Introduction to Urban Big Data and Machine Learning

Lecture 15 Machine Learning (II)

Wenzheng Li

Announcement

- All assignments (1–4) should be submitted no later than Friday.
- Friday Afternoon: 7–10 minutes presentation
- Final poster due: Sunday night

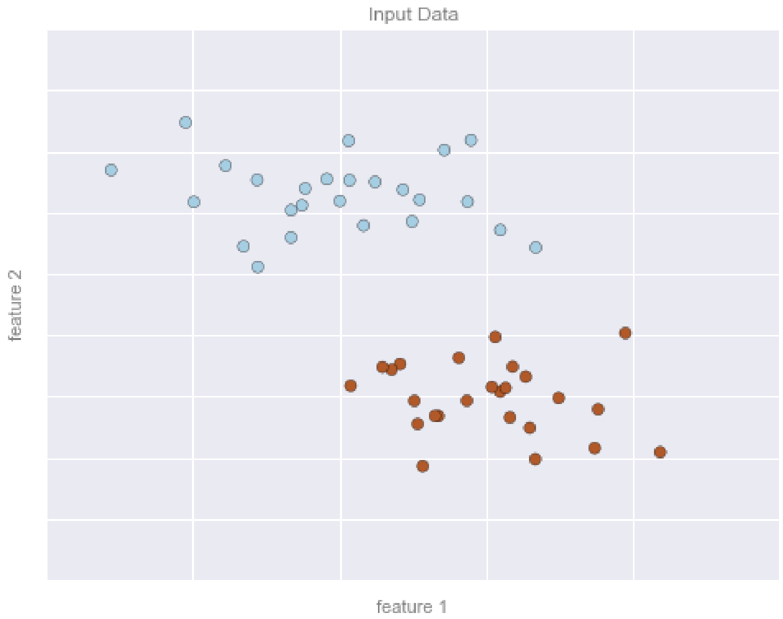
OUTLINE

- Introduction to Machine Learning
  - What is machine learning?
  - Machine learning types
- Supervised Learning
  - Classification and Regression
- Unsupervised Learning
  - Clustering

1 Quick Review

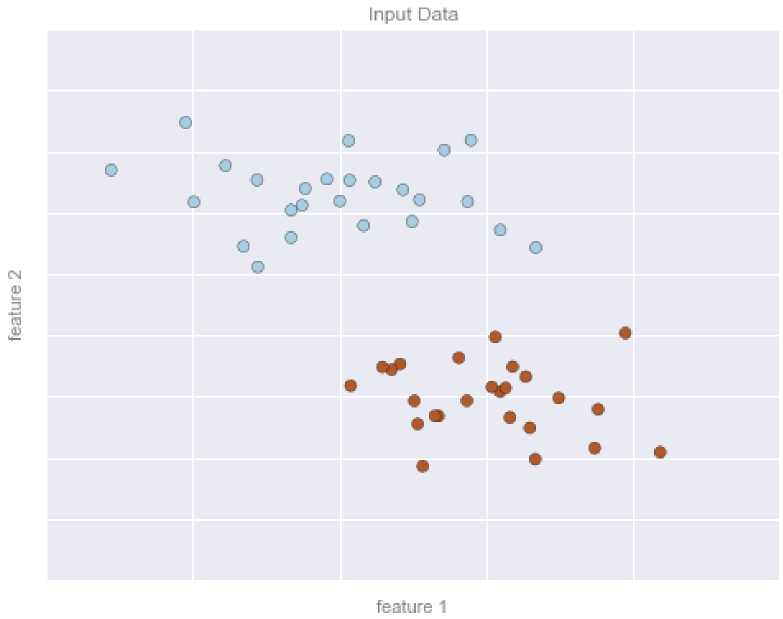
Supervised learning: Classification

- Let’s look at some examples



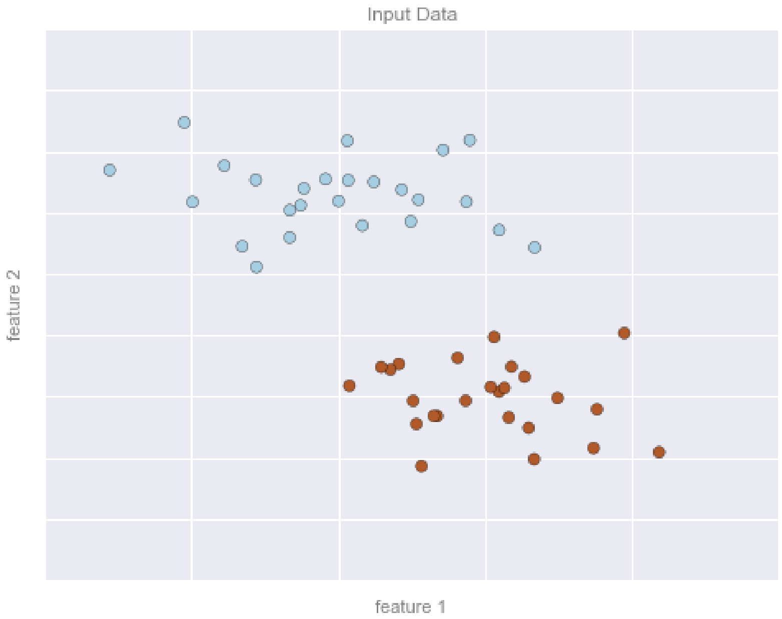
Supervised learning: Classification

- Let’s look at some examples
- Here, we have a two-dimensional (read: two columns) dataset
- We know which dots are blue and which are red.



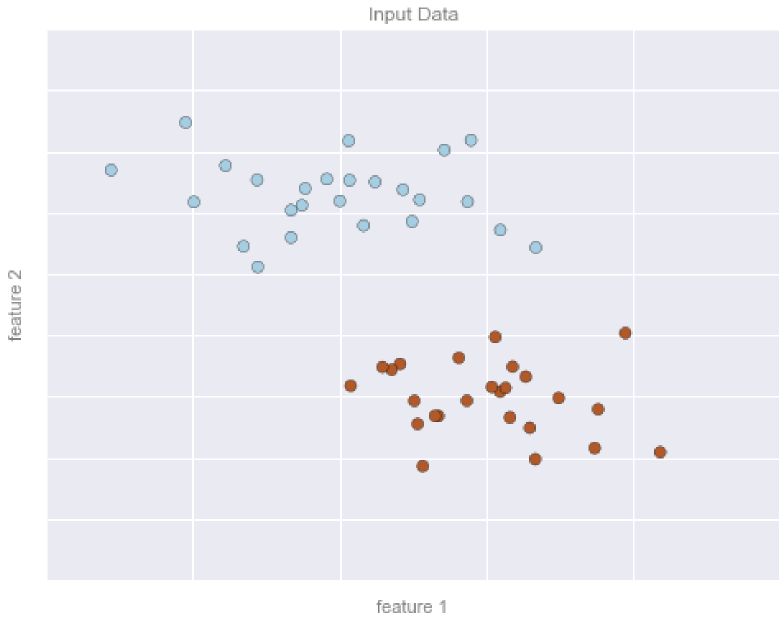
Supervised learning: Classification

- Let's look at some examples
- Here, we have a two-dimensional (read: two columns) dataset
- We know which dots are blue and which are red.
- The classification question then asks:
    - Can we create a "separator" model that separates these two points?



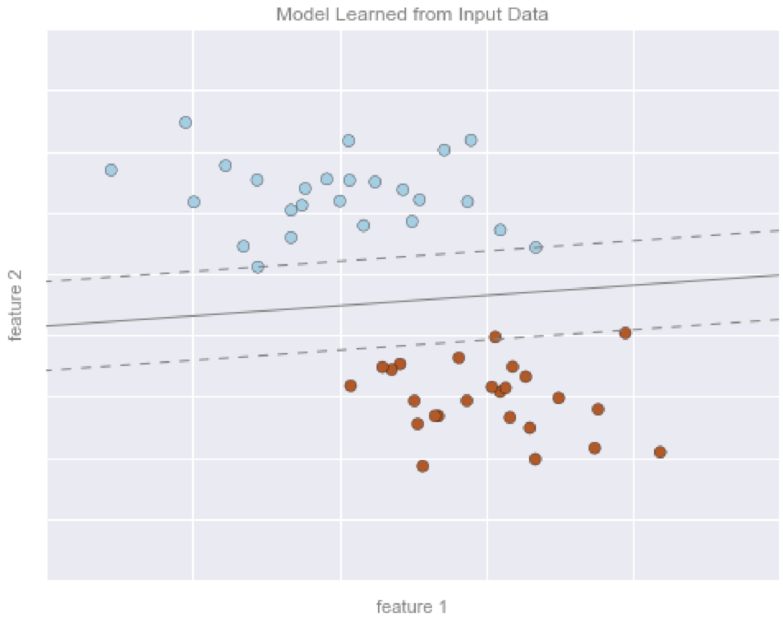
Supervised learning: Classification

- Model: the quantitative version of “there is a straight line that can separate the two classes”.
- Parameters: the intercept and axis (or however else we want to define our line) that describes the line



Supervised learning: Classification

- So our model could look like this on the right.



Which is classification problem?

