Unsupervised Machine Learning with Python

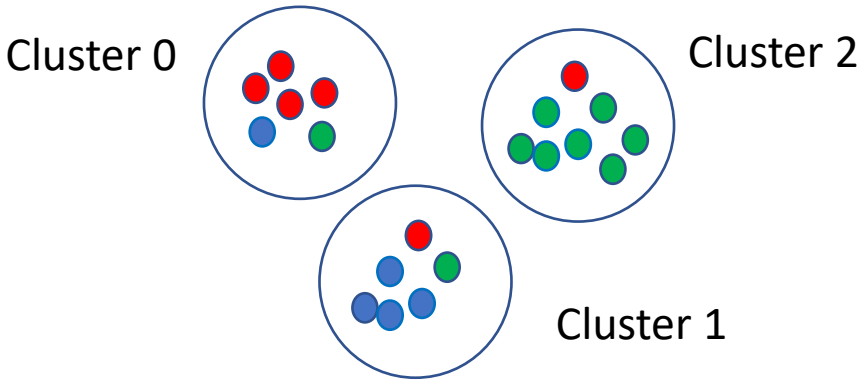
Section 10.1: Clustering Quality Measures

Clustering Quality Measures

- Previously looked at Davies-Bouldin and Silhouette Index measures. These are “internal evaluation methods” based solely on clustering results
- If some outside information is available (such as predetermined class labels), then can use alternative approaches to measure clustering quality
- In subsequent sections we will look at 3 case studies where class labels are available
  - Iris Flower Clustering
  - MNIST Digits Image Grouping
  - BBC Text Clustering
  - Class labels will be used for clustering quality assessment and not for actual clustering

Clustering Quality Measures

- Consider clustering example where each data point has specified class assignment:



How can we assess the clustering quality in this example?

- 3 actual classes (red, blue, green data points)
- 3 clusters are found, each with more than 1 class
- In perfect world clustering will identify clusters that have exactly 1 class
  - There should be red cluster, blue cluster, and green cluster

Purity Measure

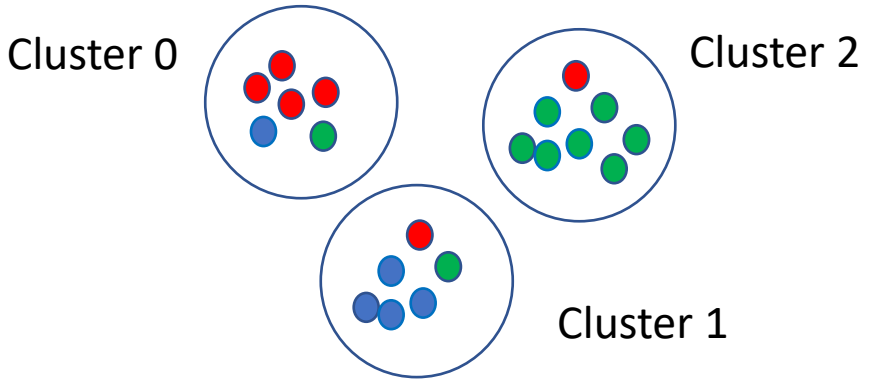
- Purity measures extent to which clusters contain a single class
- M is number of data points, C is set of clusters, D is set of classes
- For each cluster: determine maximum number of data points from any class
- Purity is sum of these maximums divided by total number of data points

$$P = \frac{1}{M}\sum_{c \in C} max_{d \in D}|d \cap c|$$

- Purity satisfies $0 < P \leq 1$ (P=1 for "perfect" clustering)

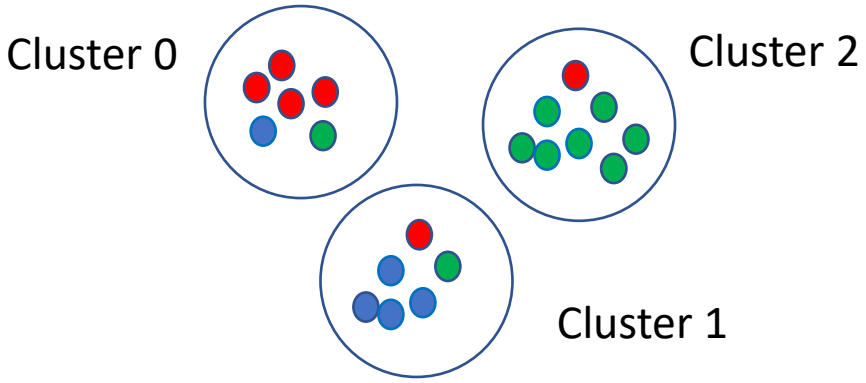
Example

- 20 data points and 3 clusters
- 3 actual classes: red, blue, green
- Max number from any class:
  - Cluster 0: 4 red, Cluster1: 4 blue, Cluster 2: 7 green
- $Purity = \frac{1}{20}(4 + 4 + 7) = 0.75$

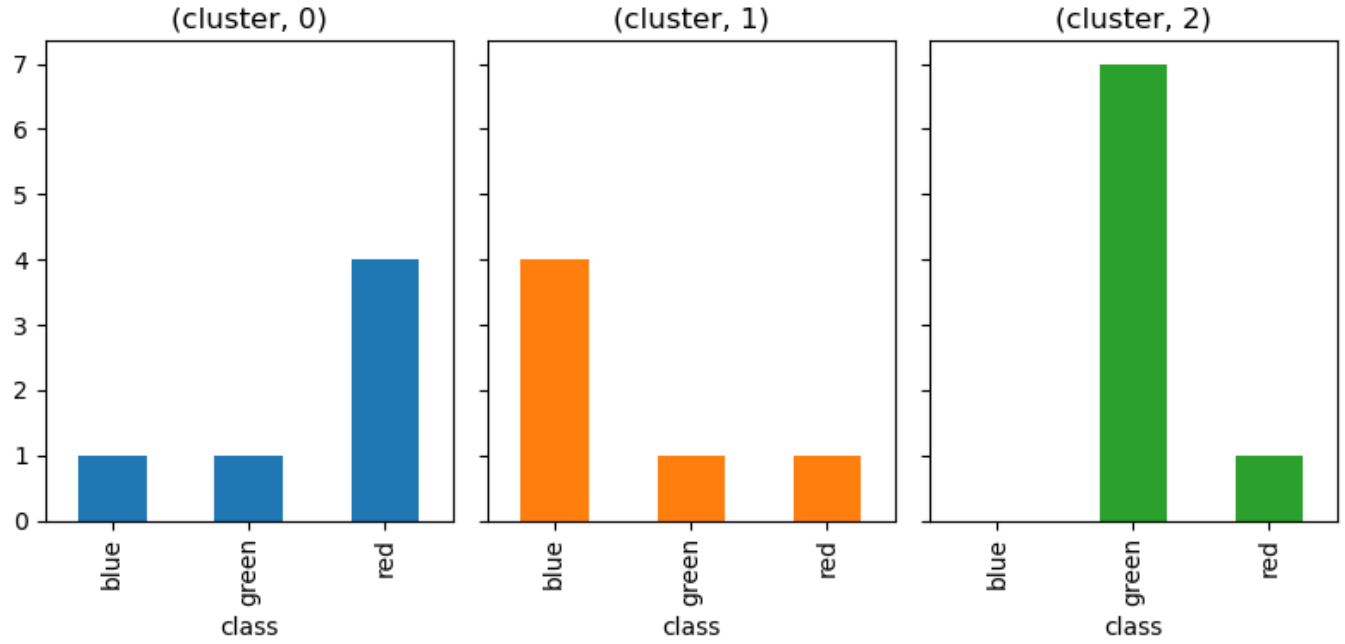


Bar Chart

- Given data set:



- Can also represent clustering results using bar chart



Clustering Quality Code Design

Function	Input	Description
purity	cluster_assignment (1d numpy array) class_assignment (1d numpy array)	Computes purity value given cluster and class assignments Return: purity See UnsupervisedML/Examples/Section10/ClusteringQuality.ipynb
plot_cluster_distribution	cluster_assignment (1d numpy array) class_assignment (1d numpy array) figsize (tuple) figrow (integer)	Creates bar charts given cluster and class assignments. figsize and figrow are used to configure the bar charts. Return: nothing See: UnsupervisedML/Examples/Section10/ClusteringQuality.ipynb

10.1 Clustering Quality DEMO

Jupyter Notebook located at:

- UnsupervisedML/Examples/Section10/ClusteringQuality.ipynb

Clustering Quality functions located at:

- UnsupervisedML/Code/Programs

Files to Review	Description
metrics.py	File containing purity and bar plot creation functions

Course Resources at:

- https://github.com/satishchandrareddy/UnsupervisedML/

- Stop video if you would like to implement code yourself first

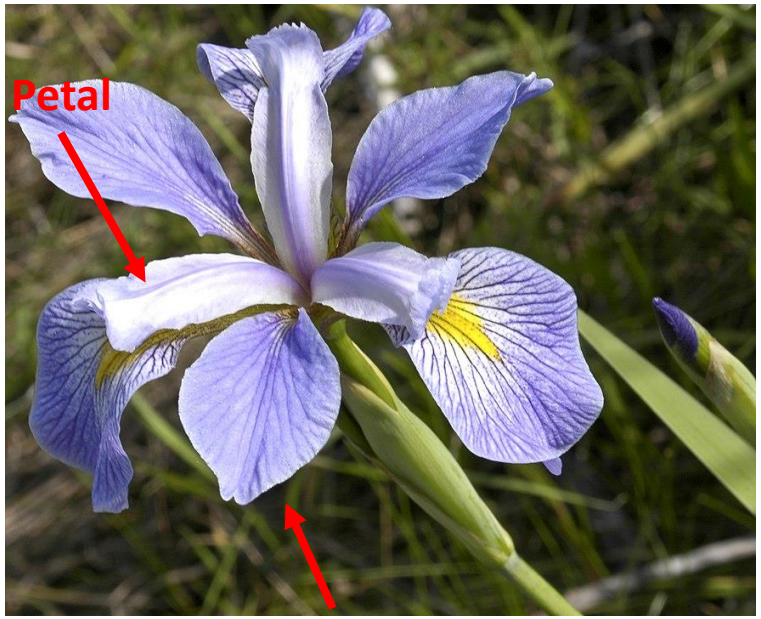
Unsupervised Machine Learning with Python

