FINAL PROJECT

I590 – PYTHON

TABLE OF CONTENTS

PROBLEM STATEMENT

Breast cancer is a rising issue among women. A cancer's stage is a crucial factor in deciding what treatment options to recommend, and in determining the patient's prognosis. Today, in the United States, approximately one in eight women over their lifetime has a risk of developing breast cancer. An analysis of the most recent data has shown that the survival rate is 88% after 5 years of diagnosis and 80% after 10 years of diagnosis. With early detection and treatment, it is possible that this type of cancer will go into remission. In such a case, the worse fear of a cancer patient is the recurrence of the cancer.

The objective of the final project is to implement one of the most popular data mining technique, the "k-means" algorithm for the Wisconsin Breast Cancer dataset. We will do this in three phases. At the end of this project, you will be able to sepperate (classify) benign and malign cells into two different groups (classes) and evaluate how well your k-means algorithm (classifier) performs. You will also write a report of your findings.

DATASET DESCRIPTION

There are total 11 column in this dataset. In clasification tasks such as this one, those columns that contribute to the classification are called "features".

Column	Name	Description
1	scn	Sample code number: id number
2	a1	Clump Thickness: 1 - 10
3	a2	Uniformity of Cell Size: 1 - 10
4	a3	Uniformity of Cell Shape: 1 - 10
5	a4	Marginal Adhesion: 1 - 10
6	a5	Single Epithelial Cell Size: 1 - 10
7	a6	Bare Nuclei: 1 - 10
8	a7	Bland Chromatin: 1 - 10
9	a8	Normal Nucleoli: 1 - 10
10	a9	Mitoses: 1 - 10
11	class	2 (benign) or 4 (malignant)

PHASE 1 (30%)

For this phase, you will familiarize yourself with the k-means algorithm perform some exploratory data analysis. Exploratory data analysis is an important first step before performing any operations on your data. Getting an overview of picture of what is "inside" the dataset will impact many decisions you will make during the process. Small biases can be amplified as you progress, therefore, it is crucial to understand the context in which your results were obtained and your classifier evaluated.

Here is an overview of the process:

- Download the data and load it into a Pandas dataframe.
- Impute missing values
- Compute data statistics
- Plot basic histograms using matplotlib

DELIVERABLES

a. Download the dataset, breaast_cancer_wisconsin.csv, from the Final Project assignment page on Canvas, and load dataset into Panda using the pd.read_csv().

   Hint: Column a7 has missing values which are indicated by a '?'. You can use the na_values argument to replace the '?' with an indicator of your choice such as NaN. Chris Ablon has a number of clear examples in his section on importing csv's into Pandas.

b. The missing values will impede our classification, so we need to impute (replace) the missing values in column a7. There are many choices including the median, mode or any other method of your choice.[1] The Pandas user guide on working with missing data is an excellant resource.

c. Plot histograms for the features a2 through10. You can play with the bin size and other aesthetic properties.

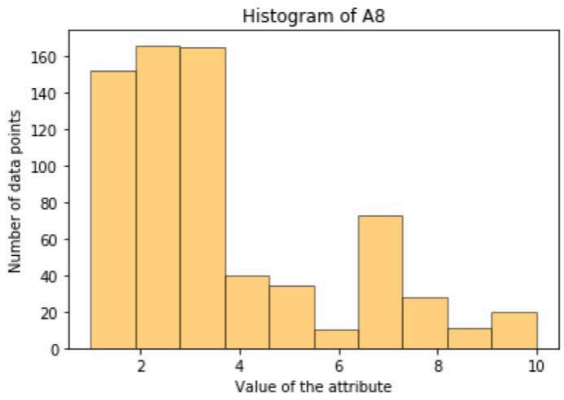
d. Obtain the mean, median, standard deviation, and variance for each feature column as well.

e. Save both your plots and descriptive statistics as a pdf. You are welcome to use any program to arrange your plots and descriptive statistics.