- An e-commerce company using labeled customer data to predict whether or not a customer will purchase a particular item

- A restaurant using review data to ascribe positive or negative sentiment to a given review

- A bike share company using time and weather data to predict the number of bikes being rented at any given hour

Which is classification problem?

- An e-commerce company using labeled customer data to predict whether or not a customer will purchase a particular item

- A restaurant using review data to ascribe positive or negative sentiment to a given review

- A bike share company using time and weather data to predict the number of bikes being rented at any given hour

Which is classification problem?

- A regression problem is when we try to use any type (continuous or categorical) of data to predict outcomes that are continuous.

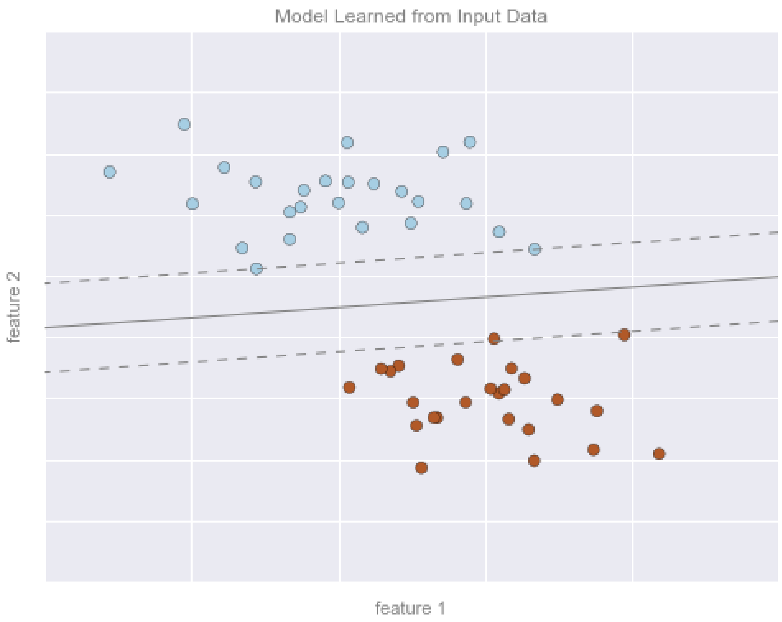
Regression vs Classification

- A regression problem is when we try to use any type (continuous or categorical) of data to predict outcomes that are continuous.

- A classification problem is when we try to use any type (continuous or categorical) of data to predict outcomes that are categorical.

Model Validation: Train-Test Split

- Once we have trained our model, i.e. described this function that best separates our two classes (in this case), we can use the model to generalize and make **new predictions**

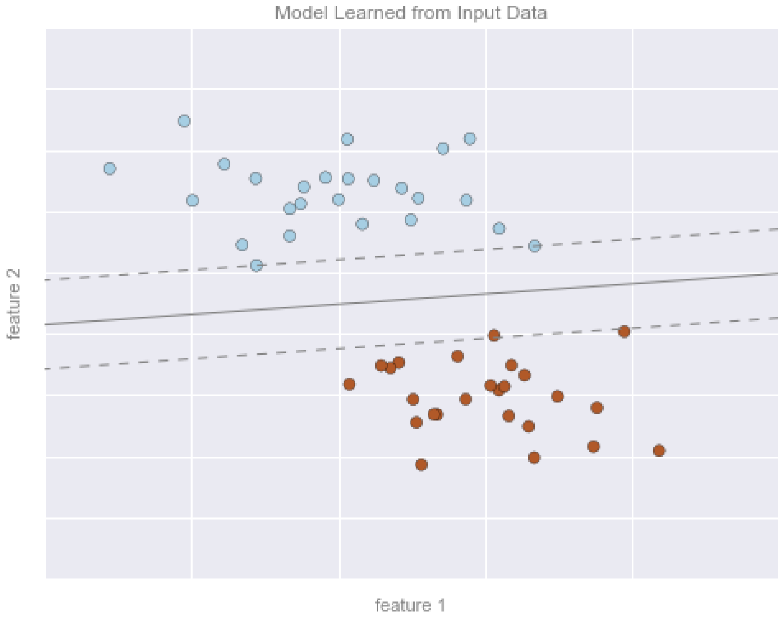


Model Validation: Train-Test Split

- We typically split our data into **train** and **test** sets to evaluate the performance of a model.

- The training set is used to find the model

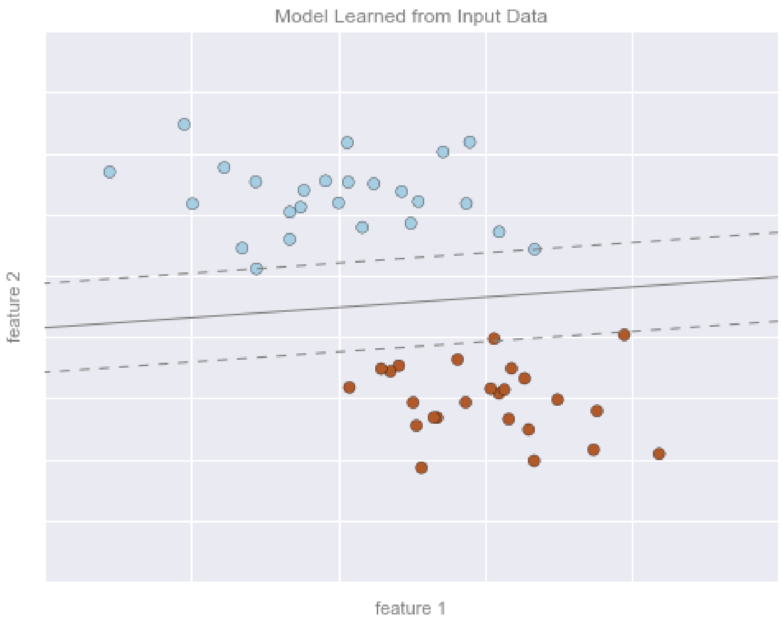
- The test set is used to evaluate the performance



Model Validation: Train-Test Split

- We want to give the model new, unseen data.

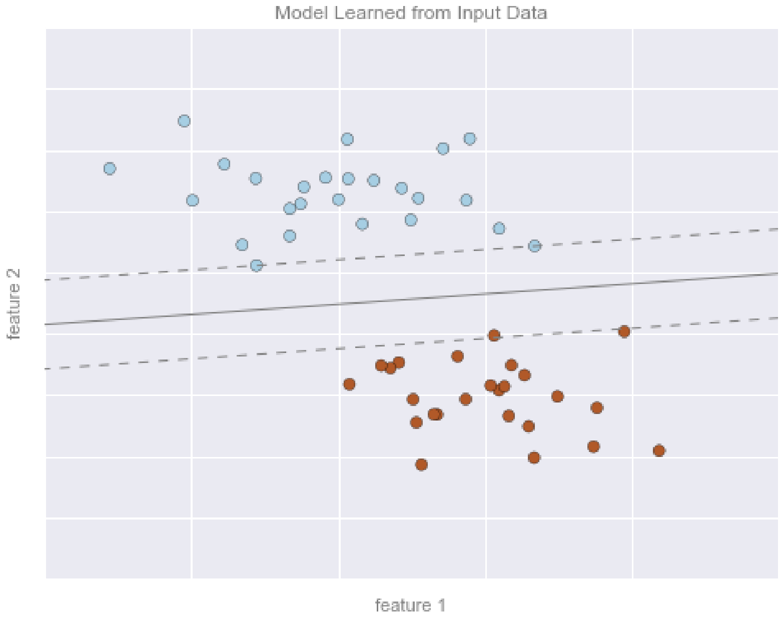
- We do this to ensure the model can generalize well.



Model Validation: Train-Test Split

- How do we divide the train and test set? No one-size-fits-all rule.

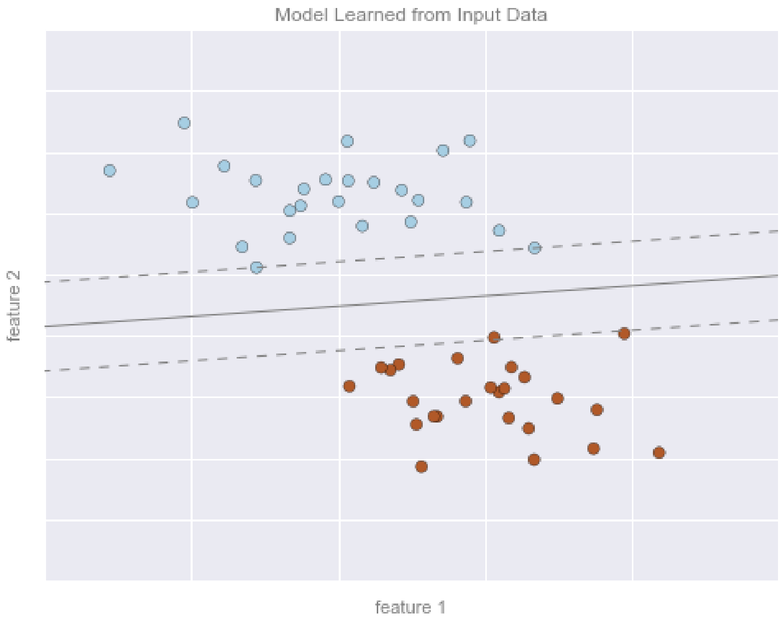
- A common breakdown is 70/30 or 80/20 train vs test