Section 10.2: Clustering for the Iris Flower Dataset

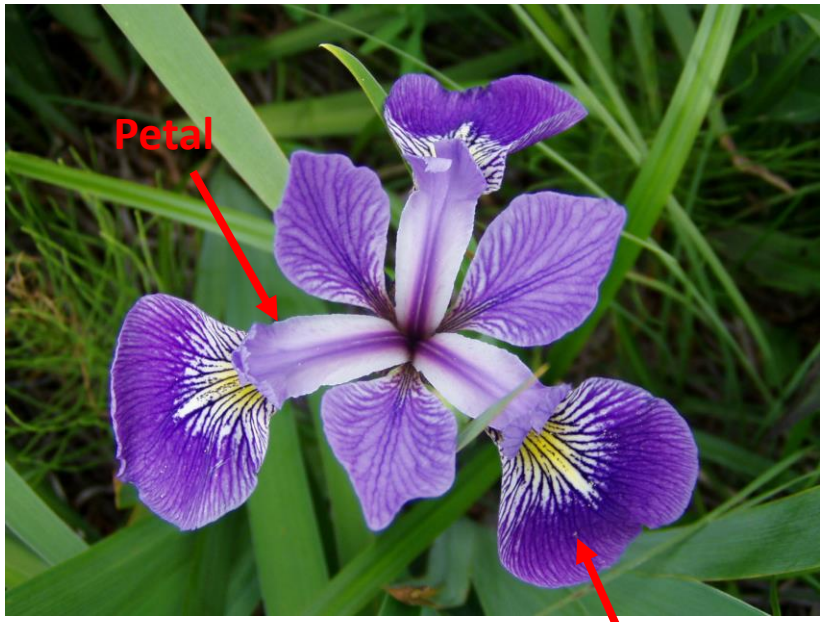
Iris Flower Dataset

Three types of iris flowers in dataset

Iris Virginica



Iris Versicolor

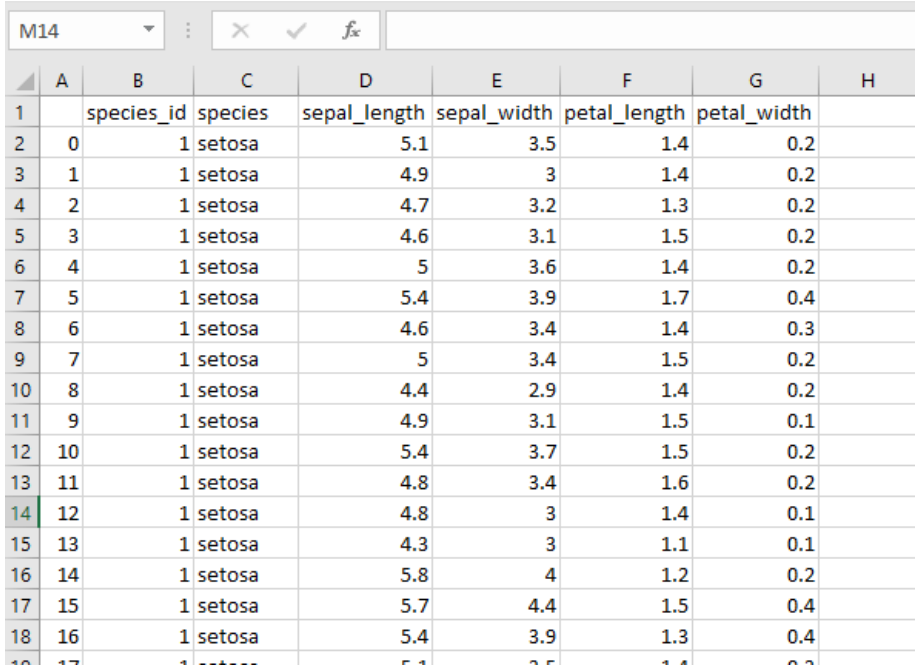


Iris Setosa

See UnsupervisedML_Resources.pdf file for links
Images reproduced here under Wikipedia Commons Copyright

Iris Dataset

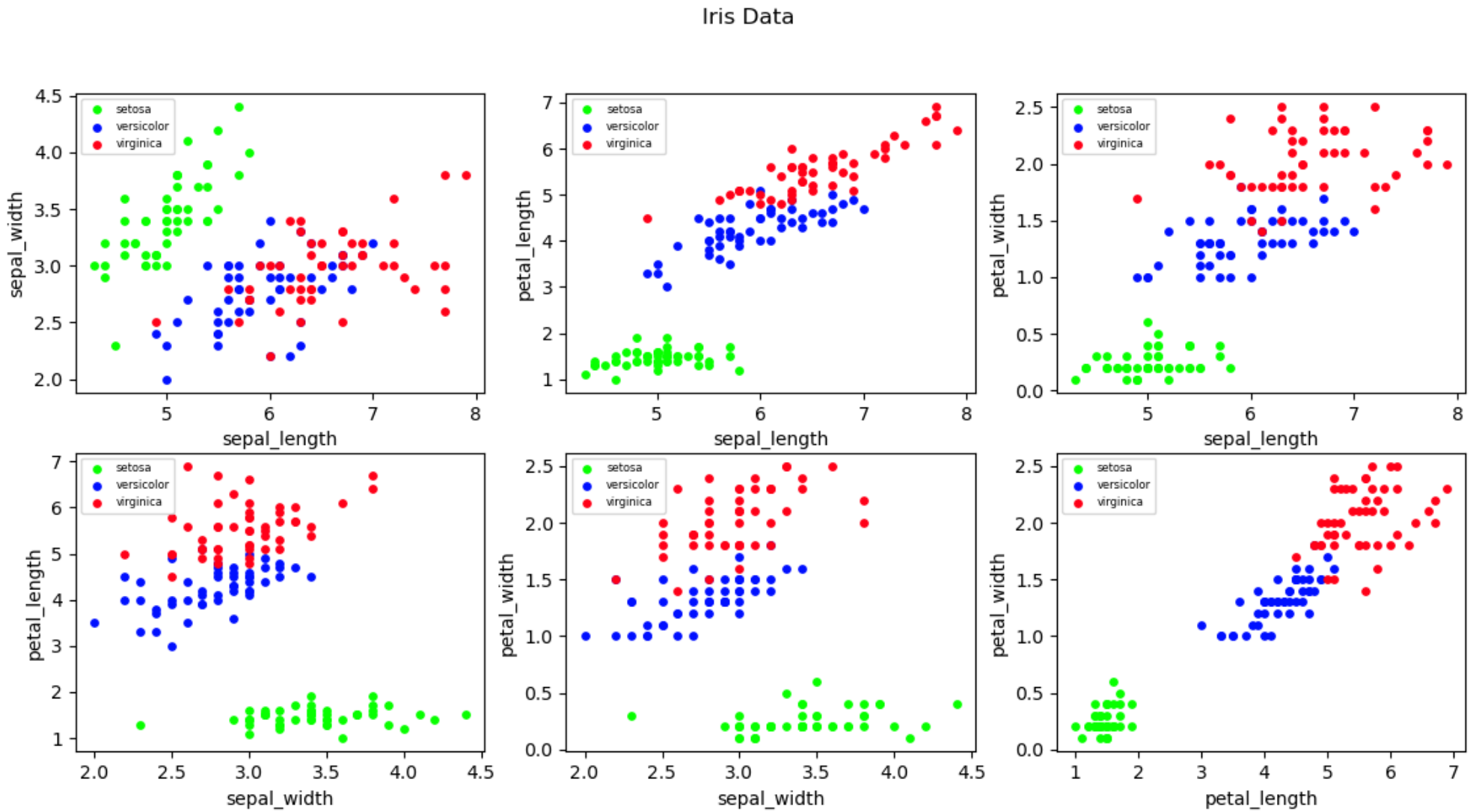
- 50 samples each of 3 types of iris flower species: virginica, versicolor, setosa
- 4 features: sepal_length, sepal_width, petal_length, petal_width
- Dataset available at UCI Machine Learning Repository https://archive.ics.uci.edu/ml/datasets/iris
- File: UnsupervisedML/Code/Data_Iris/iris.csv



M14

	A	B	C	D	E	F	G	H
1		species_id	species	sepal_length	sepal_width	petal_length	petal_width	
2	0	1	setosa	5.1	3.5	1.4	0.2	
3	1	1	setosa	4.9	3	1.4	0.2	
4	2	1	setosa	4.7	3.2	1.3	0.2	
5	3	1	setosa	4.6	3.1	1.5	0.2	
6	4	1	setosa	5	3.6	1.4	0.2	
7	5	1	setosa	5.4	3.9	1.7	0.4	
8	6	1	setosa	4.6	3.4	1.4	0.3	
9	7	1	setosa	5	3.4	1.5	0.2	
10	8	1	setosa	4.4	2.9	1.4	0.2	
11	9	1	setosa	4.9	3.1	1.5	0.1	
12	10	1	setosa	5.4	3.7	1.5	0.2	
13	11	1	setosa	4.8	3.4	1.6	0.2	
14	12	1	setosa	4.8	3	1.4	0.1	
15	13	1	setosa	4.3	3	1.1	0.1	
16	14	1	setosa	5.8	4	1.2	0.2	
17	15	1	setosa	5.7	4.4	1.5	0.4	
18	16	1	setosa	5.4	3.9	1.3	0.4	

Iris Dataset



Examples in this Section

Clustering problem:

- Employ clustering algorithm/PCA to group flowers in Iris Dataset
- How well do clustering algorithms perform?

