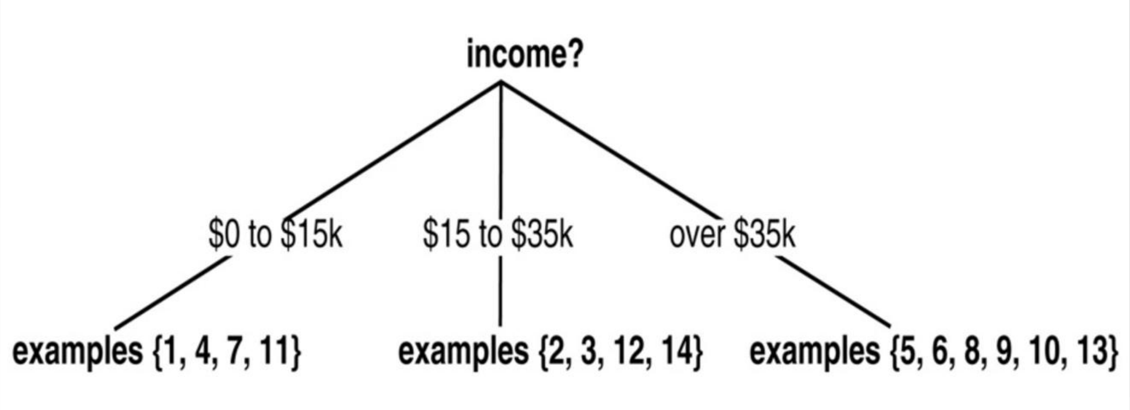
- Where:
  - C is the original instances at the node
  - C_v is the subset of C for which attribute a has value v, and
  - the entropy of partitioning the data is calculated by **weighing the entropy of each partition** by its size relative to the original set

- Partitions of low entropy lead to high gain

The Example: Root Attribute

- Let's back to our previous example
- We begin with calculation of the entropy of each attribute at the root node
- In the credit history loan table we make income the property tested at the root
- This makes the division into 3 branches

C1={1,4,7,11},C2={2,3,12,14},C3={5,6,8,9,10,13}



The Example: Root Attribute

- At the beginning, the set of instance C at root node is the whole credit table
- The table has following information
  - p(risk is high)=6/14
  - p(risk is moderate)=3/14
  - p(risk is low)=5/14

$$H(\text{credit\_table}) = \frac{6}{14}\log_2\left(\frac{6}{14}\right) + \frac{3}{14}\log_2\left(\frac{3}{14}\right) + \frac{5}{14}\log_2\left(\frac{5}{14}\right)$$

$$H(\text{credit\_table}) = 1.531 \text{ bits}$$

NO.	RISK
1.	high
2.	high
3.	moderate
4.	high
5.	low
6.	low
7.	high
8.	moderate
9.	low
10.	low
11.	high
12.	moderate
13.	low
14.	high

The Example: Root Attribute

C1={1,4,7,11},C2={2,3,12,14},C3={5,6,8,9,10,13}

{h,h,h,h} {h,m,m,h} {l,l,m,l,l,l}

$$H(income) = \frac{4}{14}\mathrm{H(C1)} + \frac{4}{14}\mathrm{H(C2)} + \frac{6}{14}\mathrm{H(C3)}$$

$$H(C1) = \frac{4}{4}\log_2\left(\frac{4}{4}\right) = 0$$

$$H(C2) = \frac{2}{4}\log_2\left(\frac{2}{4}\right) + \frac{2}{4}\log_2\left(\frac{2}{4}\right) = 1.0$$

$$H(C3) = \frac{5}{6}\log_2\left(\frac{5}{6}\right) + \frac{1}{6}\log_2\left(\frac{1}{6}\right) = 0.65$$

$$H(income) = \frac{4}{14}0 + \frac{4}{14}1.0 + \frac{6}{14}0.65$$

$$H(income) = 0.564 \quad bits$$

NO.	RISK
1.	high
2.	high
3.	moderate
4.	high
5.	low
6.	low
7.	high
8.	moderate
9.	low
10.	low
11.	high
12.	moderate
13.	low
14.	high

The Example: Root Attribute

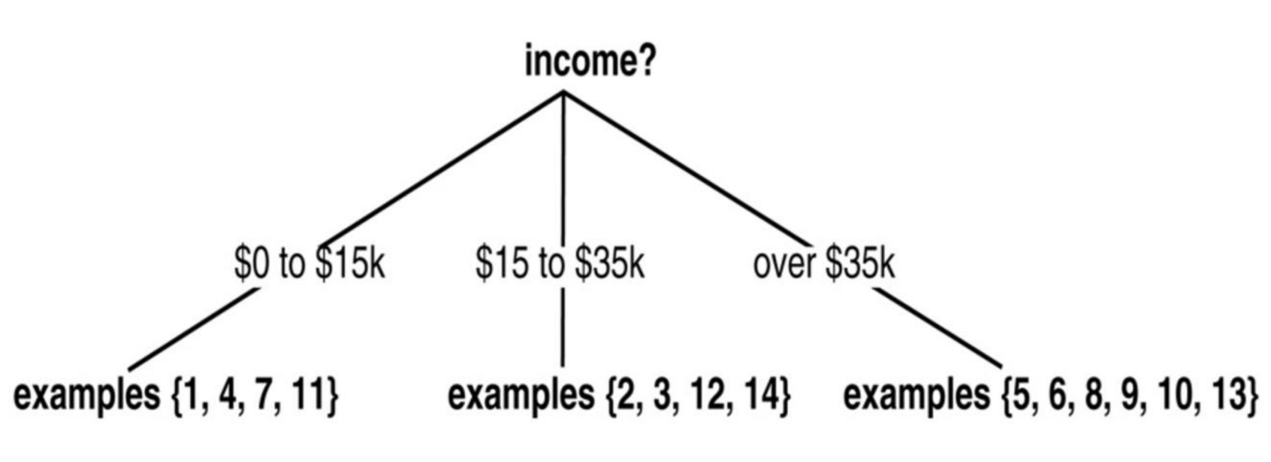
- gain(income)=H(credit_table)- H(income)
  gain(income)=1.531-0.564
  gain(income)=0.967 bits

Similarly, we can check the information gain of other attributes:

- gain(credit history)=0.266
- gain(debt)=0.581
- gain(collateral)=0.756

The Example: Root Attribute

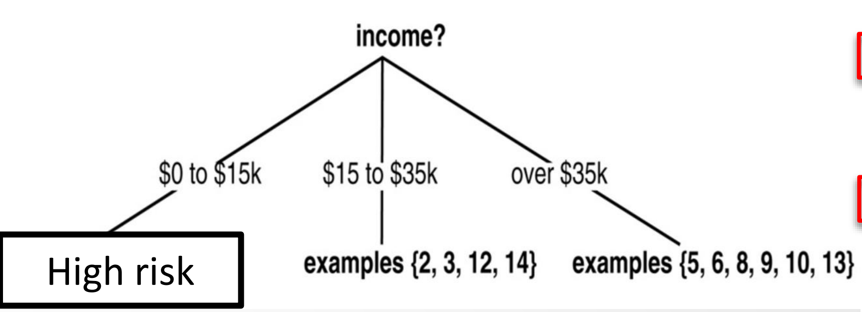
- Because "Income" provides the greatest information gain, ID3 will select it as the root



- The algorithm continues to apply this analysis recursively to each subtree, until it has completed the tree.