Model Validation: Train-Test Split

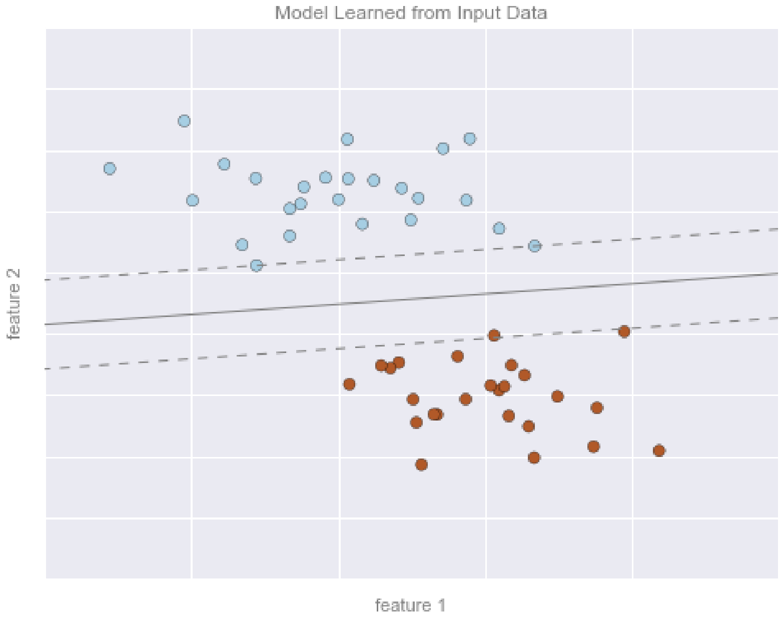
- The split might be determined by:
-
- How big the overall data is. If the data is already pretty small, then you might want a larger training set to get enough data to learn.

- If the data is large, perhaps a smaller split is fine.



Model Validation: Train-Test Split

- You need to make sure your training data is representative of the larger sample.



Confusion Matrix

- Describe the performance of a **classification** model

Predicted

Total = 1000	Spam	Non-Spam
Actual Spam	330	70
Actual Non-Spam	90	510

Predicted

	Spam	Non-Spam
Actual Spam	True Positive	False Negative
Actual Non-Spam	False Positive	True Negative

Type II error

Type I error

Overall, how often is the model correct?

Metric 1: Accuracy

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total}$$

$$= \frac{330 + 510}{1000}$$

$$= 0.84$$

Confusion Matrix

How often does the model correctly identify positives (spam emails)?

		Predicted	
	Total = 1000	Spam	Non-Spam
Actual	Spam	330	70
	Non-Spam	90	510

		Predicted	
		Spam	Non-Spam
Actual	Spam	True Positive	False Negative
	Non-Spam	False Positive	True Negative

Metric 2: Recall

(Sensitivity or True Positive Rate)

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$= \frac{330}{330 + 70}$$

$$= 0.825$$

Confusion Matrix

When the model predicts positive, how often is it correct?

Total = 1000	Predicted: Spam	Predicted: Non-Spam
Actual: Spam	330	70
Actual: Non-Spam	90	510

	Predicted: Spam	Predicted: Non-Spam
Actual: Spam	True Positive	False Negative
Actual: Non-Spam	False Positive	True Negative

Metric 3: Precision

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$= \frac{330}{330 + 90}$$

$$\approx 0.786$$

Cross-Validation

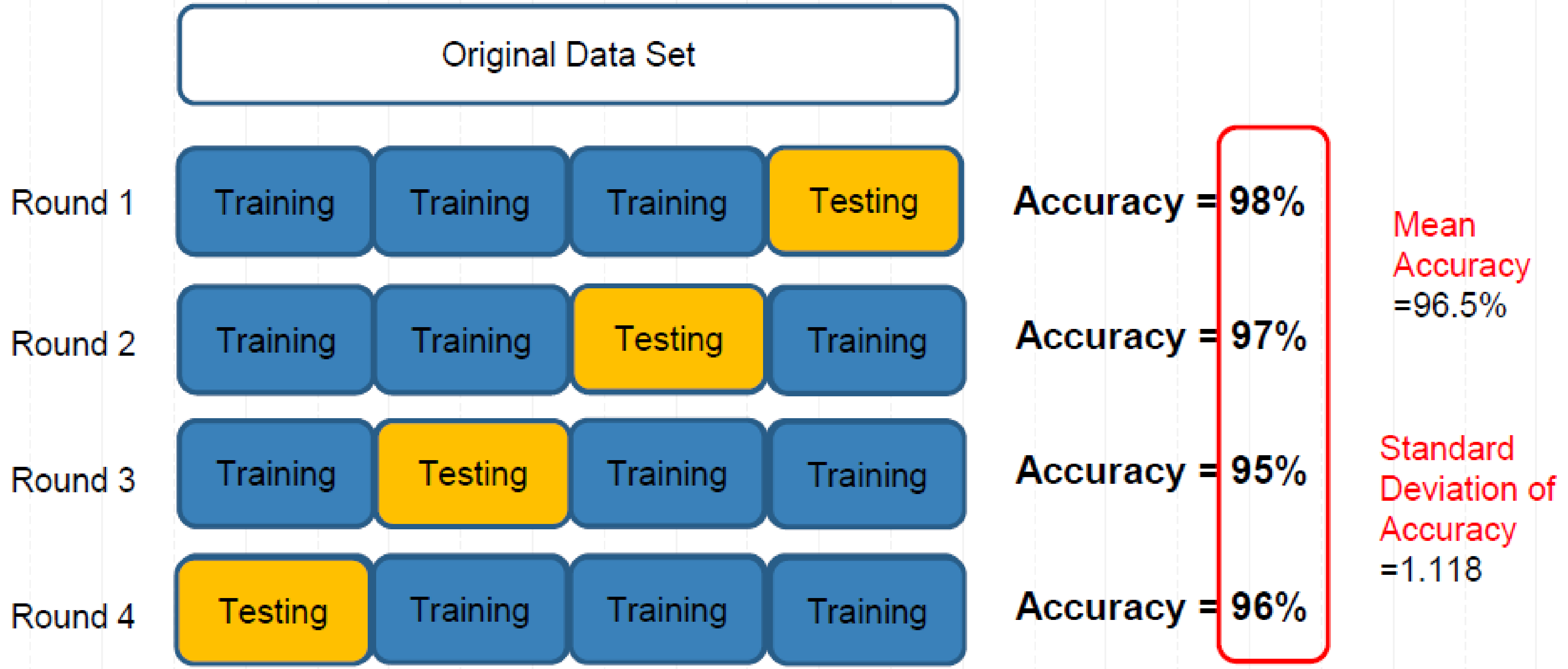
- **Cross-validation** is a resampling procedure used to evaluate machine learning models on a limited data sample.

1. Splitting the Data: The dataset is split into K equal-sized (or nearly equal-sized) subsets.

2. Training and Validation: The model is trained K times, each time using K-1 folds for training and the remaining 1 fold for validation.

3. Rotation: The validation fold is rotated such that each of the K folds is used exactly once as the validation set.

4. Averaging the Results: The performance metric (e.g., accuracy, precision, recall) is averaged across all K trials to give an overall performance estimate.

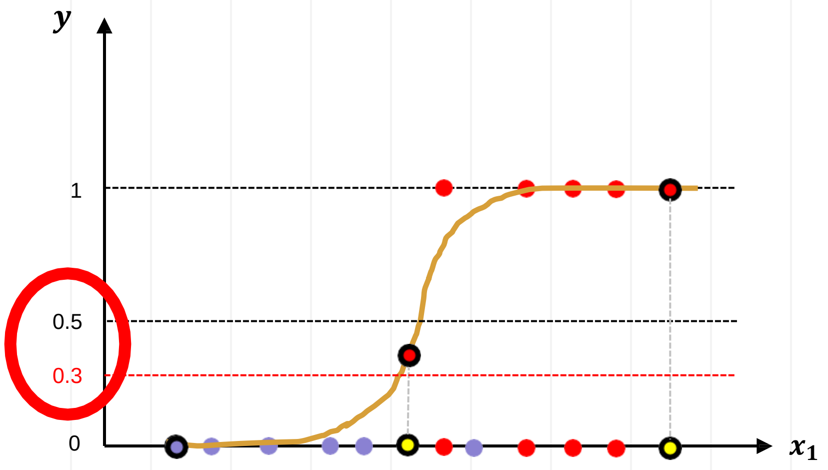


Four-Fold Cross-Validation

Hyperparameter Tuning

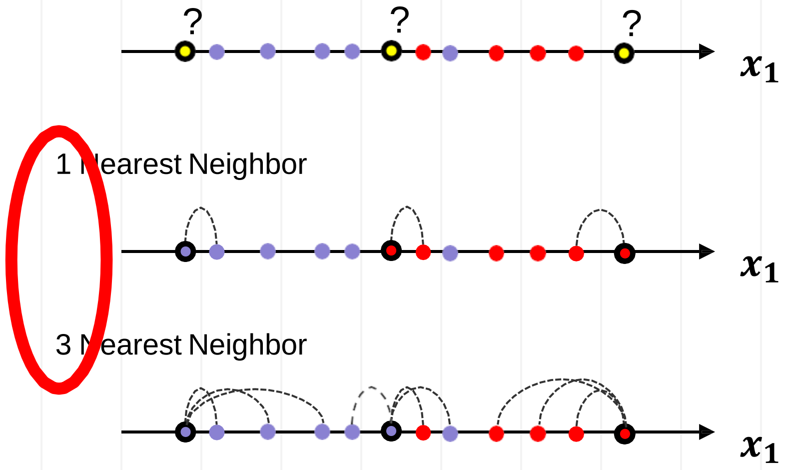
- **Hyperparameter**: model parameters specified in advance (before training)

What **threshold** to use?