Example 1

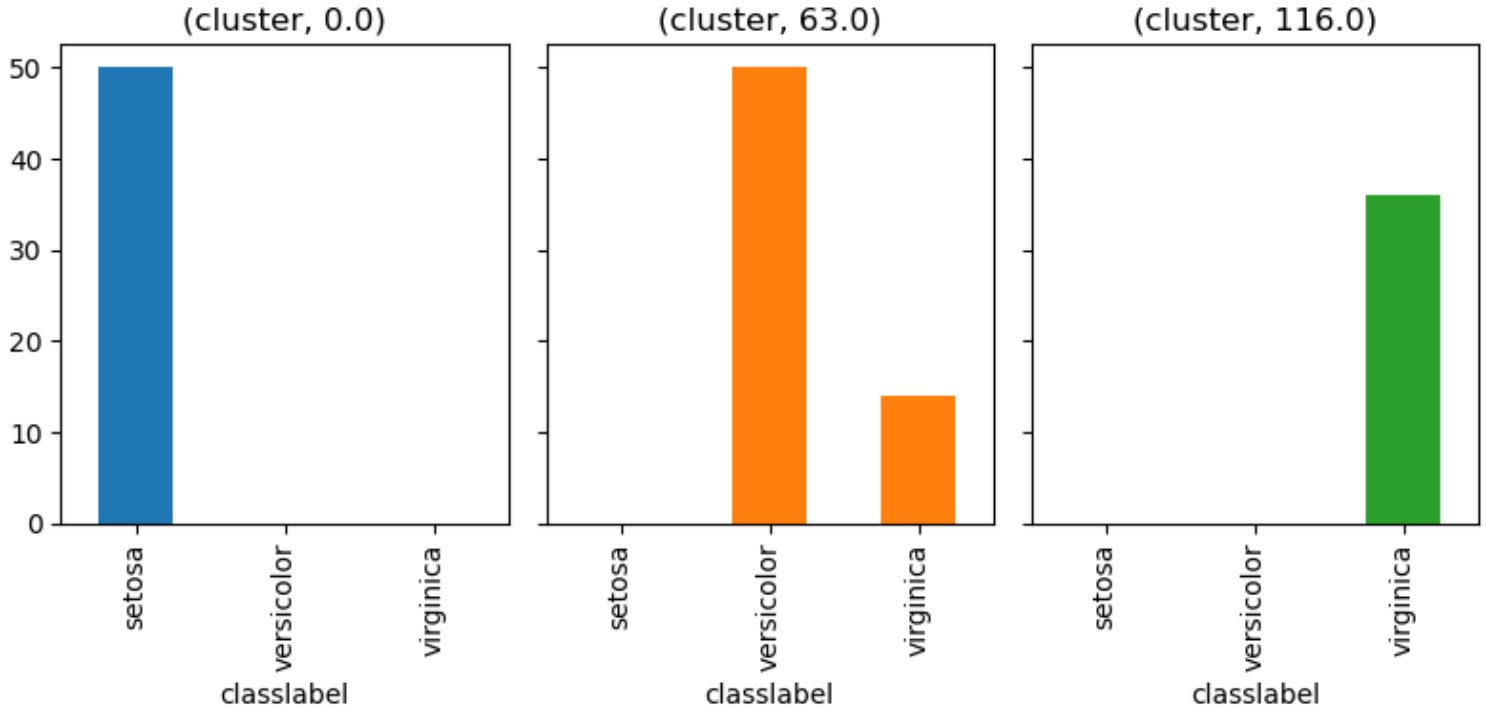
- Hierarchical Clustering for Iris dataset

Example 2:

- Hierarchical Clustering for Iris dataset after using PCA to reduce dataset to 2 dimensions

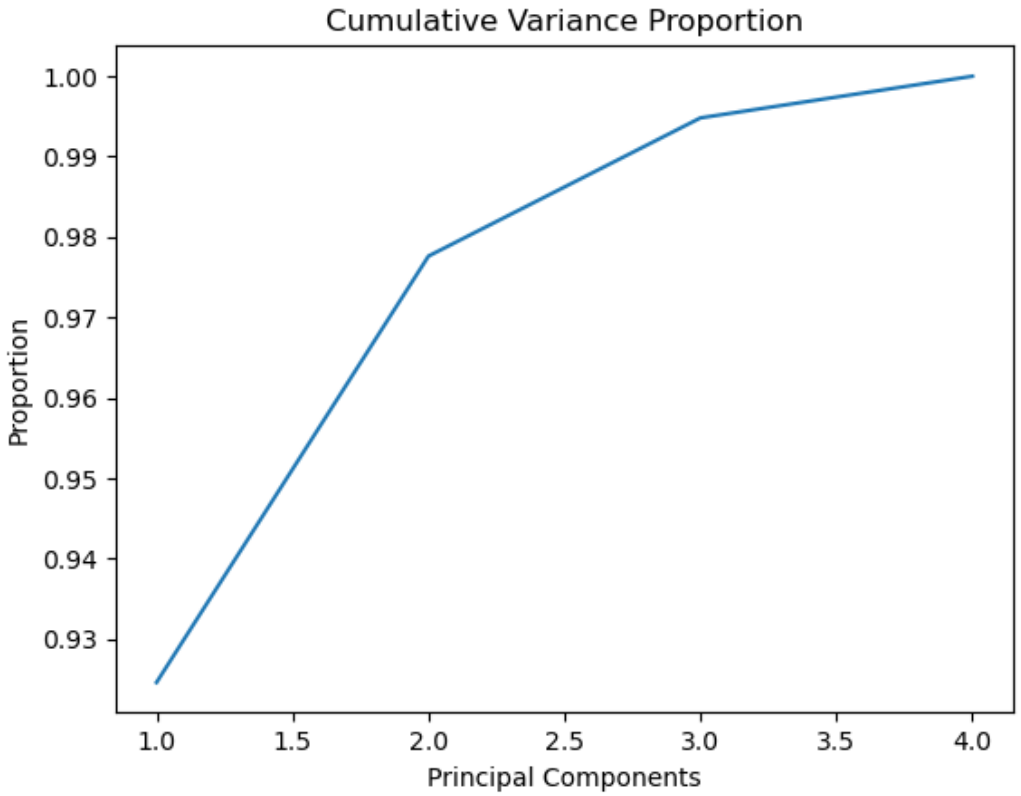
Example 1: Clustering of Iris Dataset

- Dataset: Feature matrix X (4 dimensions x 150 data points)
- Algorithm: Hierarchical Clustering (stop at 3 clusters)
- Metrics:
  - Purity: 0.907
  - Davies-Bouldin: 0.659
  - Silhouette: 0.554



PCA for Iris Dataset

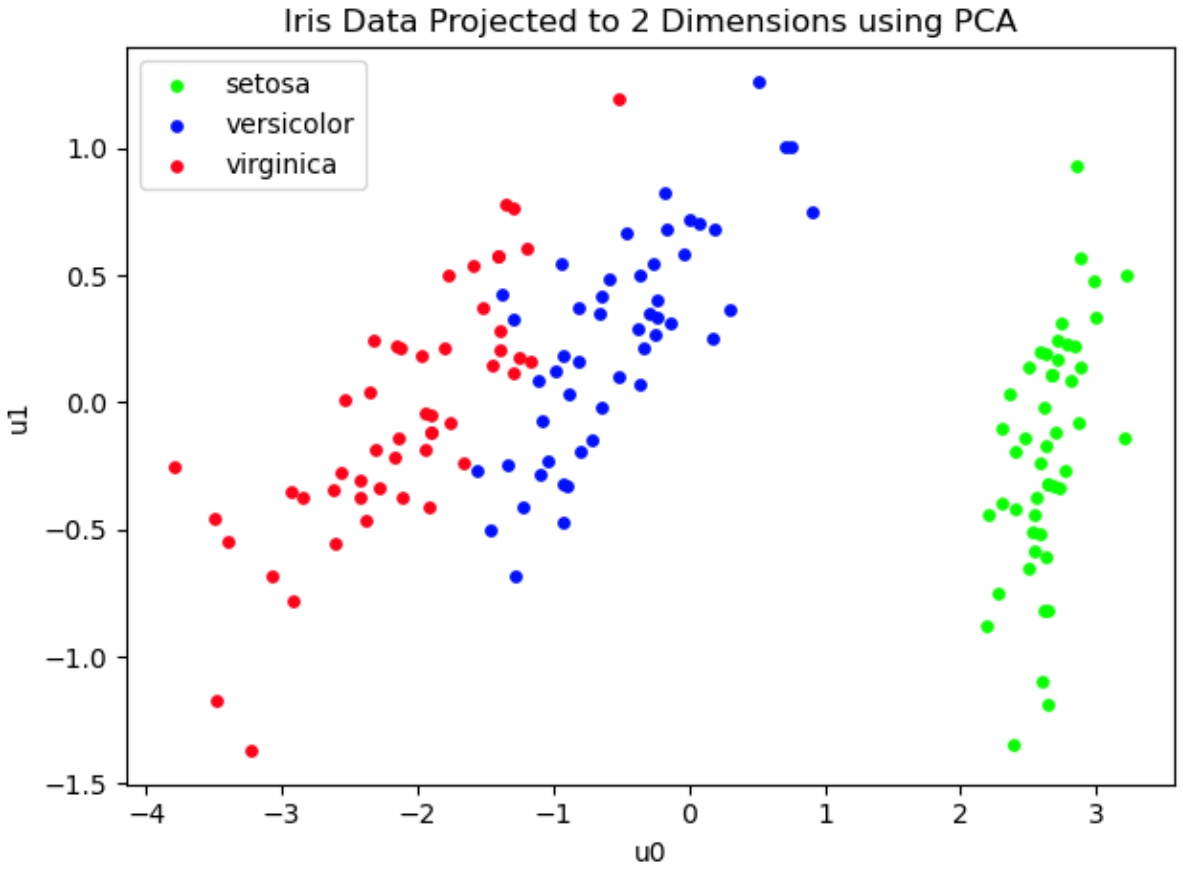
- Perform PCA for Iris Dataset
- Review Cumulative Variance Proportion



