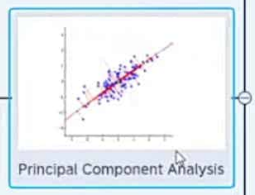
4 Algorithm



Principal Component Analysis

Definition — Dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set

1 — $z = \frac{value - mean}{standard\ deviation}$ — Standardization — All the variables need to be transformed to the same scale

2 — $\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$ — Covariance

How the variables of the input data are varying from the mean with respect to each other - Relationship between each other features

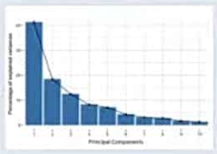
Condition
- +ve — Two variables increase or decrease together — Correlated
- -ve — One increases when the other decreases — Inversely correlated

New variables that are constructed combinations or mixtures of the in

Principal Component Analysis

Steps

3 — Eigen Values & Eigen vectors — To find the Principle Components, EG & EV need to be computed on the covariance matrix

- Principle Components
  - New variables that are constructed as linear combinations or mixtures of the initial variables
  - Computing the eigenvectors and ordering them by their eigenvalues in descending order
  - **Example**: 10-dimensional data gives you 10 principal components, but PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on
  - 
    Output
- Eigen Vector — Directions of the axes where there is the most variance(most information)
- Eigen Values — Amount of variance car… Principal Component

Dimensionality reduction

$$\begin{bmatrix} 0.6778736 & -0.7351785 \end{bmatrix}$$

File Edit Insert Tools View Help 21_ClusterringPlantIrisUsingPrincipleComponentAn... — Edited

Mind Map Outliner

Topic Subtopic Relationship Summary Boundary Insert ZEN Pitch Panel

Dimensionality reduction

4 — Feature Vector — To choose whether to keep all these components or discard those of lesser eigenvalues to form matrix with remaining ones - Feature Vector

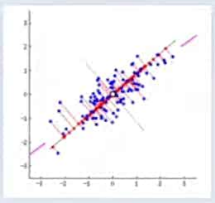
$$\begin{bmatrix} 0.6778736 & -0.7351785 \\ 0.7351785 & 0.6778736 \end{bmatrix}$$

Feature Vector - Both Eigen vectors

$$\begin{bmatrix} 0.6778736 \\ 0.7351785 \end{bmatrix}$$

Feature Vector - Only one Eigen vectors

5 —



Recasting Data along the axes of Principle components

Reorient the data from the original axes to the ones represented by the principal components