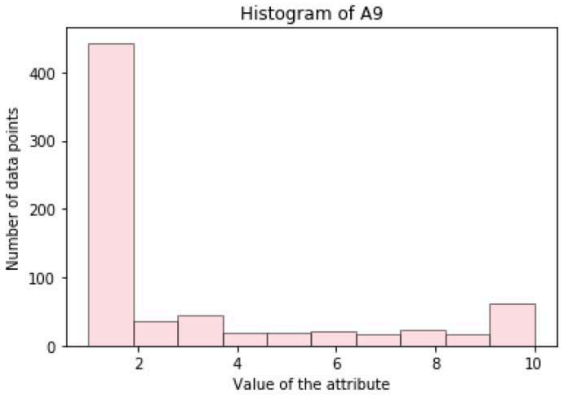
   Note: *your pdf **does not** have to look like the sample output below. Using the standard theme, putting your plots on one page, and your statistics on another page is entirely acceptable.*

Sample output:

Attribute 8 ----------
Mean: 3.4
Median 3
Variance: 5.9
Standard Deviation: 2.4



Attribute 9 ----------
Mean: 2.9
Median 1
Variance: 9.3
Standard Deviation: 3.0



1. Often there isn't a clear reason for choosing one over another. The important thing is to have a reason,and keep in mind how it may influence your results. In some cases, you may be able to justify the removal of incomplete rows entirely.

SUBMISSION GUIDELINES

- For this phrase, please name your program FP_P1.py and your pdf FP_P1.pdf.
- ***Do not upload a zip file.*** Please upload separate files for your code and pdf.
- Make sure your code is properly formatted, structured and commented.
- Include a header to program file with your name, date, question number and program description.
- Make sure that your program compiles. A program that doesn't compile may be scored 0 points.
- Each program should have a 'main()' function from which the entire code for a particular question can by run by directly executing the file.

PHASE 2 (40%)

You will implement k-means algorithm in the second phase:

- Revise k-means algorithm before writing code.
- Write code for 'Initialization' step
- Write code for 'Assignment' step
- Write code for 'Recalculation'

Note: Do not worry if you do not understand the k-means algorithm right away or at all. We will be providing extra support through supplemental material and recitation.

DELIVERABLES

In this dataset, 'id' is the first and 'class' the column. We do not consider these two columns for k-means computation. However, you will need these two columns for printing results and of course in phase 3. Use columns a2 to 10 for k-means computation.

a. Use Dataset with imputed missing values from phase 1.

b. Write code for 'Initialization' step for $K = 2$.

Steps: Choose any two points randomly from dataset as the initial means. Since you only consider column a2 to a10, a mean is nothing but a nine dimensional vector. Give first mean variable name μ_2 and second mean variable name μ_4. The two means represent two clusters.

Example:

Let's say randomly selected datapoints are 6 and 246 as two initial means. So values of μ_2 and μ_4 are:

$$\mu_2 = (8, 10, 10, 8, 7, 10, 9, 7, 1)\ ;\ \ \mu_4 = (5, 1, 1, 2, 2, 2, 3, 1, 1)$$

c. Write code for 'Assignment' step

You have defined two means in the previous step. Now, for each one of 699 datapoints compute euclidian distance from the two means.

For each datapoint you will have two distances. Assign a datapoint to cluster 2 if its distance from μ_2 is closer than its distance from μ_4, assign the datapoint to cluster 4 otherwise.

At the end of this step, you have each point assigned to one of the two clusters.

Example:

Let's take datapoint at row number 375 and compute its distance from both means.

d375 = (3, 1, 2, 1, 2, 1, 2, 1, 1)

$$d(375, \mu_2) = \sqrt{(3-8)^2 + (1-10)^2 + (2-10)^2 \ldots\ldots + (1-1)^2} = 20.25$$
$$d(375, \mu_4) = \sqrt{(3-5)^2 + (1-1)^2 + (2-1)^2 \ldots\ldots + (1-1)^2} = 2.83$$

Since $d(375, \mu_4) < d(375, \mu_2)$, point at row number 375 will be assigned to cluster 4.

d. Write code for 'Recalculation' step

So far, you have got every datapoint assigned to one of the two clusters. Next, you will update the means.

Example:

Let's say after performing step *b*, you have 300 datapoints assigned to cluster 2 and remaining 399 datapoints assigned to cluster 4. Update μ_2 by computing the mean from cluster 2 datapoints and update μ_4 by computing the mean from cluster 4 datapoints.

e. Iterate steps *c* and *d* untill any one of the following conditions is true:

1. All the datapoints do not change their cluster assignment compared to the previous iteration.
2. Steps *c* and *d* iterated 1500 times.

In step *c* use μ_2 *and* μ_4 values you computed in step *d* of the previous iteration.