- 1 principal component captures nearly 93% of the variance
- 2 principal components capture nearly 98% of the variance

PCA for Iris Dataset

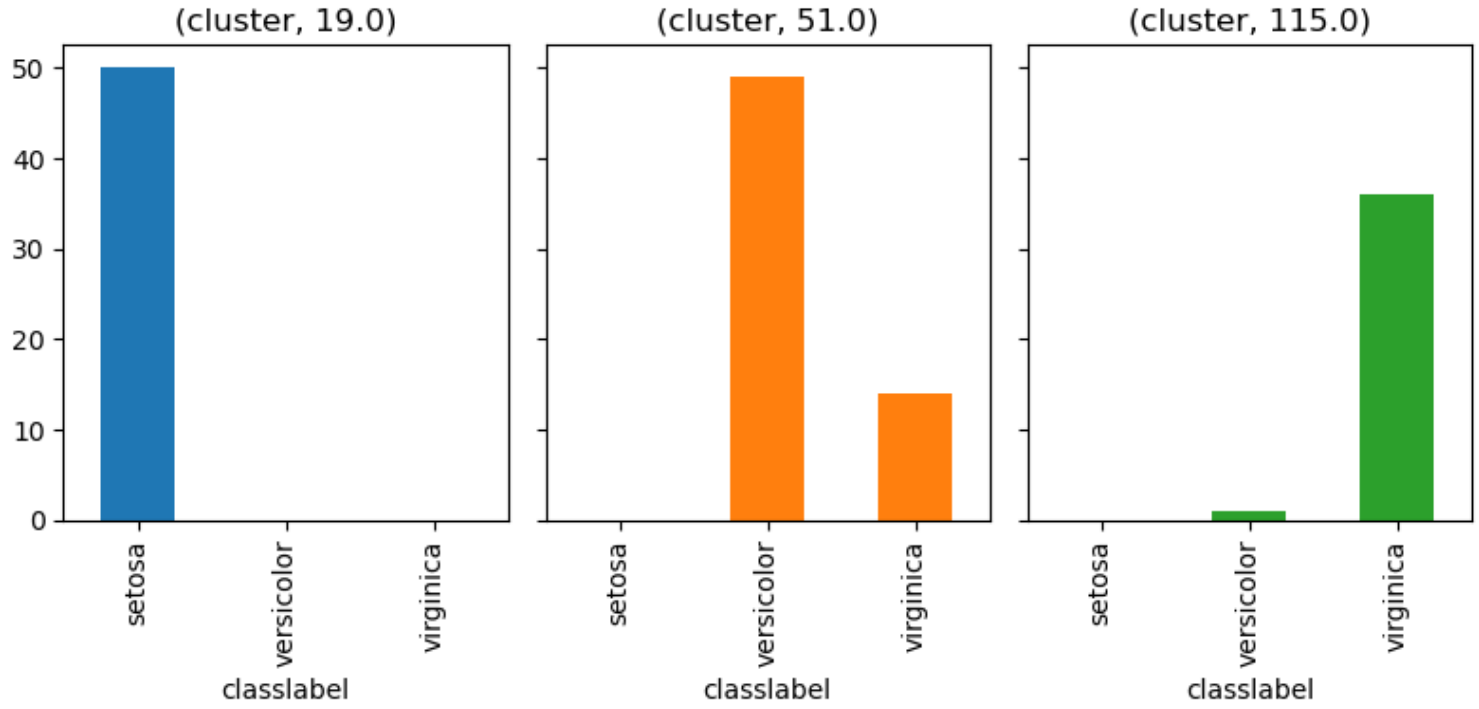
- Project data from 4 dimensions to 2 dimension using PCA
- Variance capture is 97.8%



- New basis vectors/features u0 and u1 do not correspond to actual measurable quantities, such as sepal width or length or petal width or length

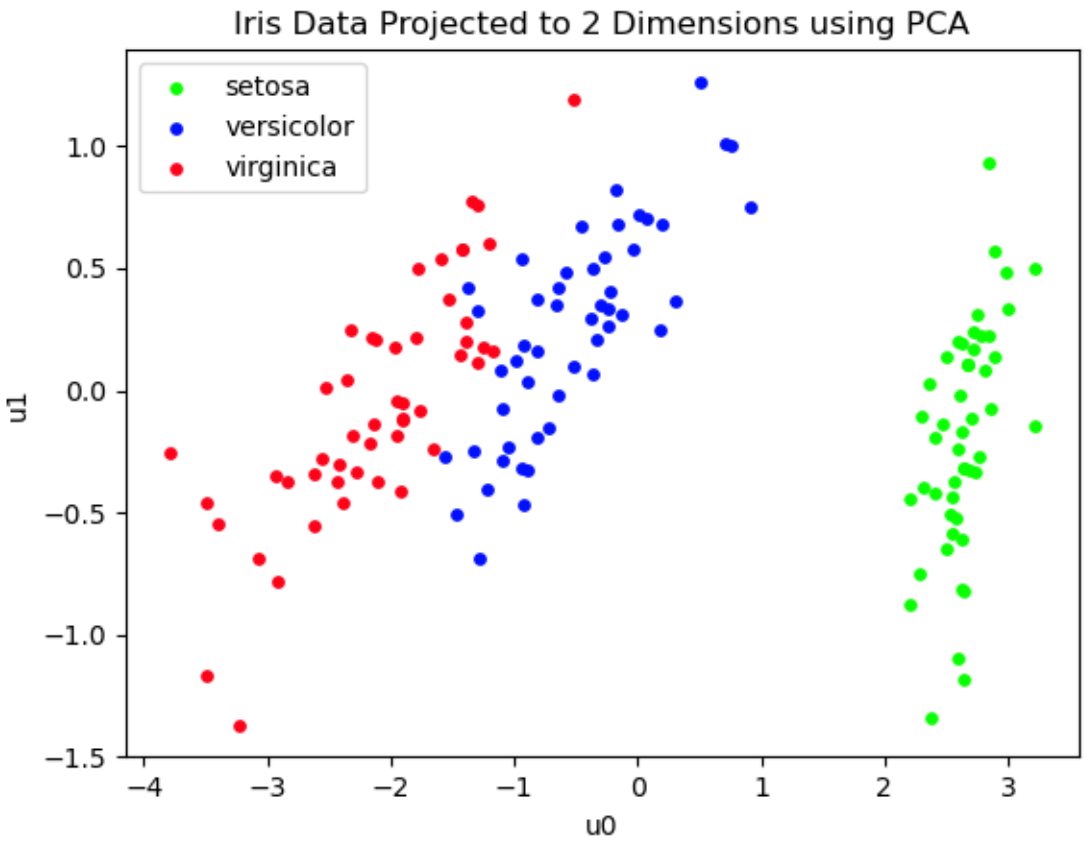
Example 2: Clustering for Iris Dataset using PCA

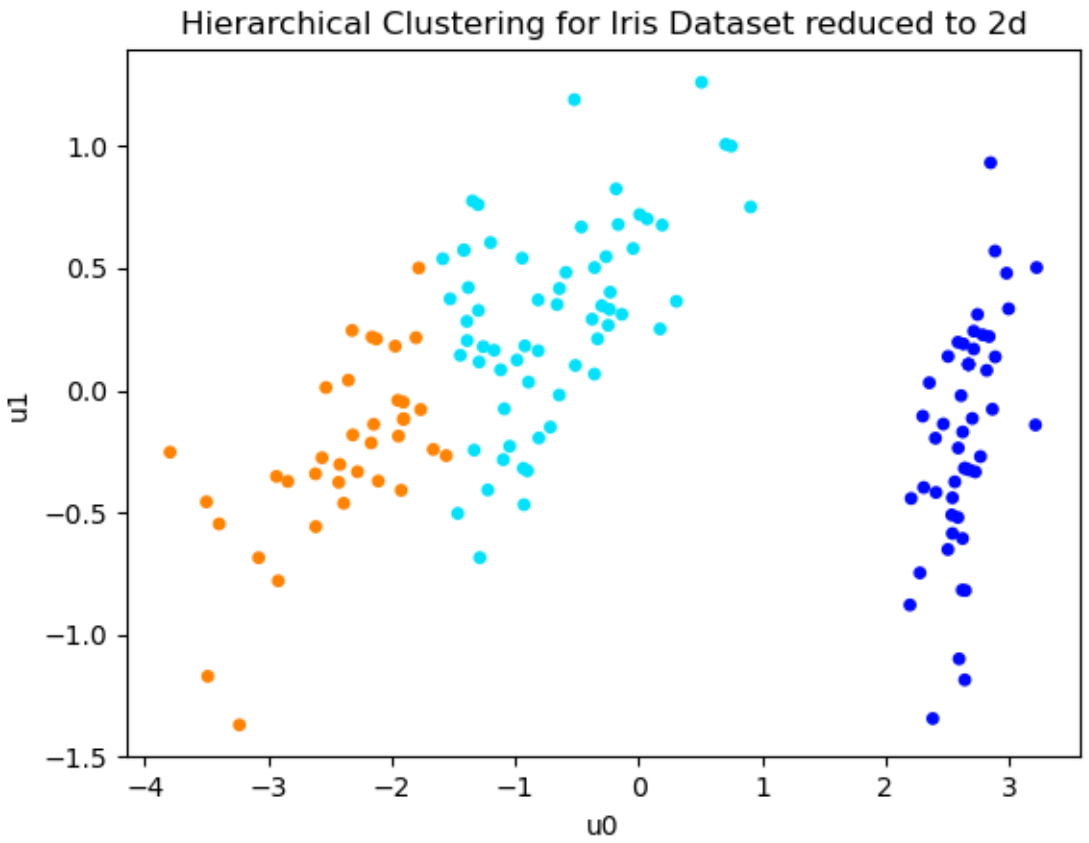
- Dataset: reduced dimension feature matrix R (2 x 150)
- Algorithm: Hierarchical Clustering (stop at 3 clusters)
- Metrics:
  - Purity: 0.900
  - Davies-Bouldin: 0.561
  - Silhouette: 0.598



