LECTURE 3 PREDICTIVE ANALYTICS II

LEK HSIANG HUI

OUTLINE

Bayesian Methods

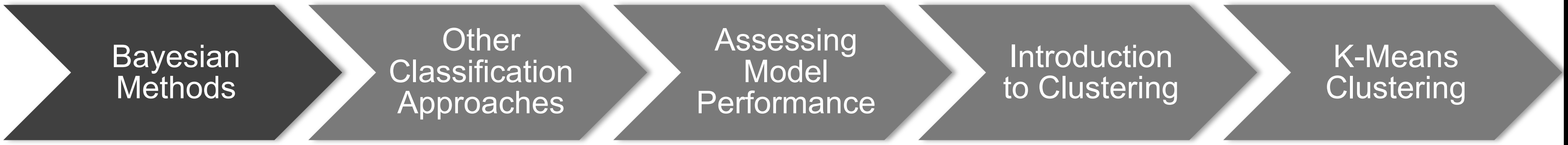
Other Classification Approaches

Assessing Model Performance

Introduction to Clustering

K-Means Clustering

BAYESIAN METHODS



Bayesian Methods

Other Classification Approaches

Assessing Model Performance

Introduction to Clustering

K-Means Clustering

BAYESIAN METHODS

Bayesian methods belong to the family of probabilistic classification models

Denote:

- *x* = explanatory variables (predictors)
- *y* = target class

How to do classification?

- *P(y|x)* = probability that the instance belong to class *y* given the data instance *x*
- This is also known as the posterior probability
- E.g. assume there're 3 classes (*c1*, *c2*, *c3*), we do classification by calculating *P(y=c1|x)* , *P(y=c2|x)*, *P(y=c3|x)*
- The instance is then classified *c1*, *c2*, *c3* by finding the maximum of these posterior probabilities

$$\boxed{y_{\max} = \arg\max_{y \in \{c1, c2, c3\}} P(y \mid \mathrm{x})}$$

Want y_{max}

BAYESIAN METHODS

To calculate the posterior probability *P(y|x)*, Bayesian methods make use of the Bayes' theorem to transform this into another form

Bayes' theorem:

$$P(y \mid \mathrm{x}) = \frac{P(\mathrm{x} \mid y)P(y)}{\sum_{l=1}^{H} P(\mathrm{x} \mid y)P(y)} = \frac{P(\mathrm{x} \mid y)P(y)}{P(\mathrm{x})}$$

where

- *x* = predictors
- *y* = target
- *H* = number of distinct values for *y*

$$P(y \mid \mathrm{x}) = \frac{P(\mathrm{x} \mid y)P(y)}{P(\mathrm{x})}$$

BAYESIAN METHODS

How to do classification (same example)?

- E.g. assume there’re 3 classes (*c1*, *c2*, *c3*), we do classification by calculating *P(y=c1|x)* , *P(y=c2|x)*, *P(y=c3|x)*
- The instance is then classified *c1*, *c2*, *c3* by finding the maximum of these posterior probabilities **which we will transform using the Bayes’ theorem**

$$\left.\begin{aligned}
P(y = c1 \mid \mathrm{x}) &= \frac{P(\mathrm{x} \mid y = c1)P(y = c1)}{P(\mathrm{x})} \\
P(y = c2 \mid \mathrm{x}) &= \frac{P(\mathrm{x} \mid y = c2)P(y = c2)}{P(\mathrm{x})} \\
P(y = c3 \mid \mathrm{x}) &= \frac{P(\mathrm{x} \mid y = c3)P(y = c3)}{P(\mathrm{x})}
\end{aligned}\right\}$$

Want to find which one is the largest

$$\boxed{P(y \mid \mathrm{x}) = \frac{P(\mathrm{x} \mid y)P(y)}{P(\mathrm{x})}}$$

BAYESIAN METHODS

How to do classification (same example)?

- E.g. assume there're 3 classes (*c1*, *c2*, *c3*), we do classification by calculating *P(y=c1|x)* , *P(y=c2|x)*, *P(y=c3|x)*
- The instance is then classified *c1*, *c2*, *c3* by finding the maximum of these posterior probabilities **which we will transform using the Bayes' theorem**

$$\left.\begin{aligned}
P(y=c1 \mid \mathrm{x}) &= \frac{P(\mathrm{x} \mid y=c1)P(y=c1)}{P(\mathrm{x})} \\
P(y=c2 \mid \mathrm{x}) &= \frac{P(\mathrm{x} \mid y=c2)P(y=c2)}{P(\mathrm{x})} \\
P(y=c3 \mid \mathrm{x}) &= \frac{P(\mathrm{x} \mid y=c3)P(y=c3)}{P(\mathrm{x})}
\end{aligned}\right\} \quad \underset{y \in \{c1,c2,c3\}}{\arg\max}\, P(y \mid \mathrm{x}) = \underset{y \in \{c1,c2,c3\}}{\arg\max} \frac{P(\mathrm{x} \mid y)P(y)}{P(\mathrm{x})}$$

$$P(y \mid \mathrm{x}) = \frac{P(\mathrm{x} \mid y)P(y)}{P(\mathrm{x})}$$

BAYESIAN METHODS

How to do classification (same example)?

- E.g. assume there're 3 classes (*c1*, *c2*, *c3*), we do classification by calculating *P(y=c1|x)* , *P(y=c2|x)*, *P(y=c3|x)*
- The instance is then classified *c1*, *c2*, *c3* by finding the maximum of these posterior probabilities **which we will transform using the Bayes' theorem**

$$P(y = c1 \mid \mathrm{x}) = \frac{P(\mathrm{x} \mid y = c1)P(y = c1)}{\boxed{P(\mathrm{x})}}$$

$$P(y = c2 \mid \mathrm{x}) = \frac{P(\mathrm{x} \mid y = c2)P(y = c2)}{\boxed{P(\mathrm{x})}}$$

$$P(y = c3 \mid \mathrm{x}) = \frac{P(\mathrm{x} \mid y = c3)P(y = c3)}{\boxed{P(\mathrm{x})}}$$

Notice that all of them are divided by the same denominator

i.e. doesn't affect the y_{max} decision

$$\boxed{P(y \mid \mathrm{x}) = \frac{P(\mathrm{x} \mid y)P(y)}{P(\mathrm{x})}}$$

BAYESIAN METHODS

How to do classification (same example)?

- E.g. assume there’re 3 classes (*c1*, *c2*, *c3*), we do classification by calculating *P(y=c1|x)* , *P(y=c2|x)*, *P(y=c3|x)*
- The instance is then classified *c1*, *c2*, *c3* by finding the maximum of these posterior probabilities **which we will transform using the Bayes’ theorem**

$$\arg\max_{y \in \{c1,c2,c3\}} P(y \mid \mathrm{x}) = \arg\max_{y \in \{c1,c2,c3\}} \frac{P(\mathrm{x} \mid y)P(y)}{P(\mathrm{x})} = \arg\max_{y \in \{c1,c2,c3\}} \boxed{P(\mathrm{x} \mid y)}\boxed{P(y)}$$

Conditional Probability / Likelihood function

Prior probability

NAÏVE BAYES CLASSIFIERS

$$\underset{y\in\{c1,c2,c3\}}{\operatorname{argmax}} P(\mathrm{x} \mid y)P(y)$$

Using chain rule:

$$\begin{aligned}
P(\mathrm{x} \mid y)P(y) &= P(y)P(x_1,\ldots,x_n \mid y) \\
&= P(y)P(x_1 \mid y)P(x_2,\ldots,x_n \mid y,x_1) \\
&= P(y)P(x_1 \mid y)P(x_2 \mid y,x_1)P(x_3,\ldots,x_n \mid y,x_1,x_2) \\
&= \ldots \\
&= P(y)P(x_1 \mid y)P(x_2 \mid y,x_1)\ldots P(x_n \mid y,x_1,x_2,\ldots,x_{n-1})
\end{aligned}$$

NAÏVE BAYES CLASSIFIERS

Naïve Bayes assume conditional independence (each attribute *x_i* is conditionally independent of every other attribute *x_j* for *$i \neq j$*)

$$P(x_2 \mid y, x_1) = P(x_2 \mid y)$$

$$P(x_3 \mid y, x_1, x_2) = P(x_3 \mid y)$$

$$P(x_i \mid y, x_1, \ldots, x_{i-1}) = P(x_i \mid y)$$

After conditional independence assumption:

$$P(\mathrm{x} \mid y) P(y) = P(y) P(x_1 \mid y) P(x_2 \mid y, x_1) \ldots P(x_n \mid y, x_1, x_2, \ldots, x_{n-1})$$

$$P(\mathrm{x} \mid y) P(y) = P(y) P(x_1 \mid y) P(x_2 \mid y) \ldots P(x_n \mid y)$$

NAÏVE BAYES CLASSIFIERS

Conditional probabilities values are calculated from the available data:

- For categorical/discrete numerical attributes:

$$P(x_j \mid y) = P(x_j = r_{jk} \mid y = v_h) = \frac{s_{jhk}}{m_h}$$

where

s_{jhk} = number of class v_h for which the variable takes value r_{jk} (based on the training data)

m_h = total number of class v_h (based on the training data)

NAÏVE BAYES CLASSIFIERS

Conditional probabilities values are calculated from the available data:

- For numerical attributes:
  - $P(x_j \mid y)$ is estimated by making some assumption regarding its distribution
  - Often, this conditional probability is assumed to follow the Gaussian distribution and we compute the Gaussian density function

NAÏVE BAYES EXAMPLE

Outlook	Temperature	Humidity	Wind	Play
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Outlook = Sunny
Temp. = Cool
Humidity = High
Wind = Strong

Play = ?