The Example: Internal Attributes

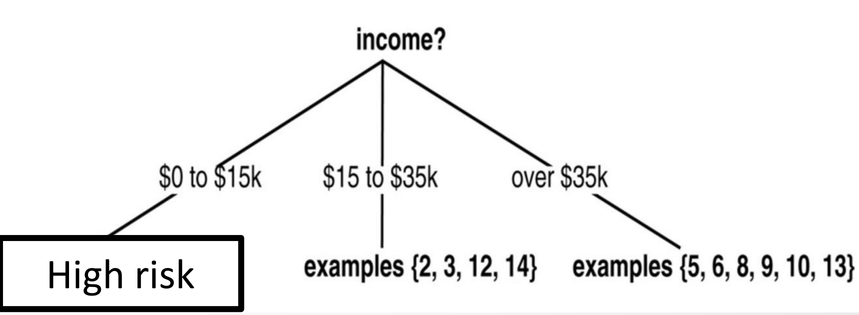
- Look at the first branch, we have data {1,4,7,11}, all of them conclude “high” risk
- So we can make this branch **a leaf node**!



NO.	RISK	CREDIT HISTORY	DEBT	COLLATERAL	INCOME
1.	high	bad	high	none	$0 to $15k
2.	high	unknown	high	none	$15 to $35k
3.	moderate	unknown	low	none	$15 to $35k
4.	high	unknown	low	none	$0 to $15k
5.	low	unknown	low	none	over $35k
6.	low	unknown	low	adequate	over $35k
7.	high	bad	low	none	$0 to $15k
8.	moderate	bad	low	adequate	over $35k
9.	low	good	low	none	over $35k
10.	low	good	high	adequate	over $35k
11.	high	good	high	none	$0 to $15k
12.	moderate	good	high	none	$15 to $35k
13.	low	good	high	none	over $35k
14.	high	bad	high	none	$15 to $35k

The Example: Internal Attributes

- The second branch, we have data {2,3,12,14},
- First, pick "Credit history" as the internal node



NO.	RISK	CREDIT HISTORY	DEBT	COLLATERAL	INCOME
1.	high	bad	high	none	$0 to $15k
2.	high	unknown	high	none	$15 to $35k
3.	moderate	unknown	low	none	$15 to $35k
4.	high	unknown	low	none	$0 to $15k
5.	low	unknown	low	none	over $35k
6.	low	unknown	low	adequate	over $35k
7.	high	bad	low	none	$0 to $15k
8.	moderate	bad	low	adequate	over $35k
9.	low	good	low	none	over $35k
10.	low	good	high	adequate	over $35k
11.	high	good	high	none	$0 to $15k
12.	moderate	good	high	none	$15 to $35k
13.	low	good	high	none	over $35k
14.	high	bad	high	none	$15 to $35k

The Example: Internal Attributes

- Again we have 3 branches

$C_{unknown}$={2,3}, C_{bad}={14}, C_{good}={12}

$$H(\text{credit\_history}) = \frac{2}{4}H(C_{\text{unknown}}) + \frac{1}{4}H(C_{\text{bad}}) + \frac{1}{4}H(C_{\text{good}})$$

$$H(\text{credit\_history}) = \frac{2}{4}1.0 + \frac{1}{4}0.0 + \frac{1}{4}0.0 = 0.5$$

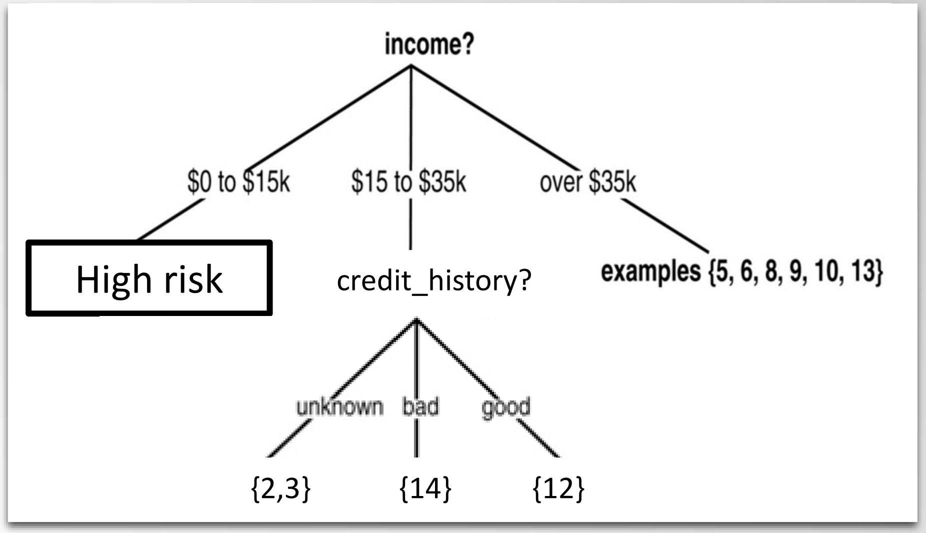
- gain(credit_history)=1.0 - H(credit_history) =0.5

Similarly

- gain(debt)= 1.0 – 3/4* (0.92)-1/4*0.0 = 0.31
- gain(collateral)= 1.0 – 1.0 = 0.0

