FINAL PROJECT REPORT

NAME : RADHIKA Y KULKARNI

EMAIL: rykulkar@iu.edu

SU17-BL-INFO-I590-13949

Title : Implementation of "k- means" algorithm for Wisconsin Breast Cancer data using Python

Abstract:

The project is designed to address the issue of Breast cancer early detection and treatment. To achieve this, implementation is done for one of the most popular data mining technique- k means clustering on very famous Wisconsin Breast Cancer Data. It also helps in classification of benign and malign cells in two different groups.

Overview:

This project is divided into 3 phases.

Phase 1-Deals with data analysis tasks.

Most of the data provided is structured data with few places those need to be replaced with relevant data eg: ? at some places need to be replaced with proper value

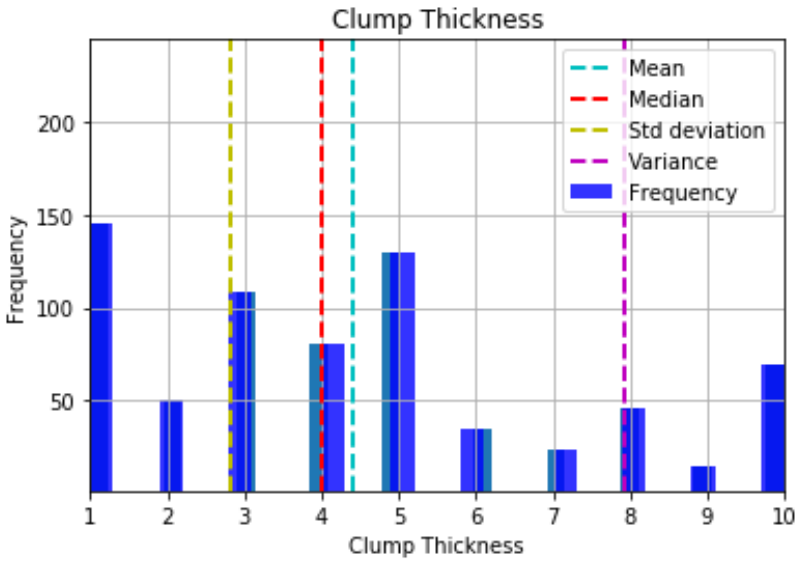
Steps followed to plot the final histogram which gives bar plot of discretized value frequency in each category used in cancer diagnosis:

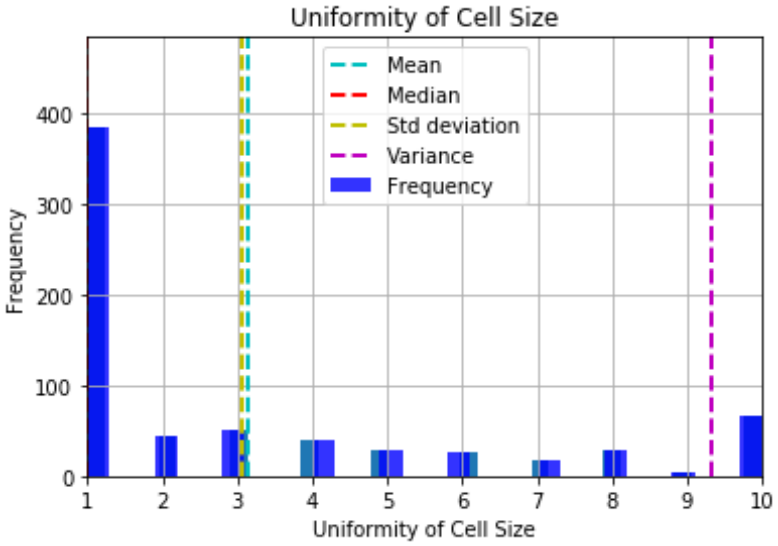
- Fetching of data from Breast cancer data from UCI machine learning repository.

  https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancerwisconsin.data

- Impute missing values
  This has been achieved by replacing "?" with mean value method.
- Plot basic graphs
  Histograms for attributes A2 to A10 are plotted using matplot libraries.
- Compute description of data
  Finding mean, median, standard deviation and variance of each of the attributes A2 to A10.