NAÏVE BAYES EXAMPLE

Outlook	Temp.	Hum.	Wind	Play
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

14 rows

Outlook = Sunny
Temp. = Cool
Humidity = High
Wind = Strong

Play = ?

$$P\left(\{Sunny,Cool,High,Strong\}\mid y\right)P\left(y\right)=P\left(y\right)P(Sunny\mid y)P(Cool\mid y)P(High\mid y)P(Strong\mid y)$$

Prior probabilities

$$P\left(y=Yes\right)=\frac{9}{14} \qquad P\left(y=No\right)=\frac{5}{14}$$

NAÏVE BAYES EXAMPLE

Outlook	Temp.	Hum.	Wind	Play
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Conditional probabilities

$$P(Outlook = Sunny \mid y = Yes) = \frac{2}{9}$$

$$P(Outlook = Sunny \mid y = No) = \frac{3}{5}$$

$$P(\{Sunny, Cool, High, Strong\} \mid y) P(y) = P(y) P(Sunny \mid y) P(Cool \mid y) P(High \mid y) P(Strong \mid y)$$

Prior probabilities

$$P(y = Yes) = \frac{9}{14} \qquad P(y = No) = \frac{5}{14}$$

NAÏVE BAYES EXAMPLE

Outlook	Temp.	Hum.	Wind	Play
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

$$P(Yes)P(Sunny \mid Yes)P(Cool \mid Yes)P(High \mid Yes)P(Strong \mid Yes)$$

$$=\left(\frac{9}{14}\right)\left(\frac{2}{9}\right)\left(\frac{3}{9}\right)\left(\frac{3}{9}\right)\left(\frac{3}{9}\right)=0.00529$$

$$P(No)P(Sunny \mid No)P(Cool \mid No)P(High \mid No)P(Strong \mid No)$$

$$=\left(\frac{5}{14}\right)\left(\frac{3}{5}\right)\left(\frac{1}{5}\right)\left(\frac{4}{5}\right)\left(\frac{3}{5}\right)=0.02057$$

{Outlook = Sunny, Temp. = Cool, Humidity = High, Wind = Strong}

Play = No

NAÏVE BAYES CLASSIFIERS

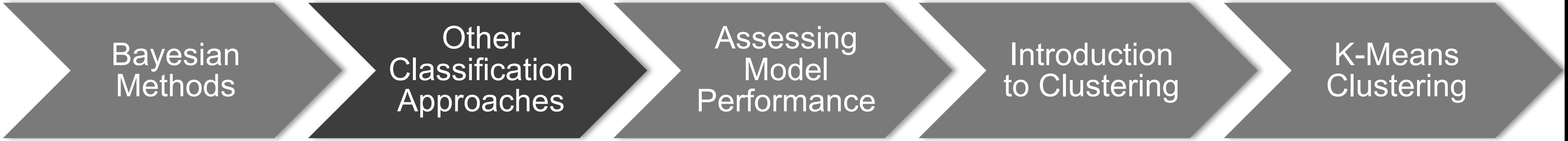
Conditional independence assumption:

- Training becomes very easy and fast - just need to consider each attribute in each class separately and build up a table of the prior probabilities and conditional probabilities
- Testing is also easy – just look up the tables and calculate the probability for each class

Naïve Bayes

- A popular machine learning algorithm which is fast with competitive performance (accuracy) compared to the other state-of-the-art classifiers