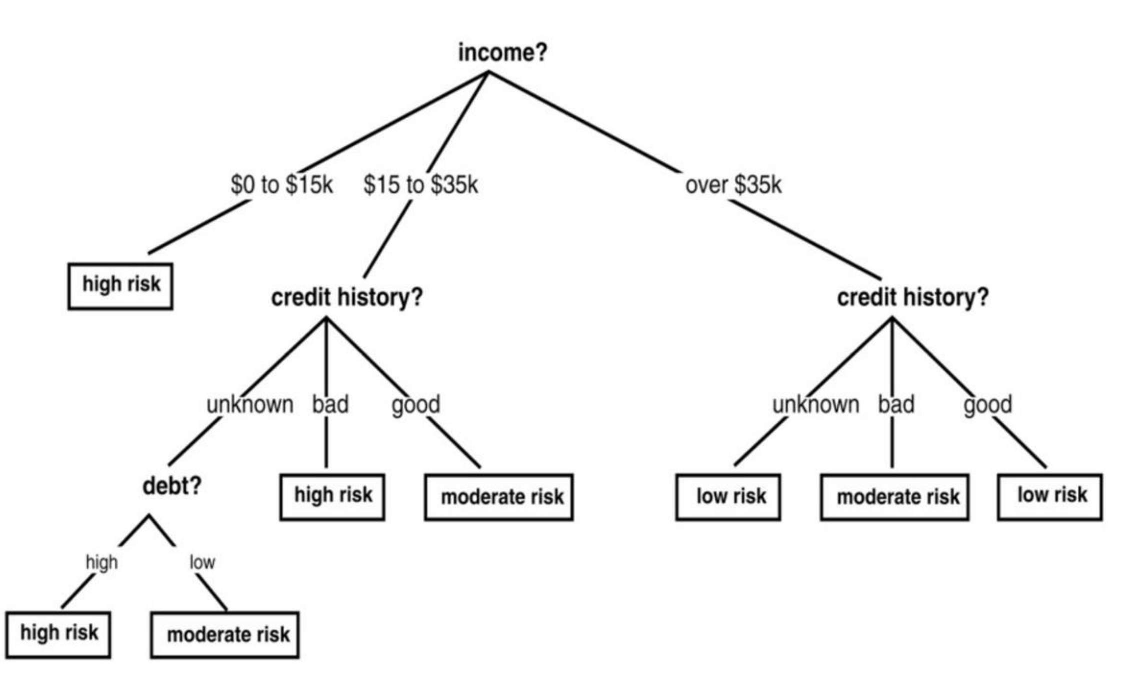
The Example: Internal Attributes

- So credit_history will be selected, the tree is updated as



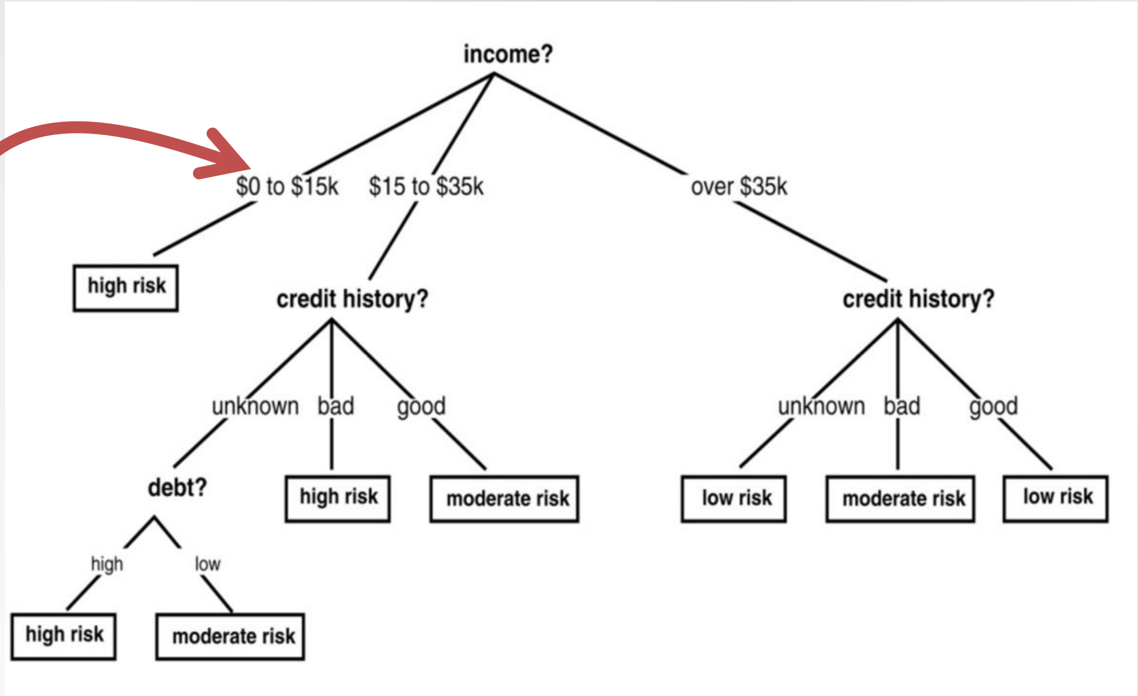
The Example

- We loop for other uncompleted branches and have the final tree



The Example

- Back to our earlier question, the new comer will be classified as…

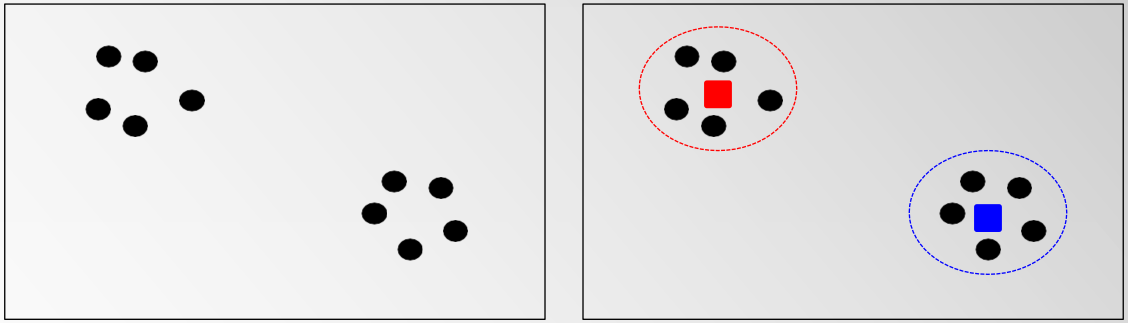


Income	Credit History	Debt	Collateral	Risk
$0-15K	bad	low	adequate	High Risk

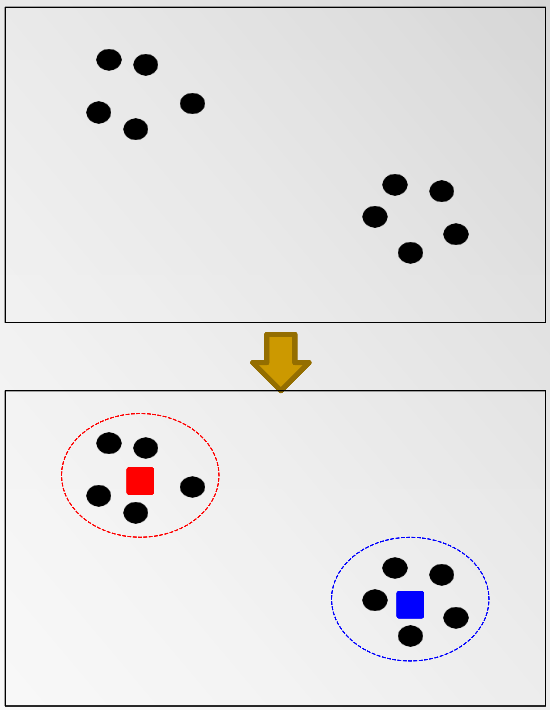
Clustering

Clustering

- Clustering
  - Partition a given dataset into groups based on specified features so that the data points within a group are more similar to each other than the points in different groups
  - A typical unsupervised learning problem
  - Different clustering algorithms lead to different results



K-means clustering



Given

a set of observation $(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N)$,
where each observation is
a *d*-dimensional real vector

Aim to

partition observations into *K* sets
(*K* < *N*), $\mathbf{S} = \{S_1, \cdots S_K\}$
so as to minimize
the within-cluster sum of squares:

$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{K} \sum_{\mathbf{x}_j \in S_i} \| \mathbf{x}_j - \mu_i \|^2$$

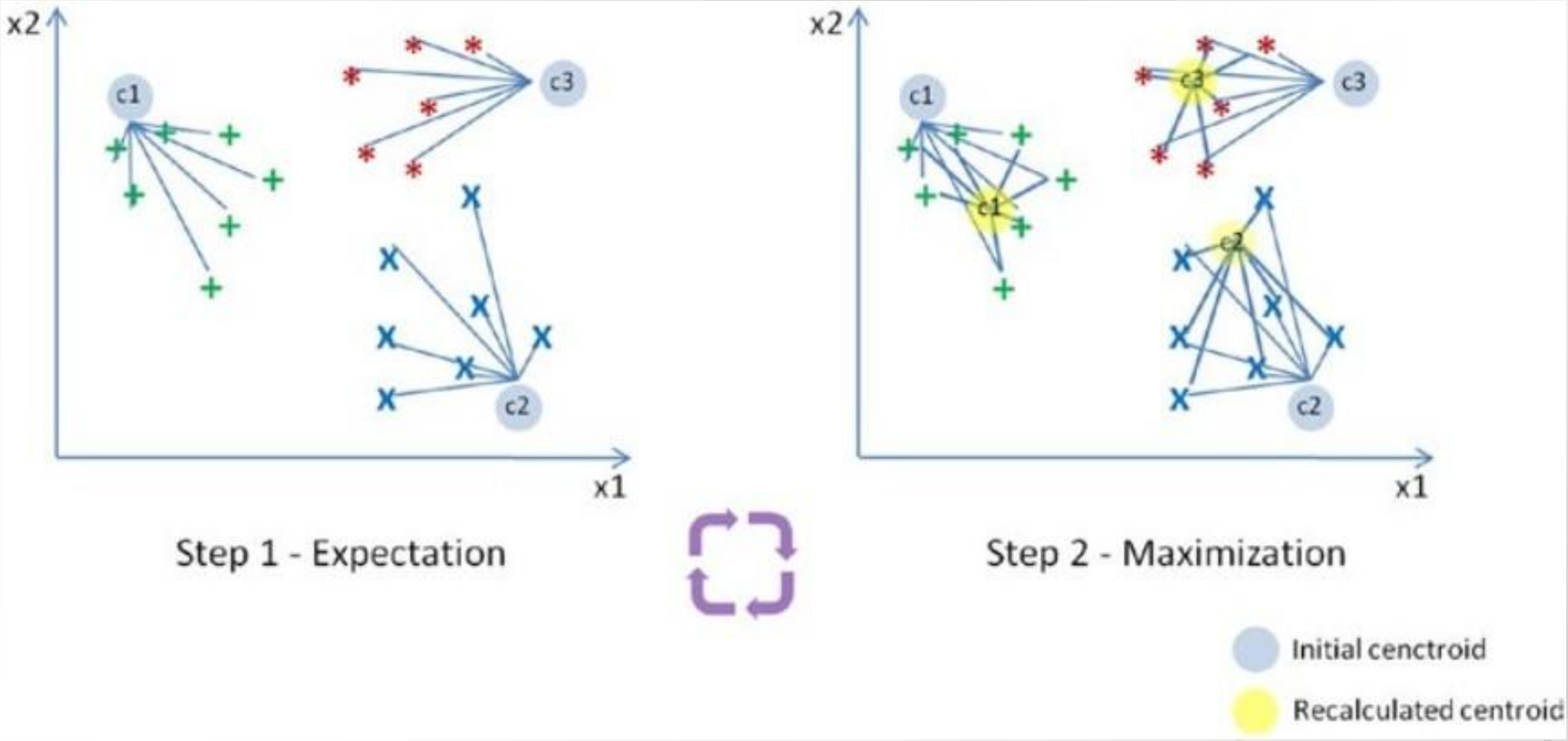
where μ_i is the mean of S_i .

