**Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same datasetforclusteringusingk-Meansalgorithm.Comparetheresultsofthesetwo algorithmsandcommentonthequalityofclustering.YoucanaddJava/PythonML libraryclasses/APIintheprogram**

**K-Means Algorithm**

1. Load data set
2. Clusters the data into k groups where k is predefined.
3. Select k points at random as cluster centers.
4. Assign objects to their closest cluster center according to the Euclidean distance function.
5. Calculate the centroid or mean of all objects in each cluster.
6. Repeat steps 3, 4 and 5 until the same points are assigned to each cluster in consecutive rounds.

**EM algorithm**

These are the two basic steps of the EM algorithm, namely E Step or Expectation Step or Estimation Step and M Step or Maximization Step.

Estimation step:

* Initialize µk, ∑k and πk by some random values, or by K means clustering results or by hierarchical clustering results.
* Then for those given parameter values, estimate the value of the latent variables (i.eγk)
* Maximization Step:
* Update the value of the parameters( i.e. µk, ∑k and πk) calculated using ML method.

1. Select the number of seeds. Let this number be k
2. Pick k seeds as centroids of the k clusters. The seeds may be randomly picked.
3. Compute Euclidean/Manhattan distance of each object in the dataset from each of the centroids.
4. Allocate each object to the cluster it is nearest to based on the distances computed in the previous step.
5. Compute the centroids of the clusters by computing the means of the attribute values of the objects in each cluster.
6. Check if the stopping criterion has been met. (ex: cluster membership is unchanged). If so go to step 7. if not got to step 3
7. [optional] one may decide to stop at this stage or to split a cluster or combine two clusters heuristically until a stopping criterion is met.

**Iris Species**

Classify iris plants into three species in this classic dataset

It includes three iris species with 50 samples eachwhere each class refers to a type of iris plant.**Attribute Information:**

1. sepal length in cm  
2. sepal width in cm  
3. petal length in cm  
4. petal width in cm  
5. class:  
-- Iris Setosa  
-- Iris Versicolour  
-- Iris Virginica

**scikit-learn**

Machine Learning in Python

* Simple and efficient tools for predictive data analysis
* Accessible to everybody, and reusable in various contexts
* Built on NumPy, SciPy, and matplotlib
* Open source, commercially usable - BSD license

**What is model fit in Sklearn?**

* Plenty of models have fit methods in scikit-learn. When you call fit method it **estimates the best representative function for the data points** (could be a line, polynomial or discrete borders around). With that representation, you can calculate new data points.

**Models\_labels**

**labels\_*ndarray of shape (n\_samples,)***

Labels of each point

**Gaussian Mixture.**

Representation of a Gaussian mixture model probability distribution. This class allows to estimate the parameters of a Gaussian mixture distribution.

*class*sklearn.**preprocessing.StandardScaler**(*\**, *copy=True*, *with\_mean=True*, *with\_std=True*)[[source]](https://github.com/scikit-learn/scikit-learn/blob/0d378913b/sklearn/preprocessing/_data.py#L632)

Standardize features by removing the mean and scaling to unit variance.

The standard score of a sample x is calculated as:

z = (x - u) / s

where u is the mean of the training samples or zero if with\_mean=False, and s is the standard deviation of the training samples or one if with\_std=False.

**transform**(*X*, *copy=None*)[[source]](https://github.com/scikit-learn/scikit-learn/blob/0d378913b/sklearn/preprocessing/_data.py#L955)

Perform standardization by centering and scaling.

**Parameters**

**X*{array-like, sparse matrix of shape (n\_samples, n\_features)***

The data used to scale along the features axis.

**copy*bool, default=None***

Copy the input X or not.

**Returns**

**X\_tr*{ndarray, sparse matrix} of shape (n\_samples, n\_features)***

Transformed array.

**Homogeneity** metric of a cluster labeling given a ground truth.

A clustering result satisfies homogeneity if all of its clusters contain only data points which are members of a single class.

**Completeness metric** of a cluster labeling given a ground truth.

A clustering result satisfies completeness if all the data points that are members of a given class are elements of the same cluster.

**V-measure** cluster labeling given a ground truth.

This score is identical to **[normalized\_mutual\_info\_score](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.normalized_mutual_info_score.html" \l "sklearn.metrics.normalized_mutual_info_score" \o "sklearn.metrics.normalized_mutual_info_score)** with the 'arithmetic' option for averaging.

The V-measure is the harmonic mean between homogeneity and completeness:

v=(1+beta)\*homogeneity\*completeness

/(beta\*homogeneity+completeness)

**Rand index adjusted for chance.**

The Rand Index computes a similarity measure between two clusterings by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusterings.

The raw RI score is then “adjusted for chance” into the ARI score using the following scheme:

ARI=(RI-Expected\_RI)/(max(RI)-Expected\_RI)