Logistic Regression

What **k** to use?



K-Nearest Neighbors

Machine Learning Steps

- Gathering and loading data
- Exploring data (e.g., pandas and visualization)
- Transforming data (e.g., string to numeric)
- Splitting data for training and testing
- Choosing and creating a model
- Training
- Testing (evaluating accuracy)
- Tuning the model (hyperparameters)
- Making predictions on new data

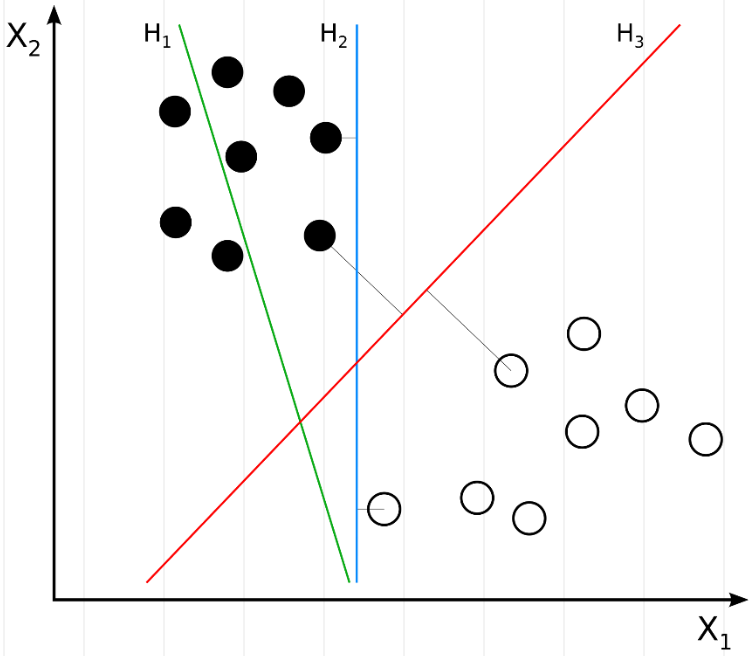
Supervised Learning Algorithms 2

Support Vector Machine

- **Objective**: To find a line or plane (in higher dimensions) that maximizes the distance between the line/plane and the nearest training data points of any class

- Key Concept:
  - The maximum-margin **hyperplane**.
  - **Margin:** the distance between the hyperplane and the closest data points from either class.
  - These closest points are called **support vectors**.

- margin is maximized equally for both classes.

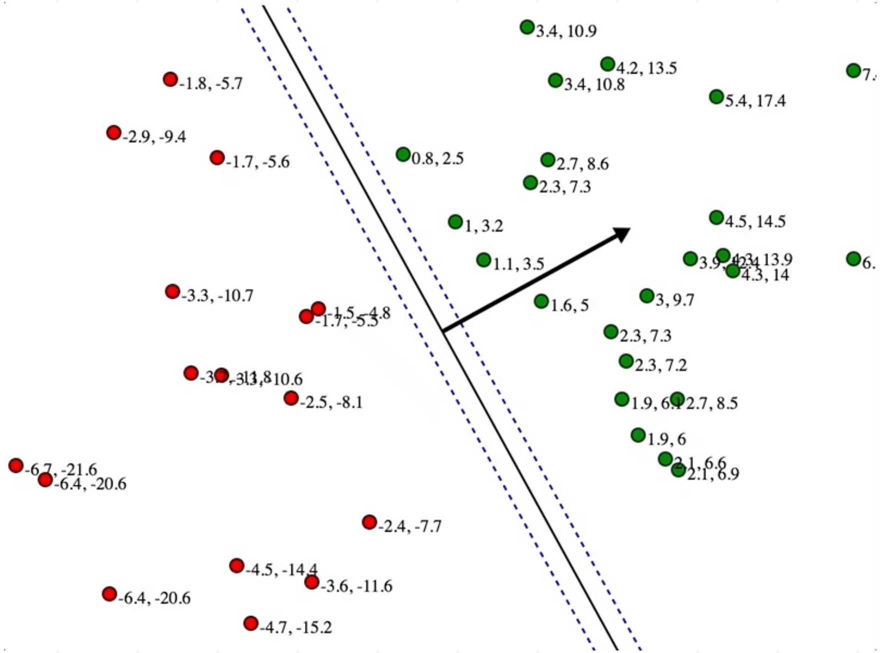


Support Vector Machine

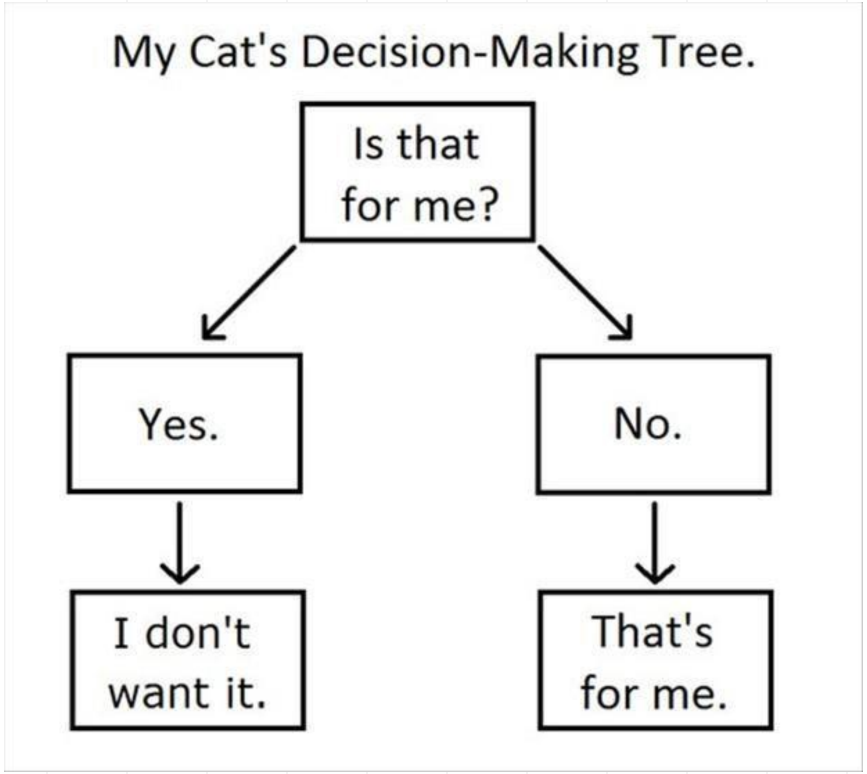
- Here, these numbers represent distances to the line

Initial State:

- The algorithm starts with an initial guess for the hyperplane.
- The algorithm iteratively adjusts the position and orientation of the hyperplane. Tweaking the parameters to improve the separation.



Decision Trees



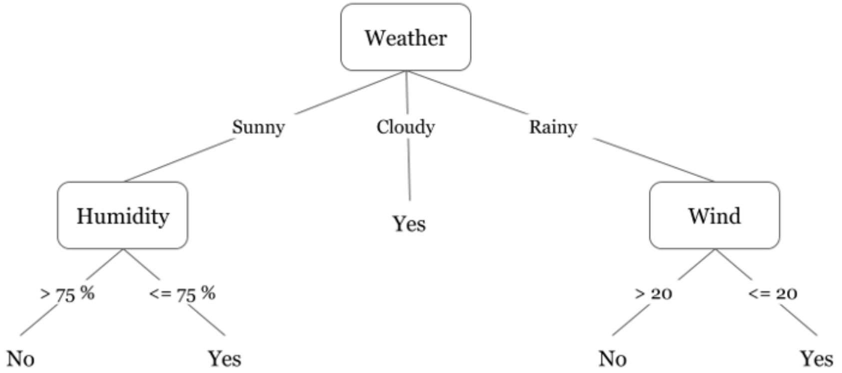
Decision Trees

Did I play badminton for each day of the week?