Algorithm workflow

- Our algorithm works as follows, assuming we have inputs x1, x2, x3,…,xn and value of K
    - Step 1 - Pick K random points as cluster centers called centroids.
    - Step 2 - Assign each xi to nearest cluster by calculating its distance to each centroid.
    - Step 3 - Find new cluster center by taking the average of the assigned points.
    - Step 4 - Repeat Step 2 and 3 until none of the cluster assignments change.

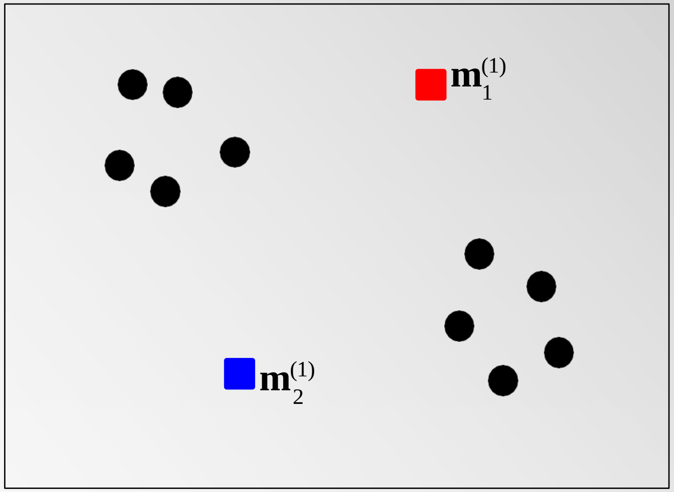
Algorithm workflow

- Categorized as an expectation maximization algorithm



K means: details

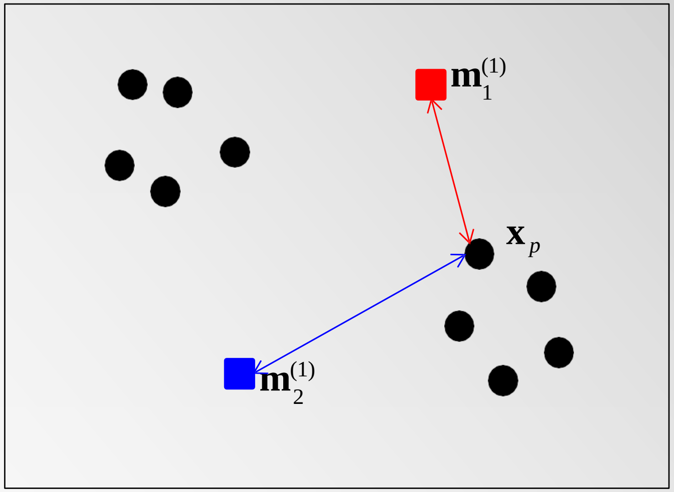
First iteration



K means begin with an initial guess to the centers: $\mathbf{m}_1^{(1)}$ and $\mathbf{m}_2^{(1)}$

K means: details

First iteration



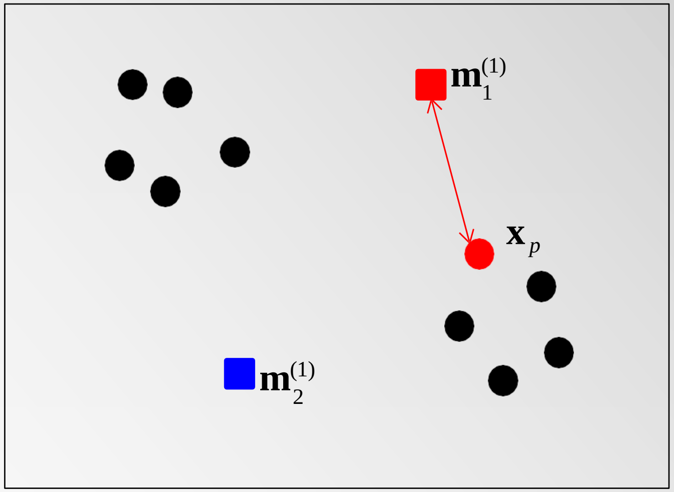
Calculate the Euclidean distance between each point and the centers

$$d^2(\mathbf{x}_p, \mathbf{m}_1^{(1)}) = \| \mathbf{x}_p - \mathbf{m}_1^{(1)} \|^2 = \sum_i (x_{pi} - m_{1i}^{(1)})^2$$

$$d^2(\mathbf{x}_p, \mathbf{m}_2^{(1)}) = \| \mathbf{x}_p - \mathbf{m}_2^{(1)} \|^2 = \sum_i (x_{pi} - m_{2i}^{(1)})^2$$

K means: details

First iteration



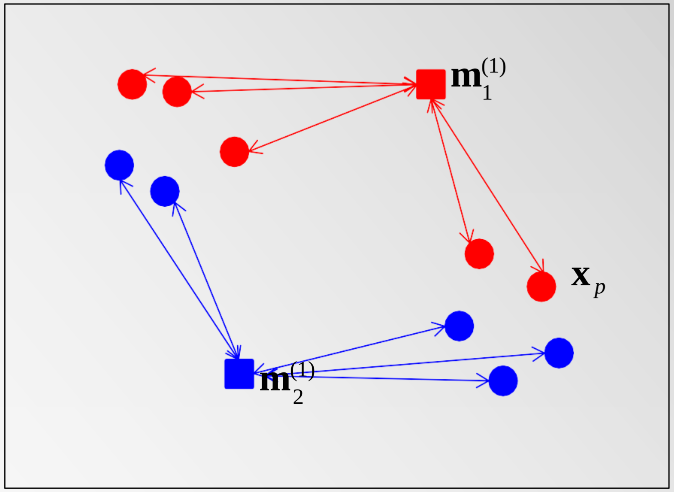
Assign the data point to the center with smallest distance:

$$d^2(\mathbf{x}_p, \mathbf{m}_1^{(1)}) < d^2(\mathbf{x}_p, \mathbf{m}_2^{(1)})$$