OTHER CLASSIFICATION APPROACHES



Bayesian Methods

Other Classification Approaches

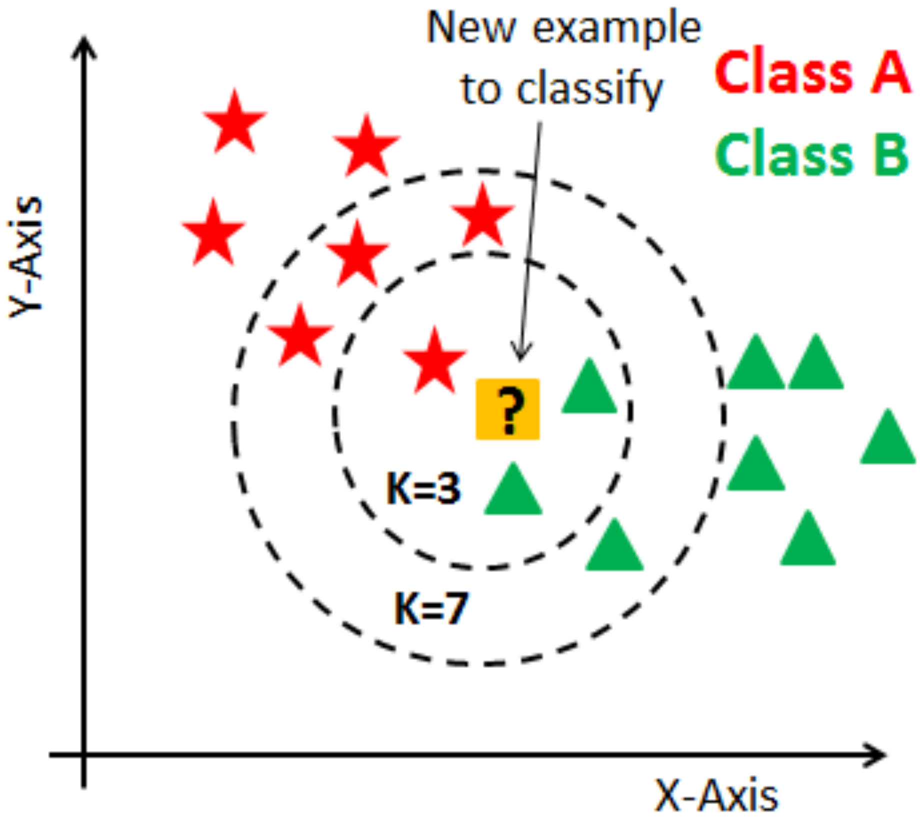
Assessing Model Performance

Introduction to Clustering

K-Means Clustering

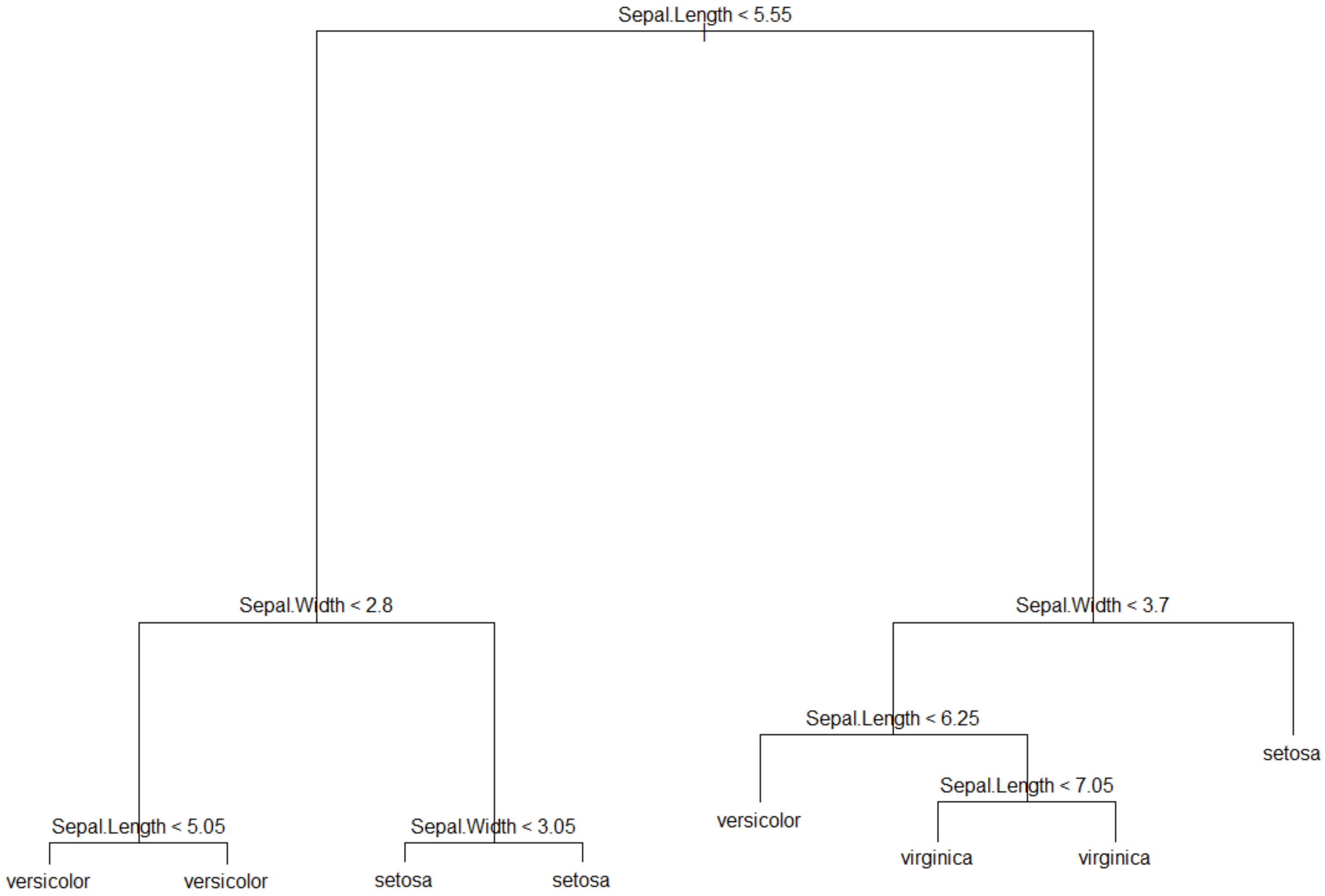
OTHER CLASSIFICATION APPROACHES

K-Nearest Neighbor (KNN)



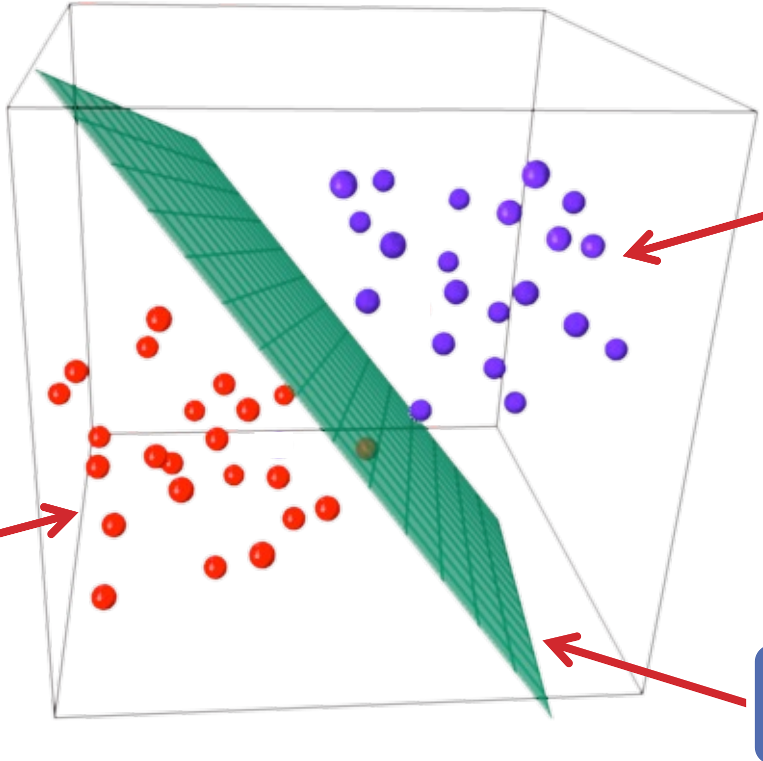
OTHER CLASSIFICATION APPROACHES

Decision Tree



OTHER CLASSIFICATION APPROACHES

Support Vector Machine (SVM)