Clustering for Iris Dataset using PCA

- Comparison of Class and Clustering Results





iris class Code Design

method	Input	Description
__init__		Constructor for iris class – saves directory Return: nothing
load		Loads all 150 samples and corresponding class labels from iris dataset Return: X (2d numpy array), class_label (1d numpy array) See UnsupervisedML/Examples/Section02/Pandas.ipynb
plot		Creates scatter plots showing classes as a function of all possible 2 variable combinations of sepal width, sepal length, petal width & petal length Return: nothing See UnsupervisedML/Examples/Section02/MatplotlibAdvanced.ipynb

Iris Clustering Code Walkthrough

Code and data located at:

- UnsupervisedML/Code/Programs
- UnsupervisedML/Code/Data_Iris

Files to Review	Description
Data_Iris/iris.csv	Iris dataset
Programs/data_iris.py	Class for loading and processing iris data
Programs/plot_data.py	Functions for creating basic scatter plots
Programs/casestudy_iris.py	Driver for hierarchical clustering
Programs/casestudy_iris_pca.py	Driver for hierarchical clustering using pca to reduce dimension

Course Resources at:

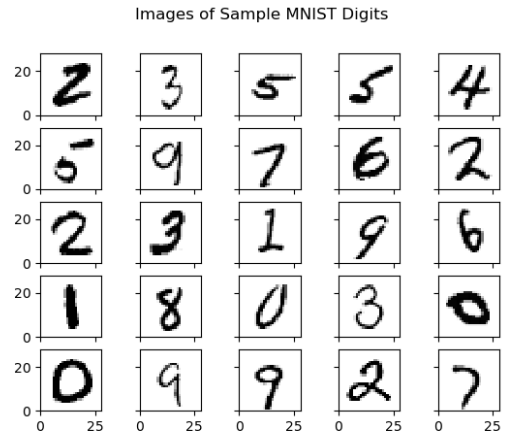
- https://github.com/satishchandrareddy/UnsupervisedML/

- Stop video if you would like to implement code yourself first

Unsupervised Machine Learning with Python

Section 10.3: Clustering for MNIST Digits Dataset

Examples in this Section



Goal:

- Employ clustering /PCA to group images in MNIST Dataset
- See how well clustering algorithms create clusters with the same digits

Example 1:

- K Means Clustering

Example 2:

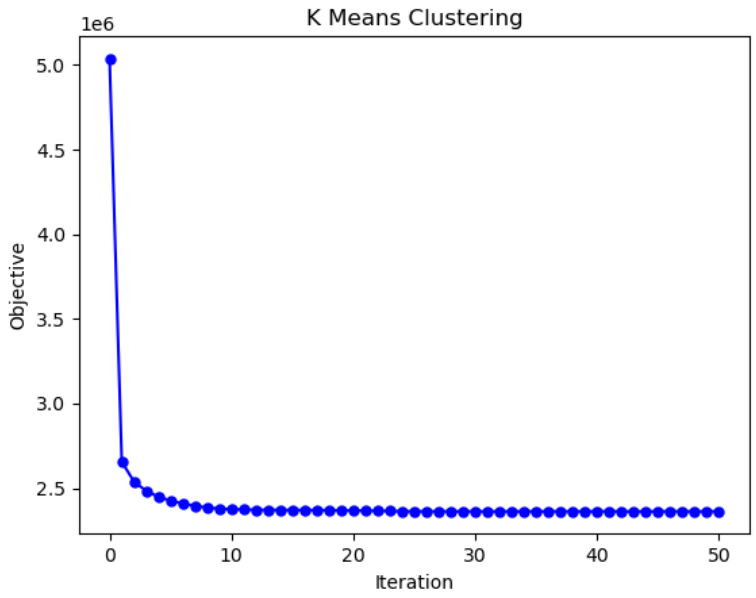
- K Means Clustering after using PCA to capture 90% of variance

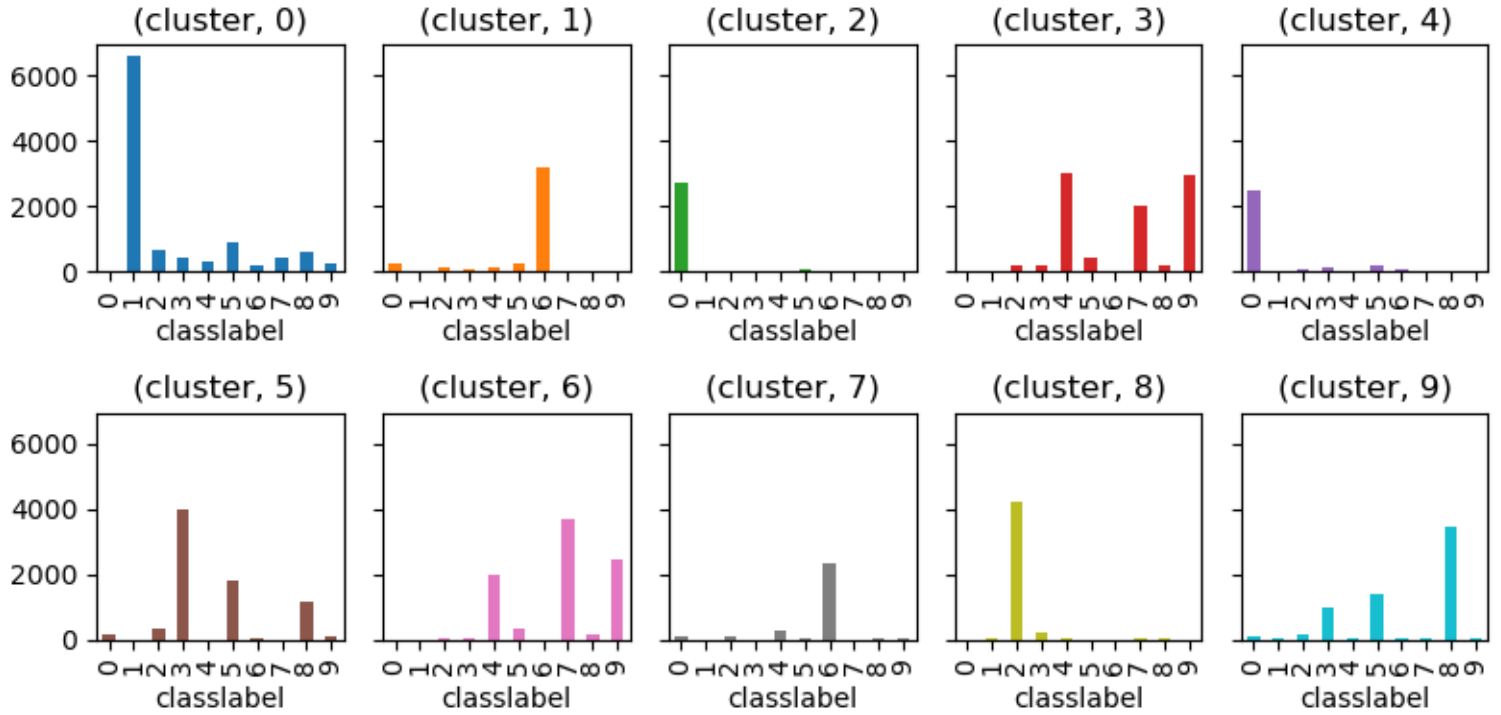
Example 3:

- Gaussian MM Clustering after using PCA to capture 90% of variance

Example 1: K Means Clustering

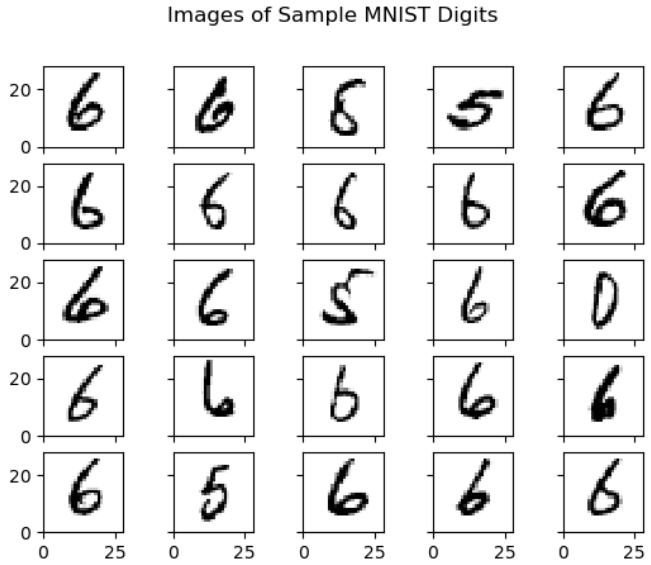
- Dataset: Feature matrix X (784 dimensions x 60000 images)
- Algorithm: K Means with 10 clusters, kmeans++ for initialization, 100 iterations maximum, tolerance of 10^{-4}
- Metrics:
  - Purity: 0.596
  - Davies-Bouldin: 2.82
  - Clustering Time: 196 seconds