Then $\mathbf{x}_p$ is assigned to $\mathbf{m}_1^{(1)}$

K means: details

First iteration



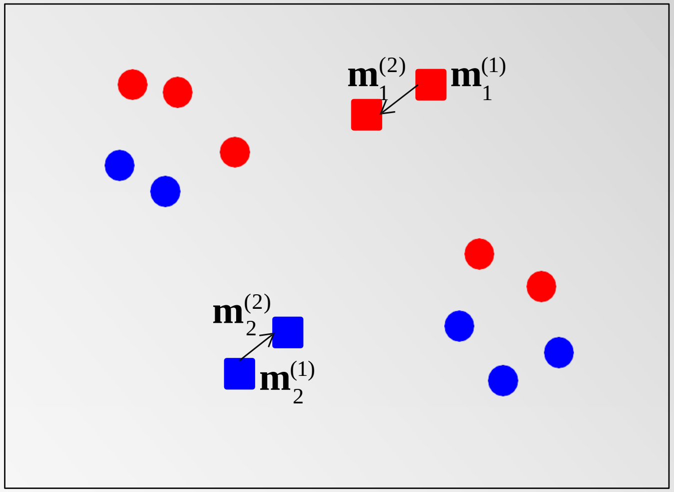
Assign all the other points to the centers in the same way as $\mathbf{x}_p$

$$S_1^{(1)} = \{\mathbf{x}_p : \|\mathbf{x}_p - \mathbf{m}_1^{(1)}\|^2 \le \|\mathbf{x}_p - \mathbf{m}_2^{(1)}\|^2\}$$

$$S_2^{(1)} = \{\mathbf{x}_p : \|\mathbf{x}_p - \mathbf{m}_2^{(1)}\|^2 \le \|\mathbf{x}_p - \mathbf{m}_1^{(1)}\|^2\}$$

K means: details

First iteration



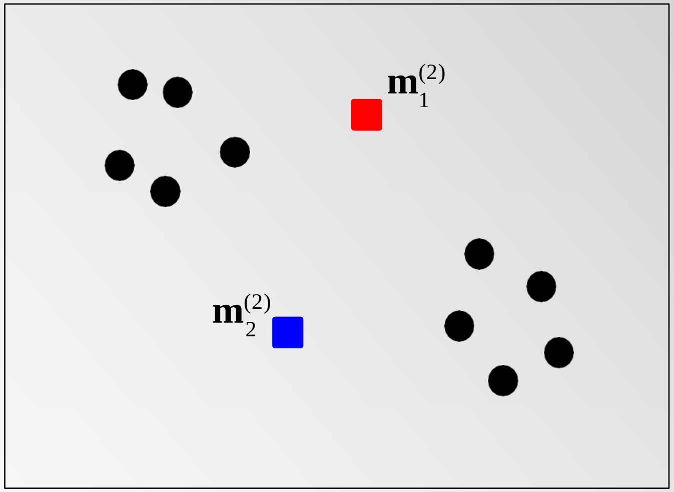
Update new centers by averaging the data points in each set:

$$\mathbf{m}_1^{(2)} = \sum_{\mathbf{x}_p \in S_1^{(1)}} \mathbf{x}_p / | S_1^{(1)} |$$

$$\mathbf{m}_2^{(2)} = \sum_{\mathbf{x}_p \in S_2^{(1)}} \mathbf{x}_p / | S_2^{(1)} |$$

K means: details

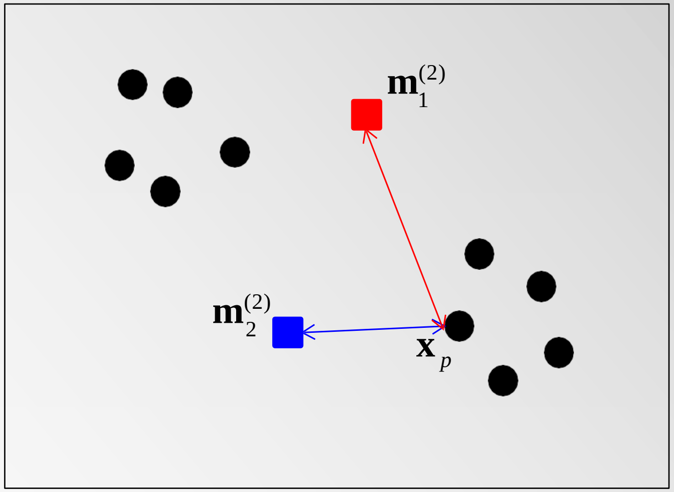
First iteration



The result after first iteration

K means: details

Second iteration

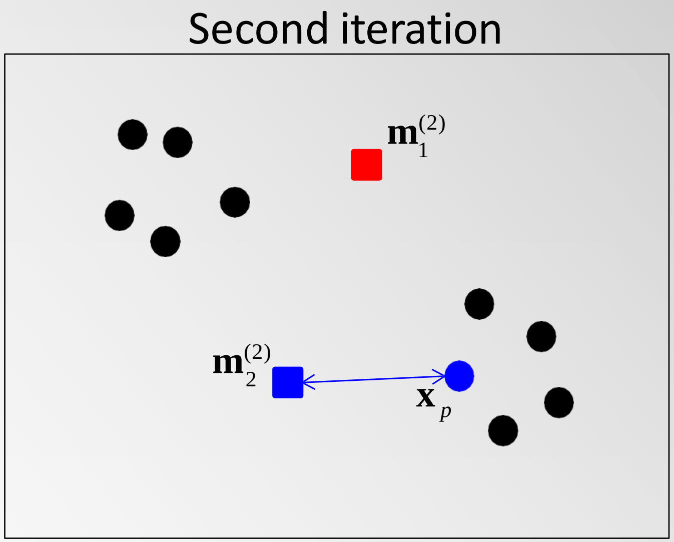


Calculate the Euclidean distance between each point and the centers

$$d^2(\mathbf{x}_p, \mathbf{m}_1^{(2)}) = \| \mathbf{x}_p - \mathbf{m}_1^{(2)} \|^2 = \sum_i (x_{pi} - m_{1i}^{(2)})^2$$

$$d^2(\mathbf{x}_p, \mathbf{m}_2^{(2)}) = \| \mathbf{x}_p - \mathbf{m}_2^{(2)} \|^2 = \sum_i (x_{pi} - m_{2i}^{(2)})^2$$

K means: details



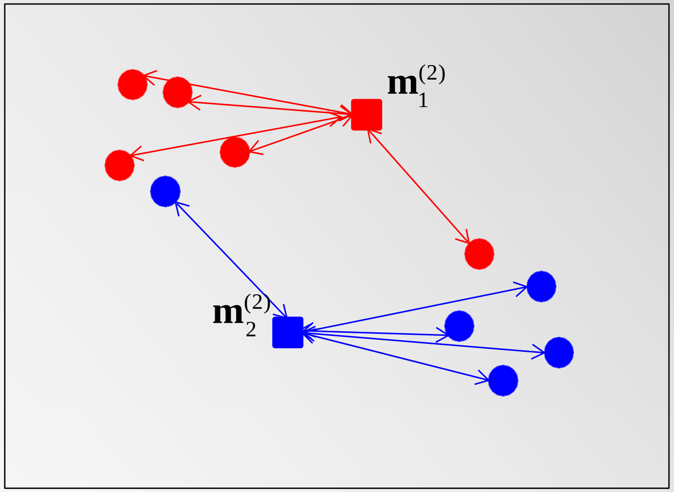
Assign the data point to the center with smallest distance:

$$d^2(\mathbf{x}_p, \mathbf{m}_2^{(2)}) < d^2(\mathbf{x}_p, \mathbf{m}_1^{(2)})$$

Then $\mathbf{x}_p$ is assigned to $\mathbf{m}^{(2)}_2$

K means: details

Second iteration



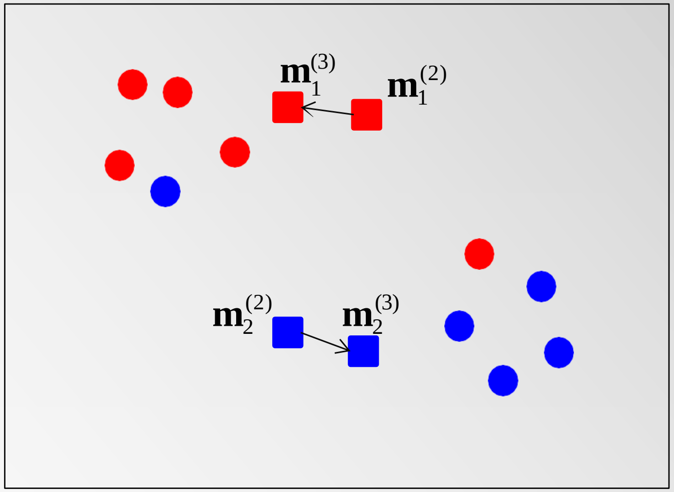
Assign all the other points to the centers in the same way as $\mathbf{x}_p$

$$S_1^{(2)} = \{\mathbf{x}_p : \| \mathbf{x}_p - \mathbf{m}_1^{(2)} \|^2 \leq \| \mathbf{x}_p - \mathbf{m}_2^{(2)} \|^2\}$$

$$S_2^{(2)} = \{\mathbf{x}_p : \| \mathbf{x}_p - \mathbf{m}_2^{(2)} \|^2 \leq \| \mathbf{x}_p - \mathbf{m}_1^{(2)} \|^2\}$$

