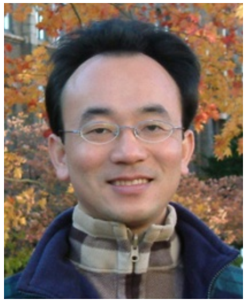
資料探勘
(Data Mining)

非監督學習：
集群分析，行銷市場區隔
(Unsupervised Learning: Cluster Analysis, Market Segmentation)

1092DM06

MBA, IM, NTPU (M5026) (Spring 2021)

Tue 2, 3, 4 (9:10-12:00) (B8F40)



Min-Yuh Day
戴敏育
Associate Professor
副教授
Institute of Information Management, National Taipei University
國立臺北大學 資訊管理研究所
https://web.ntpu.edu.tw/~myday

2021-04-13

課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
1	2021/02/23	資料探勘介紹 (Introduction to data mining)
2	2021/03/02	ABC：人工智慧，大數據，雲端運算 (ABC: AI, Big Data, Cloud Computing)
3	2021/03/09	Python資料探勘的基礎 (Foundations of Data Mining in Python)
4	2021/03/16	資料科學與資料探勘：發現，分析，可視化和呈現數據 (Data Science and Data Mining: Discovering, Analyzing, Visualizing and Presenting Data)
5	2021/03/23	非監督學習：關聯分析，購物籃分析 (Unsupervised Learning: Association Analysis, Market Basket Analysis)
6	2021/03/30	資料探勘個案研究 I (Case Study on Data Mining I)

課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
7	2021/04/06	放假一天 (Day off)
8	2021/04/13	非監督學習：集群分析，行銷市場區隔 (Unsupervised Learning: Cluster Analysis, Market Segmentation)
9	2021/04/20	期中報告 (Midterm Project Report)
10	2021/04/27	監督學習：分類和預測 (Supervised Learning: Classification and Prediction)
11	2021/05/04	機器學習和深度學習 (Machine Learning and Deep Learning)
12	2021/05/11	卷積神經網絡 (Convolutional Neural Networks)

課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
13	2021/05/18	資料探勘個案研究 II (Case Study on Data Mining II)
14	2021/05/25	遞歸神經網絡 (Recurrent Neural Networks)
15	2021/06/01	強化學習 (Reinforcement Learning)
16	2021/06/08	社交網絡分析 (Social Network Analysis)
17	2021/06/15	期末報告 I (Final Project Report I)
18	2021/06/22	期末報告 II (Final Project Report II)

Unsupervised Learning: Cluster Analysis, Market Segmentation

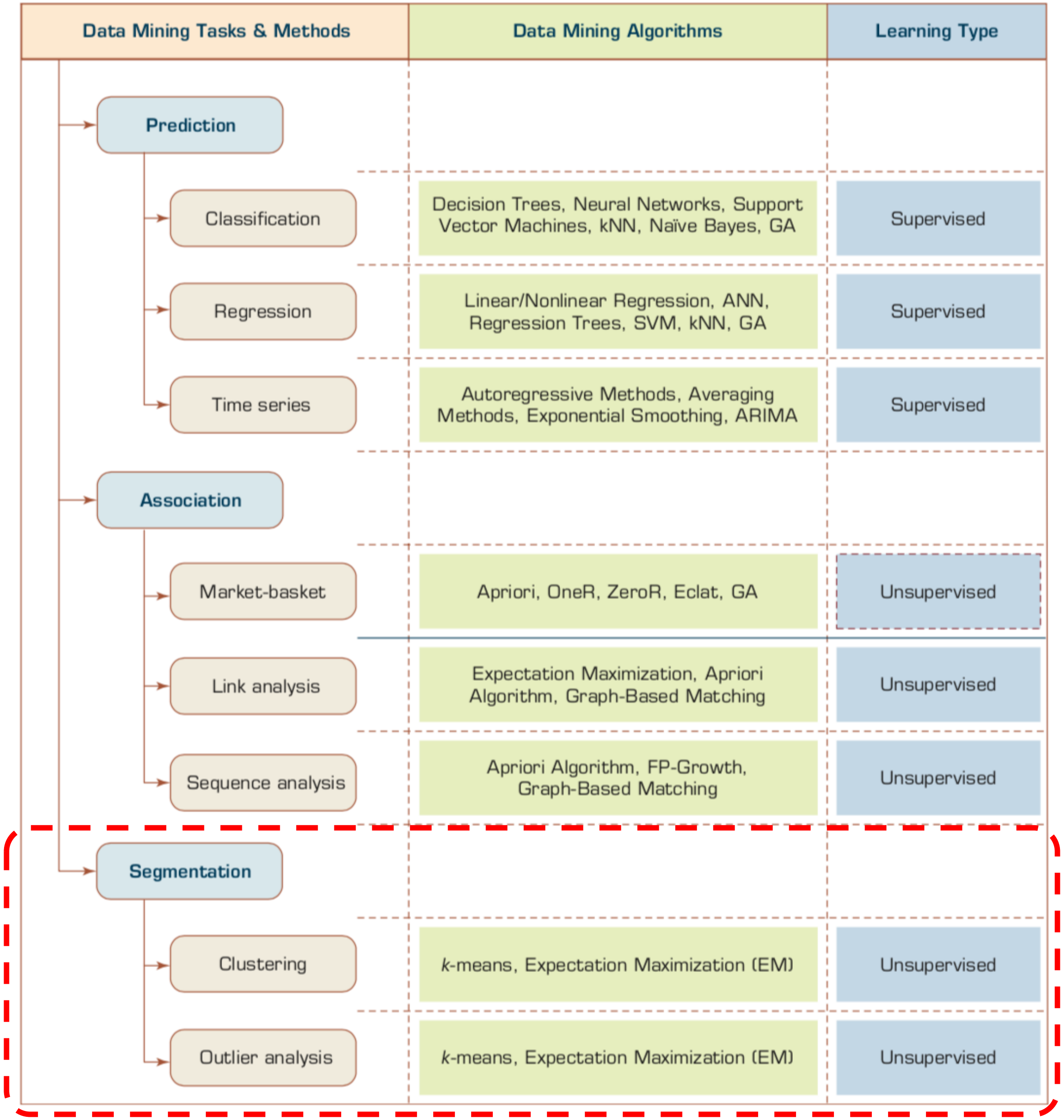
Outline

- Unsupervised Learning
- Cluster Analysis
- Market Segmentation
- K-Means Clustering

Data Mining Tasks & Methods

Unsupervised Learning:
Cluster Analysis,
Market Segmentation

Segmentation



Data Mining Tasks & Methods	Data Mining Algorithms	Learning Type
Prediction		
Classification	Decision Trees, Neural Networks, Support Vector Machines, kNN, Naïve Bayes, GA	Supervised
Regression	Linear/Nonlinear Regression, ANN, Regression Trees, SVM, kNN, GA	Supervised
Time series	Autoregressive Methods, Averaging Methods, Exponential Smoothing, ARIMA	Supervised
Association		
Market-basket	Apriori, OneR, ZeroR, Eclat, GA	Unsupervised
Link analysis	Expectation Maximization, Apriori Algorithm, Graph-Based Matching	Unsupervised
Sequence analysis	Apriori Algorithm, FP-Growth, Graph-Based Matching	Unsupervised
Segmentation		
Clustering	*k*-means, Expectation Maximization (EM)	Unsupervised
Outlier analysis	*k*-means, Expectation Maximization (EM)	Unsupervised

Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson

Example of Cluster Analysis

Point	P	P(x,y)
p01	a	(3, 4)
p02	b	(3, 6)
p03	c	(3, 8)
p04	d	(4, 5)
p05	e	(4, 7)
p06	f	(5, 1)
p07	g	(5, 5)
p08	h	(7, 3)
p09	i	(7, 5)
p10	j	(8, 5)

K-Means Clustering

Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	1.95	3.78	Cluster1
p02	b	(3, 6)	0.69	4.51	Cluster1
p03	c	(3, 8)	2.27	5.86	Cluster1
p04	d	(4, 5)	0.89	3.13	Cluster1
p05	e	(4, 7)	1.22	4.45	Cluster1
p06	f	(5, 1)	5.01	3.05	Cluster2
p07	g	(5, 5)	1.57	2.30	Cluster1
p08	h	(7, 3)	4.37	0.56	Cluster2
p09	i	(7, 5)	3.43	1.52	Cluster2
p10	j	(8, 5)	4.41	1.95	Cluster2

m1	(3.67, 5.83)
m2	(6.75, 3.50)

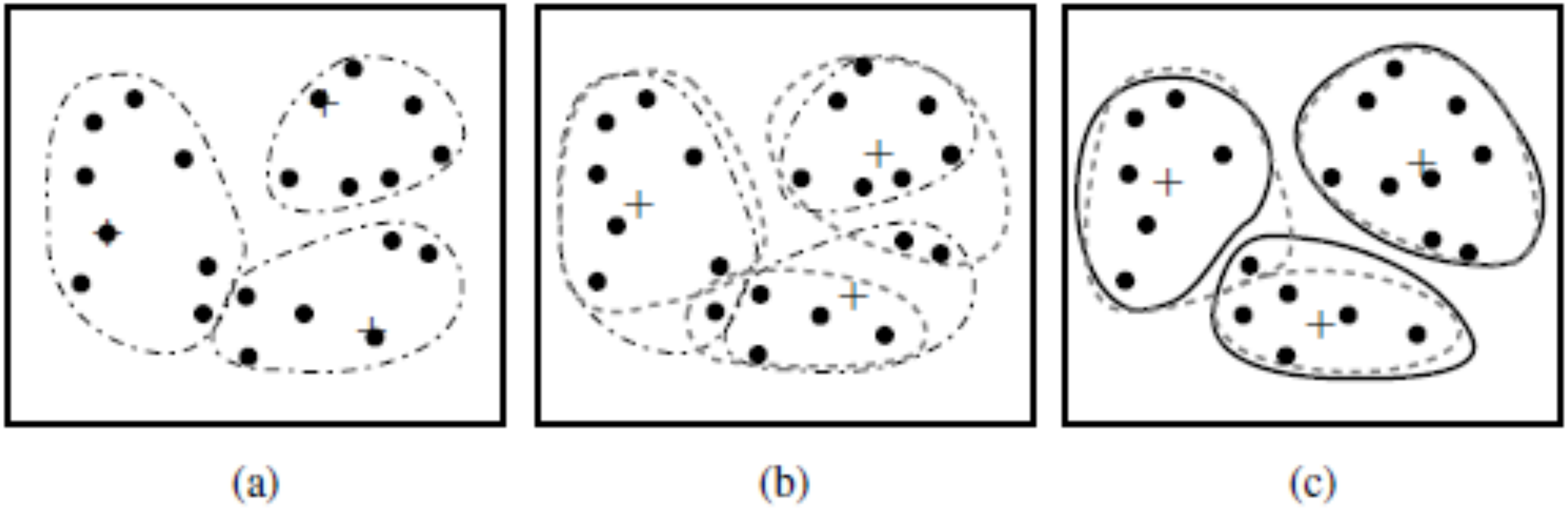
Cluster Analysis

Cluster Analysis

- Used for automatic identification of natural groupings of things
- Part of the machine-learning family
- Employ unsupervised learning
- Learns the clusters of things from past data, then assigns new instances
- There is not an output variable
- Also known as segmentation

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

Cluster Analysis



Clustering of a set of objects based on the *k-means method.*
(The mean of each cluster is marked by a “+”.)

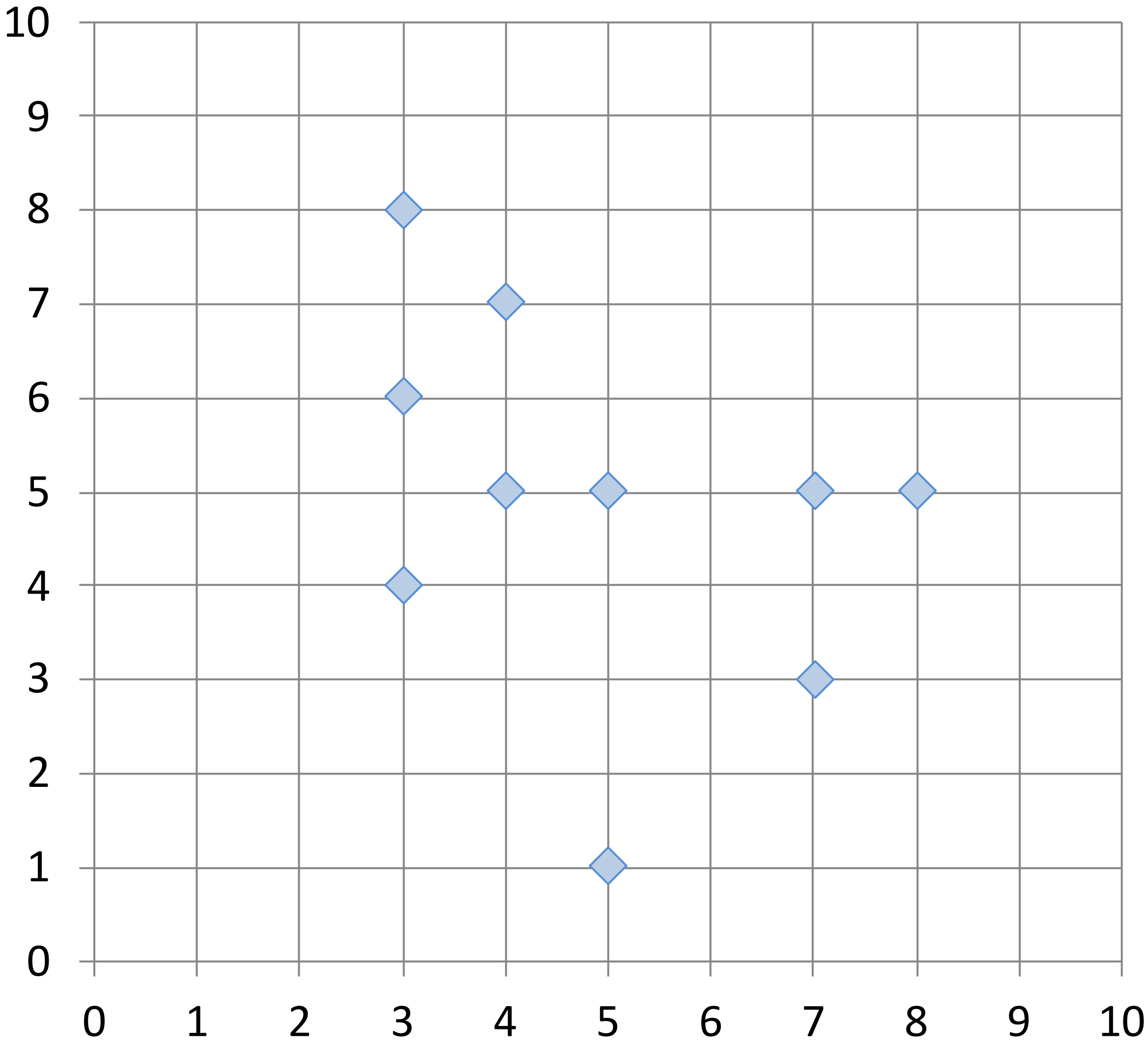
Source: Han & Kamber (2006)