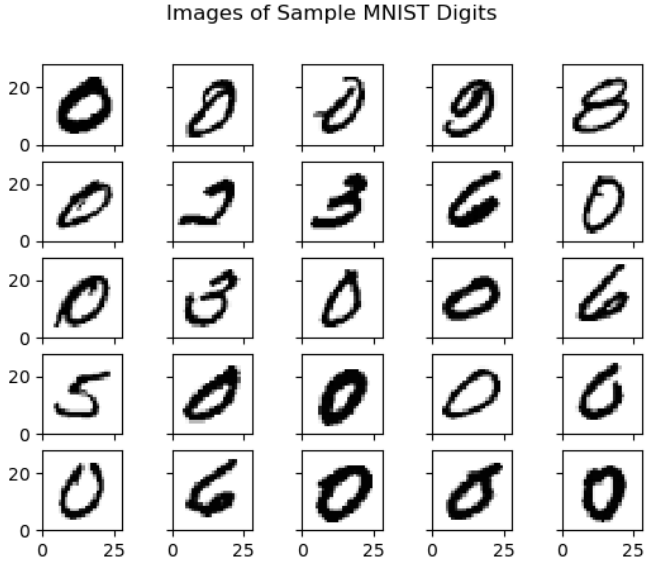


Example 1: K Means Clustering

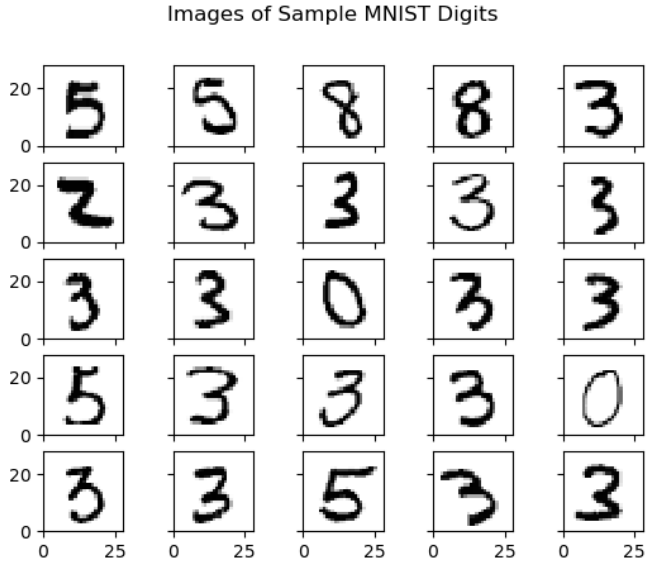
Cluster 1



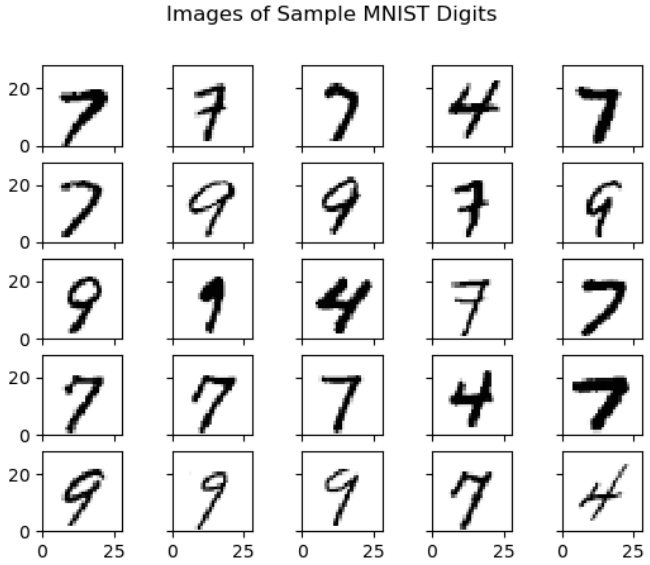
Cluster 4



Cluster 5

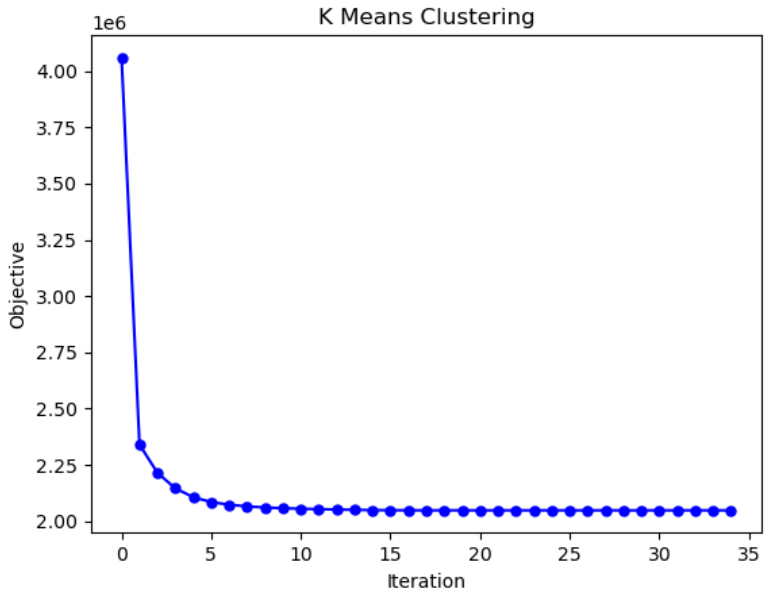


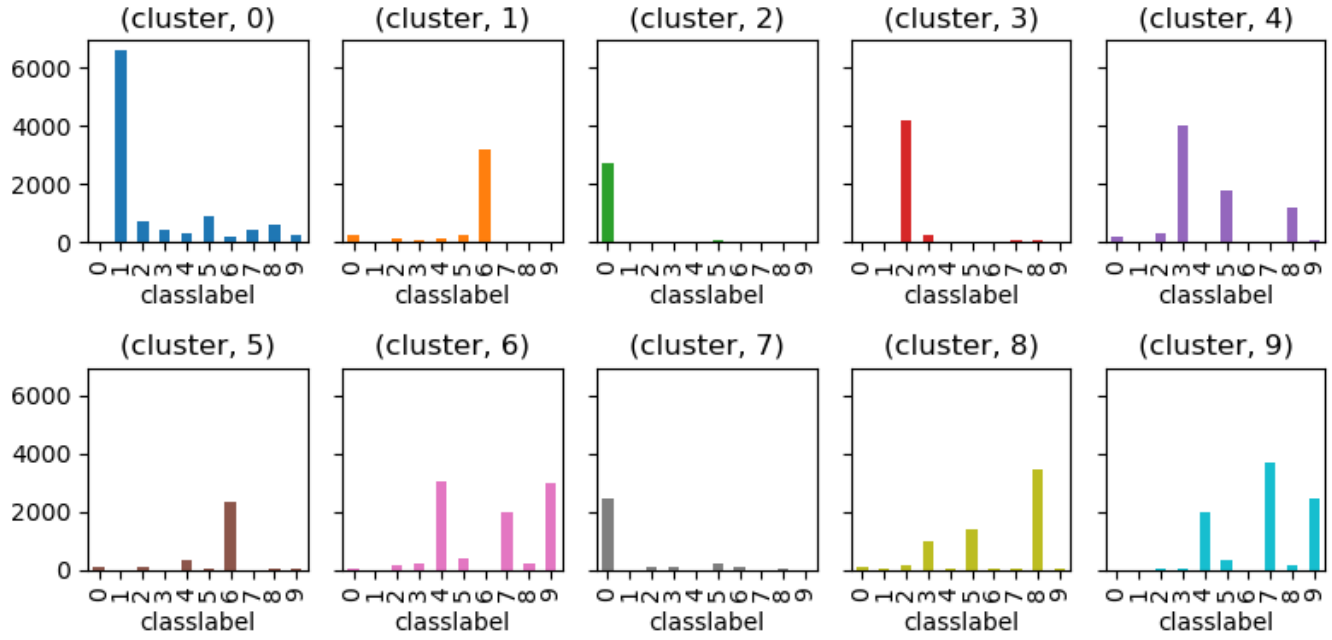
Cluster 6



Example 2: K Means Clustering with PCA

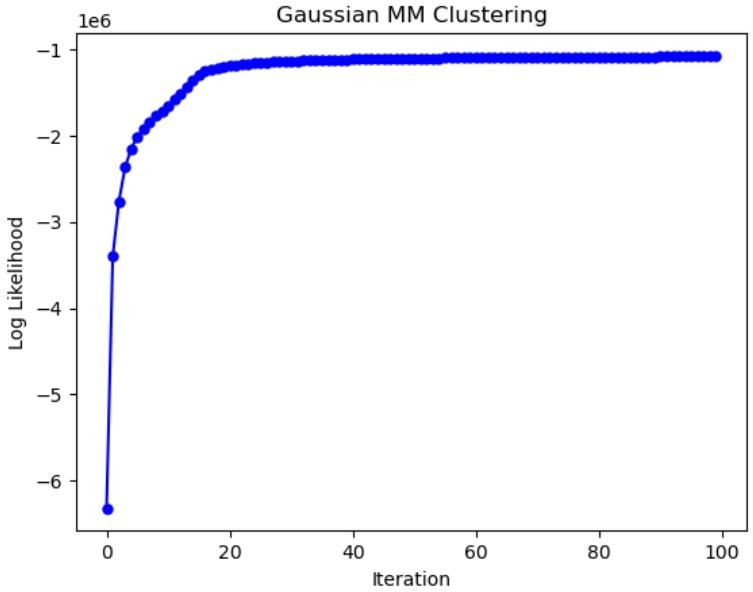
- Dataset: apply PCA with 90% variance capture to 60000 images resulting in feature matrix R (87 dimensions x 60000 images)
- Algorithm: K Means with 10 clusters, kmeans++ for initialization, 100 iterations maximum, 10^{-4} tolerance
- Metrics:
  - Purity: 0.596
  - Davies-Bouldin: 2.62
  - PCA Time: 7.6 seconds + Clustering Time: 16.2 seconds