K means: details

Second iteration



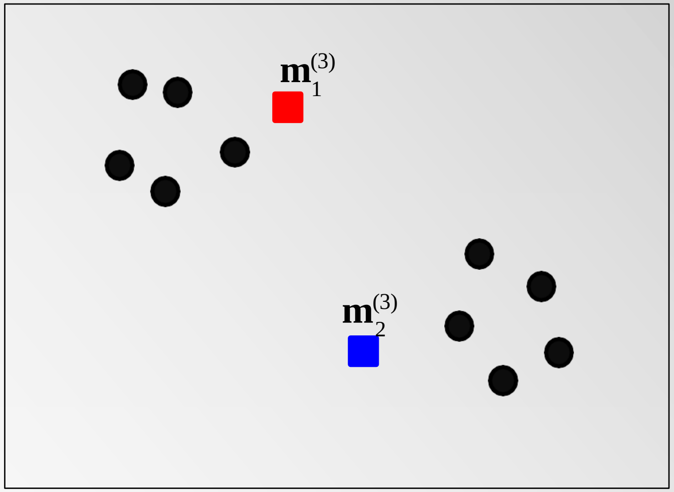
Update new centers by averaging the data points in each set:

$$\mathbf{m}_1^{(3)} = \sum_{\mathbf{x}_p \in S_1^{(2)}} \mathbf{x}_p / |S_1^{(2)}|$$

$$\mathbf{m}_2^{(3)} = \sum_{\mathbf{x}_p \in S_2^{(2)}} \mathbf{x}_p / |S_2^{(2)}|$$

K means: details

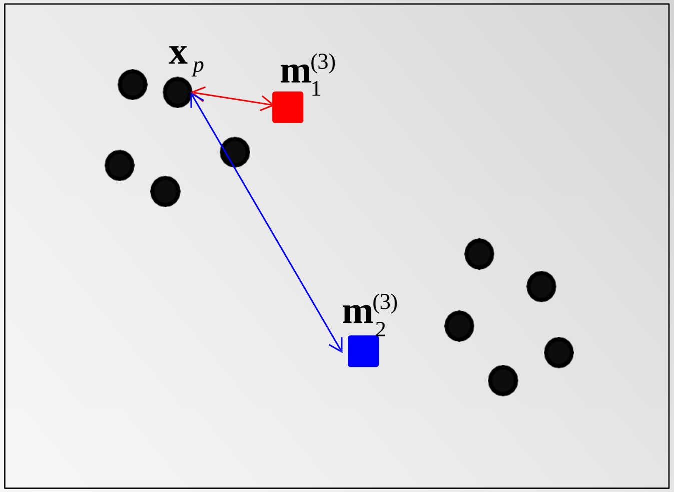
Second iteration



The result after second iteration

K means: details

Third iteration



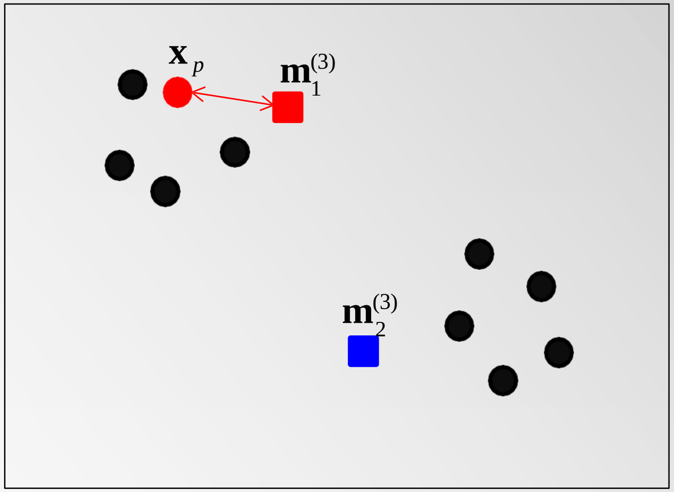
Calculate the Euclidean distance between each point and the centers

$$d^2(\mathbf{x}_p, \mathbf{m}_1^{(3)}) = \| \mathbf{x}_p - \mathbf{m}_1^{(3)} \|^2 = \sum_i (x_{pi} - m_{1i}^{(3)})^2$$

$$d^2(\mathbf{x}_p, \mathbf{m}_2^{(3)}) = \| \mathbf{x}_p - \mathbf{m}_2^{(3)} \|^2 = \sum_i (x_{pi} - m_{2i}^{(3)})^2$$

K means: details

Third iteration



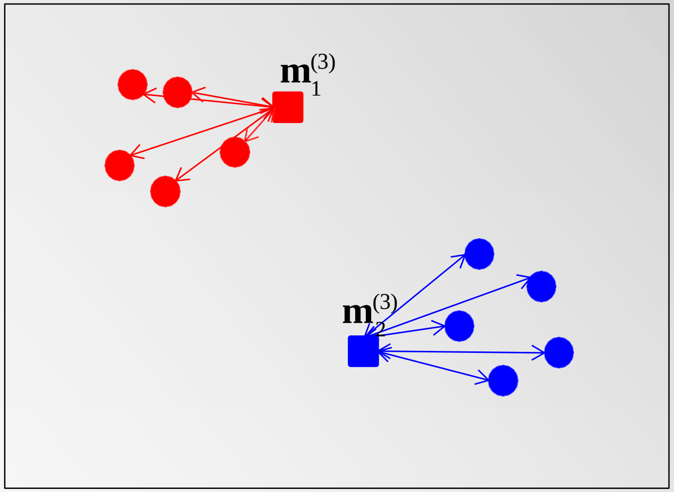
Assign the data point to the center with smallest distance:

$$d^2(\mathbf{x}_p, \mathbf{m}_1^{(3)}) < d^2(\mathbf{x}_p, \mathbf{m}_2^{(3)})$$

Then $\mathbf{x}_p$ is assigned to $\mathbf{m}_1^{(3)}$

K means: details

Third iteration



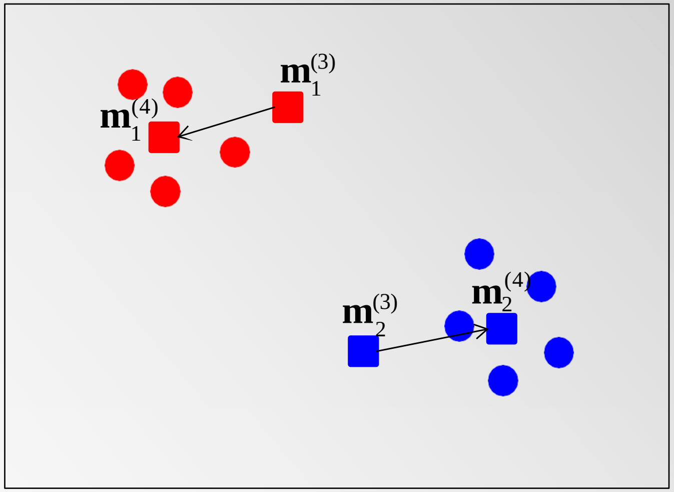
Assign all the other points to the centers in the same way as $\mathbf{x}_p$

$$S_1^{(3)} = \{\mathbf{x}_p : \| \mathbf{x}_p - \mathbf{m}_1^{(3)} \|^2 \leq \| \mathbf{x}_p - \mathbf{m}_2^{(3)} \|^2\}$$

$$S_2^{(3)} = \{\mathbf{x}_p : \| \mathbf{x}_p - \mathbf{m}_2^{(3)} \|^2 \leq \| \mathbf{x}_p - \mathbf{m}_1^{(3)} \|^2\}$$