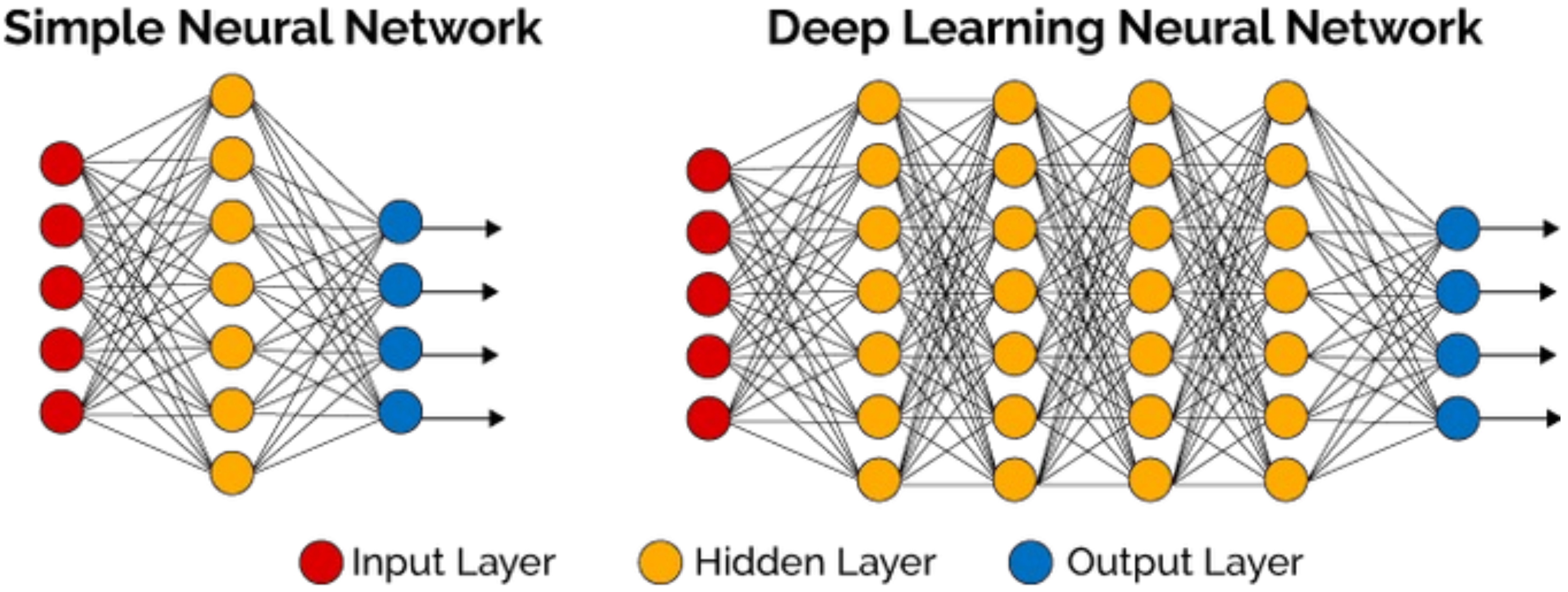
Class 1

Class 2

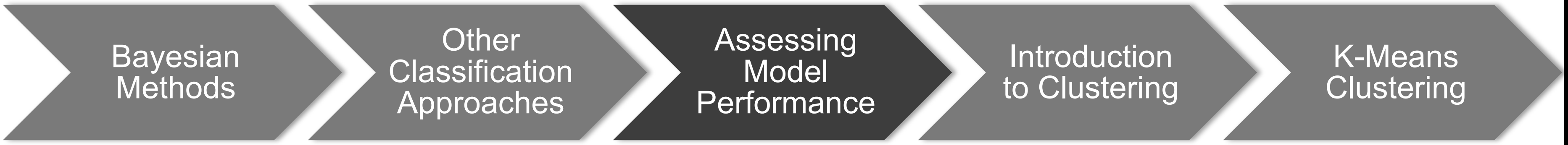
Separating Hyperplane

OTHER CLASSIFICATION APPROACHES

Neural Network



ASSESSING MODEL PERFORMANCE



Bayesian Methods

Other Classification Approaches

Assessing Model Performance

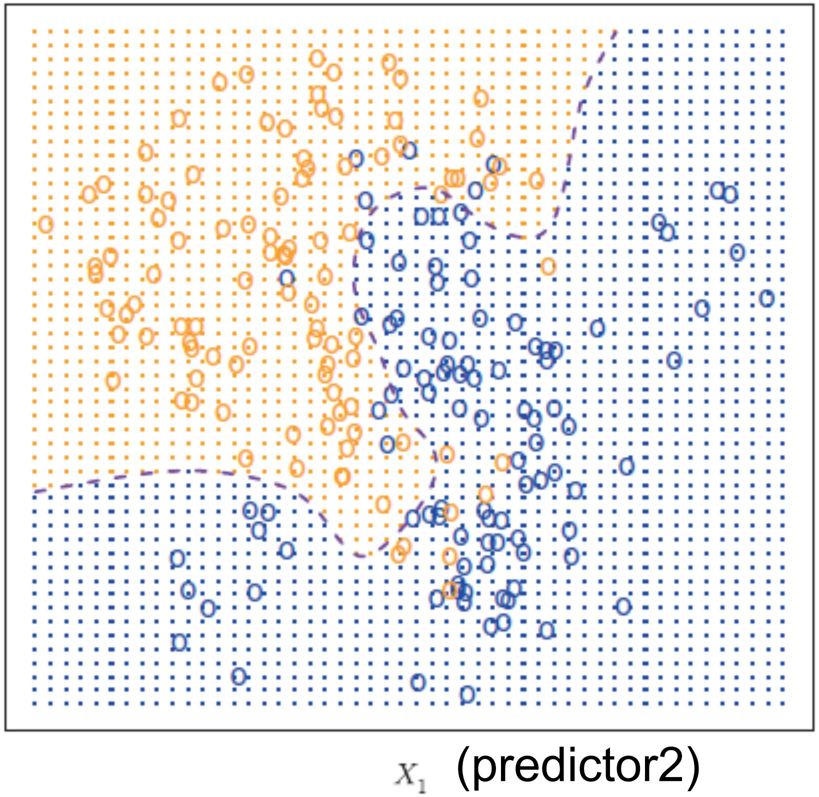
Introduction to Clustering

K-Means Clustering

CLASSIFICATION SETTINGS

In the classification setting, we have a list of pre-defined class/categories that we want to classify each observation to

(predictor1)



(predictor2)

For any point, is it the orange class or blue class?

ACCURACY

For classification, one common measure for assessing the model accuracy is Accuracy

$$Accuracy = \frac{1}{n}\sum_{i=1}^{n} I(y_i = \hat{y}_i)$$

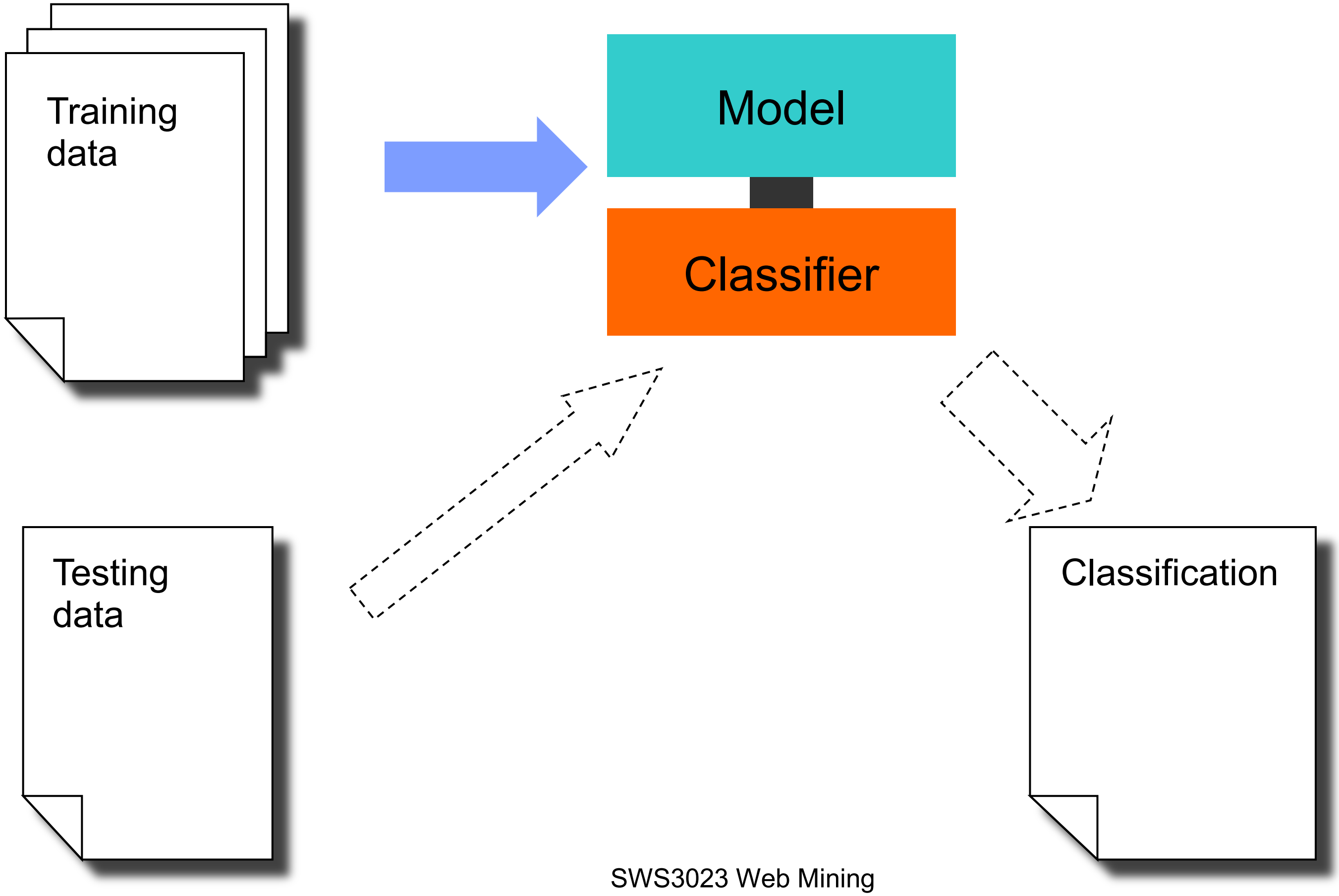
- where

$$I(y_i = \hat{y}_i) = \begin{cases} 1 & \text{if } (y_i = \hat{y}_i) \\ 0 & \text{otherwise} \end{cases}$$

Basically the proportion of correct classification instances over all the instances

MODEL EVALUATION

So far what we have seen is to:



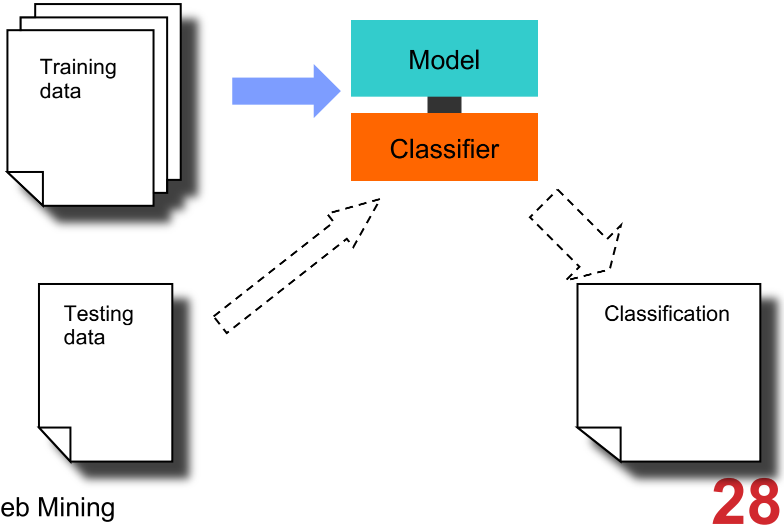
MODEL EVALUATION

Problems with such an approach:

- Difficult to evaluate the accuracy/performance of a model or a learning method (since testing data is not readily available)
- Performance of a model depends heavily on the training data

Solution:

- Make use of Resampling Methods



RESAMPLING METHODS

Idea:

- Find the performance by running the experiment multiple times (using the same training set *S*)
- For each run:
  - Draw a subset of *S* for training (S_{train}) and a subset of *S* for testing (S_{test})
  - Generate a model using S_{train} and evaluate on S_{test}
- Performance = average of the multiple runs