Cluster Analysis

- Clustering results may be used to
  - Identify natural groupings of customers
  - Identify rules for assigning new cases to classes for targeting/diagnostic purposes
  - Provide characterization, definition, labeling of populations
  - Decrease the size and complexity of problems for other data mining methods
  - Identify outliers in a specific domain (e.g., rare-event detection)

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

Example of Cluster Analysis



Point	P	P(x,y)
p01	a	(3, 4)
p02	b	(3, 6)
p03	c	(3, 8)
p04	d	(4, 5)
p05	e	(4, 7)
p06	f	(5, 1)
p07	g	(5, 5)
p08	h	(7, 3)
p09	i	(7, 5)
p10	j	(8, 5)

Cluster Analysis for Data Mining

- Analysis methods
  - Statistical methods (including both hierarchical and nonhierarchical), such as *k*-means, *k*-modes, and so on
  - Neural networks (adaptive resonance theory [ART], self-organizing map [SOM])
  - Fuzzy logic (e.g., fuzzy c-means algorithm)
  - Genetic algorithms
- Divisive versus Agglomerative methods

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

Cluster Analysis for Data Mining

- How many clusters?
  - There is not a “truly optimal” way to calculate it
  - Heuristics are often used
    1. Look at the sparseness of clusters
    2. Number of clusters = $(n/2)^{1/2}$ (n: no of data points)
    3. Use Akaike information criterion (AIC)
    4. Use Bayesian information criterion (BIC)
- Most cluster analysis methods involve the use of a distance measure to calculate the closeness between pairs of items
  - Euclidian versus Manhattan (rectilinear) distance

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

k-Means Clustering Algorithm

- *k* : pre-determined number of clusters
- Algorithm (Step 0: determine value of *k*)

Step 1: Randomly generate *k* random points as initial cluster centers

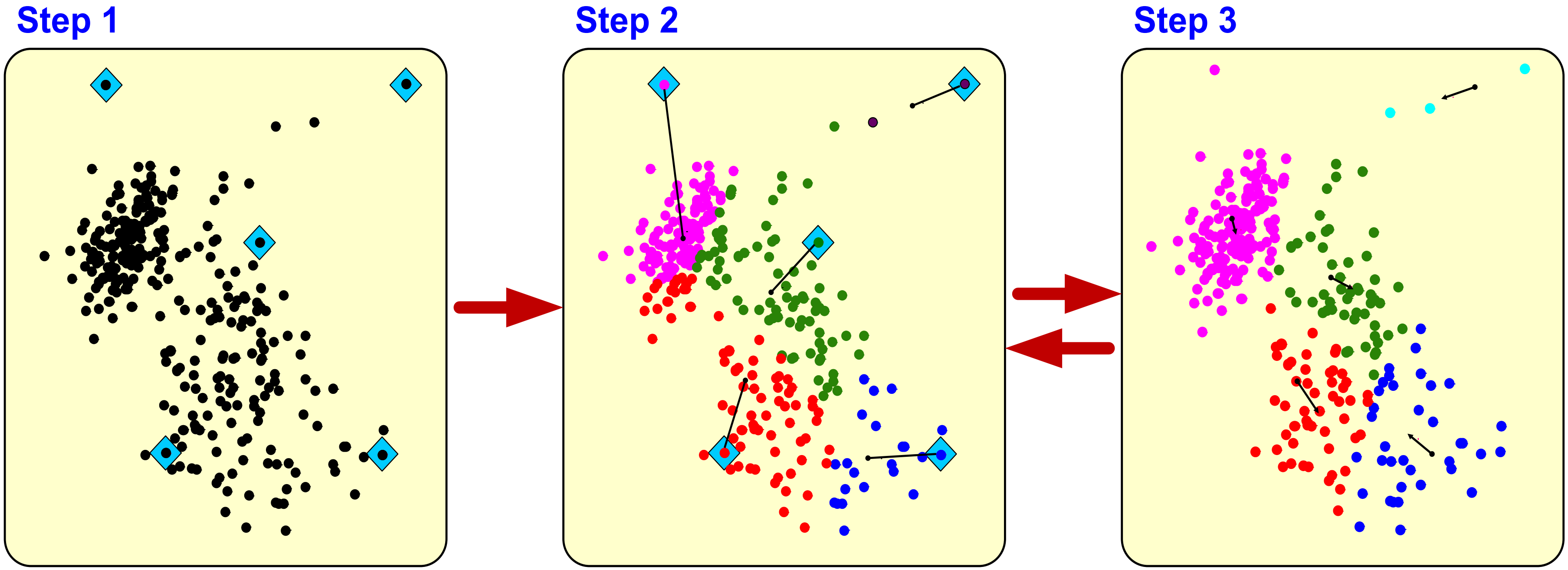
Step 2: Assign each point to the nearest cluster center

Step 3: Re-compute the new cluster centers

Repetition step: Repeat steps 2 and 3 until some convergence criterion is met (usually that the assignment of points to clusters becomes stable)

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

Cluster Analysis for Data Mining - *k*-Means Clustering Algorithm



Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

Similarity

Distance

Similarity and Dissimilarity Between Objects

- Distances are normally used to measure the similarity or dissimilarity between two data objects
- Some popular ones include: *Minkowski distance*:

$$d(i,j) = \sqrt[q]{(|x_{i_1} - x_{j_1}|^q + |x_{i_2} - x_{j_2}|^q + ... + |x_{i_p} - x_{j_p}|^q)}$$

where $i = (x_{i1}, x_{i2}, ..., x_{ip})$ and $j = (x_{j1}, x_{j2}, ..., x_{jp})$ are two p-dimensional data objects, and q is a positive integer

- If $q = 1$, d is Manhattan distance

$$d(i,j) = |x_{i_1} - x_{j_1}| + |x_{i_2} - x_{j_2}| + ... + |x_{i_p} - x_{j_p}|$$

Source: Han & Kamber (2006)

Similarity and Dissimilarity Between Objects (Cont.)

- *If q = 2, d* is Euclidean distance:

$$d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + ... + |x_{i_p} - x_{j_p}|^2)}$$

  – Properties

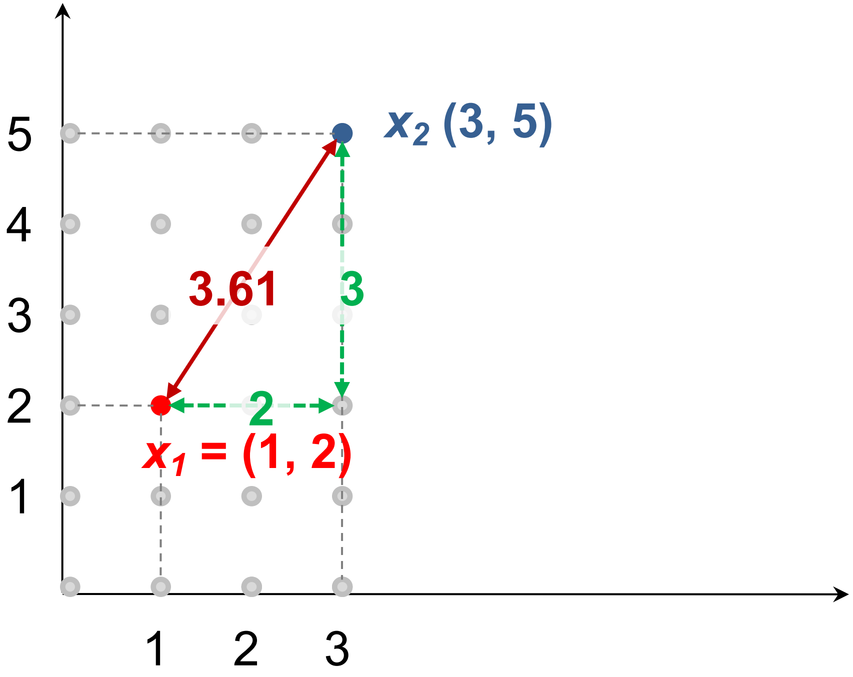
    - $d(i,j) \geq 0$
    - $d(i,i) = 0$
    - $d(i,j) = d(j,i)$
    - $d(i,j) \leq d(i,k) + d(k,j)$

- Also, one can use weighted distance, parametric Pearson product moment correlation, or other disimilarity measures

Source: Han & Kamber (2006)

Euclidean distance vs Manhattan distance

- Distance of two point x_1 = (1, 2) and x_2 (3, 5)

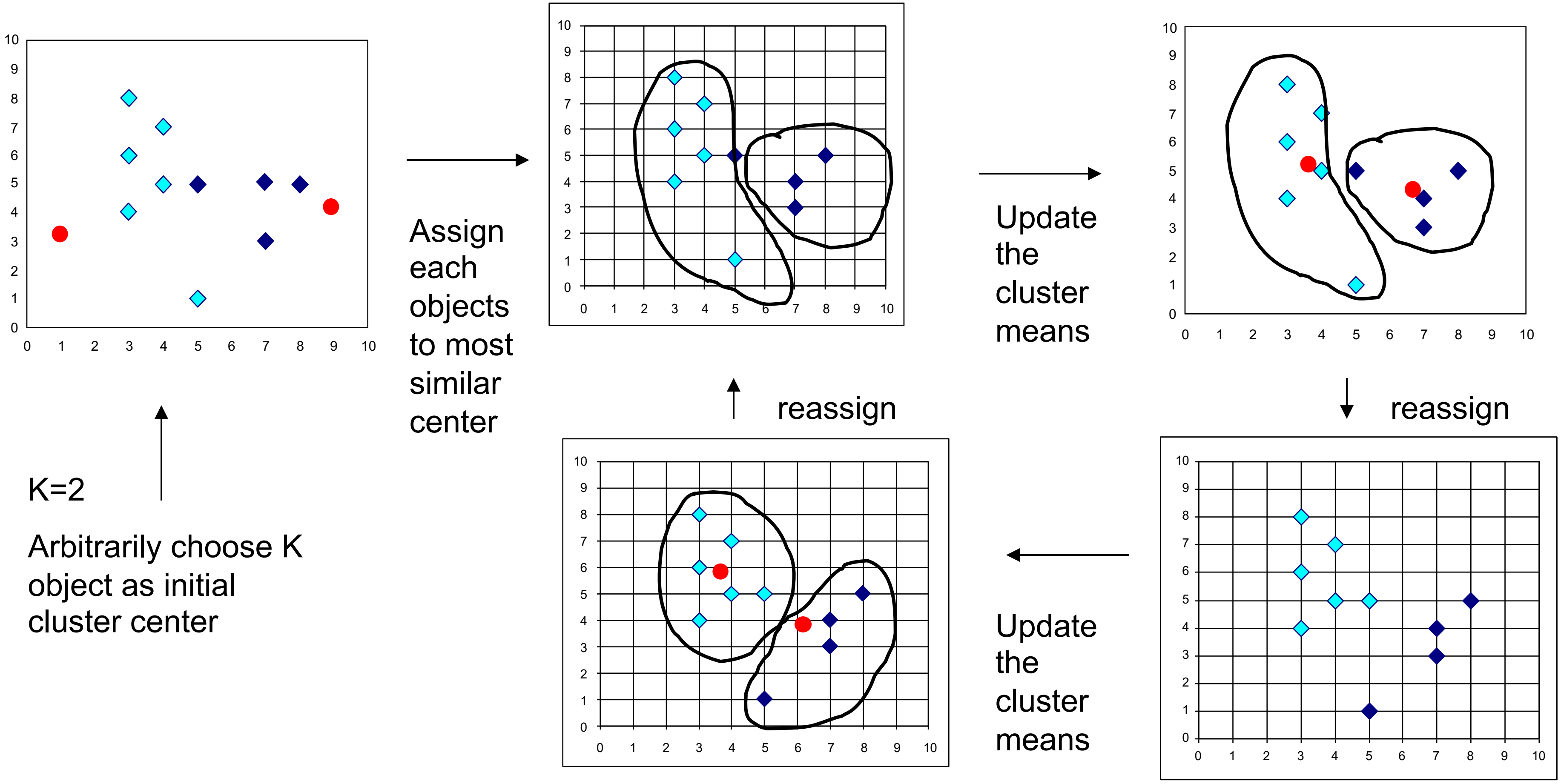


Euclidean distance:
$= ((3-1)^2 + (5-2)^2\ )^{1/2}$
$= (2^2 + 3^2)^{1/2}$
$= (4 + 9)^{1/2}$
$= (13)^{1/2}$
= 3.61

Manhattan distance:
= (3-1) + (5-2)
= 2 + 3
= 5

The *K-Means* Clustering Method

- Example



K=2

Arbitrarily choose K object as initial cluster center

Assign each objects to most similar center

Update the cluster means

reassign

Update the cluster means

reassign

Source: Han & Kamber (2006)