K means: details

Third iteration

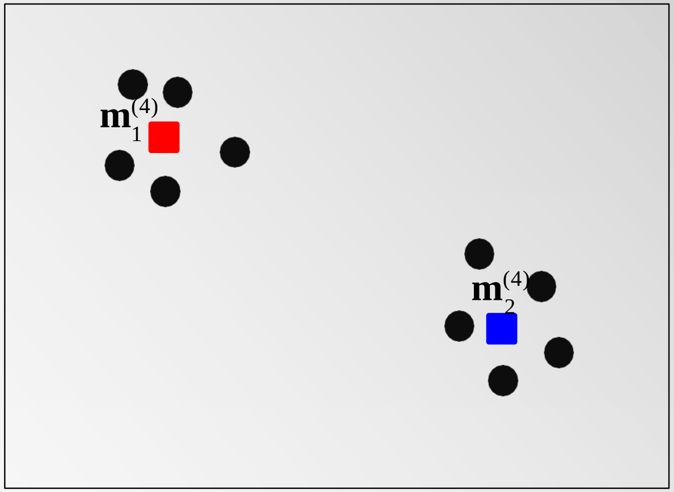


Update new centers by averaging the data points in each set:

$$\mathbf{m}_1^{(4)} = \sum_{\mathbf{x}_p \in S_1^{(3)}} \mathbf{x}_p / |S_1^{(3)}|$$

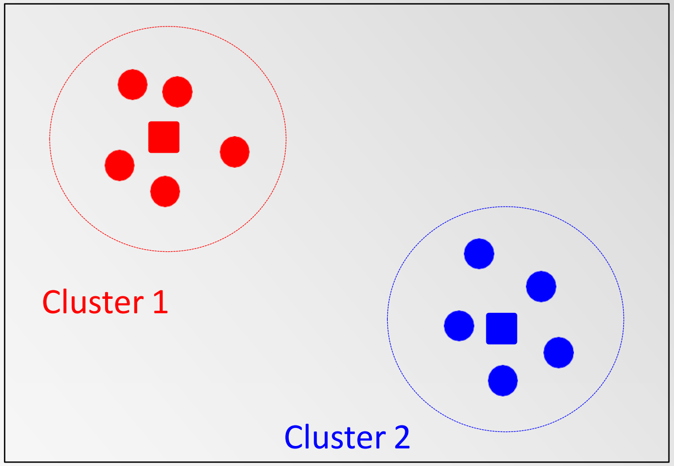
$$\mathbf{m}_2^{(4)} = \sum_{\mathbf{x}_p \in S_2^{(3)}} \mathbf{x}_p / |S_2^{(3)}|$$

K means: details



Converge !

K means: details

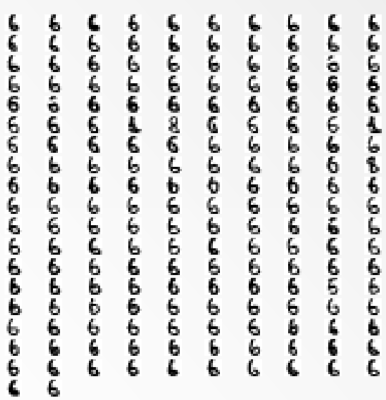


Clustering result

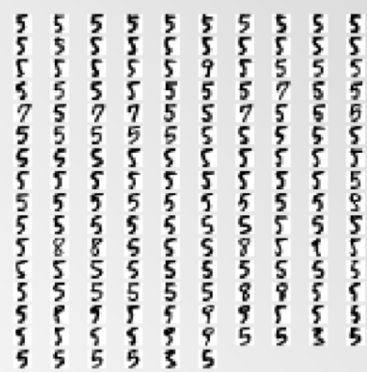
Application of K means: Classification

- Digit image classification with k-means
- MNIST dataset: 8x8 image of handwritten digits

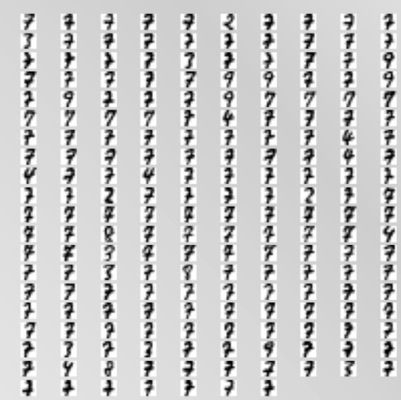
Cluster 1



Cluster 2



Cluster 3 ……

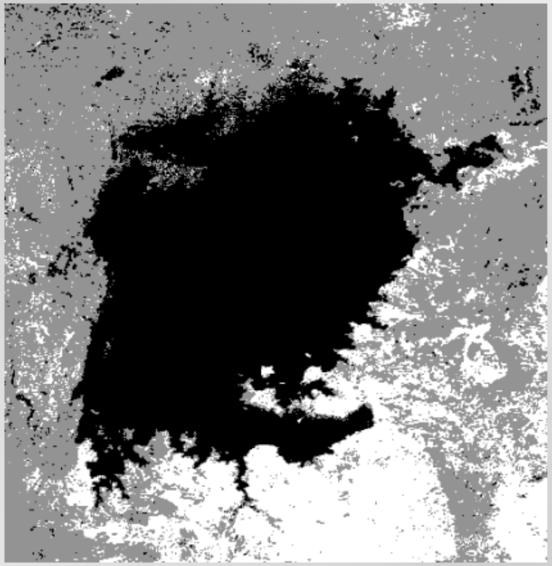


https://www.unioviedo.es/compnum/labs/new/kmeans.html

Application of K means: Segmentation

- Segmentation : separate into regions
- Based on both locality and color

Segment into 3 regions
with K-mean (K=3)



Application of K means: Data Analysis

- To analyze similar customers, we can make use of K-mean clustering .
- Following table contains information of Clients that subscribe to Membership card, including their age, income, and spending score depending on number times in week the show up in Mall, total expense in same mall and etc.

	A	B	C	D	E
1	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
2	1	Male	19	15	39
3	2	Male	21	15	81
4	3	Female	20	16	6
5	4	Female	23	16	77
6	5	Female	31	17	40
7	6	Female	22	17	76
8	7	Female	35	18	6
9	8	Female	23	18	94
10	9	Male	64	19	3
11	10	Female	30	19	72

Elbow Method

- Before using K-Mean Clustering, we need to decide value K, i.e. how many clusters to use?
- There are two commonly used methods to determine the ideal number of clusters possible in K-means
  - Elbow Method
  - Silhouette Method
- Here, we will introduce Elbow method

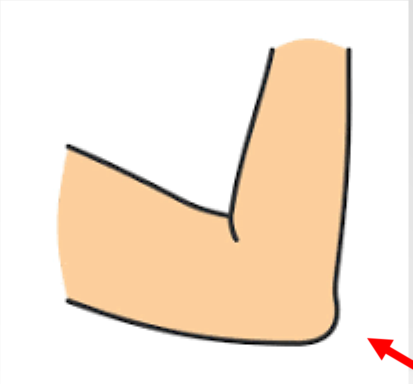
Elbow Method

- First of all, compute the sum of squared error (SSE) for some values of k (for example 2, 4, 6, 8, etc.).
- The SSE is defined as the sum of the squared distance between each member of the cluster and its centroid.