$$FinalDataSet = FeatureVector^T * StandardizedOriginalData$$

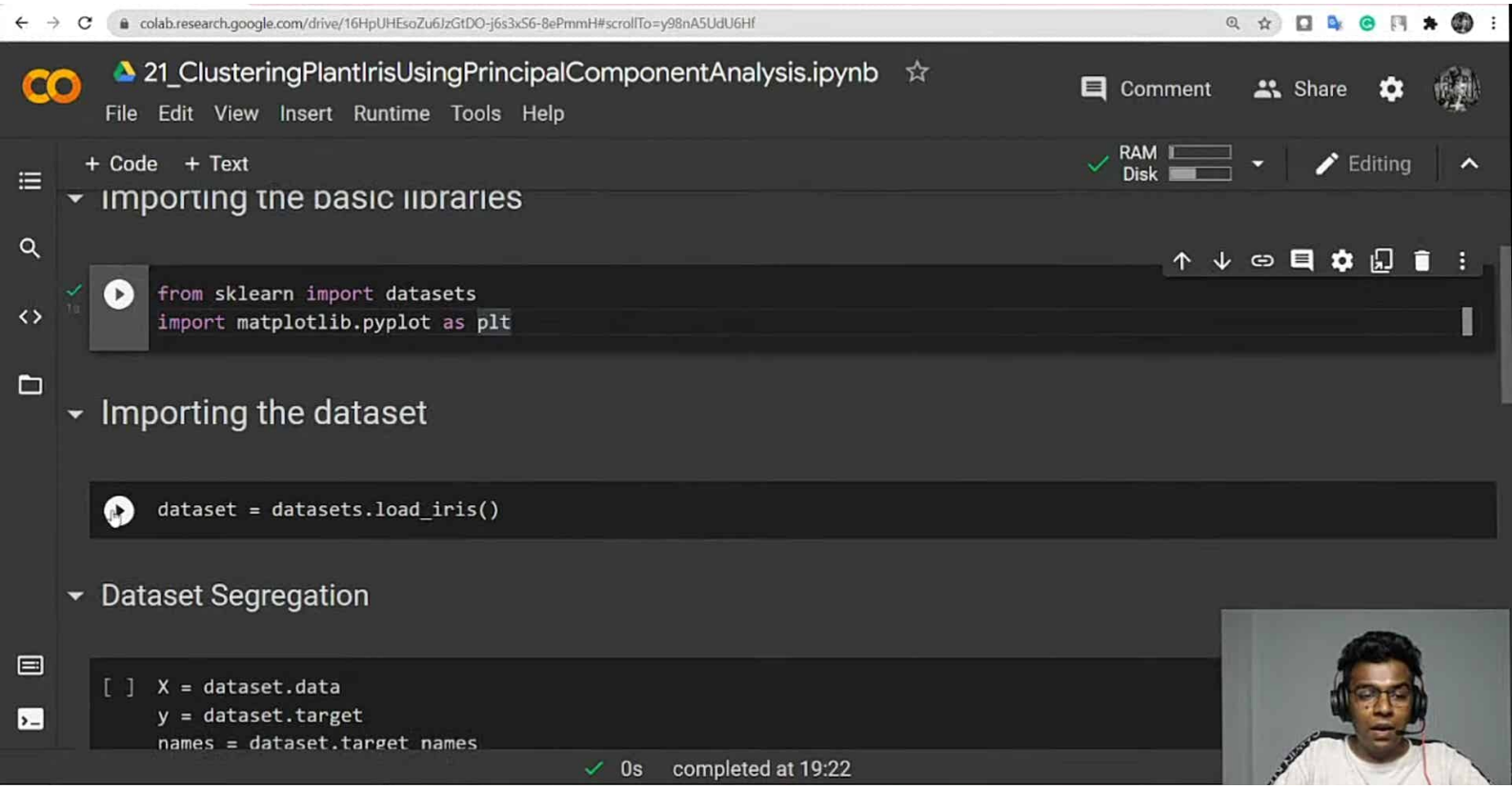
Formula - PCA

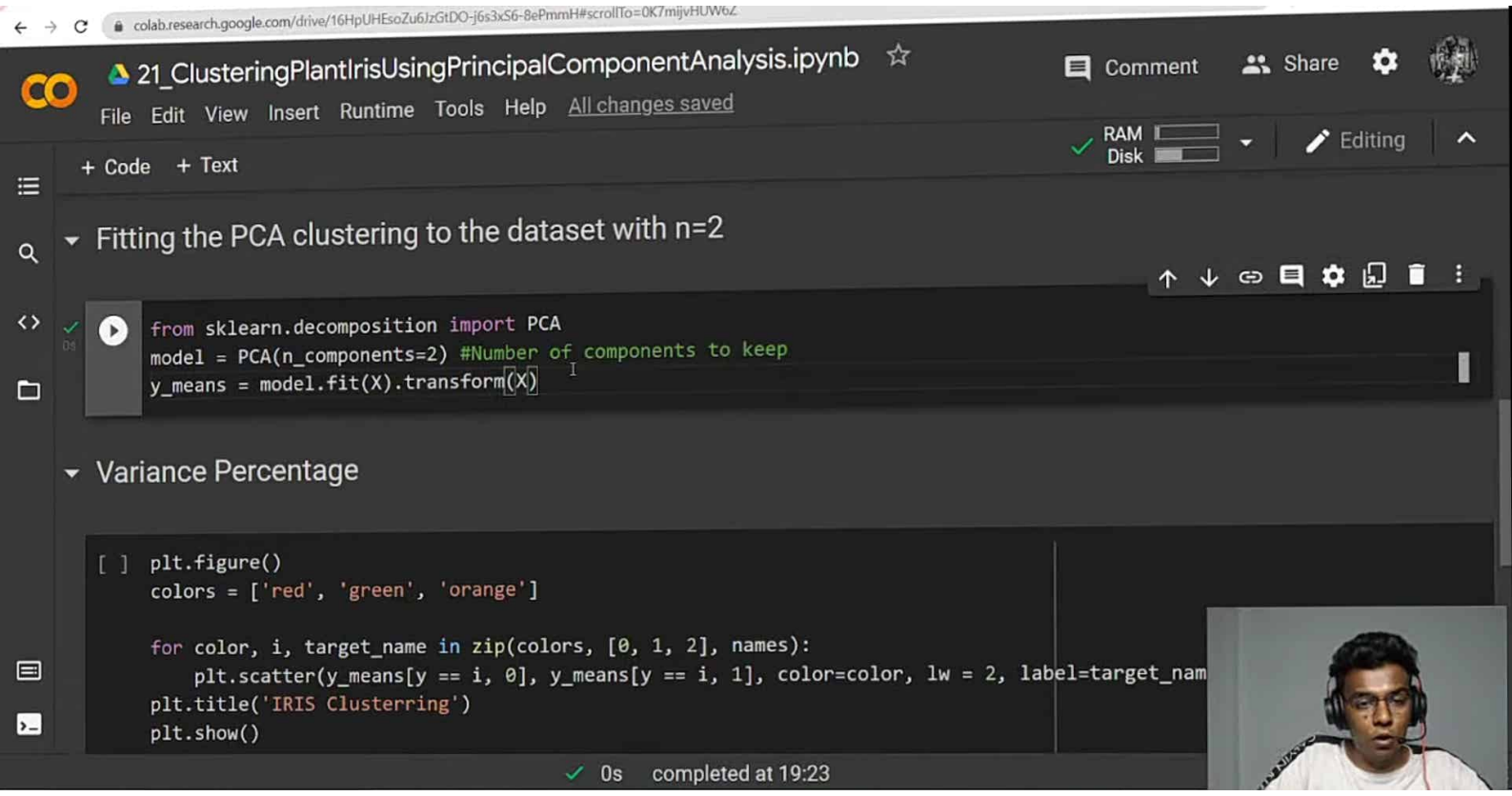
5 — Fitting the PCA clustering to the dataset by specifying the Number of components to keep