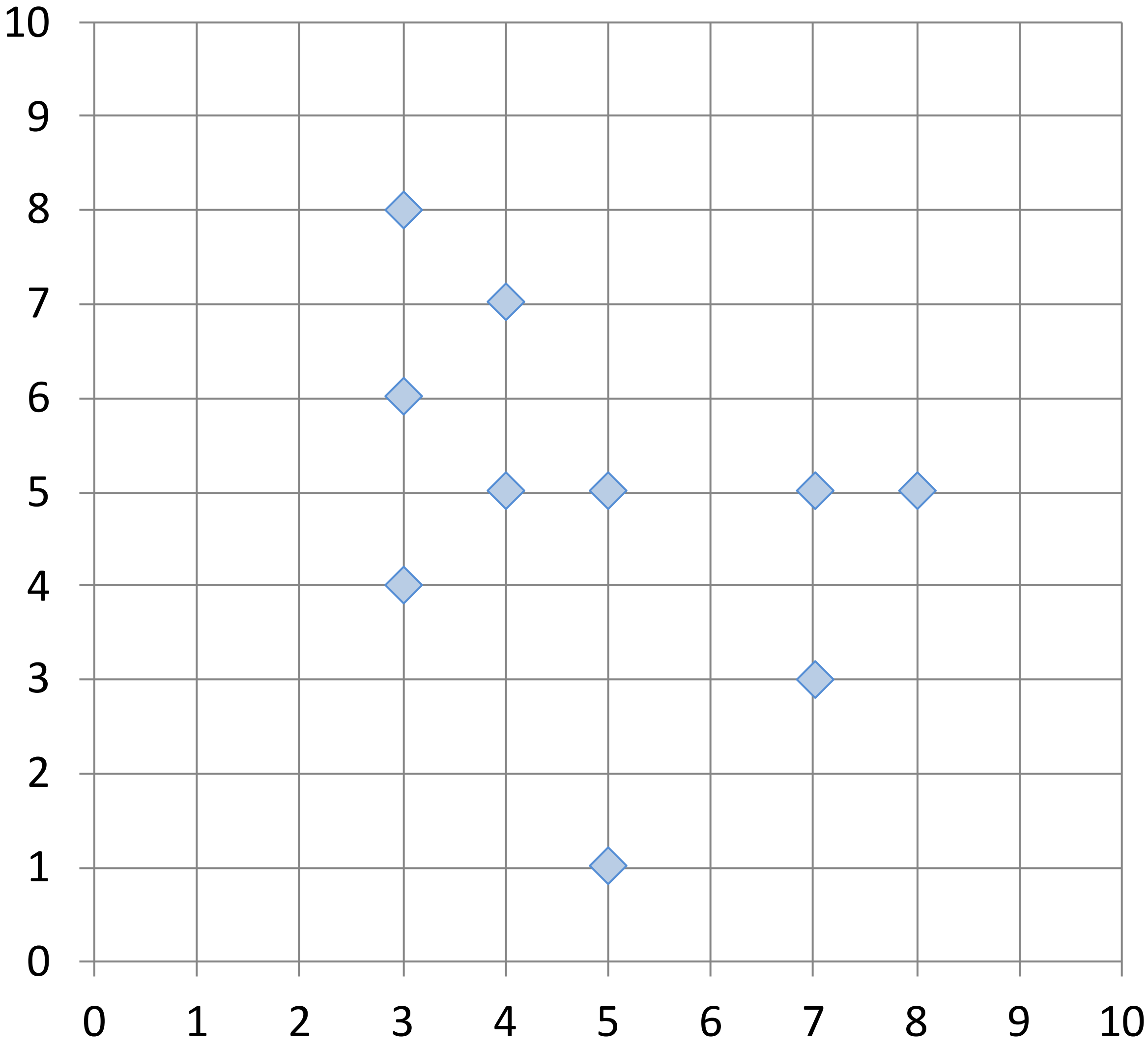
K-Means Clustering

Example of Cluster Analysis

Point	P	P(x,y)
p01	a	(3, 4)
p02	b	(3, 6)
p03	c	(3, 8)
p04	d	(4, 5)
p05	e	(4, 7)
p06	f	(5, 1)
p07	g	(5, 5)
p08	h	(7, 3)
p09	i	(7, 5)
p10	j	(8, 5)

K-Means Clustering

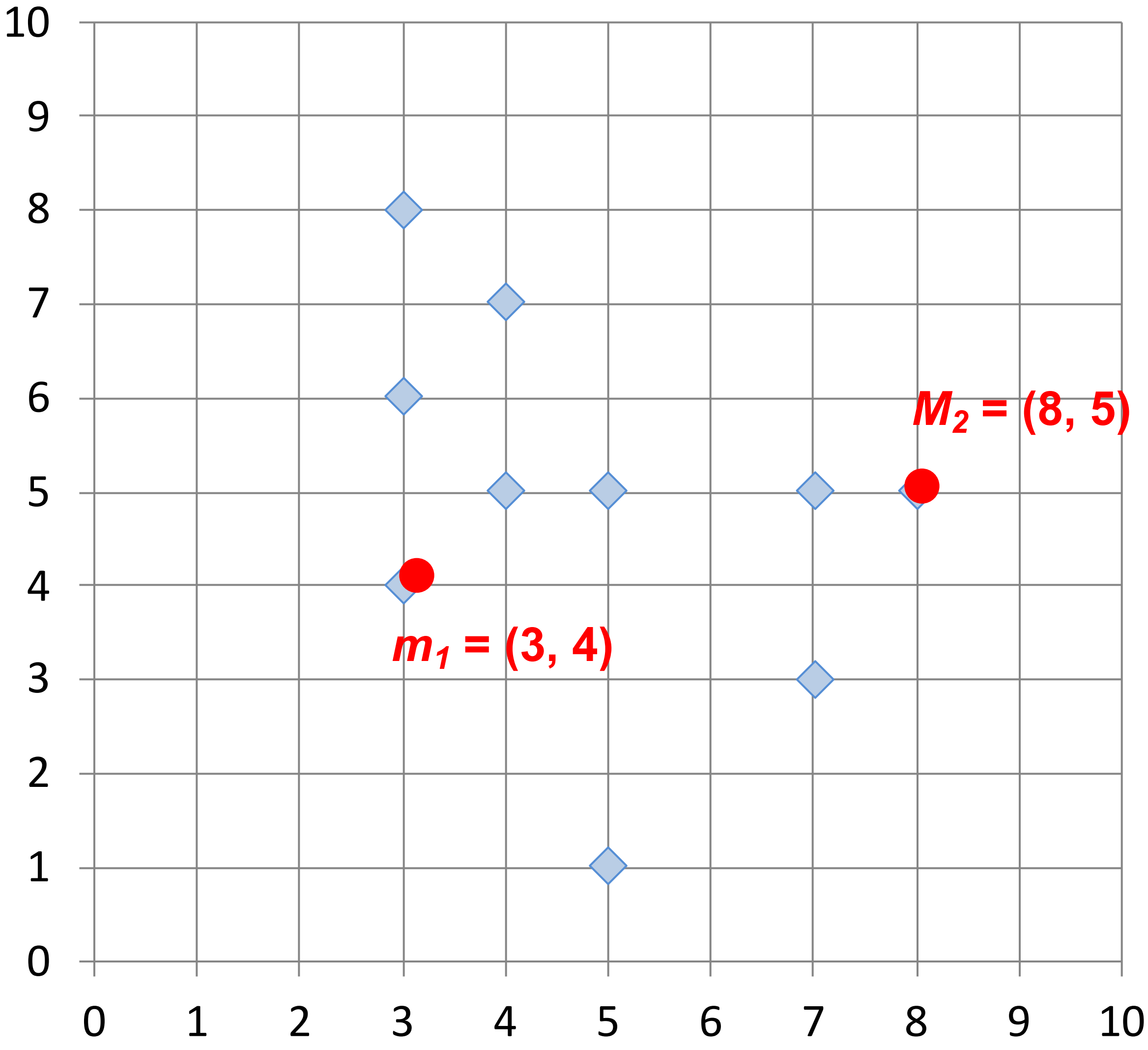
Step by Step



Point	P	P(x,y)
p01	a	(3, 4)
p02	b	(3, 6)
p03	c	(3, 8)
p04	d	(4, 5)
p05	e	(4, 7)
p06	f	(5, 1)
p07	g	(5, 5)
p08	h	(7, 3)
p09	i	(7, 5)
p10	j	(8, 5)

K-Means Clustering

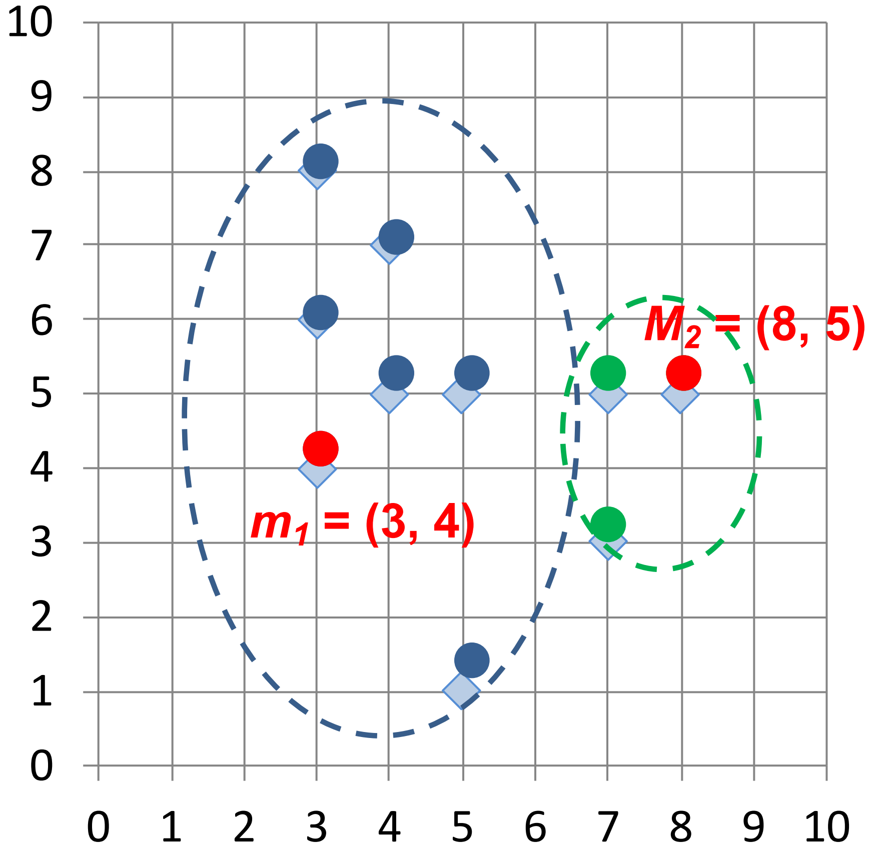
Step 1: K=2, Arbitrarily choose K object as initial cluster center



Point	P	P(x,y)
p01	a	(3, 4)
p02	b	(3, 6)
p03	c	(3, 8)
p04	d	(4, 5)
p05	e	(4, 7)
p06	f	(5, 1)
p07	g	(5, 5)
p08	h	(7, 3)
p09	i	(7, 5)
p10	j	(8, 5)
Initial	m1	(3, 4)
Initial	m2	(8, 5)

Step 2: Compute seed points as the centroids of the clusters of the current partition

Step 3: Assign each objects to most similar center



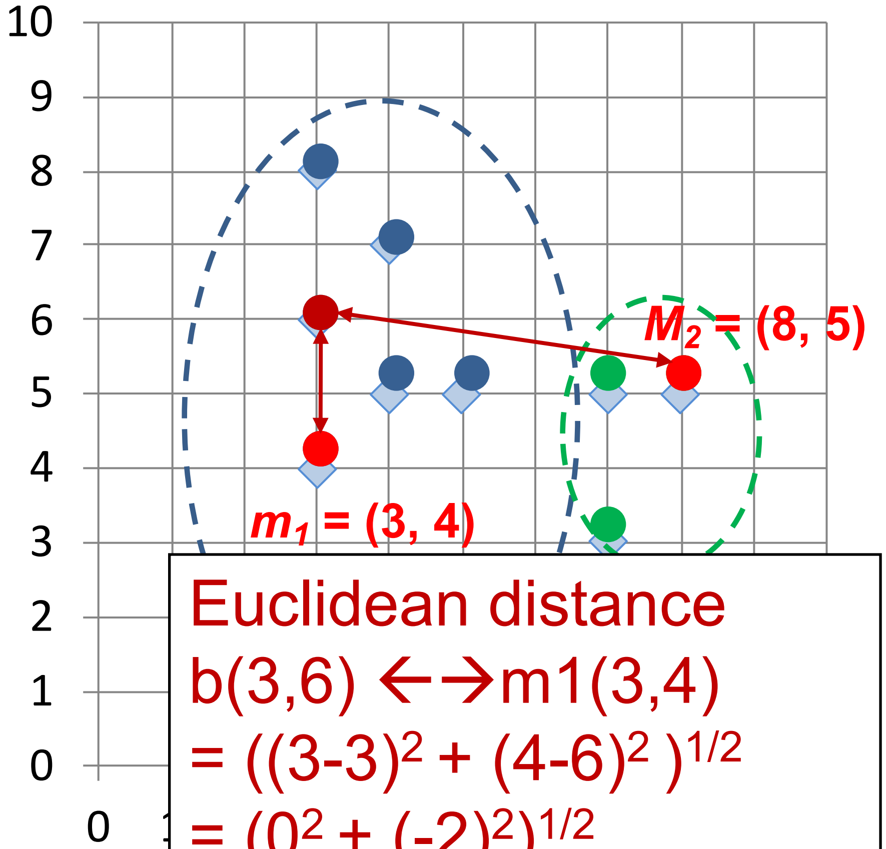
Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	0.00	5.10	Cluster1
p02	b	(3, 6)	2.00	5.10	Cluster1
p03	c	(3, 8)	4.00	5.83	Cluster1
p04	d	(4, 5)	1.41	4.00	Cluster1
p05	e	(4, 7)	3.16	4.47	Cluster1
p06	f	(5, 1)	3.61	5.00	Cluster1
p07	g	(5, 5)	2.24	3.00	Cluster1
p08	h	(7, 3)	4.12	2.24	Cluster2
p09	i	(7, 5)	4.12	1.00	Cluster2
p10	j	(8, 5)	5.10	0.00	Cluster2

K-Means Clustering

Initial	m1	(3, 4)
Initial	m2	(8, 5)

Step 2: Compute seed points as the centroids of the clusters of the current partition

Step 3: Assign each objects to most similar center



Euclidean distance

b(3,6) ←→m1(3,4)

$= ((3-3)^2 + (4-6)^2)^{1/2}$

$= (0^2 + (-2)^2)^{1/2}$

$= (0 + 4)^{1/2}$

$= (4)^{1/2}$

$= 2.00$

K-

Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	0.00	5.10	Cluster1
p02	b	(3, 6)	2.00	5.10	Cluster1
p03	c	(3, 8)	4.00	5.83	Cluster1
p04	d	(4, 5)	1.41	4.00	Cluster1
p05					ster1
p06					ster1
p07					ster1
p08					ster2
p09					ster2
p10					ster2

Euclidean distance

b(3,6) ←→m2(8,5)

$= ((8-3)^2 + (5-6)^2)^{1/2}$

$= (5^2 + (-1)^2)^{1/2}$

$= (25 + 1)^{1/2}$

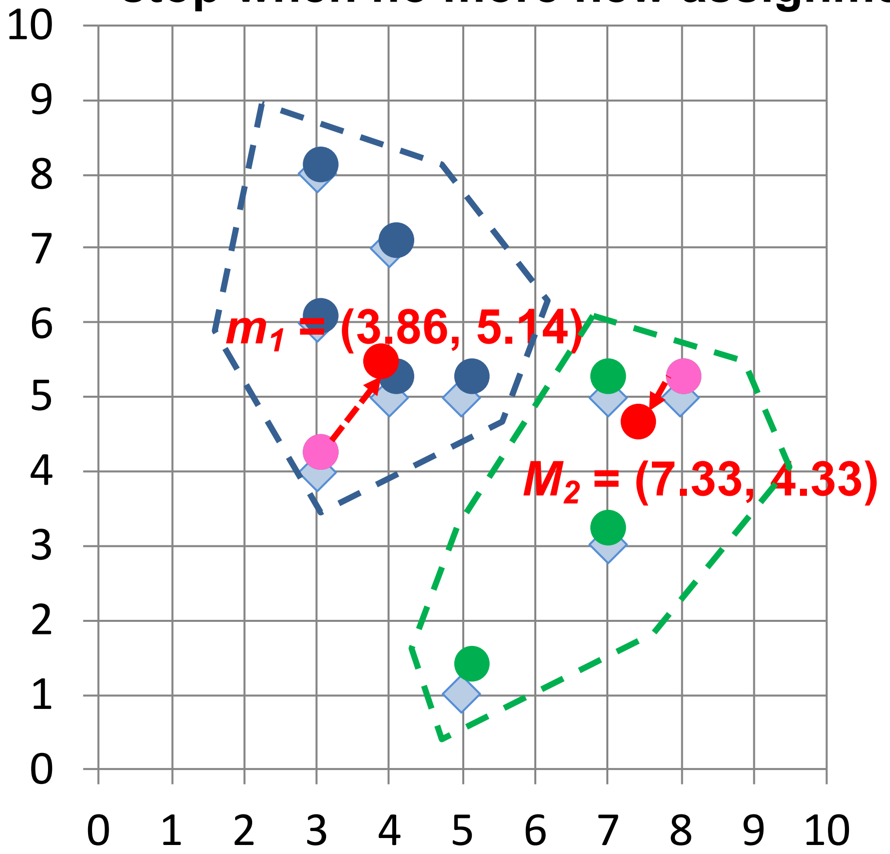
$= (26)^{1/2}$

$= 5.10$

Initial m1 (3, 4)