RESAMPLING METHODS

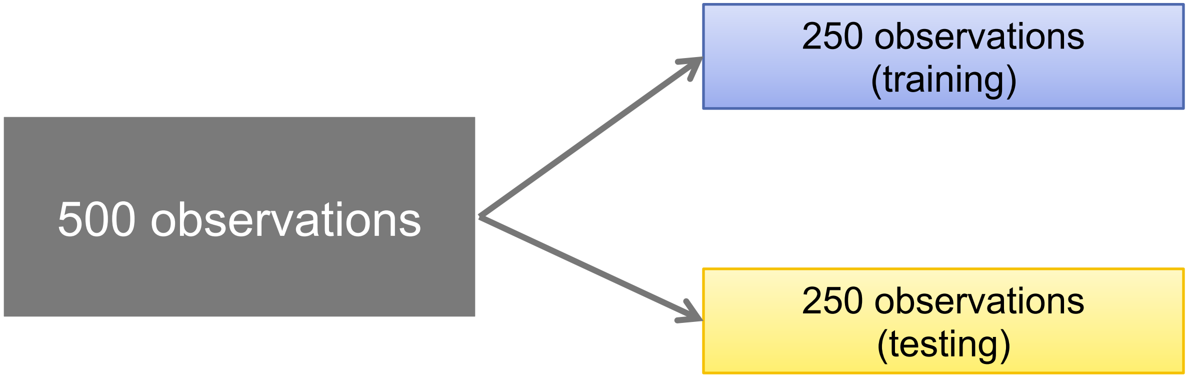
Advantages:

- Can evaluate the model even though we do not have any testing data
- Can ensure that the training data we are using is good (i.e. not biased or noisy)
- Allows us to better evaluate a model’s performance

VALIDATION SET APPROACH

Randomly split the training data into <u>training</u> and <u>validation</u> (testing) datasets

- 50% for training, 50% for testing (sample with replacement)



Train a model using this "new" training dataset, and evaluate the performance on this "new" testing dataset

VALIDATION SET APPROACH

Advantages:

- Simple
- Easy to implement

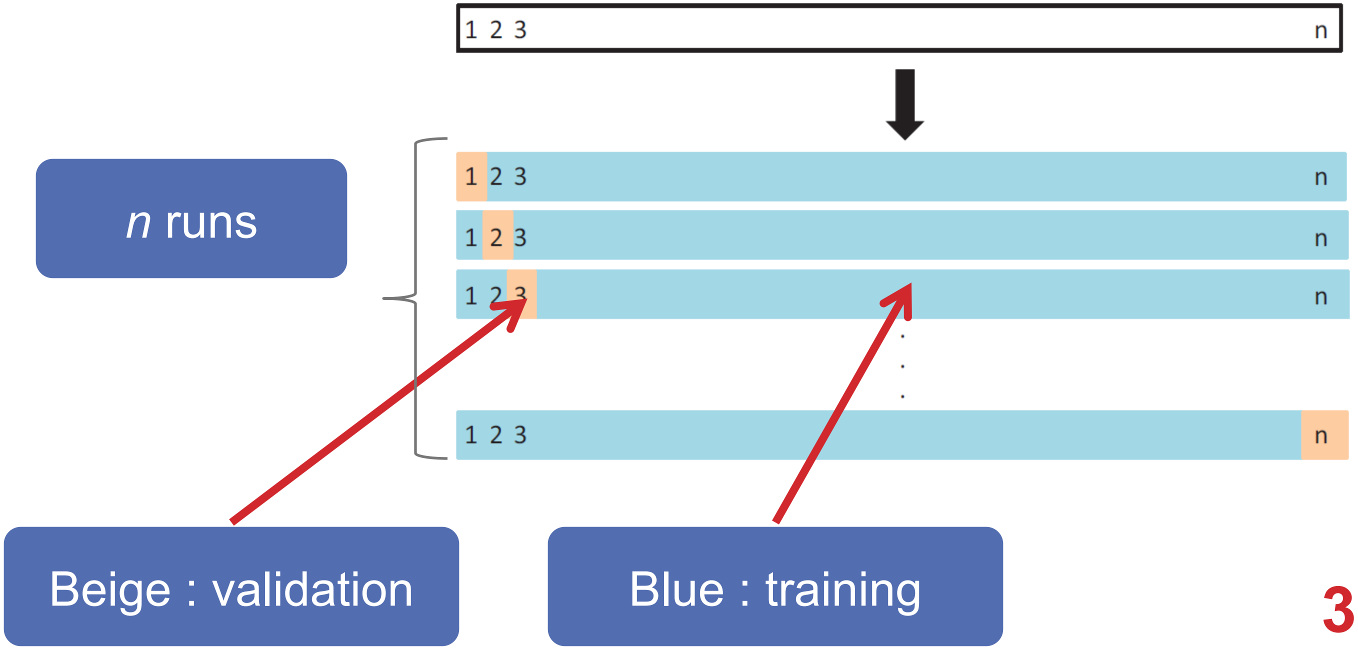
Disadvantages:

- The validation estimation of the test error rate can be highly variable
  - Depends on which observations are included in the training and which are included in the validation set
- Only a subset of observations (training data) are used to fit the model.
  - Models trained on a smaller training dataset (fewer observations) tend to perform worse

LEAVE-ONE-OUT CROSS-VALIDATION (LOOCV)

Idea:

- Similar to the validation set approach except that only 1 observation is used for validation and the remaining *n*-1 observations are used for training
- The error rate is again the average of the n runs



n runs

Beige : validation

Blue : training

LEAVE-ONE-OUT CROSS-VALIDATION (LOOCV)

Advantages:

- Less bias (compared to the validation set approach)
  - Repeatedly fit the statistical learning method using *n*-1 observations (almost all the data set is used)
- Less variable error rate/MSE

Disadvantages:

- Computationally expensive. Need to run *n* times

K-FOLD CROSS-VALIDATION

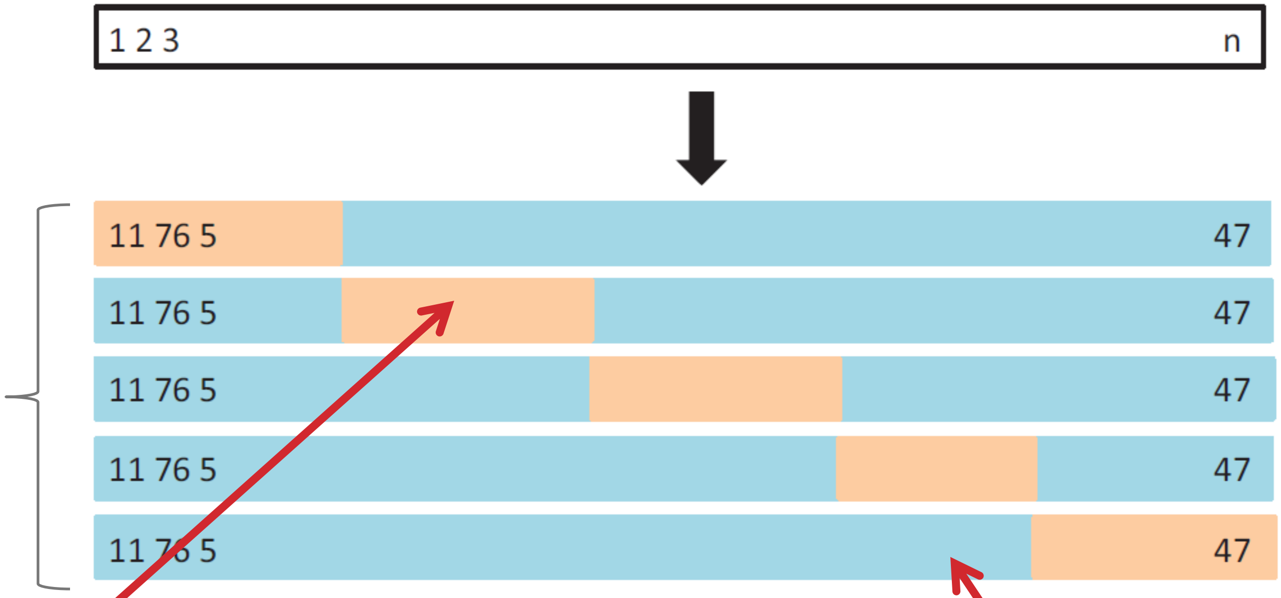
An alternative to LOOCV

Instead of splitting the data (1, *n*-1), randomly divide the set of observations into *k* groups (or *k*-fold) each with approximately equal size

The first fold is treated as a validation set, and the remaining *k*-1 folds are used for training