$$SSE = \sum_{i=1}^{K} \sum_{x \in c_i} dist(x, c_i)^2$$

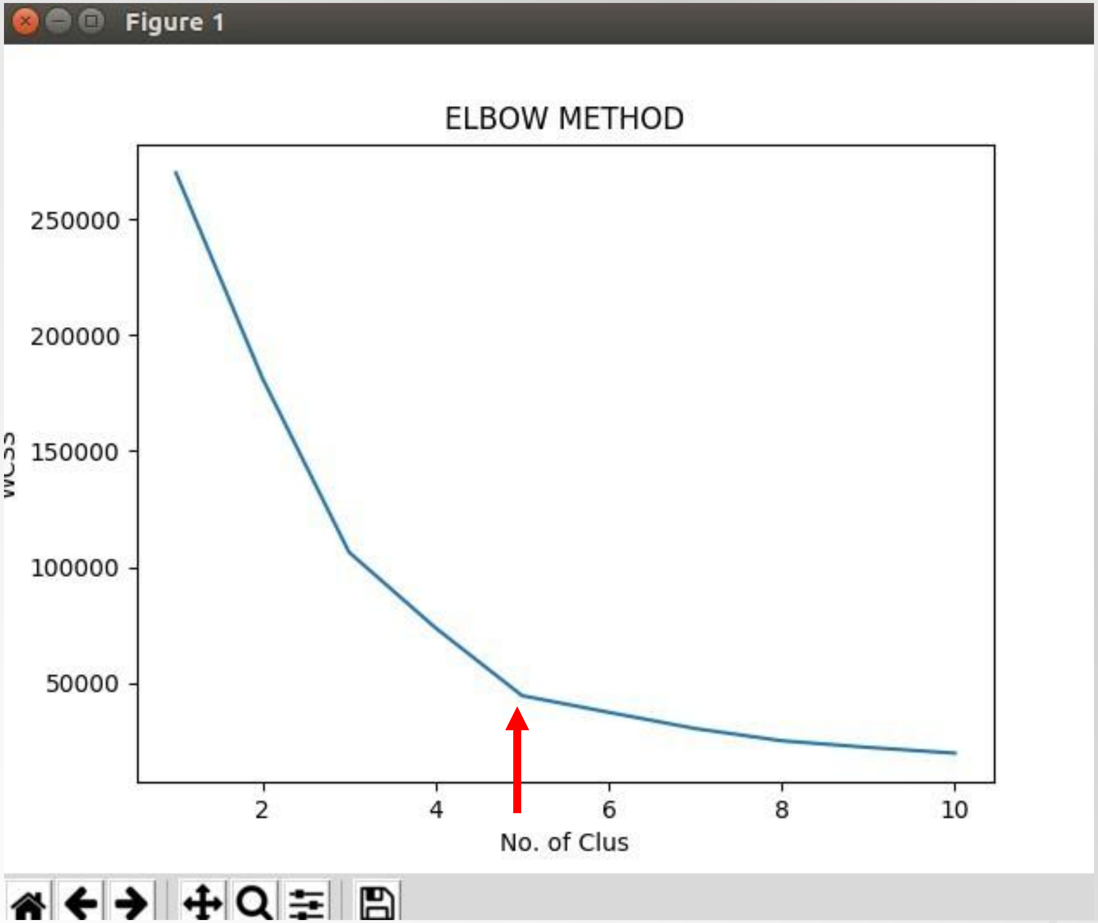
Elbow Method

- If you plot k against the SSE, you will see that the error decreases as k gets larger;
- This is because when the number of clusters increases, they should be smaller, so distortion is also smaller.
- The idea of the elbow method is to choose the k at which the SSE decreases abruptly. This produces an "elbow effect" in the graph



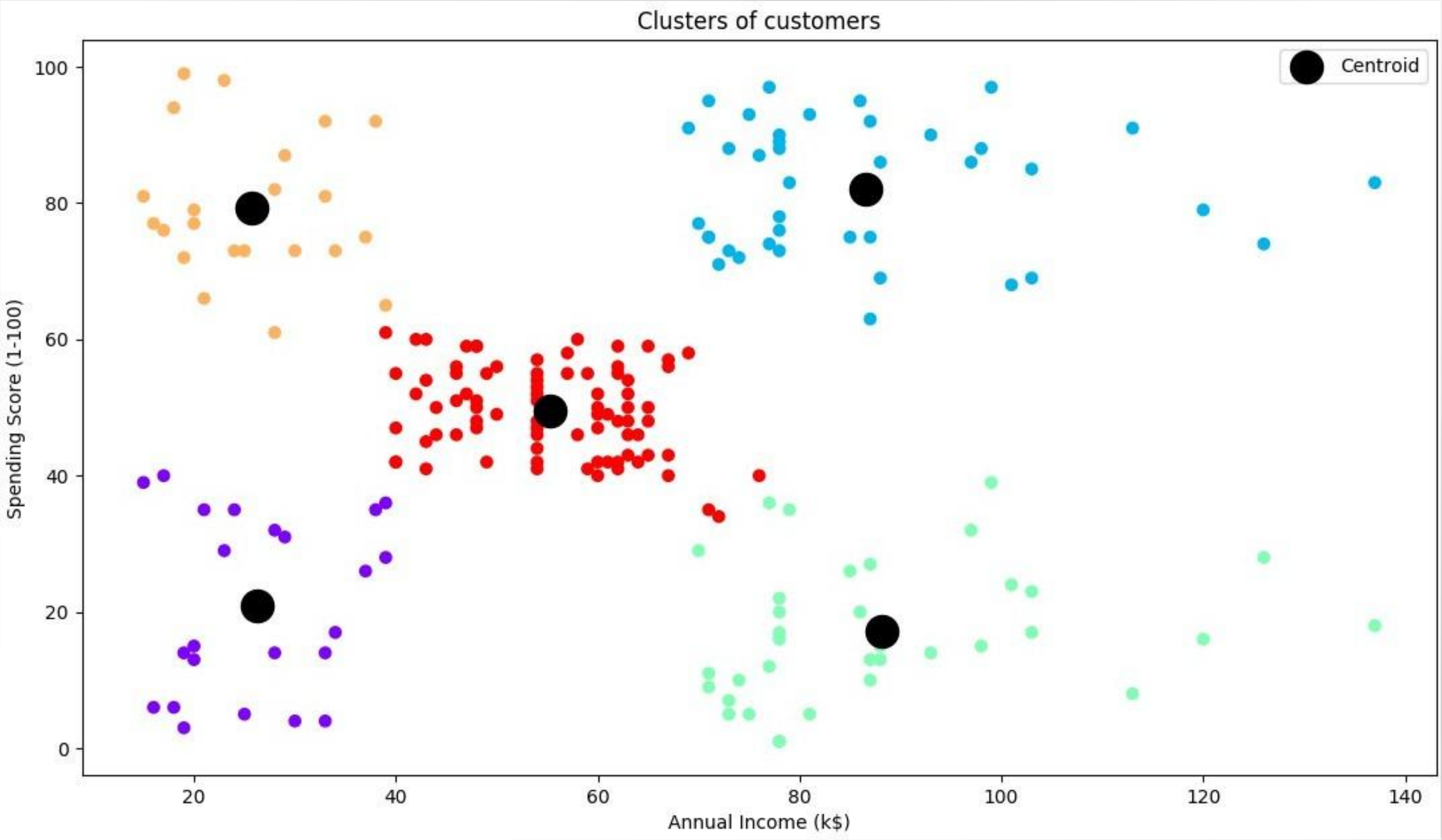
Elbow method

- In this case, k=5 is the value that the Elbow method has selected.



Visualizing the clusters

- We can check with each cluster to find similar customers



Limitations

- K-means clustering needs the number of clusters to be specified.
  - Although elbow method can be used, it also depends on nature of the application itself
- K-means has problems if the “true” clusters are of differing sized, densities, and non-globular shapes.
- Presence of outlier can skew the results.

Summary

- Machine learning
- Basics of Classification
- Decision Tree
- Clustering