Initial m2 (8, 5)

Step 4: Update the cluster means,
Repeat Step 2, 3,
stop when no more new assignment

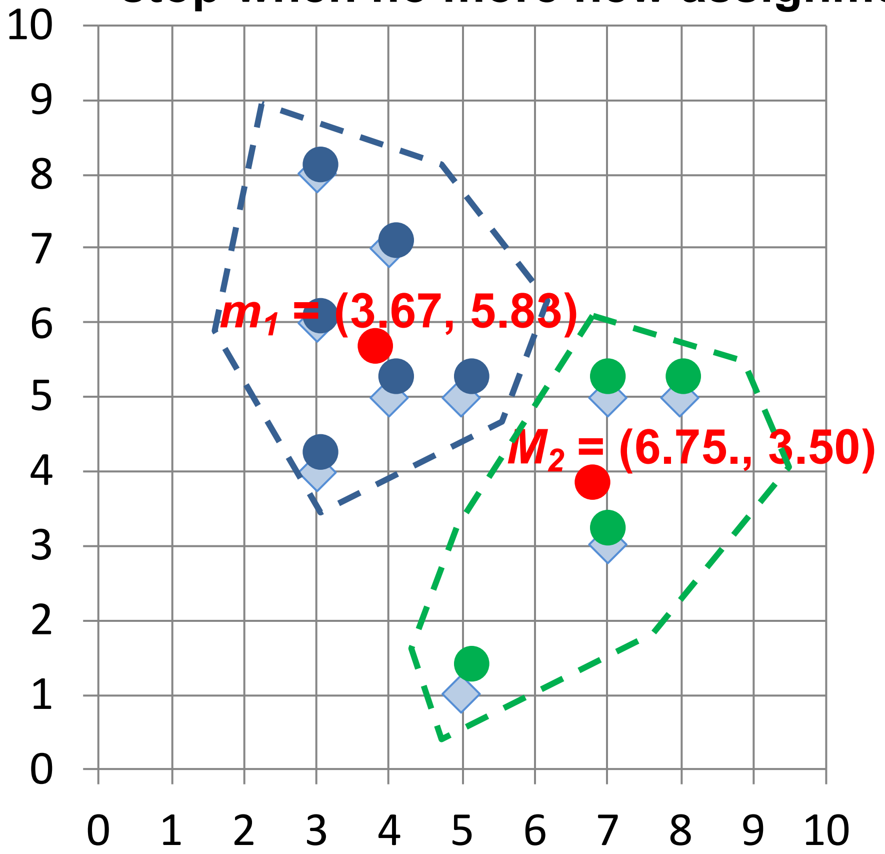


Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	1.43	4.34	Cluster1
p02	b	(3, 6)	1.22	4.64	Cluster1
p03	c	(3, 8)	2.99	5.68	Cluster1
p04	d	(4, 5)	0.20	3.40	Cluster1
p05	e	(4, 7)	1.87	4.27	Cluster1
p06	f	(5, 1)	4.29	4.06	Cluster2
p07	g	(5, 5)	1.15	2.42	Cluster1
p08	h	(7, 3)	3.80	1.37	Cluster2
p09	i	(7, 5)	3.14	0.75	Cluster2
p10	j	(8, 5)	4.14	0.95	Cluster2

m1	(3.86, 5.14)
m2	(7.33, 4.33)

K-Means Clustering

Step 4: Update the cluster means,
Repeat Step 2, 3,
stop when no more new assignment

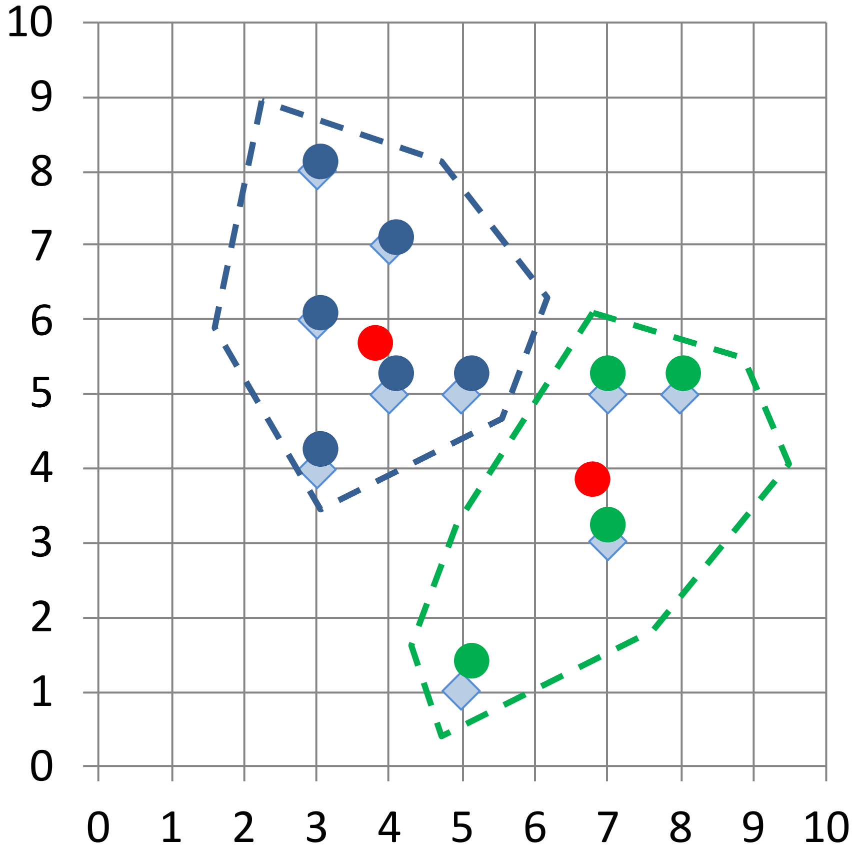


Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	1.95	3.78	Cluster1
p02	b	(3, 6)	0.69	4.51	Cluster1
p03	c	(3, 8)	2.27	5.86	Cluster1
p04	d	(4, 5)	0.89	3.13	Cluster1
p05	e	(4, 7)	1.22	4.45	Cluster1
p06	f	(5, 1)	5.01	3.05	Cluster2
p07	g	(5, 5)	1.57	2.30	Cluster1
p08	h	(7, 3)	4.37	0.56	Cluster2
p09	i	(7, 5)	3.43	1.52	Cluster2
p10	j	(8, 5)	4.41	1.95	Cluster2

m1	(3.67, 5.83)
m2	(6.75, 3.50)

K-Means Clustering

stop when no more new assignment



Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	1.95	3.78	Cluster1
p02	b	(3, 6)	0.69	4.51	Cluster1
p03	c	(3, 8)	2.27	5.86	Cluster1
p04	d	(4, 5)	0.89	3.13	Cluster1
p05	e	(4, 7)	1.22	4.45	Cluster1
p06	f	(5, 1)	5.01	3.05	Cluster2
p07	g	(5, 5)	1.57	2.30	Cluster1
p08	h	(7, 3)	4.37	0.56	Cluster2
p09	i	(7, 5)	3.43	1.52	Cluster2
p10	j	(8, 5)	4.41	1.95	Cluster2

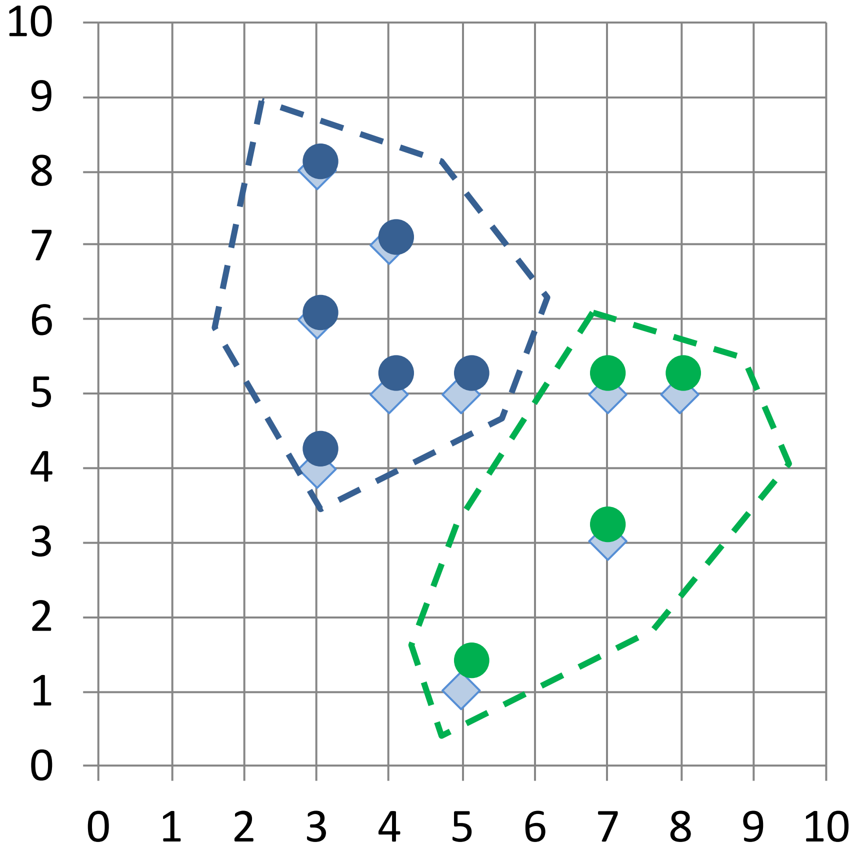
m1 (3.67, 5.83)

m2 (6.75, 3.50)

K-Means Clustering

K-Means Clustering (*K*=2, two clusters)

stop when no more new assignment



Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	1.95	3.78	Cluster1
p02	b	(3, 6)	0.69	4.51	Cluster1
p03	c	(3, 8)	2.27	5.86	Cluster1
p04	d	(4, 5)	0.89	3.13	Cluster1
p05	e	(4, 7)	1.22	4.45	Cluster1
p06	f	(5, 1)	5.01	3.05	Cluster2
p07	g	(5, 5)	1.57	2.30	Cluster1
p08	h	(7, 3)	4.37	0.56	Cluster2
p09	i	(7, 5)	3.43	1.52	Cluster2
p10	j	(8, 5)	4.41	1.95	Cluster2

m1 (3.67, 5.83)

m2 (6.75, 3.50)

K-Means Clustering

K-Means Clustering

Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	1.95	3.78	Cluster1
p02	b	(3, 6)	0.69	4.51	Cluster1
p03	c	(3, 8)	2.27	5.86	Cluster1
p04	d	(4, 5)	0.89	3.13	Cluster1
p05	e	(4, 7)	1.22	4.45	Cluster1
p06	f	(5, 1)	5.01	3.05	Cluster2
p07	g	(5, 5)	1.57	2.30	Cluster1
p08	h	(7, 3)	4.37	0.56	Cluster2
p09	i	(7, 5)	3.43	1.52	Cluster2
p10	j	(8, 5)	4.41	1.95	Cluster2

m1	(3.67, 5.83)
m2	(6.75, 3.50)

Market Segmentation

Marketing

Marketing

“Meeting needs profitably”

Source: Philip Kotler and Kevin Lane Keller (2016), Marketing Management, 15th edition, Pearson.

Marketing

"**Marketing** is an organizational function and a set of processes for creating, communicating, and delivering **value** to customers and for managing customer **relationships** in ways that benefit the organization and its stakeholders."

Source: Philip Kotler and Kevin Lane Keller (2016), Marketing Management, 15th edition, Pearson.

Marketing Management

Marketing Management

"**Marketing management** is the art and science of choosing target markets and getting, keeping, and growing customers through creating, delivering, and communicating superior **customer value**."

Source: Philip Kotler and Kevin Lane Keller (2016), Marketing Management, 15th edition, Pearson.

Marketing Management Tasks

1. Developing market strategies and plans
2. Capturing marketing insights
3. Connecting with customers
4. Building strong brands
5. Creating value
6. Delivering value
7. Communicating value
8. Creating successful long-term growth

Source: Philip Kotler and Kevin Lane Keller (2016), Marketing Management, 15th edition, Pearson.

The Essence of Strategic Marketing (STP)

Segmentation

Targeting

Positioning

Source: Philip Kotler and Kevin Lane Keller (2016), Marketing Management, 15th edition, Pearson.