- The accuracy is calculated based on the average of the *k* runs

K-FOLD CROSS-VALIDATION



k runs

Beige : validation

Blue : training

K-FOLD CROSS-VALIDATION

Commonly used for evaluation model performance

Hybrid approach between validation set approach and LOOCV

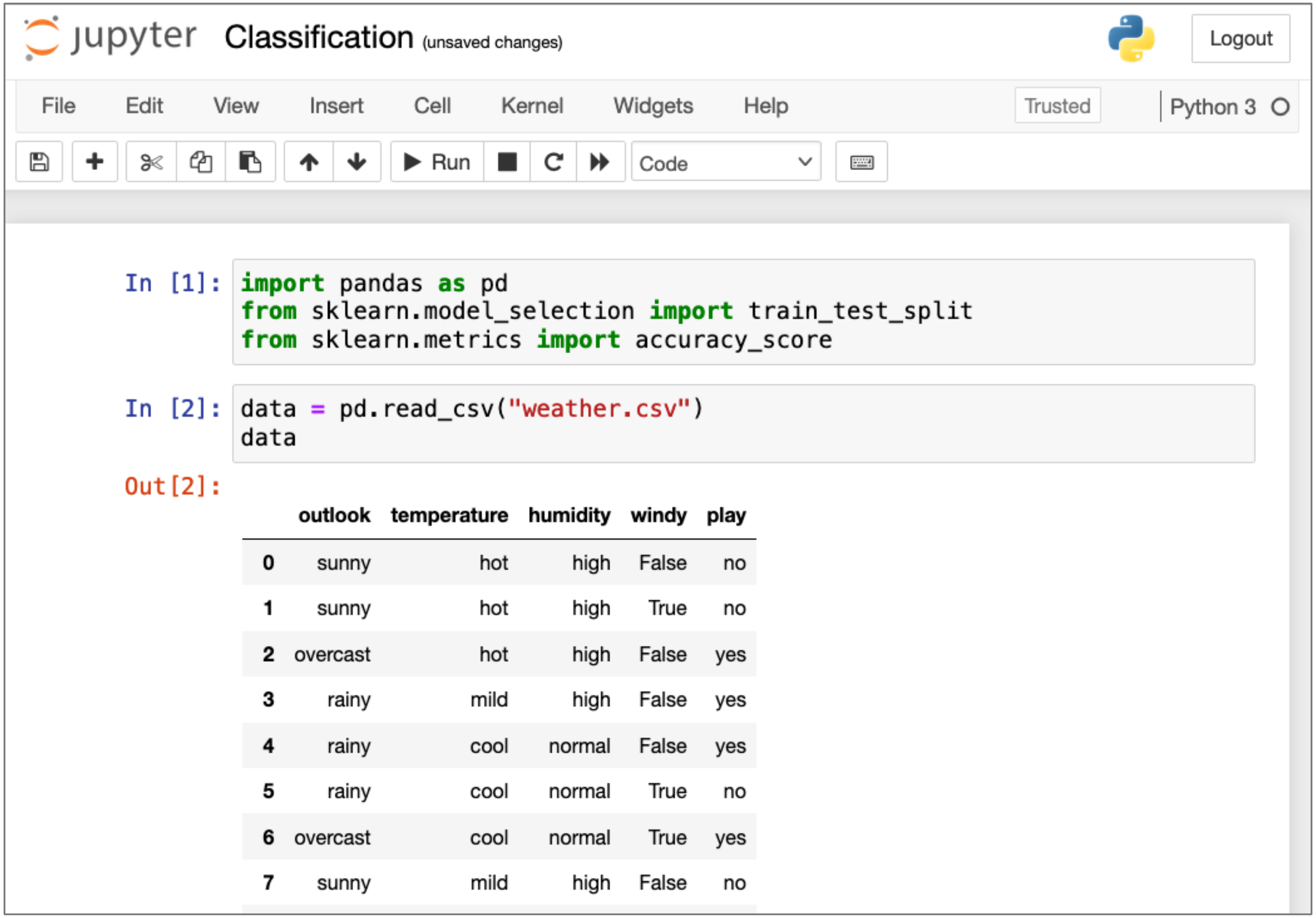
Less computationally expensive compared to LOOCV

Common value: K = 10 or 10-Fold Cross Validation

LOOCV is actually a special case of *k*-fold cross-validation where *k*=*n*

HANDS-ON: CLASSIFICATION

Download and access:
Classification.ipynb



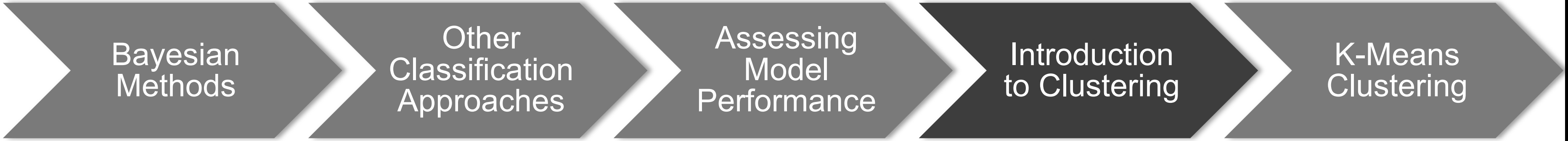
```
In [1]: import pandas as pd
        from sklearn.model_selection import train_test_split
        from sklearn.metrics import accuracy_score
```

```
In [2]: data = pd.read_csv("weather.csv")
        data
```

Out[2]:

	outlook	temperature	humidity	windy	play
0	sunny	hot	high	False	no
1	sunny	hot	high	True	no
2	overcast	hot	high	False	yes
3	rainy	mild	high	False	yes
4	rainy	cool	normal	False	yes
5	rainy	cool	normal	True	no
6	overcast	cool	normal	True	yes
7	sunny	mild	high	False	no

INTRODUCTION TO CLUSTERING



Bayesian Methods

Other Classification Approaches

Assessing Model Performance

Introduction to Clustering

K-Means Clustering

SUPERVISED VS UNSUPERVISED LEARNING

Supervised Learning

- We have labeled (training) data
- Use the labeled data to train a model for predicting the response on the unlabeled (testing) data

Unsupervised Learning

- We do not have labeled data, we only have access to unlabeled data (usually in large amount)
- We want to either label the unlabeled observations, or group up the observations into subgroups

AN EXAMPLE OF UNSUPERVISED LEARNING

Assume we have access to a large collection of tweets, can we group up the tweets to similar interests?

e.g. Tech, Politics, Singapore, NUS, smartphone etc (the list is not pre-defined)

WHY UNSUPERVISED LEARNING?

Since we already have the supervised learning approach, why do we still need unsupervised learning?

- Unlabeled data is usually readily available (easy problem + able to get large amount)
- Access to (large amount of) training data is rarely available
- Creating training data (i.e. label the unlabeled data manually by hand) is very tedious and not scalable
- Unsupervised learning approaches do not require data to be labeled but still can perform the same task as supervised learning approaches

SUPERVISED VS UNSUPERVISED LEARNING

Supervised Learning

- Accuracy is always better than (or at least as good as) unsupervised learning
- Not scalable (large amount of labeled data is not available – nice idea but not practical)

Unsupervised Learning

- Accuracy is not as good as supervised learning
- More scalable (large amount of unlabeled data available)

CLUSTERING

Clustering is an example of an unsupervised learning approach

Refers to the broad set of techniques for finding similar subgroups, or clusters in a data set

- A good clustering is when the observations within the same cluster is similar to each other but different compared to other clusters

APPLICATIONS OF CLUSTERING

Marketing

- Discover distinct groups in customer bases and develop targeted marketing

Medical

- Discover different types of cancer by clustering the data

Web Search

- Grouping words together to suggest related terms

Social Network

- Identifying trending topics

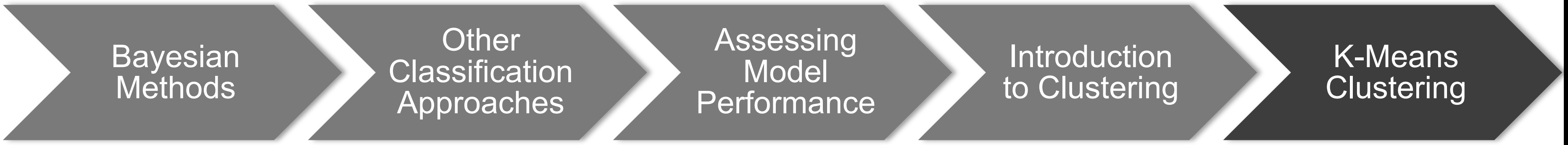
CLUSTERING METHODS

There are many clustering methods

But we will be only focusing on the most commonly used approach:

- K-means Clustering

K-MEANS CLUSTERING



Bayesian Methods

Other Classification Approaches

Assessing Model Performance

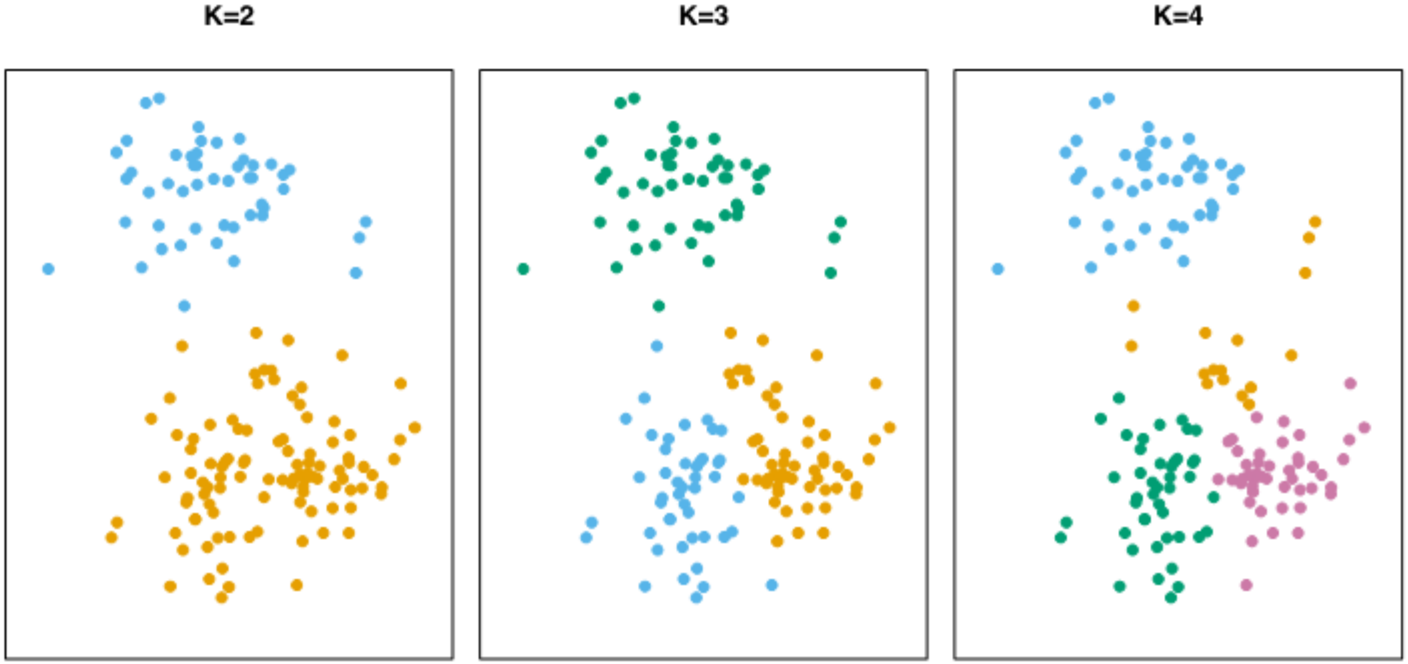
Introduction to Clustering

K-Means Clustering

K-MEANS CLUSTERING

K-means clustering requires the user to supply the desired number of clusters (*K*)

The observations are then grouped up into one of these *K* clusters



K-MEANS CLUSTERING

The K-means clustering algorithm partition the data into K clusters:

$$C_1,\ldots,C_K$$

- Each observation belong to one of the K clusters

$$C_1 \cup C_2 \cup \ldots \cup C_K = \{1,\ldots,n\}$$

- The clusters are non-overlapping. I.e. no observation belongs to more than one cluster

$$C_i \cap C_j = \emptyset$$

K-MEANS CLUSTERING

Idea: a good cluster = one where the within-cluster variation is as small as possible (i.e. elements within a cluster should be as similar as possible)

$$\underset{C_1,\ldots,C_K}{\text{minimize}}\left\{\sum_{k=1}^{K} W(C_k)\right\}$$

where

$$W(C_k) = \text{within - cluster variation for cluster } C_k$$

$$W(C_k) = \frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}(x_{ij} - x_{i'j})^2$$

K-MEANS ALGORITHM

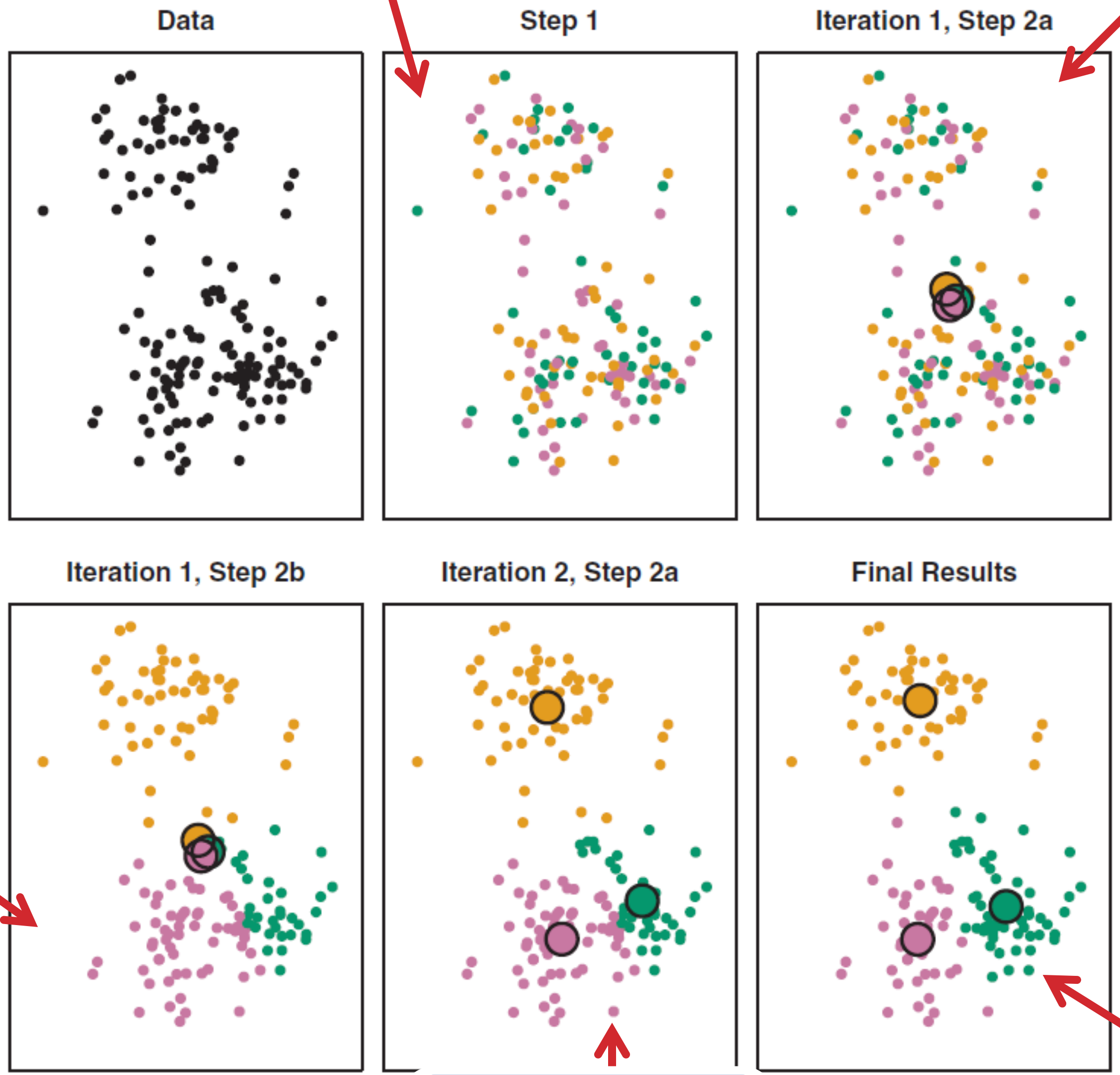
Initial Step: Randomly assign each observation to one of the *K* cluster

Iterate until cluster assignments stop changing:

- For each of the *K* clusters, compute the cluster centroid. The *k*th cluster centroid is the vector of the p feature means for the observations in the *k*th cluster
- Assign each observation to the cluster whose centroid is closest (measured using Euclidean distance)