Weather	Humidity (%)	Wind Speed	Decision
Sunny	80	10	No
Sunny	60	5	Yes
Cloudy	70	15	Yes
Rainy	85	25	No
Rainy	70	10	Yes

Decision Tree:

- a model that makes decisions by splitting data into subsets based on feature values
- each node represents a feature
- each branch represents a decision rule
- each leaf node represents an outcome



Decision Trees

Choosing the Best Feature and threshold to Split:

- Gini impurity
- Entropy (Information Gain)

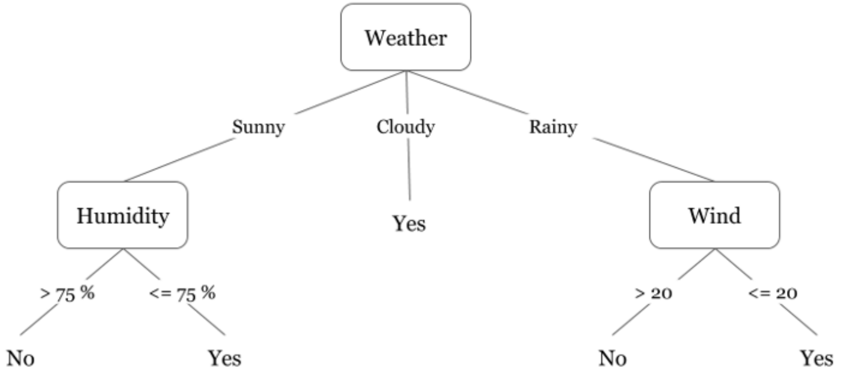
Recursive Splitting:

- Continue splitting at each node until stopping criteria are met (e.g., maximum depth, minimum samples per node)

Prediction

Did I play badminton for each day of the week?

Weather	Humidity (%)	Wind Speed	Decision
Sunny	80	10	No
Sunny	60	5	Yes
Cloudy	70	15	Yes
Rainy	85	25	No
Rainy	70	10	Yes



Decision Trees

Here's how we calculate Information Entropy for a dataset with C classes:

$$E = -\sum_{i}^{C} p_i \log_2 p_i$$

where p_i is the probability of randomly picking an element of class i (i.e. the proportion of the dataset made up of class i).

The Entropy of the root node:

There are 2 "No" and 3 "Yes" decisions.

$p_{No} = \frac{2}{5}, p_{Yes} = \frac{3}{5}$

Entropy $H(D) = -\left(\frac{2}{5}\log_2\frac{2}{5} + \frac{3}{5}\log_2\frac{3}{5}\right) \approx 0.971$

Weather	Humidity (%)	Wind Speed	Decision
Sunny	80	10	No
Sunny	60	5	Yes
Cloudy	70	15	Yes
Rainy	85	25	No
Rainy	70	10	Yes

Weighted average entropy for weather:

$H_{Weather}(D) = 0.8$

Information Gain

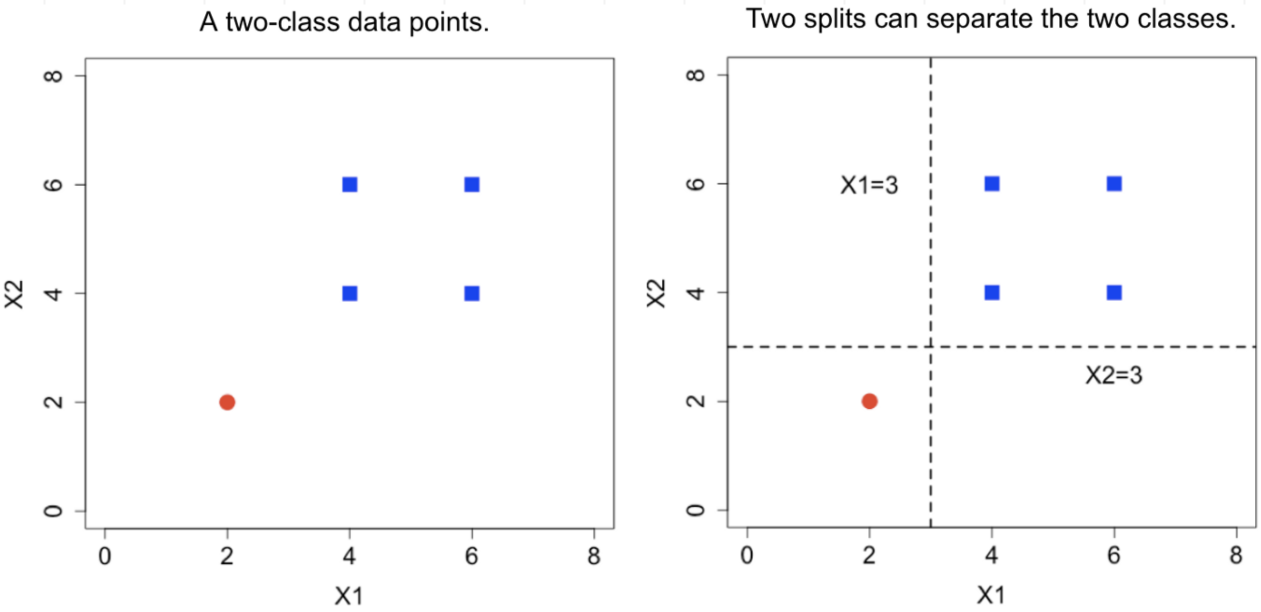
$IG(D, Weather) = H(D) - H_{Weather}(D)$

Decision Trees

First Split (X1 = 3):

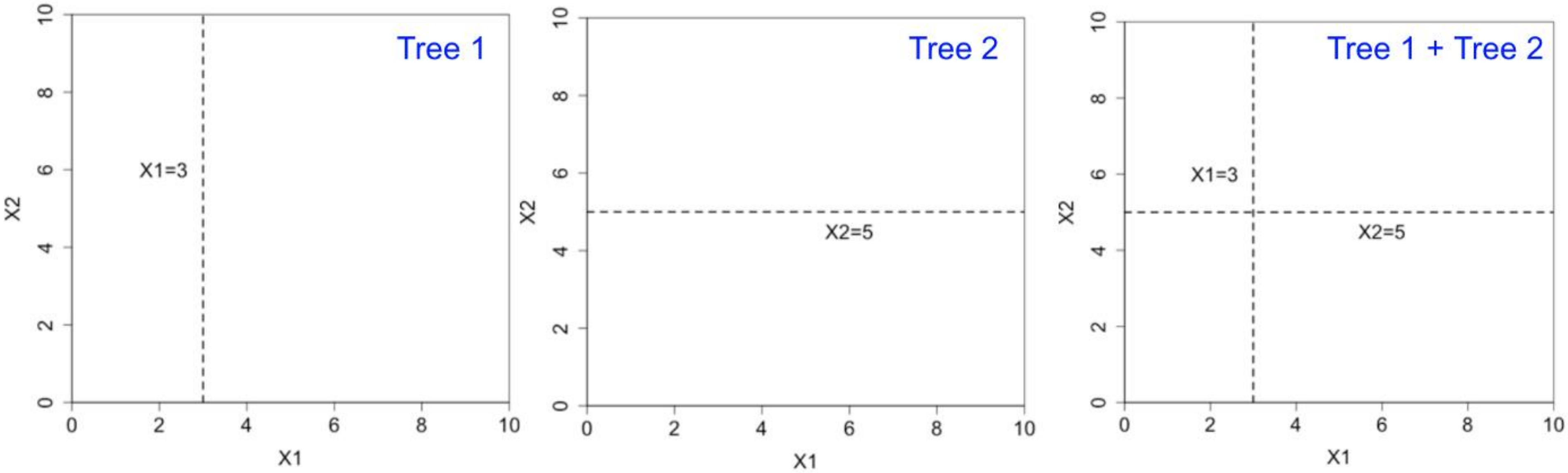
Second Split (X2 = 3)

This is another way to visualize the decision tree



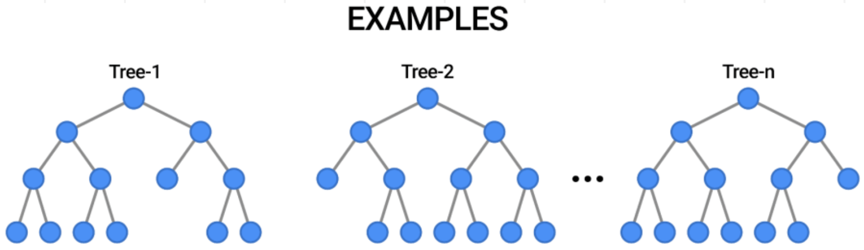
Ensemble Methods: combine the predictions of multiple models

Imagine if we had multiple (typically shallow) trees



Ensemble Methods: Random Forest

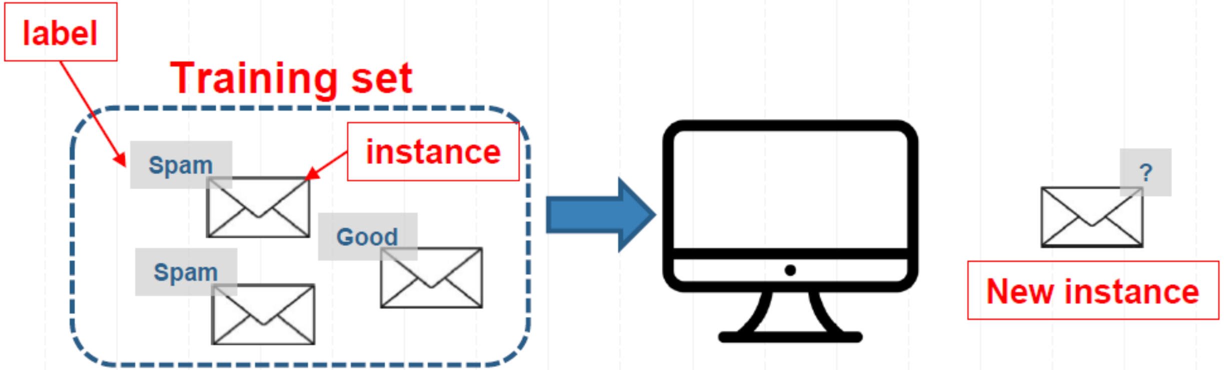
- Tree in the random forest is trained on a subset of the data and a random subset of the input features.
- So, each tree makes decisions based on a different subset of features.
- To predict a new data point, the random forest takes the majority vote of the predictions of all the individual decision trees.