Machine Learning

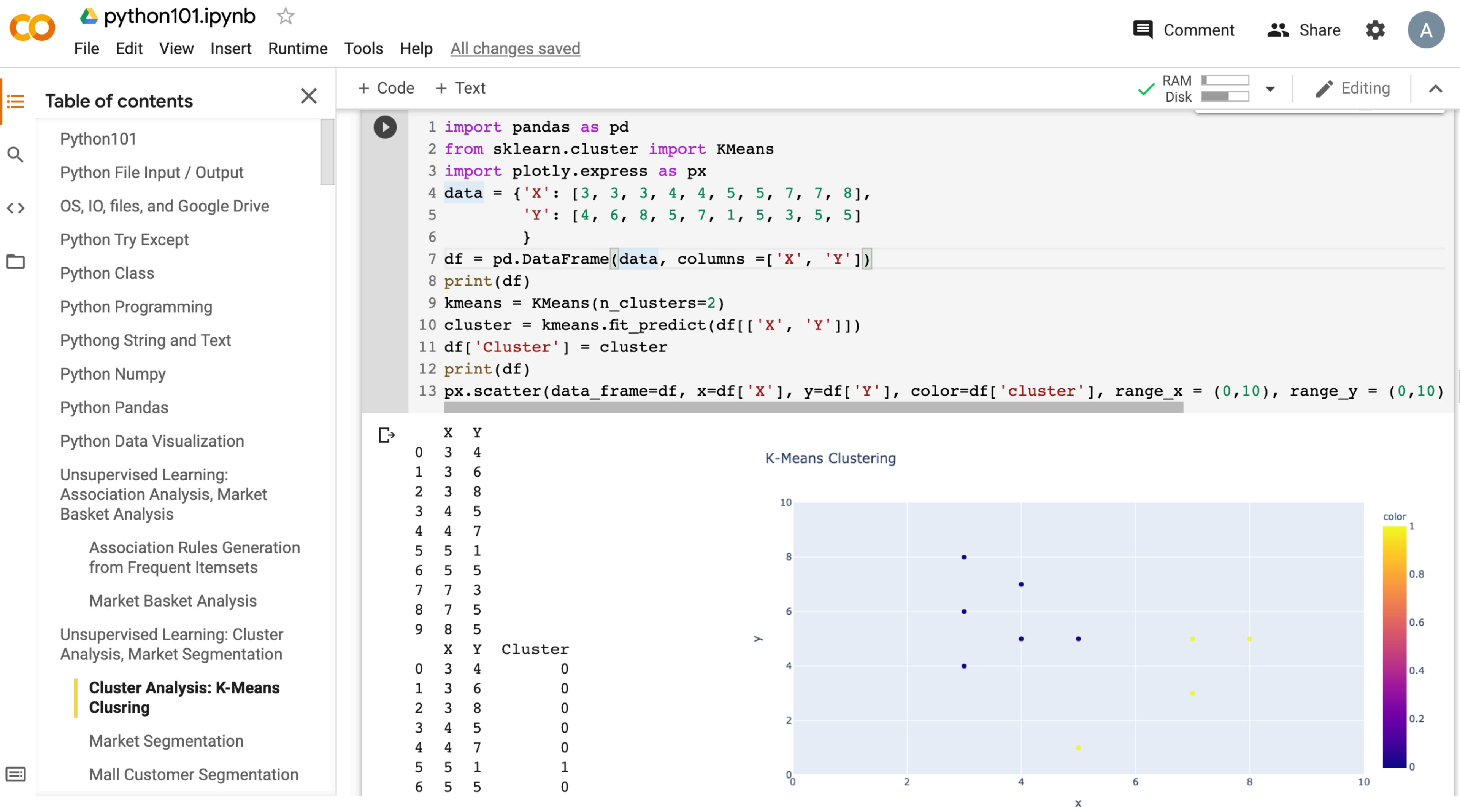
Unsupervised Learning

Cluster Analysis

K-Means Clustering

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

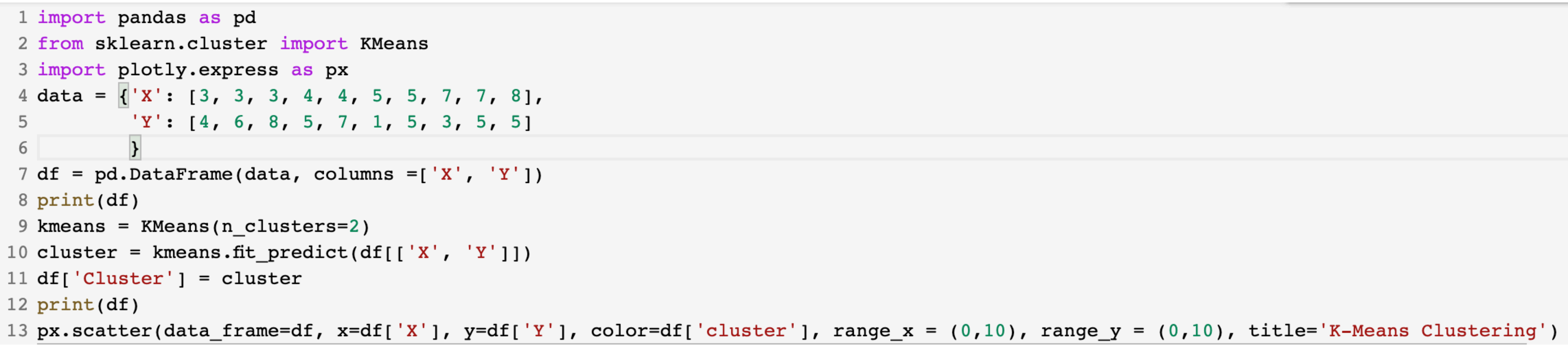

python101.ipynb
File
Edit
View
Insert
Runtime
Tools
Help
All changes saved
Comment
Share
A
Table of contents
Python101
Python File Input / Output
OS, IO, files, and Google Drive
Python Try Except
Python Class
Python Programming
Pythong String and Text
Python Numpy
Python Pandas
Python Data Visualization
Unsupervised Learning: Association Analysis, Market Basket Analysis
Association Rules Generation from Frequent Itemsets
Market Basket Analysis
Unsupervised Learning: Cluster Analysis, Market Segmentation
Cluster Analysis: K-Means Clusring
Market Segmentation
Mall Customer Segmentation
+ Code
+ Text
RAM
Disk
Editing
K-Means Clustering
y
x
0
2
4
6
8
10
color
1
0.8
0.6
0.4
0.2
0

```
import pandas as pd
from sklearn.cluster import KMeans
import plotly.express as px
data = {'X': [3, 3, 3, 4, 4, 5, 5, 7, 7, 8],
        'Y': [4, 6, 8, 5, 7, 1, 5, 3, 5, 5]
        }
df = pd.DataFrame(data, columns =['X', 'Y'])
print(df)
kmeans = KMeans(n_clusters=2)
cluster = kmeans.fit_predict(df[['X', 'Y']])
df['Cluster'] = cluster
print(df)
px.scatter(data_frame=df, x=df['X'], y=df['Y'], color=df['cluster'], range_x = (0,10), range_y = (0,10)
```

```
   X  Y
0  3  4
1  3  6
2  3  8
3  4  5
4  4  7
5  5  1
6  5  5
7  7  3
8  7  5
9  8  5
   X  Y  Cluster
0  3  4        0
1  3  6        0
2  3  8        0
3  4  5        0
4  4  7        0
5  5  1        1
6  5  5        0
```

https://tinyurl.com/aintpupython101

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=2)
cluster = kmeans.fit_predict(df[['X', 'Y']])
```

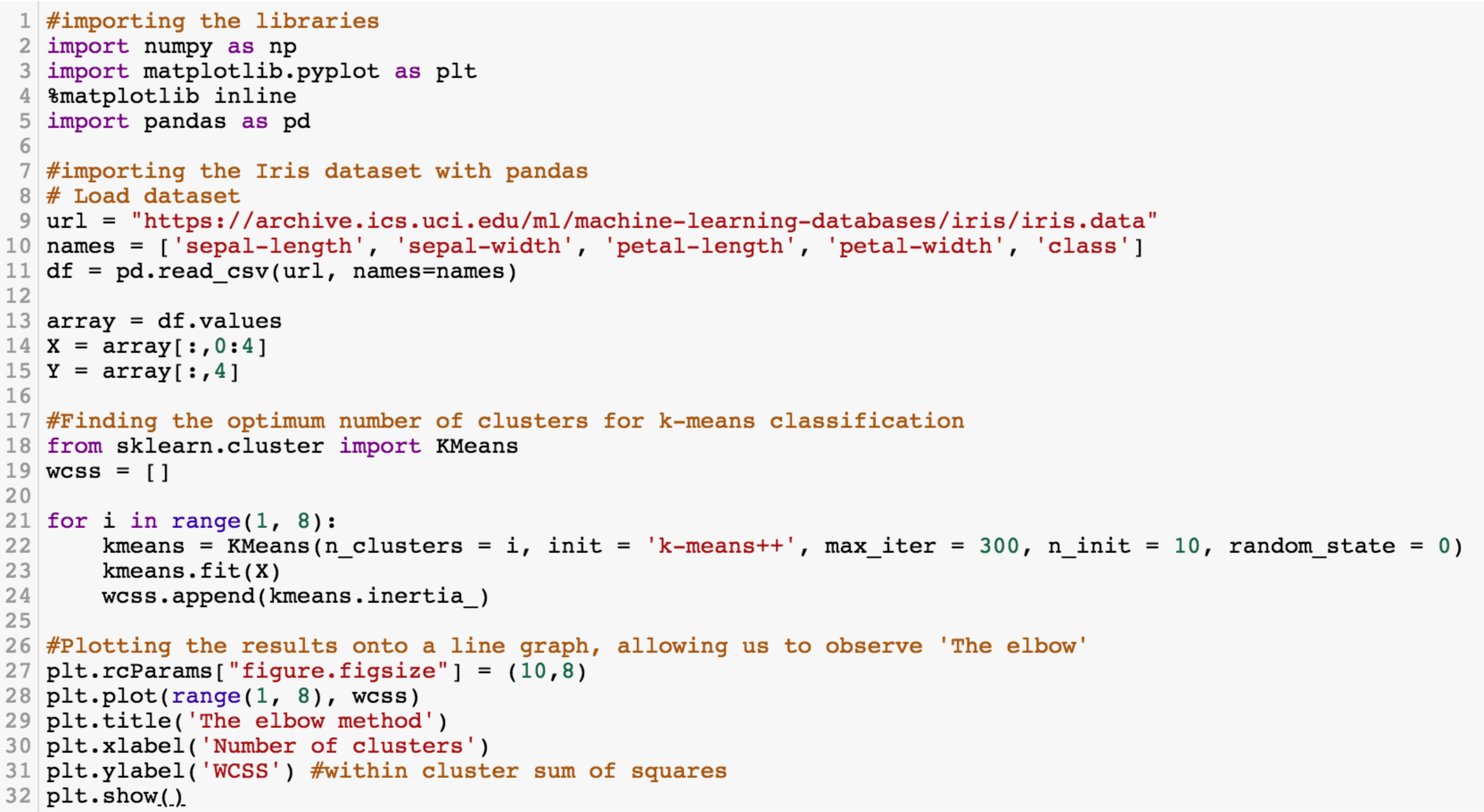


```
import pandas as pd
from sklearn.cluster import KMeans
import plotly.express as px
data = {'X': [3, 3, 3, 4, 4, 5, 5, 7, 7, 8],
        'Y': [4, 6, 8, 5, 7, 1, 5, 3, 5, 5]
        }
df = pd.DataFrame(data, columns =['X', 'Y'])
print(df)
kmeans = KMeans(n_clusters=2)
cluster = kmeans.fit_predict(df[['X', 'Y']])
df['Cluster'] = cluster
print(df)
px.scatter(data_frame=df, x=df['X'], y=df['Y'], color=df['cluster'], range_x = (0,10), range_y = (0,10), title='K-Means Clustering')
```

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=2)
cluster = kmeans.fit_predict(df[['X', 'Y']])
```

```
import pandas as pd
from sklearn.cluster import KMeans
import plotly.express as px
data = {'X': [3, 3, 3, 4, 4, 5, 5, 7, 7, 8],
'Y': [4, 6, 8, 5, 7, 1, 5, 3, 5, 5]
}
df = pd.DataFrame(data, columns =['X', 'Y'])
print(df)
kmeans = KMeans(n_clusters=2)
cluster = kmeans.fit_predict(df[['X', 'Y']])
df['Cluster'] = cluster
print(df)
px.scatter(data_frame=df, x=df['X'], y=df['Y'],
color=df['cluster'], range_x = (0,10), range_y = (0,10),
title='K-Means Clustering')
```

K-Means Clustering



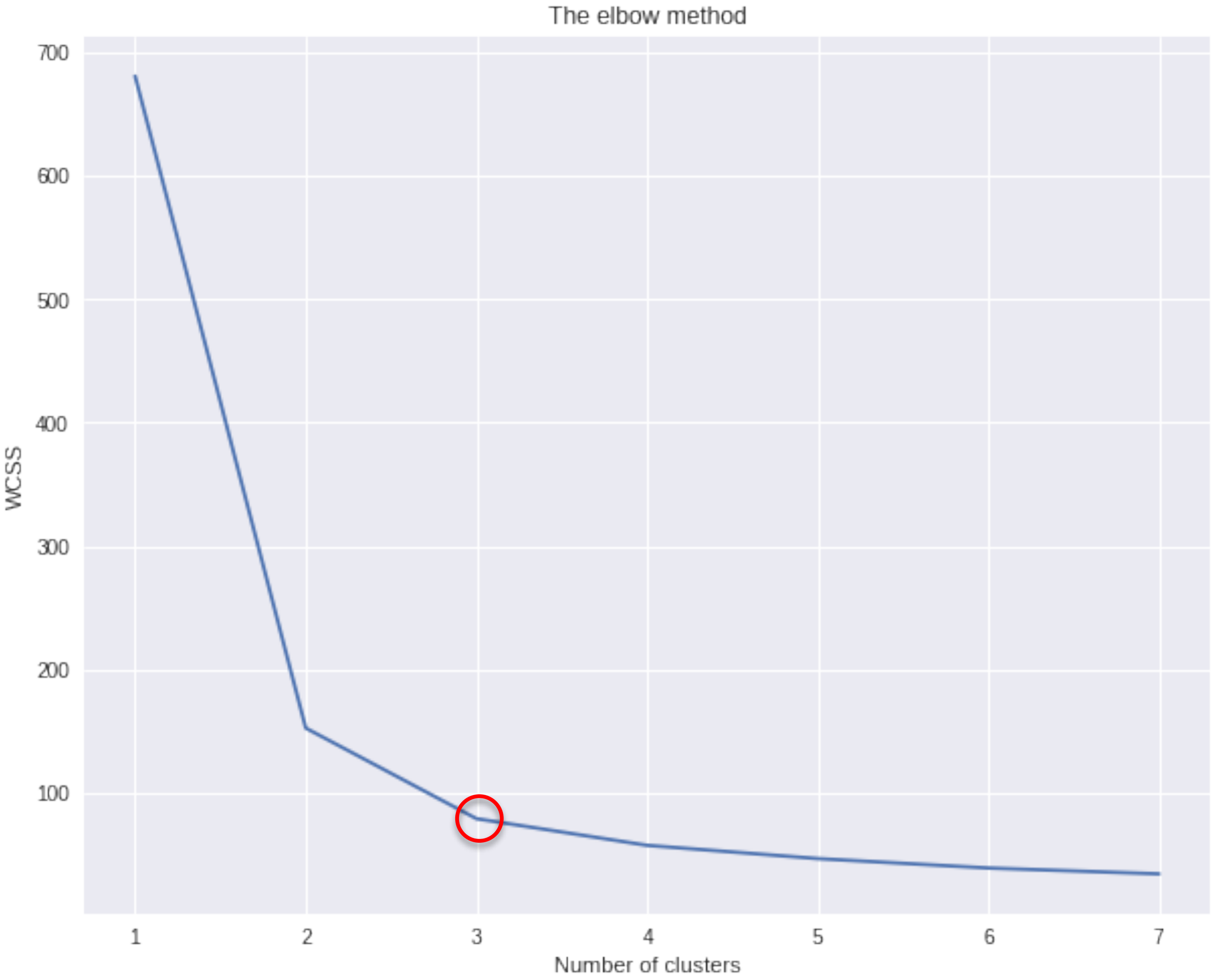
```python
#importing the libraries
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import pandas as pd

#importing the Iris dataset with pandas
# Load dataset
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'class']
df = pd.read_csv(url, names=names)

array = df.values
X = array[:,0:4]
Y = array[:,4]

#Finding the optimum number of clusters for k-means classification
from sklearn.cluster import KMeans
wcss = []

for i in range(1, 8):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', max_iter = 300, n_init = 10, random_state = 0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)

#Plotting the results onto a line graph, allowing us to observe 'The elbow'
plt.rcParams["figure.figsize"] = (10,8)
plt.plot(range(1, 8), wcss)
plt.title('The elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS') #within cluster sum of squares
plt.show()
```

```
#importing the libraries
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import pandas as pd

#importing the Iris dataset with pandas
# Load dataset
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
names = ['sepal-length', 'sepal-width',
'petal-length', 'petal-width', 'class']
df = pd.read_csv(url, names=names)

array = df.values
X = array[:,0:4]
Y = array[:,4]
```

```
#Finding the optimum number of clusters for k-means
classification
from sklearn.cluster import KMeans
wcss = []

for i in range(1, 8):
    kmeans = KMeans(n_clusters = i, init = 'k-means++',
max_iter = 300, n_init = 10, random_state = 0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)

#Plotting the results onto a line graph, allowing us to
observe 'The elbow'
plt.rcParams["figure.figsize"] = (10,8)
plt.plot(range(1, 8), wcss)
plt.title('The elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS') #within cluster sum of squares
plt.show()
```