At the end step *e*, you will have final values of means and information about which datapoints are assigned to which cluster. Print this result in console using pandas head() function. The default number of rows is 5 but you can pass 20. Additionally, copy and paste it in a document and put it aside in case you need it for your final report. ***You do not have to hand in the print out for this assignment.***

Your console output may look like this:

```
-------------------------Final mean---------------------------------
mu_2:  [3.0472103004291844, 1.3025751072961373, 1.446351931330472, 1.3433476394849786, 2.0879828326180259, 1.3800011310866602, 2.1051502145922747, 1.2618025751072961, 1.1094420600858537]
mu_4:  [7.1587982832618025, 6.7982832618025748, 6.7296137339055795, 5.733905579399142, 5.4721030042918457, 7.8739655269921256, 6.1030042918454939, 6.0772532188841204, 2.5493562231759657]


-------------------------Cluster assignment---------------------------------
         ID  Class Predicted_Class
0   1000025      2               2
1   1002945      2               4
2   1015425      2               2
3   1016277      2               4
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
```

Note: Your output may be different than the above. Include the first 20 datapoints (rows) in your report.

This version of k-means algorithm may suffer from poor initialization. Therefore, you may see different answers or swapped cluster assignments. We recommend running program multiple times and submitting the best result.

SUBMISSION GUIDELINES

- For this phrase, please name your program FP_P2.py.
- ***Do not upload a zip file.*** Please upload separate files for your code and pdf.
- Make sure your code is properly formatted, structured and commented.
- Include a header to program file with your name, date, question number and program description.
- Make sure that your program compiles. A program that doesn't compile may be scored 0 points.
- Each program should have a 'main()' function from which the entire code for a particular question can by run by directly executing the file.

PHASE 3 (30%)

During the third week, you will analyze the quality of the clustering.

- Write a code to calculate the individual and total error rate of your 2 clusters.
- Submit final report

Note: Do not worry if you do not understand the k-means algorithm right away or at all. We will be providing extra support through supplemental material and recitation. We will also be providing more information about what you might include in the report.

DELIVERABLES

Upon stopping your K means algorithm, you will have two clusters - one which contains malign cells (cluster = 4) and the other containing benign cells (cluster = 2). But there are chances that a malign cell is being clustered into a benign cluster and vice versa. To check how well your clustering worked, you need to calculate the error rate for each of your cluster.

a. Write code for 'ErrorRate' for K value 2.

Steps:
Your ErrorRate function will take two input arguments. First argument is cluster you got from your k-means algorithm, and second argument is the actual value of cluster for a particular datapoint. Actual cluster value of any datapoint is nothing but the corresponding 'class' column value.

For example: let's say the following data points are found to be part of cluster 2 i.e μ_2 after you run your k-means algorithm:

1017023,4,1,1,3,2,1,3,1,1,2	These two points belong to benign cells (cluster =2) since their class is 2. Here, you got cluster 2 and actual cluster is also 2. So No error for these two datapoints.
1018561,2,1,2,1,2,1,3,1,1,2	
1054593,10,5,5,3,6,7,7,10,1,4	This point is identified as malign cell (cluster = 4) since its class is 4. Here, you got cluster 2 and actual cluster is 4. So it is an error for this datapoint.

Now you have to calculate error rate using the following formula for each clusters:

1. For μ_2:

$$error\ B = \frac{total\ number\ of\ datapoints\ \ with\ Predicted\ class = 4\ coressponding\ to\ Actual\ class = 2}{total\ number\ of\ datapoints\ with\ Predicted\ class = 2}$$

2. For μ_4:

$$error\ M = \frac{total\ number\ of\ datapoints\ \ with\ Predicted\ class = 2\ coressponding\ to\ Actual\ class = 4}{total\ number\ of\ datapoints\ with\ Predicted\ class = 4}$$

3. Total errror rate

$$total\ error\ rate = \frac{total\ number\ of\ datapoints\ \ with\ Predicted\ class\ \neq\ Actual\ class}{total\ number\ of\ datapoints}$$

Print the above results in console.

b. Prepare final report that incorporates all the results and your comments for Phases 1 to 3. Again, will provide suggestions for what should go in it in a subsequent supplement.

SUBMISSION GUIDELINES

- For this phrase, please name your program FP_P3.py and your pdf FP_P3.pdf.
- ***Do not upload a zip file.*** Please upload separate files for your code and pdf.
- Make sure your code is properly formatted, structured and commented.
- Include a header to program file with your name, date, question number and program description.
- Make sure that your program compiles. A program that doesn't compile may be scored 0 points.
- Each program should have a 'main()' function from which the entire code for a particular question can by run by directly executing the file.