Sample output from Phase1:





Analysis:

- Clump thickness with value 1 is near to 150 count.
- Minimum number of Clump thickness of 9 is present with count less than 25.
- Overall Mean Value for Clump Thickness is 4.418
- For Uniformity of Cell size with value 1 is observed near to 400 times.
- Overall Mean value for Uniformity of Cell size is 3.134

Phase2- k-means algorithm implementation with 'Initialization', 'Assignment' and 'Recalculation'

K-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into different cells.

Steps Followed for K-means algorithm:

- Use Dataset with imputed missing values from phase 1
- Implementation for 'Initialization' step for K value 2.
- Implementation for 'Assignment' step
- Implementation for 'Recalculation' step
- Iterating above 2 steps for 1500 times.

Sample Output from Phase2:

----------Final Mean----------

mu_2: [3.39, 1.55, 1.73, 1.51, 2.26, 1.96, 2.3, 1.48, 1.13]

mu_4: [7.22, 7.48, 7.26, 6.36, 5.83, 7.88, 6.56, 6.66, 2.85]

----------Cluster Assignment----------

	ID	Class	Predicted Class
0	1000025	2	2
1	1002945	2	2
2	1015425	2	2
3	1016277	2	2
4	1017023	2	2
5	1017122	4	4
6	1018099	2	2
7	1018561	2	2
8	1033078	2	2
9	1033078	2	2
10	1035283	2	2
11	1036172	2	2

12	**1041801**	**4**	**2**
13	**1043999**	**2**	**2**
14	**1044572**	**4**	**4**
15	**1047630**	**4**	**2**
16	**1048672**	**2**	**2**
17	**1049815**	**2**	**2**
18	**1050670**	**4**	**4**
19	**1050718**	**2**	**2**
20	**1054590**	**4**	**4**

Analysis:

- Final mean calculation shows up mean for each attribute from A2-A10 after 1500 iterations of assignment and recalculation steps. Above figures shows up for cluster2, mean values varies from 1.13 -3.39
- For Cluster4, it varies from 2.85 -7.88 for different attributes from A2-A10
- Based on these means ,the data points are divided into 2 different clusters as above(shown only 20 data point records)
- Class shows the expected class of Sample code number and Predicted class is the class output from K Means Algorithm. It is not necessary that Class and Predicted class will always be the same. It depends on implementation of K means and the random mean selected in assignment step.
- With K means algorithm, we have two clusters-one which contains malign cells (cluster = 4) and the other containing benign cells (cluster = 2).

Phase3: Analyzing the quality of the centroids and of the partition

There are chances that a malign cell is being clustered into a benign cluster and vice versa. To check how well the clustering worked, we need to calculate the error rate for each of the cluster.

Steps followed for calculation of the total error rate of two clusters

- Error code is written for 2 different clusters with below sample calculation

For mu_2: $error\ B$=$total\ number\ of\ datapoints\ \ with\ Predicted\ class$=4 $coresspending\ Actual\ class$=2/ $total\ number\ of\ datapoints\ with\ Predicted\ class$=2

For mu_4: $error\ M$=$total\ number\ of\ datapoints\ with\ Predicted\ class$=2 $coresspending\ Actual$ class=4/ $total\ number\ of\ datapoints\ with\ Predicted\ class$=4

- Total error is calculated by addition of error B and error M

Sample output from Phase3:

After 1500 iterations of cluster assignments and recalculations:

Error B: 0.0078125

Error M: 0.31016042780748665

Total Error: 0.31797292780748665

Analysis:

- Error rate of Cluster containing malign cells (cluster = 4) is **high with value around 0.317**
- Error rate of Cluster containing benign cells(cluster=2) is **low with value of 0.0078125**
- It looks like data points with actual class of Benign are assigned to class of malign cells. These records need to be revisited to check on accuracy part.

References:

https://en.wikipedia.org/wiki/K-means_clustering

https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.data

https://matplotlib.org/users/pyplot_tutorial.html

https://pandas.pydata.org/pandas-docs/stable/dsintro.html