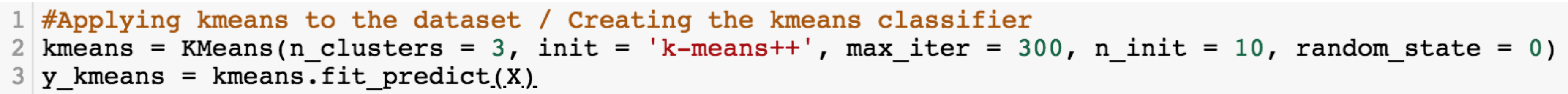
K-Means Clustering

The elbow method (*k=3*)



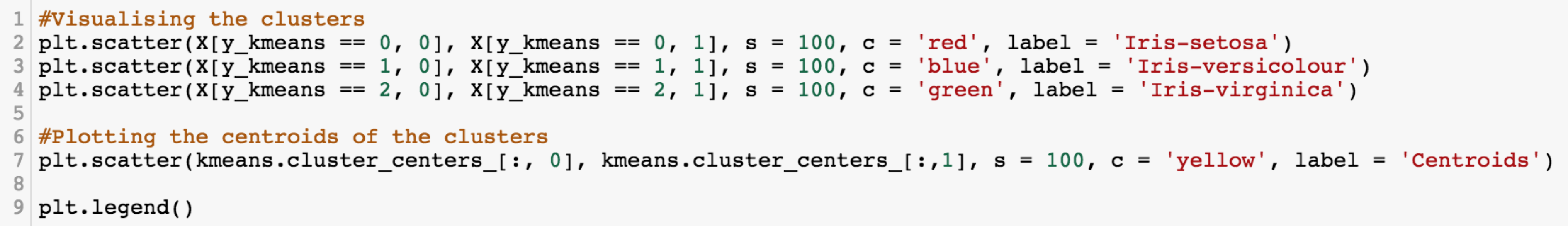
https://tinyurl.com/aintpupython101

```
kmeans = KMeans(n_clusters = 3,
init = 'k-means++', max_iter = 300,
n_init = 10, random_state = 0)
y_kmeans = kmeans.fit_predict(X)
```



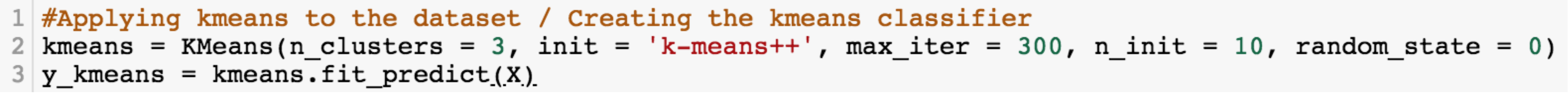
```
#Applying kmeans to the dataset / Creating the kmeans classifier
kmeans = KMeans(n_clusters = 3, init = 'k-means++', max_iter = 300, n_init = 10, random_state = 0)
y_kmeans = kmeans.fit_predict(X)
```

```
#Visualising the clusters
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100,
c = 'red', label = 'Iris-setosa')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100,
c = 'blue', label = 'Iris-versicolour')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100,
c = 'green', label = 'Iris-virginica')

#Plotting the centroids of the clusters
plt.scatter(kmeans.cluster_centers_[:, 0],
kmeans.cluster_centers_[:,1], s = 100, c = 'yellow', label =
'Centroids')

plt.legend()
```

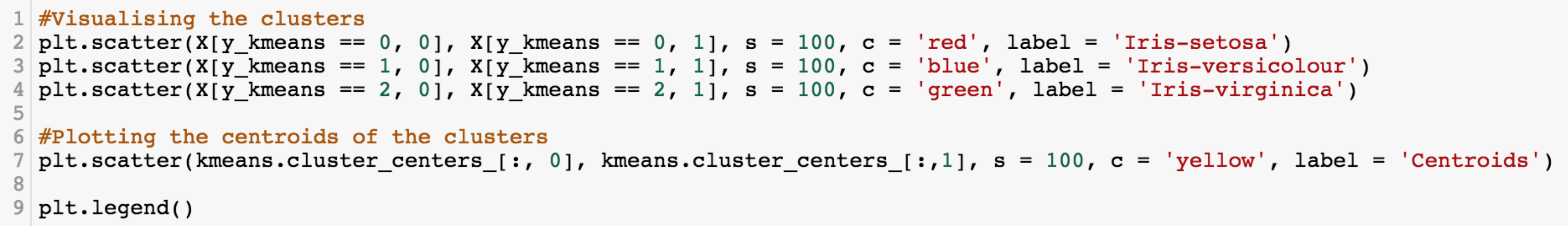


```
#Visualising the clusters
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Iris-setosa')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Iris-versicolour')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Iris-virginica')

#Plotting the centroids of the clusters
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:,1], s = 100, c = 'yellow', label = 'Centroids')

plt.legend()
```

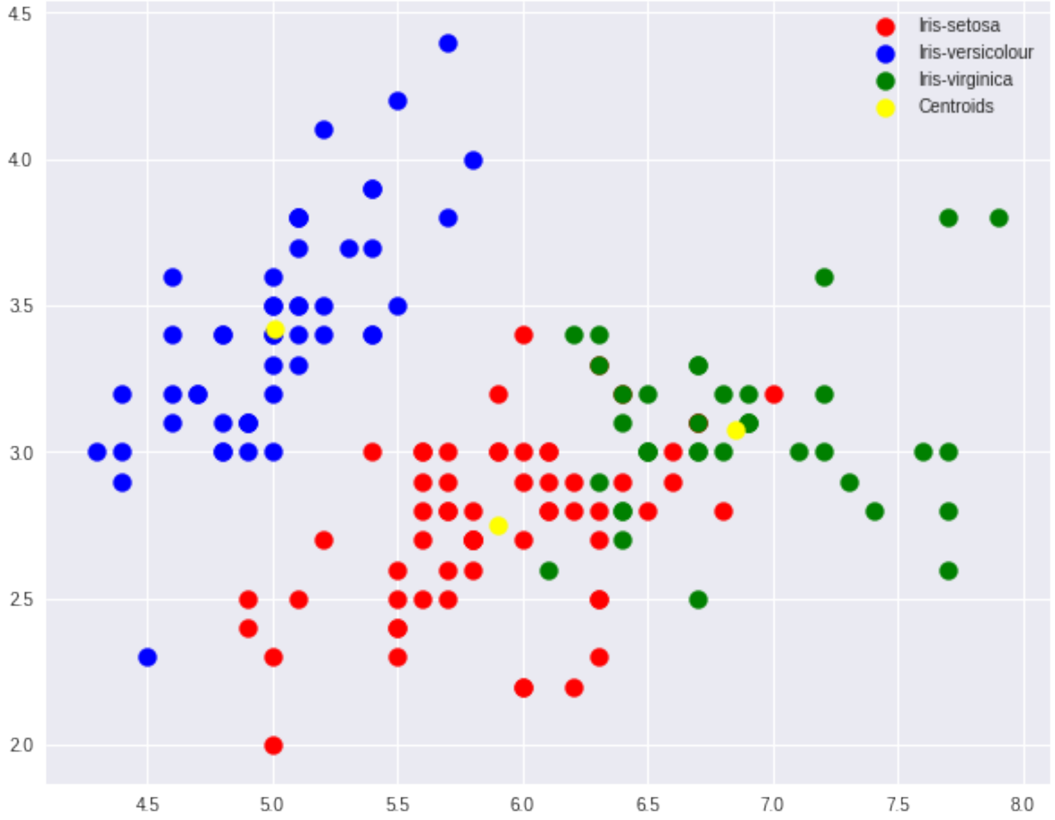
K-Means Clustering



```
#Applying kmeans to the dataset / Creating the kmeans classifier
kmeans = KMeans(n_clusters = 3, init = 'k-means++', max_iter = 300, n_init = 10, random_state = 0)
y_kmeans = kmeans.fit_predict(X)
```

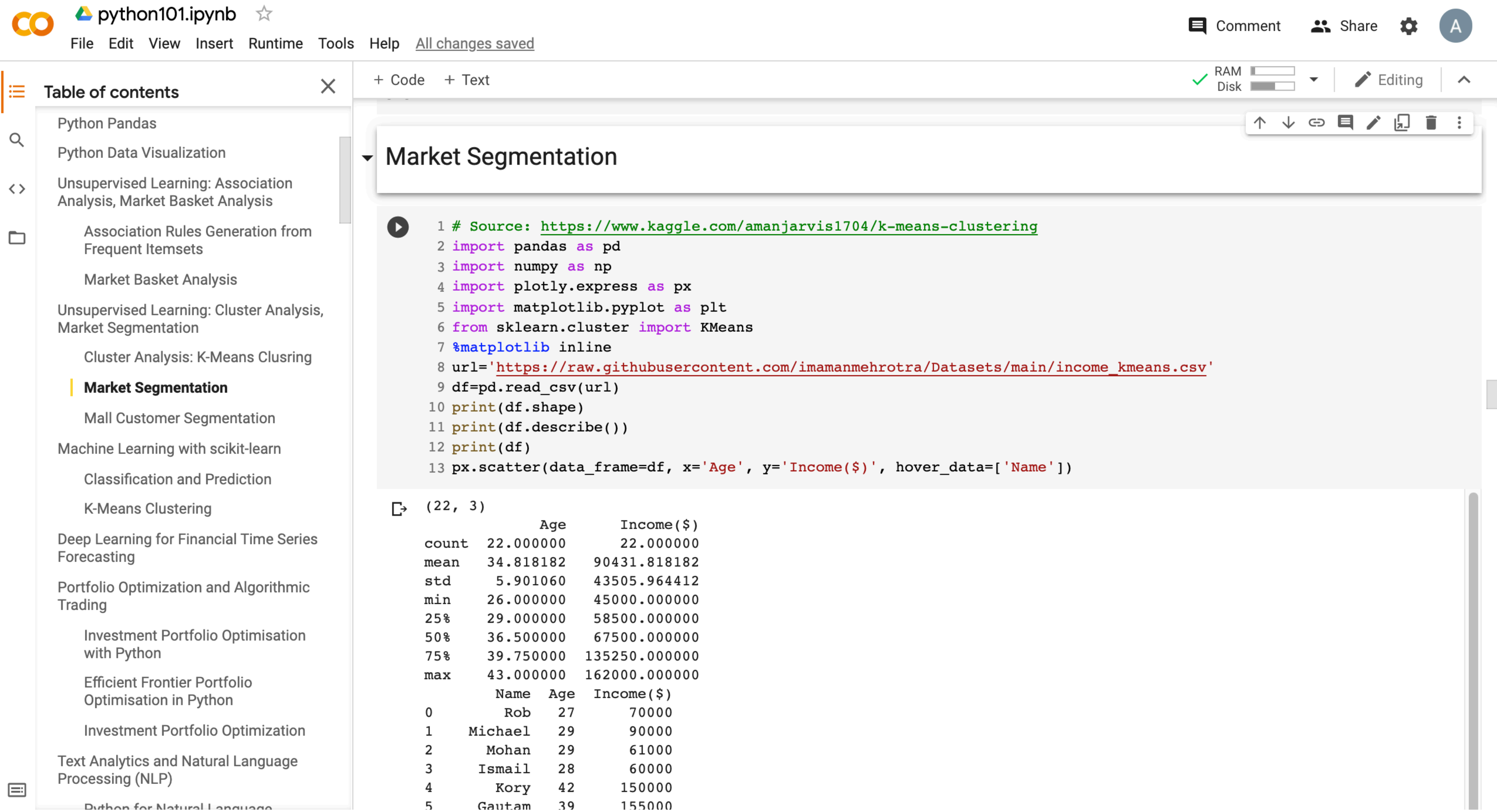


```
#Visualising the clusters
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Iris-setosa')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Iris-versicolour')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Iris-virginica')

#Plotting the centroids of the clusters
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:,1], s = 100, c = 'yellow', label = 'Centroids')

plt.legend()
```



https://tinyurl.com/aintpupython101

Market Segmentation


python101.ipynb
File
Edit
View
Insert
Runtime
Tools
Help
All changes saved
Comment
Share
A
Table of contents
Python Pandas
Python Data Visualization
Unsupervised Learning: Association Analysis, Market Basket Analysis
Association Rules Generation from Frequent Itemsets
Market Basket Analysis
Unsupervised Learning: Cluster Analysis, Market Segmentation
Cluster Analysis: K-Means Clusring
Market Segmentation
Mall Customer Segmentation
Machine Learning with scikit-learn
Classification and Prediction
K-Means Clustering
Deep Learning for Financial Time Series Forecasting
Portfolio Optimization and Algorithmic Trading
Investment Portfolio Optimisation with Python
Efficient Frontier Portfolio Optimisation in Python
Investment Portfolio Optimization
Text Analytics and Natural Language Processing (NLP)
Python for Natural Language
+ Code
+ Text
RAM
Disk
Editing