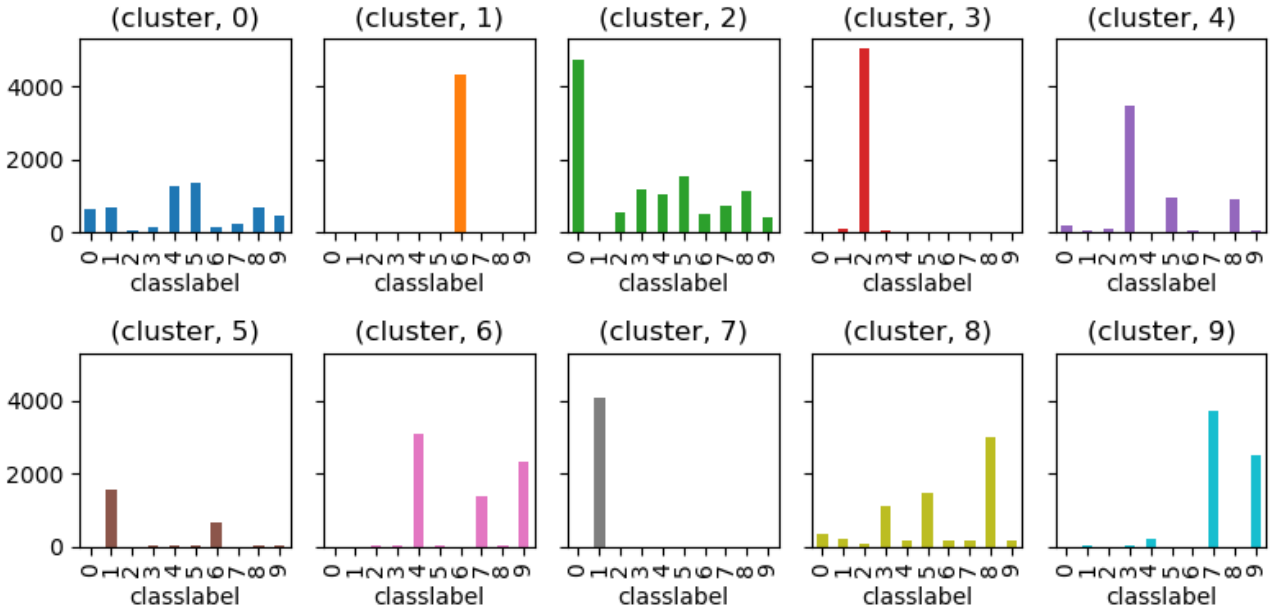




Example 3: GaussianMM Clustering with PCA

- Dataset: apply PCA with 90% variance capture to 60000 images resulting in feature matrix R (87 dimensions x 60000 images)
- Algorithm: GaussianMM with 10 clusters, kmeans++ for initialization, 100 iterations maximum, 10^{-4} tolerance
- Metrics:
  - Purity: 0.574
  - Davies-Bouldin: 2.99
  - PCA Time: 6.6 seconds + Cluster Time: 172 seconds





Comments

- K Means Clustering
  - Achieved 59.6% Purity result for clustering for 60000 image MNIST dataset with and without PCA
  - Davies-Bouldin values of 2.82 (without PCA) and 2.62 (with PCA)
  - Using PCA can significantly reduce total processing time taking into account time for PCA
  - Many more iterations are required for convergence using “random” compared to “kmeans++” initialization
- Gaussian MM Clustering
  - Don’t get convergence after 100 iterations even after applying PCA to reduce dimensions
  - Approach is slow for large numbers of dimensions
  - I am finding that method is not stable for other values of variance capture – numerical issues because determinant of covariance matrix is close to 0
    - Production level codes will have regularization functionality to deal with this situation
  - In exercises you will investigate using spherical Gaussian MM approach

MNIST Clustering Code Walkthrough

Code and data located at:

- UnsupervisedML/Code/Programs
- UnsupervisedML/Code/Data_MNIST

Files	Description
Data_MNIST/MNIST_train_set1_30K.csv Data_MNIST/MNIST_train_set2_30K.csv Data_MNIST/MNIST_valid_10K.csv	MNIST train and valid datasets
Programs/data_mnist.py	Class for loading and plotting mnist digits dataset
Programs/casestudy_mnist.py	Driver for MNIST clustering using K Means with PCA

Course Resources at:

- https://github.com/satishchandrareddy/UnsupervisedML/

- Stop video if you would like to implement code yourself first

Unsupervised Machine Learning with Python

Section 10.4: Clustering for Text Documents

BBC Text Dataset

- 2225 BBC articles in 5 categories: sport, business, tech, entertainment, politics
- Dataset: https://www.kaggle.com/yufengdev/bbc-fulltext-and-category
- File: UnsupervisedML/Code/Data_BBCText/bbc-text.csv
- Use Tfidf vectorizer in sklearn to convert text to feature matrix
- 12915 words in dictionary -> 12915 x 2225 feature matrix

	A	B	C	D	E	F	G	H
1	category	text						
2	tech	tv future in the hands of viewers with home theatre systems plasma hi						
3	business	worldcom boss left books alone former worldcom boss bernie ebbers						
4	sport	tigers wary of farrell gamble leicester say they will not be rushed into						
5	sport	yeading face newcastle in fa cup premiership side newcastle united fac						
6	entertainment	ocean s twelve raids box office ocean s twelve the crime caper sequel						
7	politics	howard hits back at mongrel jibe michael howard has said a claim by pe						
8	politics	blair prepares to name poll date tony blair is likely to name 5 may as ele						
9	sport	henman hopes ended in dubai third seed tim henman slumped to a stra						
10	sport	wilkinson fit to face edinburgh england captain jonny wilkinson will ma						
11	entertainment	last star wars not for children the sixth and final star wars movie may n						
12	entertainment	berlin cheers for anti-nazi film a german movie about an anti-nazi resist						
13	business	virgin blue shares plummet 20% shares in australian budget airline virgi						
14	business	crude oil prices back above $50 cold weather across parts of the united						

Examples in this Section

Clustering problem:

- Employ clustering algorithm/PCA to group articles in BBCText dataset
- How well can algorithm create clusters of articles in the same category?

Example 1:

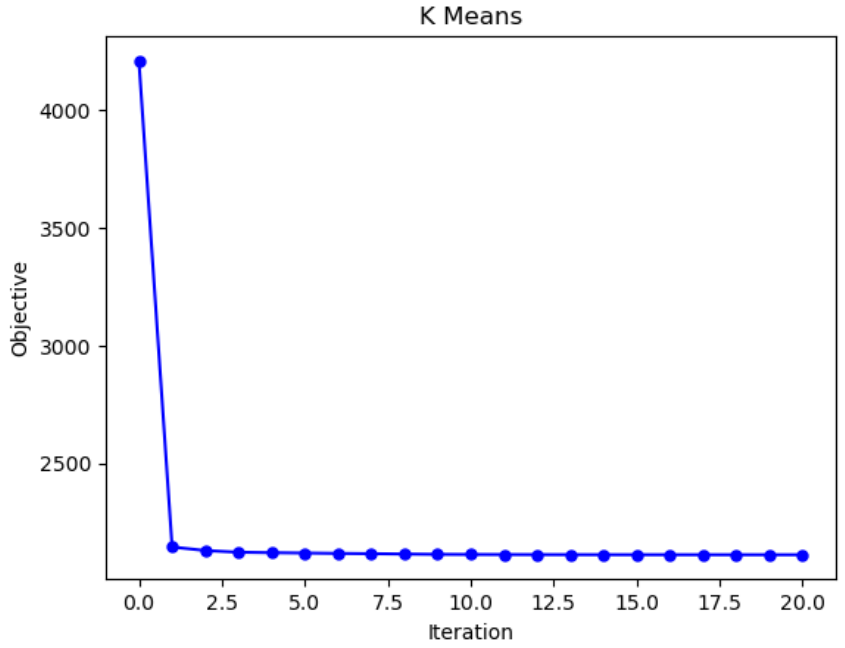
- K Means Clustering for BBCText dataset

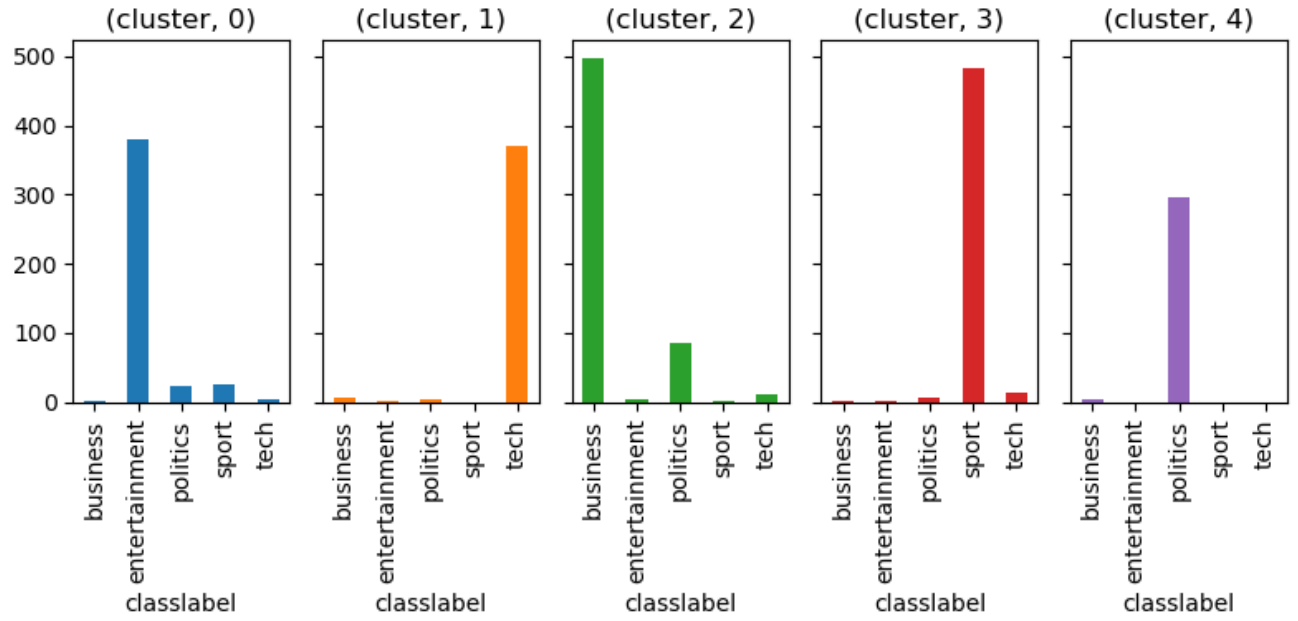
Example 2:

- K Means Clustering for BBCText dataset after using PCA to reduce dimensions and still capture 100% of variance

Example 1: K Means Clustering

- Dataset: Feature matrix X (12915 dimensions x 2225 data points)
- Algorithm: K Means with 5 clusters, “random” initialization, 50 iterations maximum, tolerance of 10^{-4}
- Metrics:
  - Purity: 0.912
  - Davies-Bouldin: 8.25
  - Clustering Time: 23.5 seconds





Example 1: Wordclouds for Clusters

Cluster 0

Cluster 1

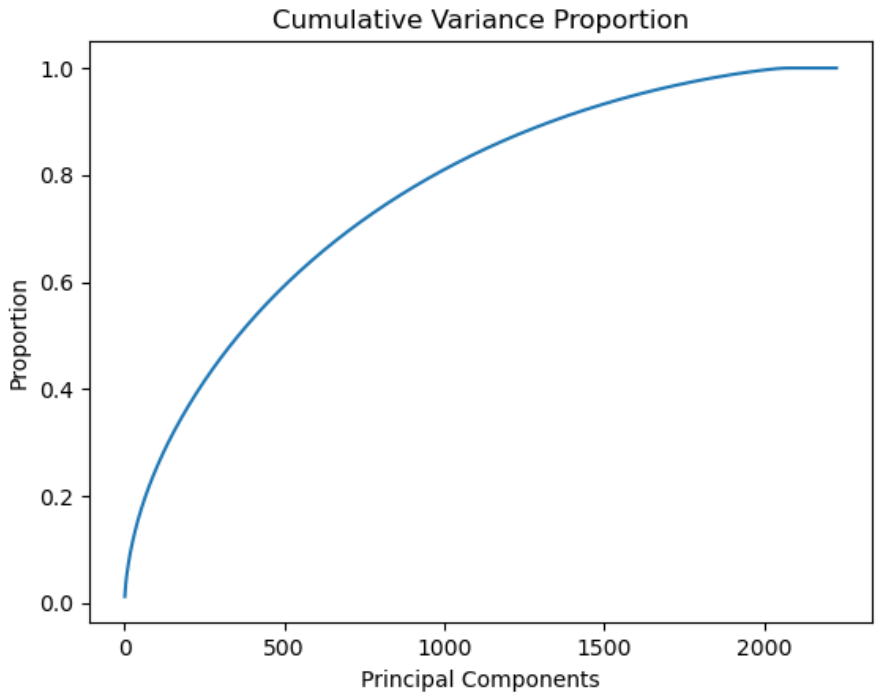
Cluster 2

Cluster 3

Cluster 4

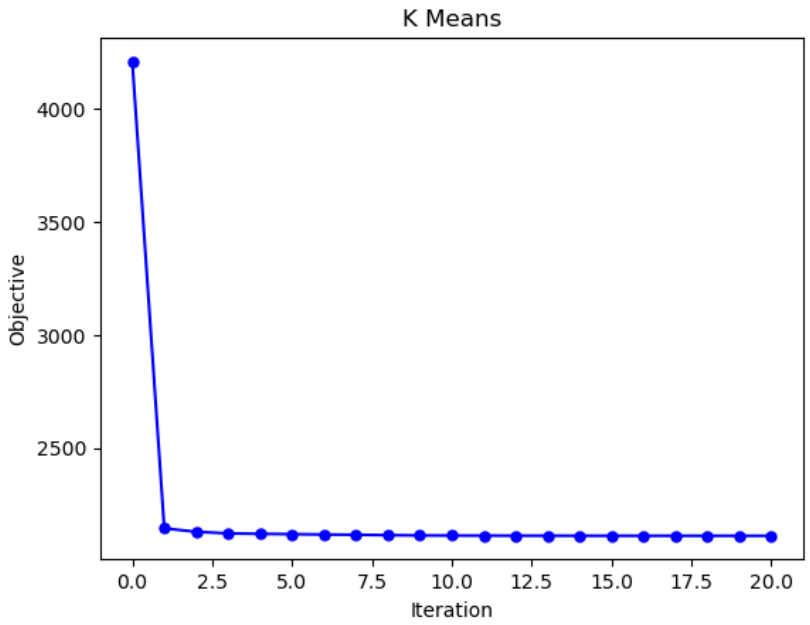
PCA for BBC Text Dataset

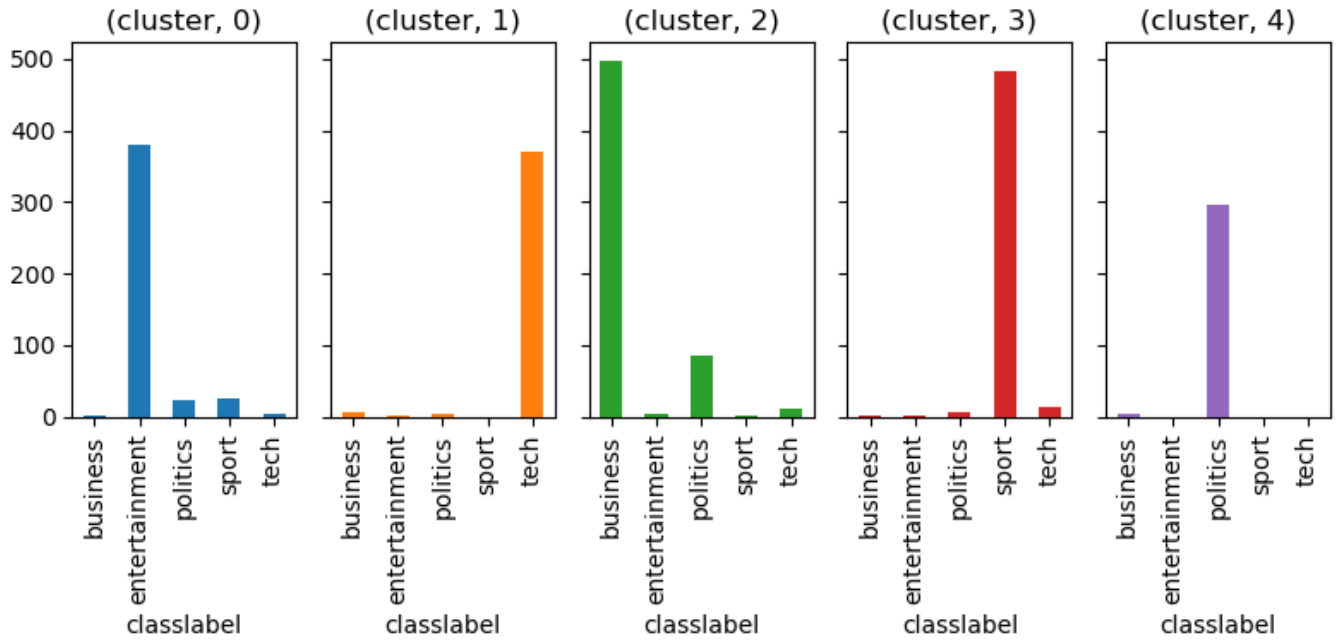
- Perform PCA for BBC Text Dataset
- Since number of dimensions (12915) > number of data points (2225), can reduce to 2225 dimensions and still retain 100% of variance
- Actually, since some singular values are 0, can retain 100% of variance with 2116 principal components



Example 2: K Means Clustering with PCA

- Dataset: use PCA to reduce dimension and still capture 100% of variance – results in feature matrix R (2116 dimensions x 2225 data points)
- Algorithm: K Means with 5 clusters, random initialization, 50 iterations maximum, tolerance of 10^{-4}
- Metrics: (clusters are exactly the same as in Example 1)
  - Purity: 0.912
  - Davies-Bouldin: 8.25
  - PCA Time: 12.9 seconds + Clustering Time: 4.5 seconds





Comments

- K Means Clustering algorithm is able to achieve greater than 91% purity measure for grouping articles in BBC Text dataset
- Can use PCA to reduce dimension and maintain 100% variance capture
  - Dimension reduced from 12915 to 2116
  - Clustering results exactly the same with and without dimension reduction
  - Combined PCA + Clustering Time < Clustering Time in no PCA Case
  - Clustering time is more than 5 times lower when PCA used

bbctext class Code Design

method	Input	Description
__init__		Constructor for bbctext class – saves directory and TFIDF vectorizer Return: nothing
load	nsample (integer)	Loads bbc text dataset for specified number of samples and applies TFIDF vectorization to create feature matrix Return: X (2d numpy array), class_label (1d numpy array) -UnsupervisedML/Examples/Section02/Pandas.ipynb -UnsupervisedML/Examples/Section03/SklearnText.ipynb
create_ wordcloud	X_tfidf (2d numpy array) cluster_assignment (1d numpy array) nword (integer)	Creates wordcloud plots for specified X_tfidf matrix, cluster assignments, and number of words Return: nothing -UnsupervisedML/Examples/Section03/SklearnText.ipynb

Text Clustering Code Walkthrough

Code and data located at:

- UnsupervisedML/Code/Programs
- UnsupervisedML/Code/Data_BBCText

Files to Review	Description
bbc-text.csv	BBC text data file
data_bbctext.py	Class for loading and processing BBC text data
casestudy_bbctext.py	Driver for bbc text clustering

Course Resources at:

- https://github.com/satishchandrareddy/UnsupervisedML/

- Stop video if you would like to implement code yourself first