3 Unsupervised Learning

Supervised Learning

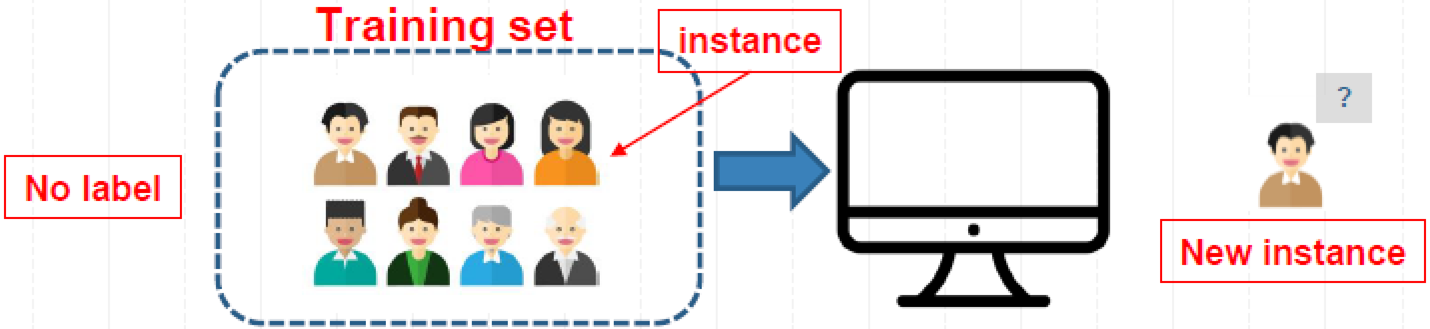
- Training



1. Supervised Learning: machine is trained with human supervision with a “teacher”, (the training set is labeled)

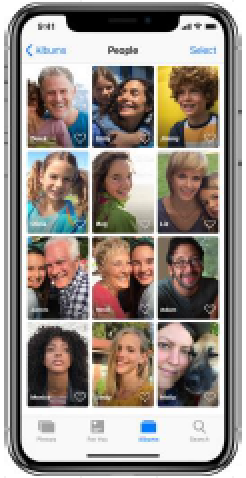
Unsupervised Learning

- Training



2. Unsupervised Learning: machine is trained without human supervision without a “teacher”, (the training set is not labeled)

Unsupervised Learning

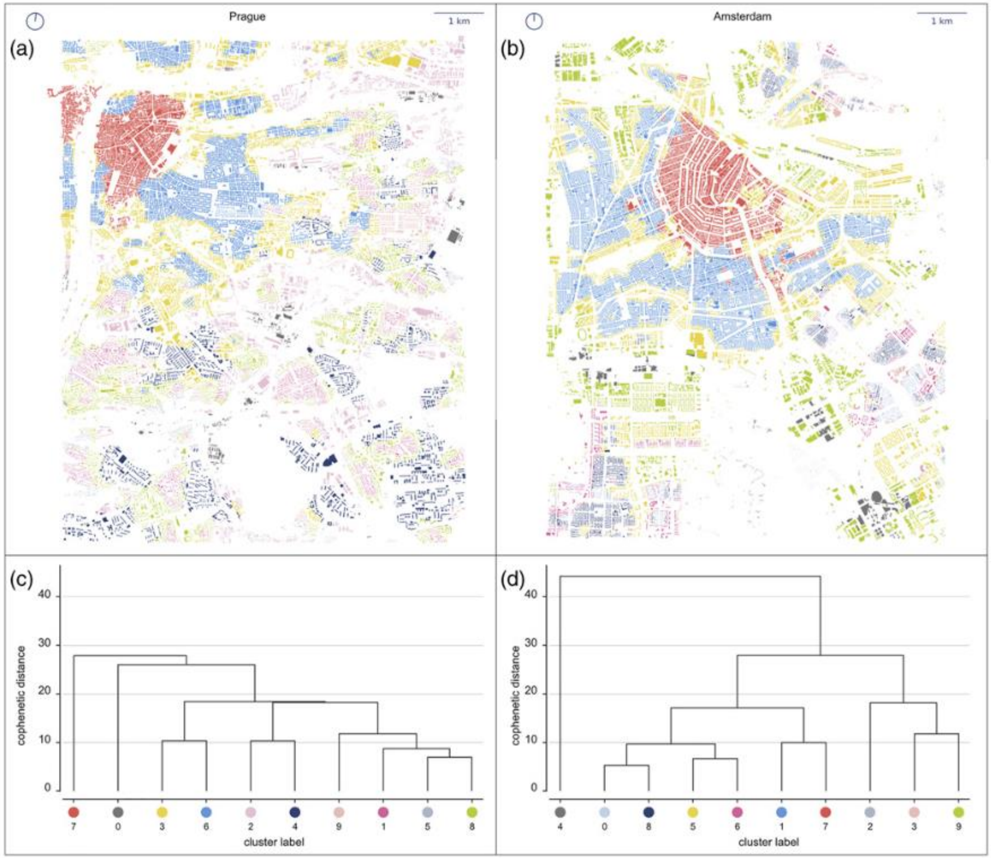


- **Clustering**: partition data into distinct groups
    - Label photos by people on iPhone
    - Recognize preference based on view/consumption history

- **Dimensionality Reduction**: Summarize a high-dimensional (many features) dataset with fewer features
    - Visualization in 2 dimension
    - Compress data, save storage space and processing time

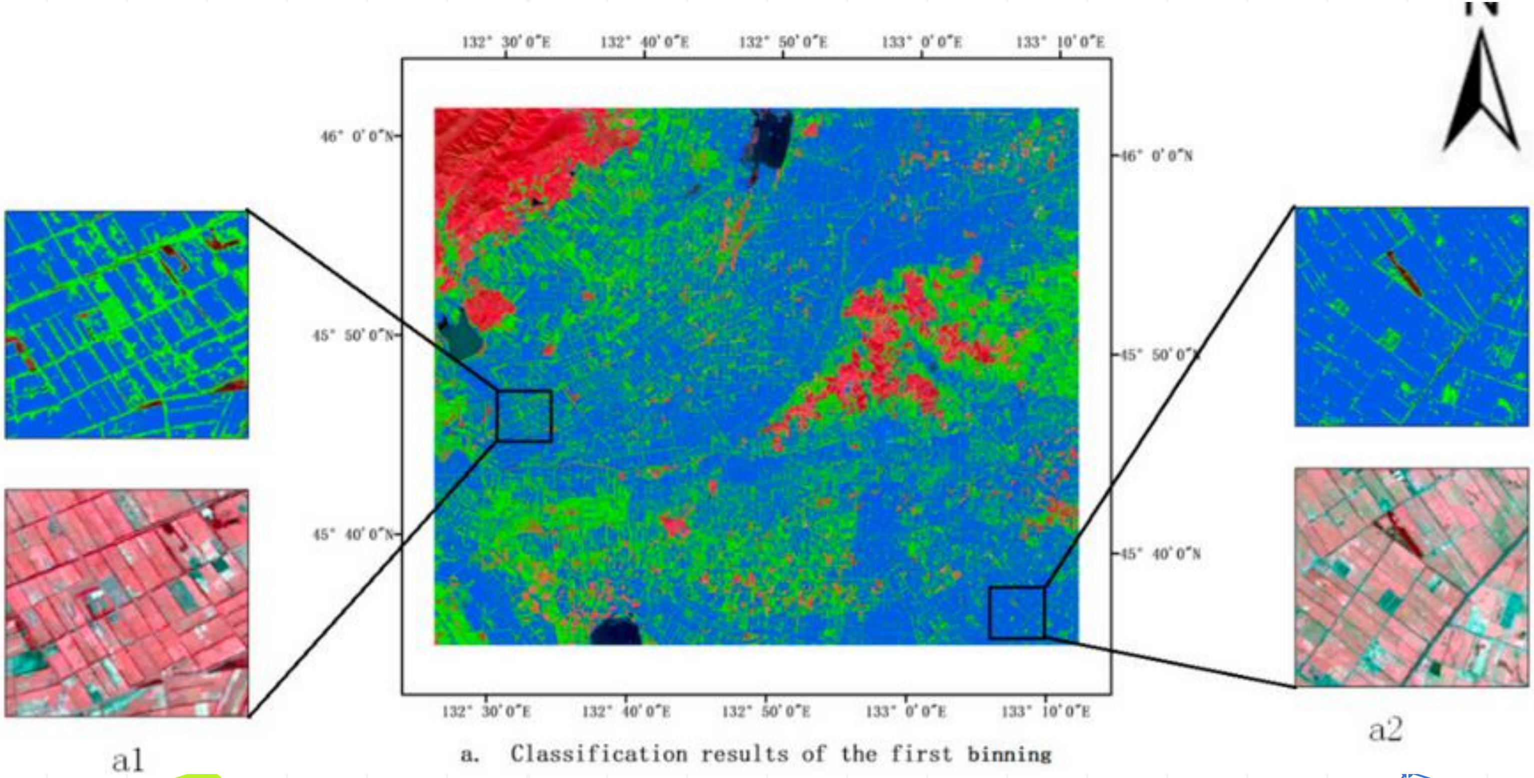
Clustering: partition data into groups

Spatial distribution of detected clusters in central Prague (a) and central Amsterdam (b) accompanied by dendrograms representing the results of Ward's hierarchical clustering of urban form types in Prague (c) and Amsterdam (d).