Market Segmentation

```python
# Source: https://www.kaggle.com/amanjarvis1704/k-means-clustering
import pandas as pd
import numpy as np
import plotly.express as px
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
%matplotlib inline
url='https://raw.githubusercontent.com/imamanmehrotra/Datasets/main/income_kmeans.csv'
df=pd.read_csv(url)
print(df.shape)
print(df.describe())
print(df)
px.scatter(data_frame=df, x='Age', y='Income($)', hover_data=['Name'])
```

```
(22, 3)
             Age      Income($)
count  22.000000      22.000000
mean   34.818182   90431.818182
std     5.901060   43505.964412
min    26.000000   45000.000000
25%    29.000000   58500.000000
50%    36.500000   67500.000000
75%    39.750000  135250.000000
max    43.000000  162000.000000
        Name  Age  Income($)
0        Rob   27      70000
1    Michael   29      90000
2      Mohan   29      61000
3     Ismail   28      60000
4       Kory   42     150000
5     Gautam   39     155000
```

https://tinyurl.com/aintpupython101

Mall Customer Segmentation

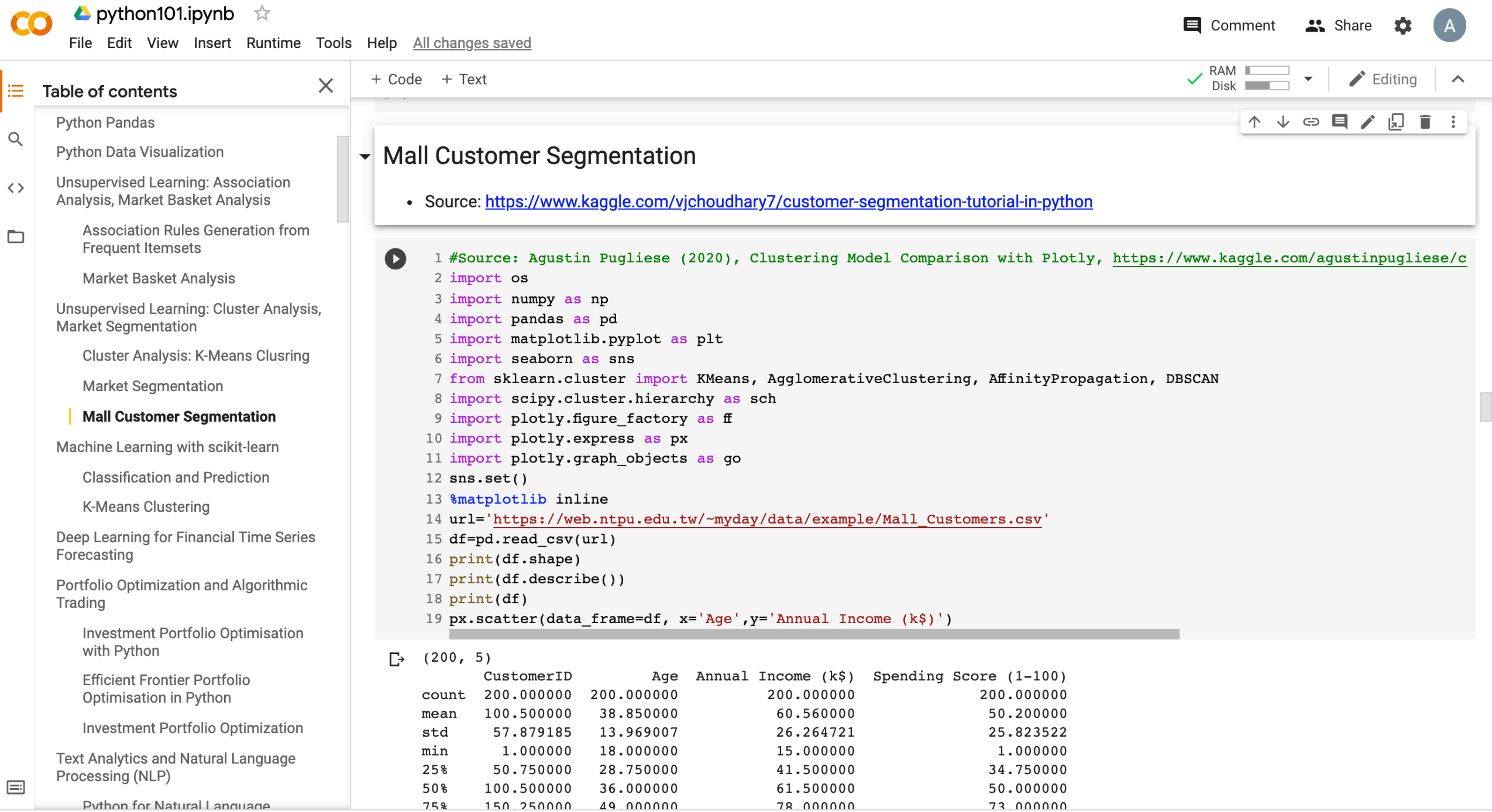


Table of contents

Mall Customer Segmentation

- Source: https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial-in-python

```python
#Source: Agustin Pugliese (2020), Clustering Model Comparison with Plotly, https://www.kaggle.com/agustinpugliese/c
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans, AgglomerativeClustering, AffinityPropagation, DBSCAN
import scipy.cluster.hierarchy as sch
import plotly.figure_factory as ff
import plotly.express as px
import plotly.graph_objects as go
sns.set()
%matplotlib inline
url='https://web.ntpu.edu.tw/~myday/data/example/Mall_Customers.csv'
df=pd.read_csv(url)
print(df.shape)
print(df.describe())
print(df)
px.scatter(data_frame=df, x='Age',y='Annual Income (k$)')
```

```
(200, 5)
       CustomerID         Age  Annual Income (k$)  Spending Score (1-100)
count  200.000000  200.000000          200.000000              200.000000
mean   100.500000   38.850000           60.560000               50.200000
std     57.879185   13.969007           26.264721               25.823522
min      1.000000   18.000000           15.000000                1.000000
25%     50.750000   28.750000           41.500000               34.750000
50%    100.500000   36.000000           61.500000               50.000000
75%    150.250000   49.000000           78.000000               73.000000
```

https://tinyurl.com/aintpupython101

Mall Customer Segmentation

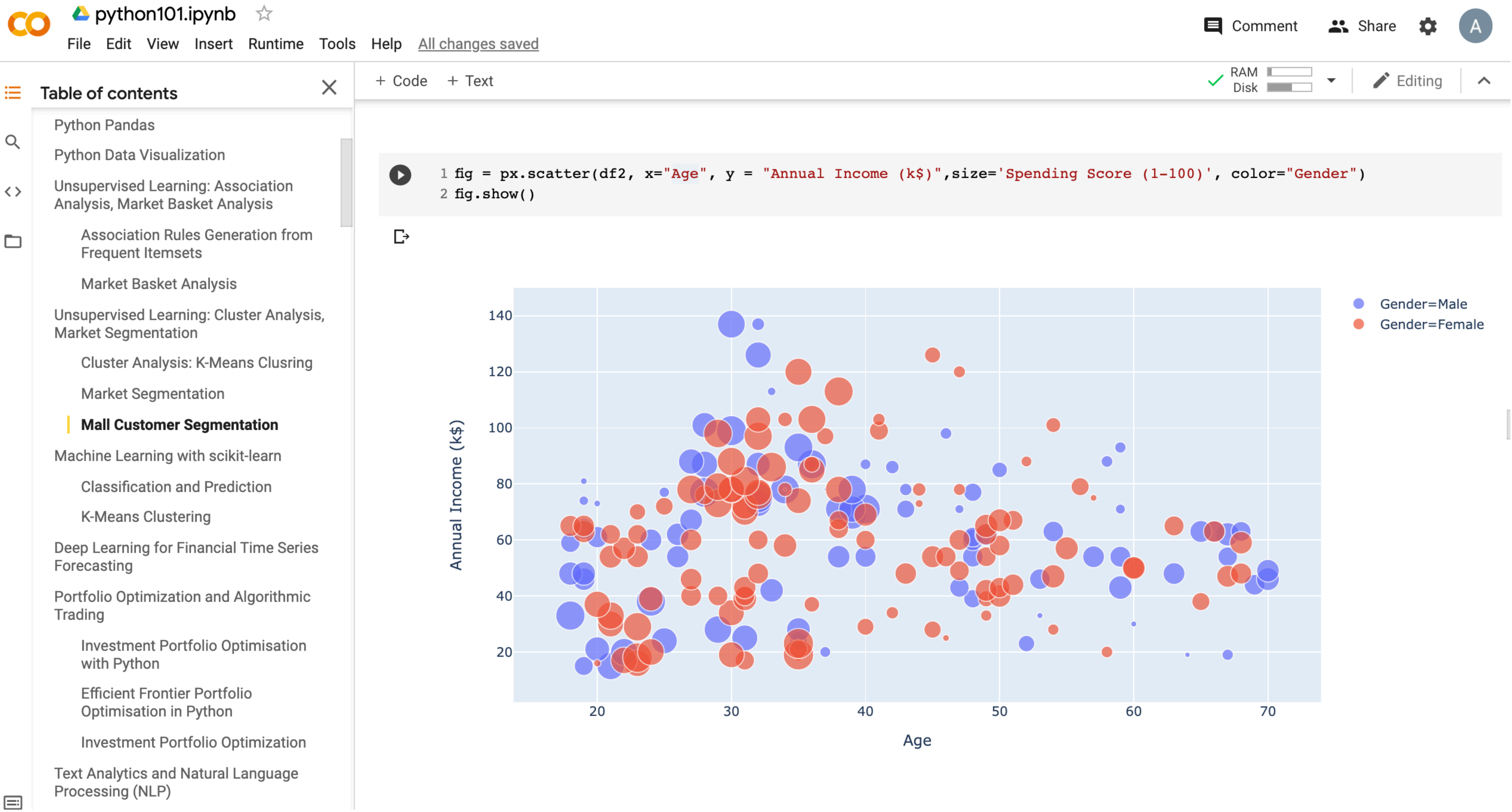
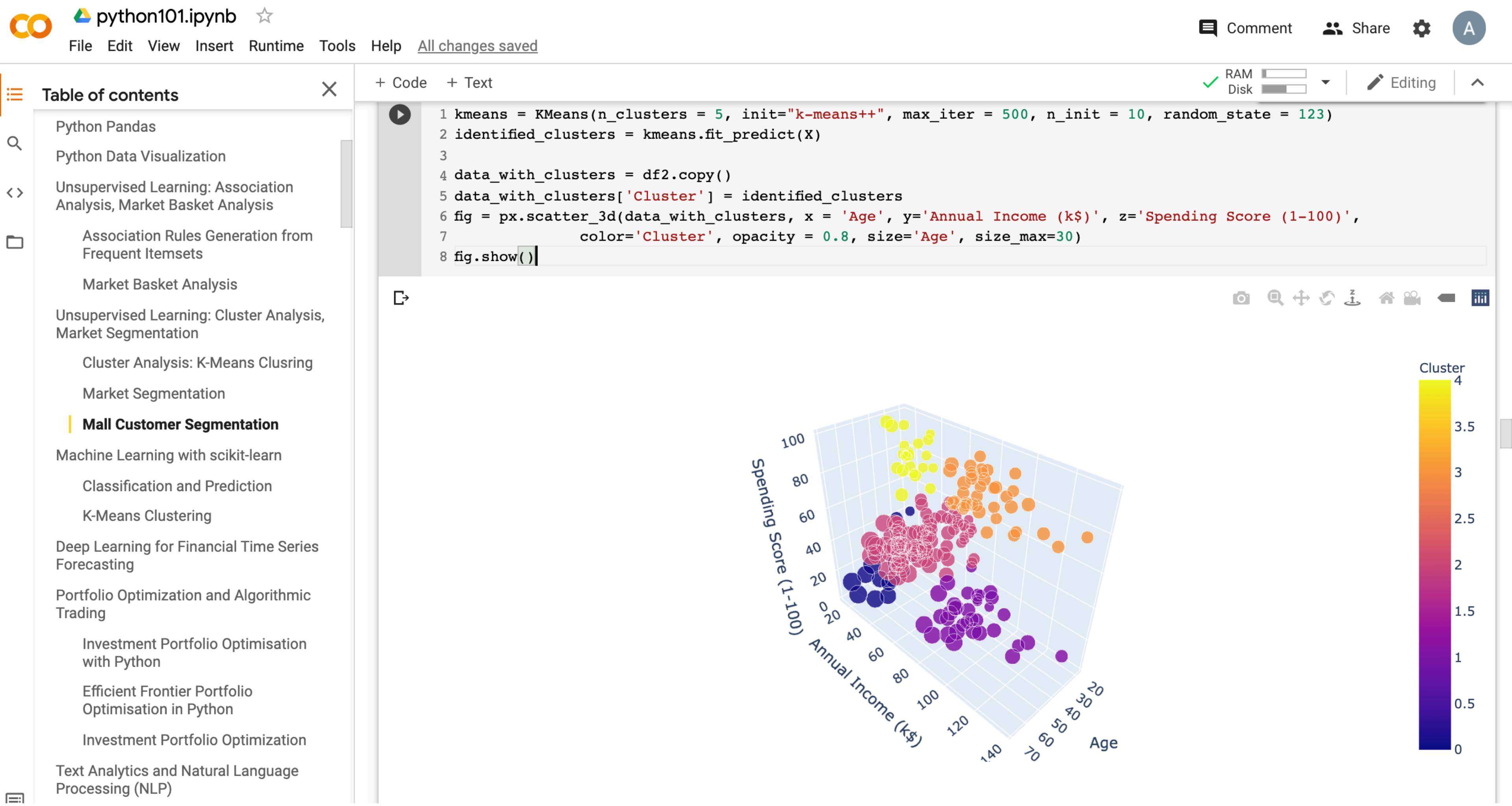


Table of contents

```
fig = px.scatter(df2, x="Age", y = "Annual Income (k$)",size='Spending Score (1-100)', color="Gender")
fig.show()
```