Randomly assign to a cluster

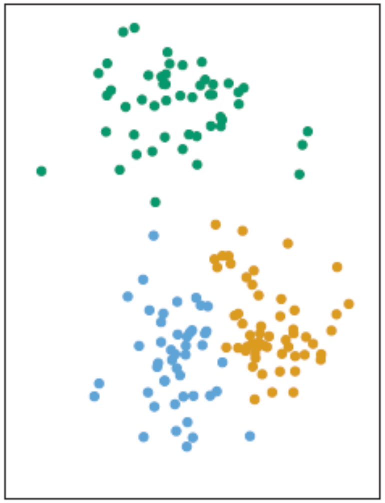
Compute the initial cluster centroids



Assign points to the closest cluster center

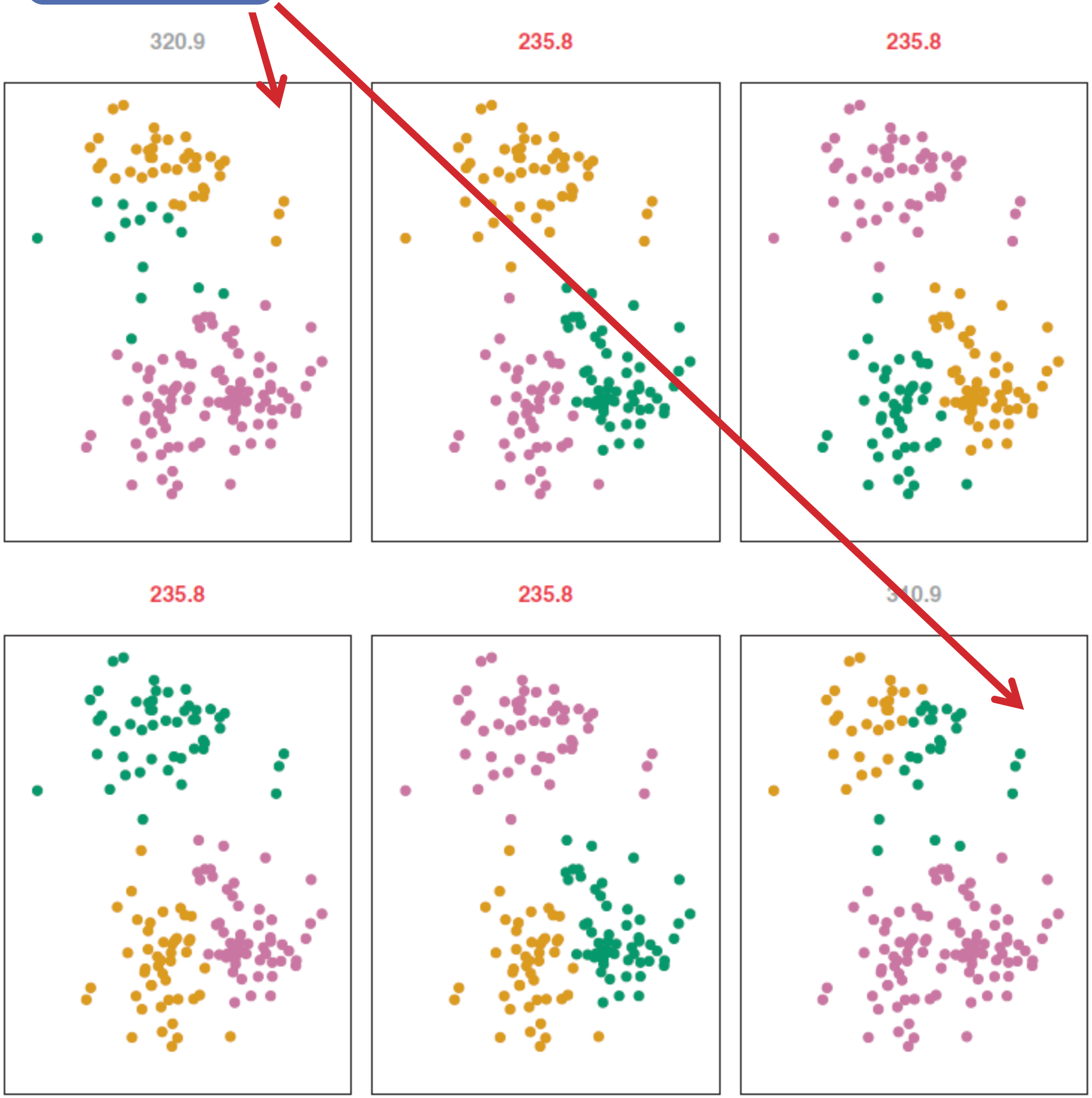
Compute the new cluster centers

Stop when there is no further change



True cluster

Bad solution



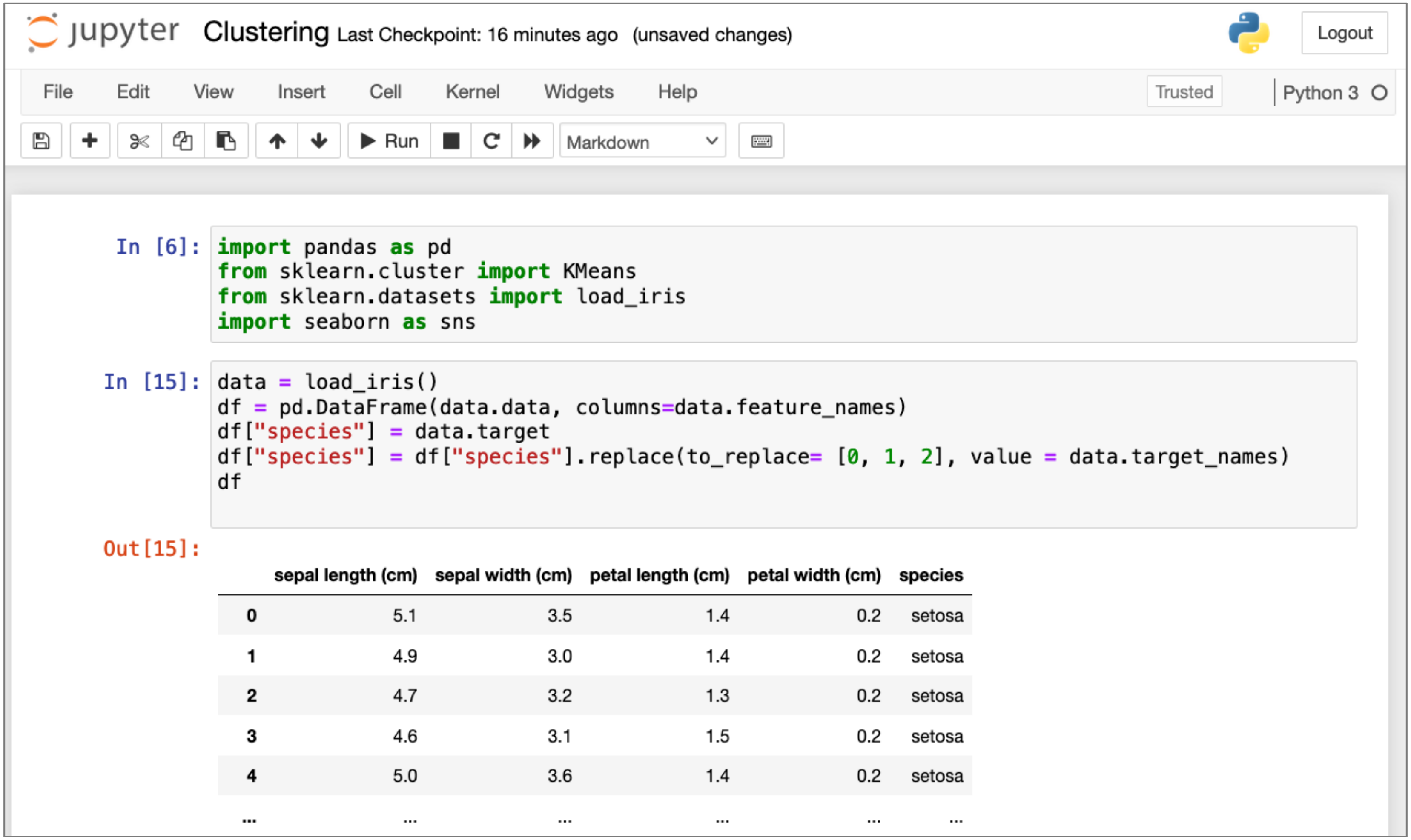
Cluster formed depends on the initial random assignment

Hence important to run the algorithm multiple times with different random starting points

K-means algorithm can get stuck in "local optimums"

HANDS-ON: CLUSTERING

Download and access: **Clustering.ipynb**



In [6]:
```
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.datasets import load_iris
import seaborn as sns
```

In [15]:
```
data = load_iris()
df = pd.DataFrame(data.data, columns=data.feature_names)
df["species"] = data.target
df["species"] = df["species"].replace(to_replace= [0, 1, 2], value = data.target_names)
df
```

Out[15]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...

WHAT'S NEXT?

Mining Web Content II