Fleischmann, M., Feliciotti, A., Romice, O., & Porta, S. (2022). Methodological foundation of a numerical taxonomy of urban form. *Environment and Planning B: Urban Analytics and City Science*, *49*(4), 1283-1299.

Clustering: partition data into groups
– IsoData imagery classification



a. Classification results of the first binning

Dimensionality Reduction: Summarize a high-dimensional (many features) dataset with fewer features

Table 4: Summary of rotated factor loadings based on PCA for 8 items on Perception of cities

	Rotated factors loadings	
	Factor 1	Factor 2
1. The city is a place with a high level of services and infrastructure.	-0.01	**0.59**
2. The city is a place with jobs and economic opportunity.	0.08	**0.67**
3. Urban way of life (*mazoea*) is a good way of life.	0.13	**0.51**
4. The city is a place with secure land tenure.	-0.28	**0.62**
5. The city is a place of chaos.	**0.71**	0.08
6. The city is a place of poverty.	**0.65**	-0.24
7. The city is a place with people who are not like me	**0.68**	0.07
8. Urban ways of life (*mazoea*) are not compatible with my tribal culture	0.37	0.32
Eigenvalue	1.63	1.60
% of total variation	20.23%	20.14%

Note: Factor loadings over 0.50 appear in bold; PCA using Varimax rotation with Kaiser's criterion.

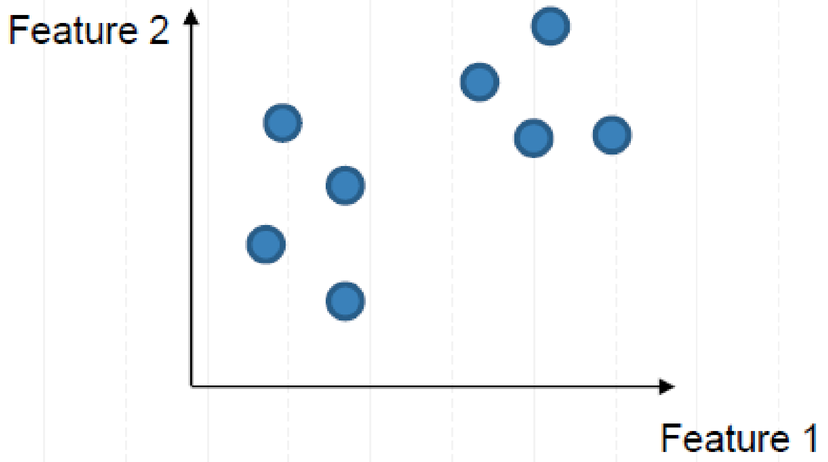
Unsupervised Learning

- **Clustering**: partition data into distinct groups
  - No Labels
  - How to find clusters?
  - How many clusters (k) to make?

K-Means Clustering

- **Step 1**: Choose k

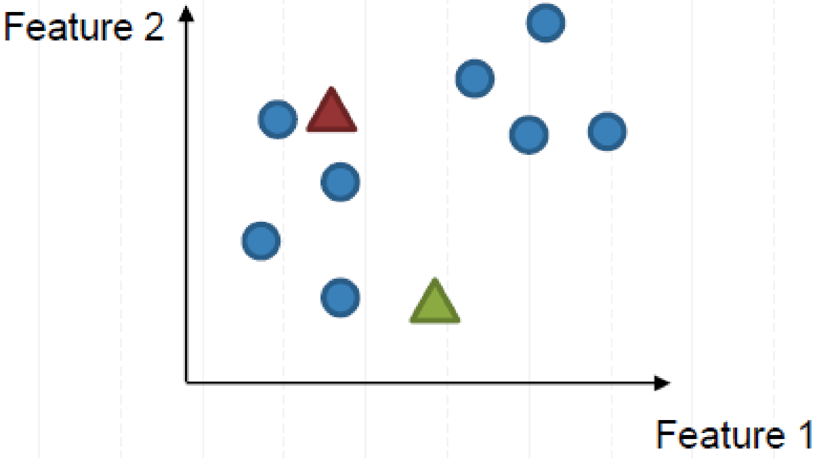
k = 2



K-Means Clustering

- **Step 2:** Randomly create k centroid points

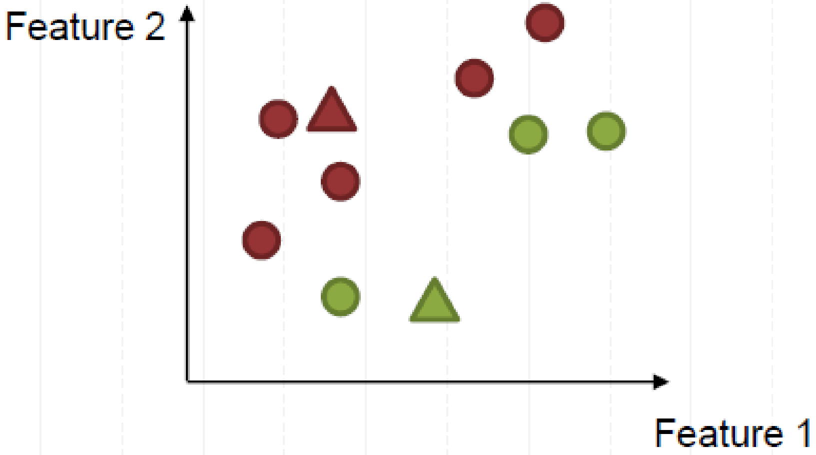
k = 2



K-Means Clustering

- **Step 3:** Cluster samples into closest centroid points

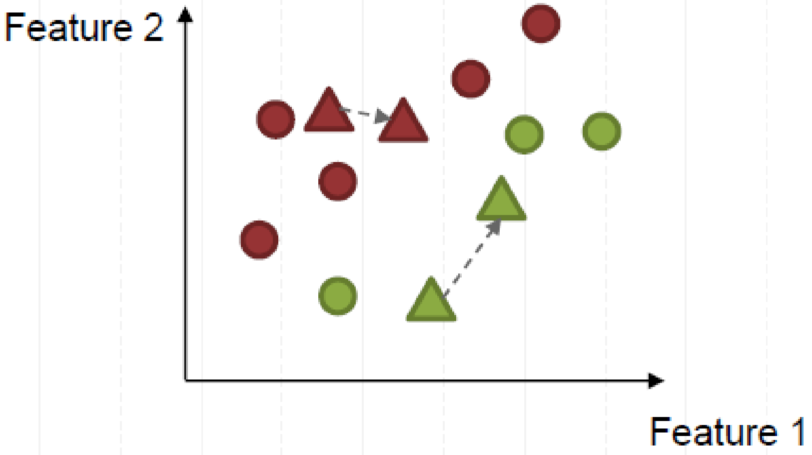
k = 2



K-Means Clustering

k = 2

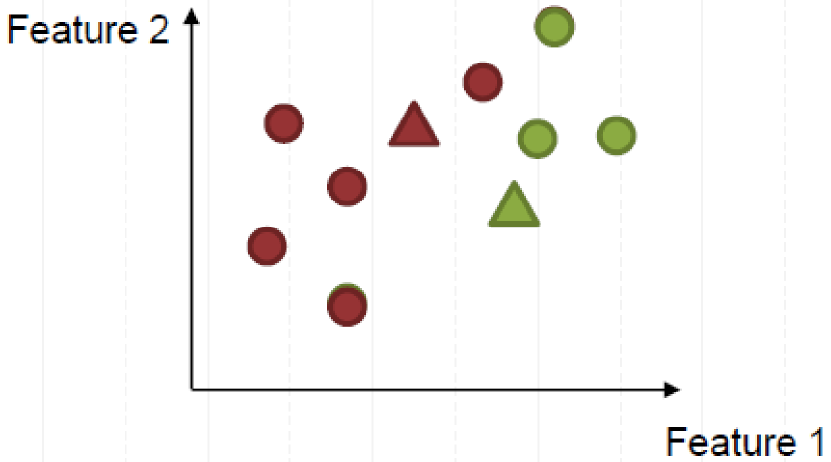
- **Step 4**: Calculate for each cluster a new centroid point based on sample features.