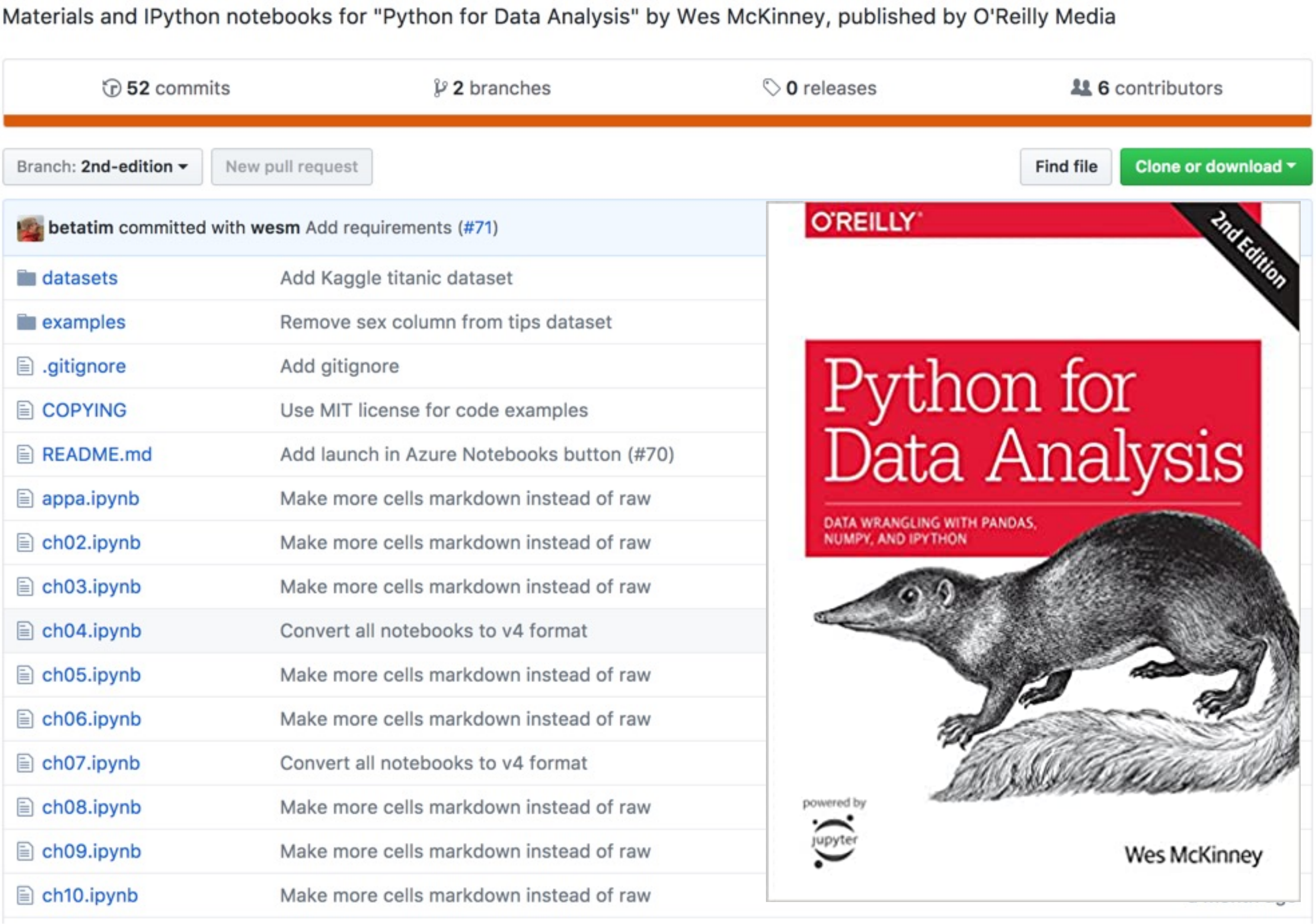
https://tinyurl.com/aintpupython101

Mall Customer Segmentation



```python
kmeans = KMeans(n_clusters = 5, init="k-means++", max_iter = 500, n_init = 10, random_state = 123)
identified_clusters = kmeans.fit_predict(X)

data_with_clusters = df2.copy()
data_with_clusters['Cluster'] = identified_clusters
fig = px.scatter_3d(data_with_clusters, x = 'Age', y='Annual Income (k$)', z='Spending Score (1-100)',
             color='Cluster', opacity = 0.8, size='Age', size_max=30)
fig.show()
```

Wes McKinney (2017), "Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython", 2nd Edition, O'Reilly Media.



Materials and IPython notebooks for "Python for Data Analysis" by Wes McKinney, published by O'Reilly Media

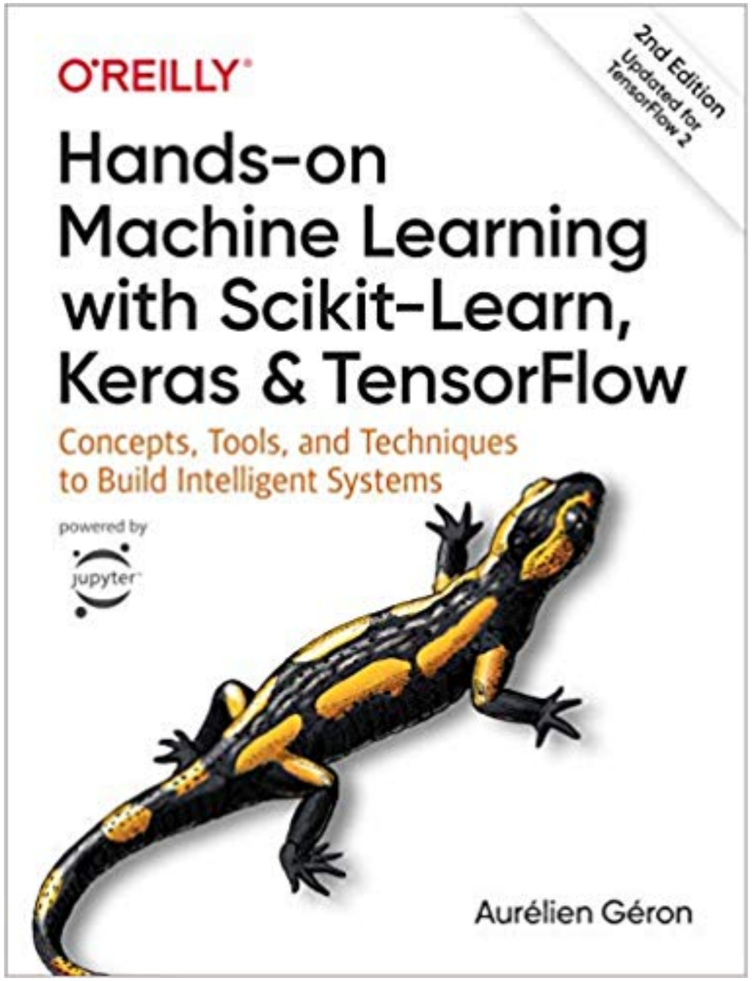
52 commits | 2 branches | 0 releases | 6 contributors

betatim committed with wesm Add requirements (#71)

datasets	Add Kaggle titanic dataset
examples	Remove sex column from tips dataset
.gitignore	Add gitignore
COPYING	Use MIT license for code examples
README.md	Add launch in Azure Notebooks button (#70)
appa.ipynb	Make more cells markdown instead of raw
ch02.ipynb	Make more cells markdown instead of raw
ch03.ipynb	Make more cells markdown instead of raw
ch04.ipynb	Convert all notebooks to v4 format
ch05.ipynb	Make more cells markdown instead of raw
ch06.ipynb	Make more cells markdown instead of raw
ch07.ipynb	Convert all notebooks to v4 format
ch08.ipynb	Make more cells markdown instead of raw
ch09.ipynb	Make more cells markdown instead of raw
ch10.ipynb	Make more cells markdown instead of raw

https://github.com/wesm/pydata-book

Aurélien Géron (2019), Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd Edition O’Reilly Media, 2019


O'REILLY
2nd Edition
Updated for TensorFlow 2
Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow
Concepts, Tools, and Techniques to Build Intelligent Systems
powered by
jupyter
Aurélien Géron

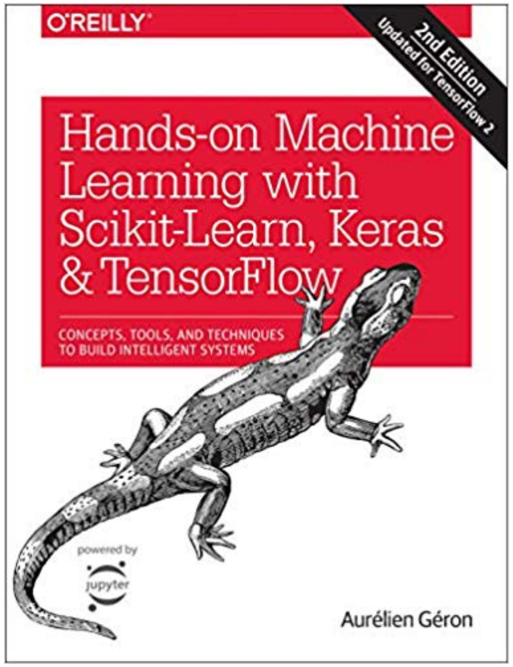
https://github.com/ageron/handson-ml2

Source: https://www.amazon.com/Hands-Machine-Learning-Scikit-Learn-TensorFlow/dp/1492032646/

Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow

Notebooks

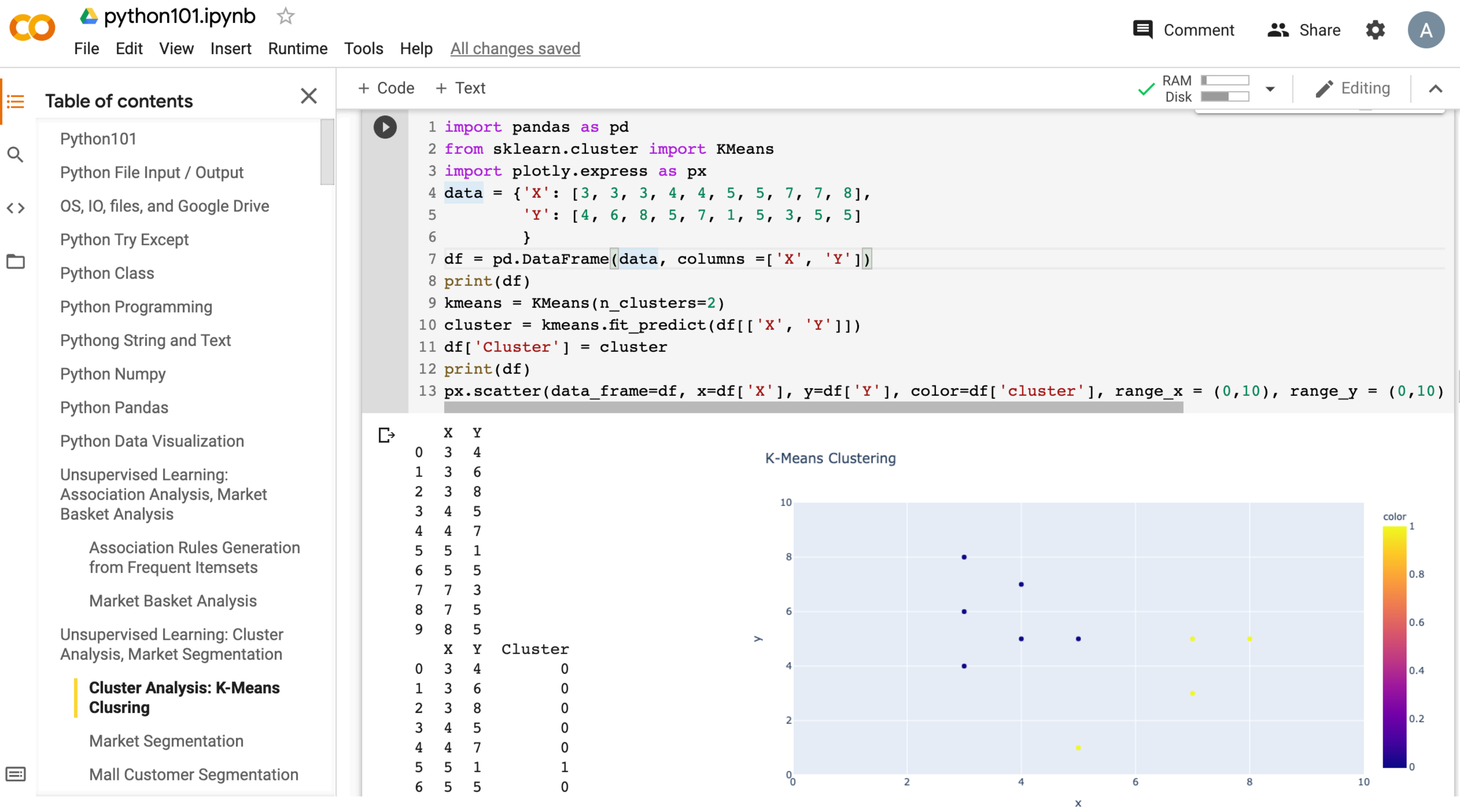
1. The Machine Learning landscape
2. End-to-end Machine Learning project
3. Classification
4. Training Models
5. Support Vector Machines
6. Decision Trees
7. Ensemble Learning and Random Forests
8. Dimensionality Reduction
9. Unsupervised Learning Techniques
10. Artificial Neural Nets with Keras
11. Training Deep Neural Networks
12. Custom Models and Training with TensorFlow
13. Loading and Preprocessing Data
14. Deep Computer Vision Using Convolutional Neural Networks
15. Processing Sequences Using RNNs and CNNs
16. Natural Language Processing with RNNs and Attention
17. Representation Learning Using Autoencoders
18. Reinforcement Learning
19. Training and Deploying TensorFlow Models at Scale


O'REILLY
2nd Edition
Updated for TensorFlow 2
Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow
CONCEPTS, TOOLS, AND TECHNIQUES TO BUILD INTELLIGENT SYSTEMS
powered by
jupyter
Aurélien Géron

https://github.com/ageron/handson-ml2

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT



```python
import pandas as pd
from sklearn.cluster import KMeans
import plotly.express as px
data = {'X': [3, 3, 3, 4, 4, 5, 5, 7, 7, 8],
        'Y': [4, 6, 8, 5, 7, 1, 5, 3, 5, 5]
       }
df = pd.DataFrame(data, columns =['X', 'Y'])
print(df)
kmeans = KMeans(n_clusters=2)
cluster = kmeans.fit_predict(df[['X', 'Y']])
df['Cluster'] = cluster
print(df)
px.scatter(data_frame=df, x=df['X'], y=df['Y'], color=df['cluster'], range_x = (0,10), range_y = (0,10)
```

```
   X  Y
0  3  4
1  3  6
2  3  8
3  4  5
4  4  7
5  5  1
6  5  5
7  7  3
8  7  5
9  8  5
   X  Y  Cluster
0  3  4        0
1  3  6        0
2  3  8        0
3  4  5        0
4  4  7        0
5  5  1        1
6  5  5        0
```

https://tinyurl.com/aintpupython101

Summary

- Unsupervised Learning
- Cluster Analysis
- Market Segmentation
- K-Means Clustering

References

- Jiawei Han and Micheline Kamber (2006), Data Mining: Concepts and Techniques, Second Edition, Elsevier, 2006.
- Jiawei Han, Micheline Kamber and Jian Pei (2011), Data Mining: Concepts and Techniques, Third Edition, Morgan Kaufmann 2011.
- Efraim Turban, Ramesh Sharda, Dursun Delen (2011), Decision Support and Business Intelligence Systems, Ninth Edition, Pearson.
- Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson.
- Jake VanderPlas (2016), Python Data Science Handbook: Essential Tools for Working with Data, O'Reilly Media.
- Robert Layton (2017), Learning Data Mining with Python - Second Edition, Packt Publishing.
- Wes McKinney (2017), "Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython", 2nd Edition, O'Reilly Media.
- Aurélien Géron (2019), Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd Edition, O’Reilly Media.
  https://github.com/wesm/pydata-book
- Min-Yuh Day (2021), Python 101, https://tinyurl.com/aintpupython101