6 — Clustering via the Variance Percentage

Type here to search

30°C Haze


colab.research.google.com/drive/16HpUHEsoZu6JzGtDO-j6s3xS6-8ePmmH#scrollTo=y98nA5UdU6Hf
21_ClusteringPlantIrisUsingPrincipalComponentAnalysis.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
Editing
Importing the basic libraries
from sklearn import datasets
import matplotlib.pyplot as plt
Importing the dataset
dataset = datasets.load_iris()
Dataset Segregation
[ ] X = dataset.data
y = dataset.target
names = dataset.target_names
0s completed at 19:22


colab.research.google.com/drive/16HpUHEsoZu6JzGtDO-j6s3xS6-8ePmmH#scrollTo=0K7mijvHUW6Z
21_ClusteringPlantIrisUsingPrincipalComponentAnalysis.ipynb
Comment
Share
File Edit View Insert Runtime Tools Help All changes saved
RAM
Disk
Editing
+ Code + Text
Fitting the PCA clustering to the dataset with n=2
from sklearn.decomposition import PCA
model = PCA(n_components=2) #Number of components to keep
y_means = model.fit(X).transform(X)
Variance Percentage
[ ] plt.figure()
colors = ['red', 'green', 'orange']
for color, i, target_name in zip(colors, [0, 1, 2], names):
plt.scatter(y_means[y == i, 0], y_means[y == i, 1], color=color, lw = 2, label=target_nam
plt.title('IRIS Clusterring')
plt.show()
0s completed at 19:23

colab.research.google.com/drive/16HpUHEsoZu6JzGtDO-j6s3xS6-8ePmmH#scrollTo=PXDHbM4aUdvc

21_ClusteringPlantIrisUsingPrincipalComponentAnalysis.ipynb

File Edit View Insert Runtime Tools Help All changes saved

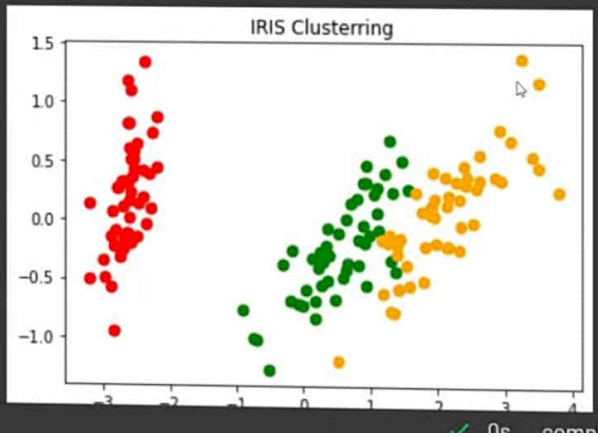
Comment Share

+ Code + Text

RAM Disk Editing

```
colors = ['red', 'green', 'orange']

for color, i, target_name in zip(colors, [0, 1, 2], names):
    plt.scatter(y_means[y == i, 0], y_means[y == i, 1], color=color, lw = 2, label=target_name)
plt.title('IRIS Clusterring')
plt.show()
```



0s completed at 10:25