K-Means Clustering

k = 2

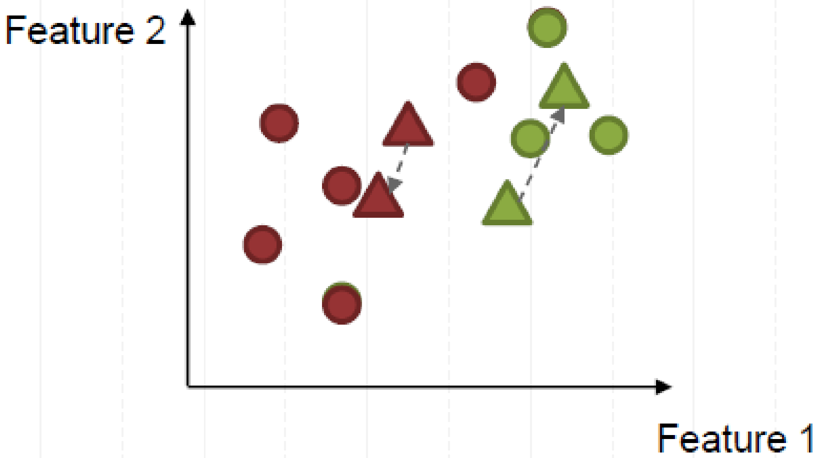
- **Step 5**: Reclassify the samples based on new centroid points



K-Means Clustering

k = 2

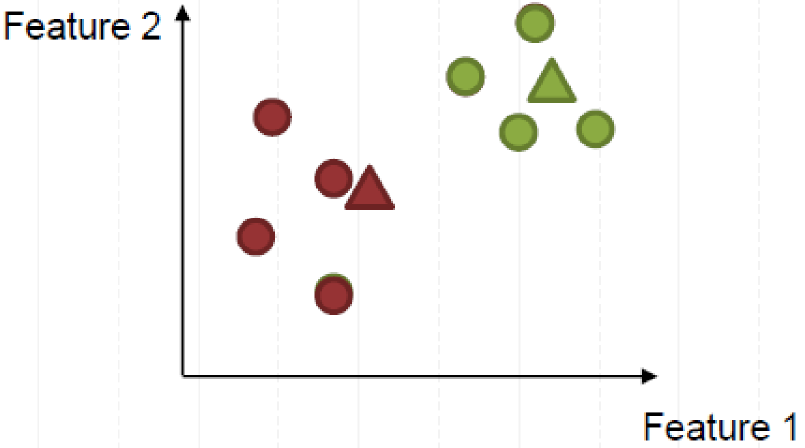
- Repeat the process until no changes in clustering



K-Means Clustering

k = 2

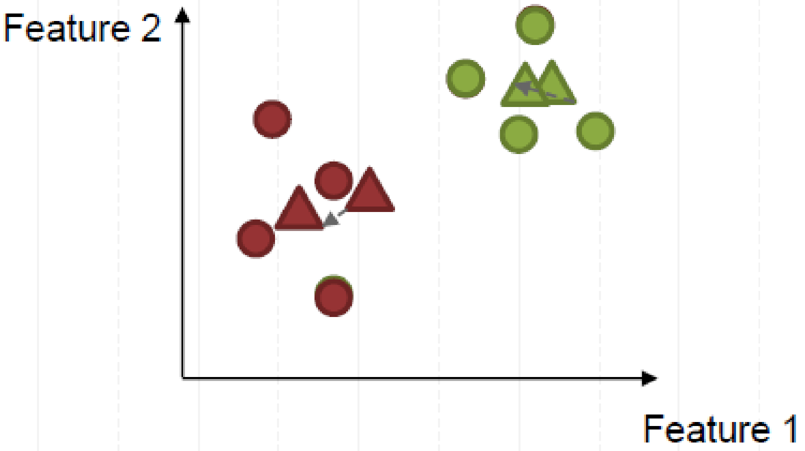
- Repeat the process until no changes in clustering



K-Means Clustering

k = 2

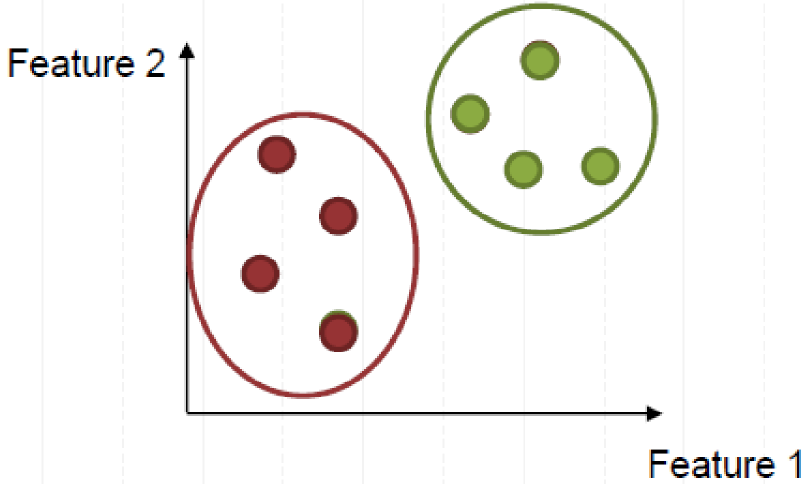
- Repeat the process until no changes in clustering



K-Means Clustering

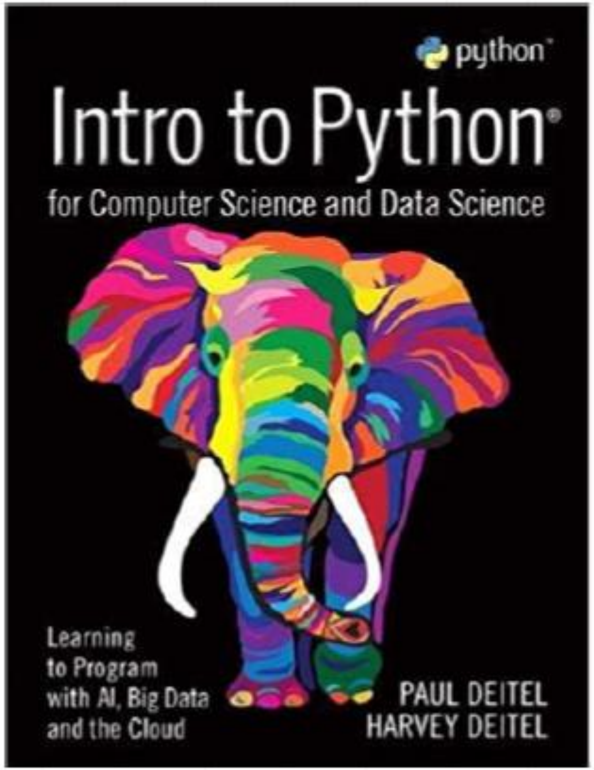
k = 2

- **Repeat the process until no changes in clustering**



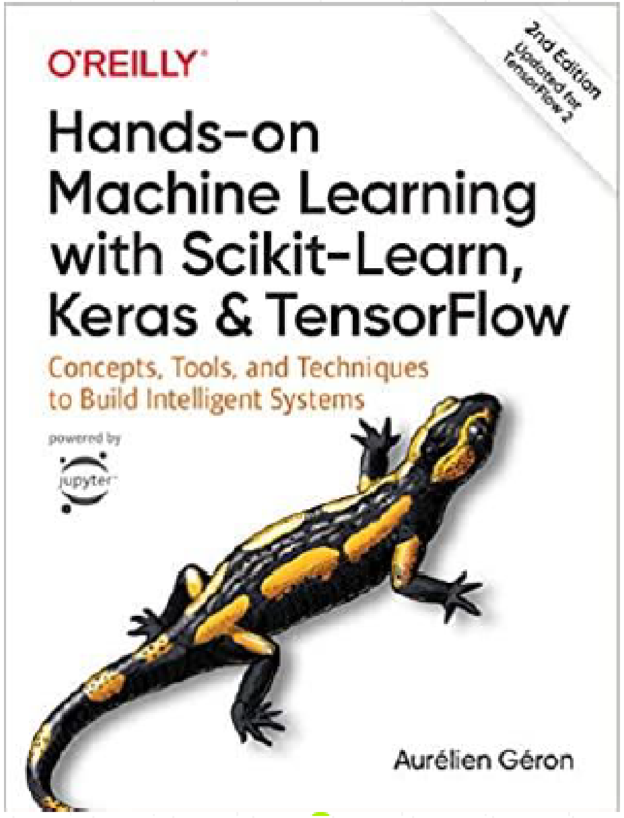
Online Courses

- Machine Learning Crash Course with TensorFlow APIs
https://developers.google.com/machine-learning/crash-course



- *Intro to Python for Computer Science and Data Science: Learning to Program with AI, Big Data and The Cloud*

  Paul J. Deitel (Author), HarveyDeitel (Author)



- *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*

  Aurélien Géron (Author)

Q&A

Any questions?

You can find me at

wl563@cornell.edu