Project 2: K-NN, PCA, and Clustering

Submitted by: Gmon Kuzhiynaikkal
NU ID : 002724506

Introduction:

The Dataset used for the project is seed dataset. I have collected from the UCI machine learning Repository and it has 7 attributes. The examined group comprised kernels belonging to three different varieties of wheat: Kama, Rosa and Canadian. The independent variables are area, perimeter, compactness, Kernel Length,
Kernel Width, coefficient, grooveLength and the independent variable is wheat Type.

In this project, I was able to apply K-nearest neighbours on the seed dataset and I was able to further explore different types of the distance metric used by the K-nearest neighbours. I was able to interlink PCA with different clustering algorithms and I was able to find differences and variation in different clustering algorithms.

As a part of further extensions, I was able to explore one of the clustering quality metrics called Within-Cluster-Sum-of-Squared-Errors and I tried to compare with Rissannon Minimum Description Length (which is another cluster quality metric method). I have tried to plot the graph between these with different numbers of clusters(k).

1.Apply Nearest Neighbor classification to your data set:

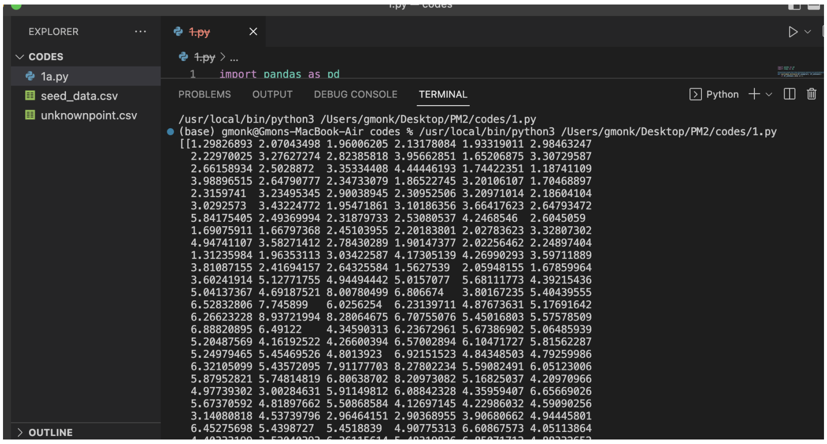
a)Write a distance metric for your data set:

I have used the euclidean distance metric for the calculation of the distance. I have created a sample data point and calculated the distance for all the data points and output of the program is given below. The equation used in the program is given below.

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2}.$$

It tries to measure the distance between the two points in a N-dimensional space. The output of the distance metric is given below. I have used one point and thus the is Ex1 vector

PS: python program to run this is given as 1a.py

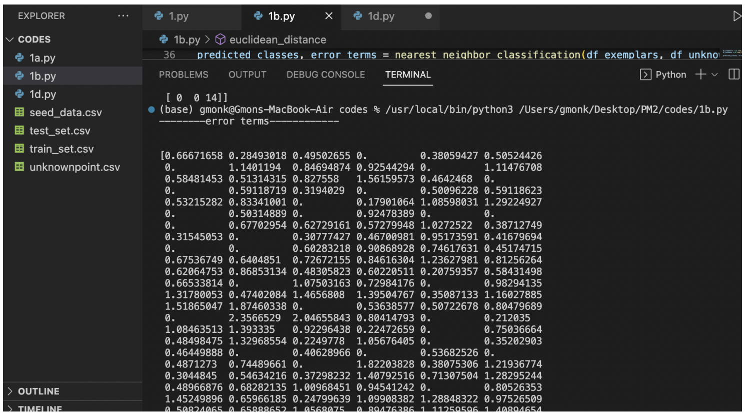


```python
import pandas as pd
```

```
/usr/local/bin/python3 /Users/gmonk/Desktop/PM2/codes/1.py
(base) gmonk@Gmons-MacBook-Air codes % /usr/local/bin/python3 /Users/gmonk/Desktop/PM2/codes/1.py
[[1.29826893 2.07043498 1.96006205 2.13178084 1.93319011 2.98463247
  2.22970025 3.27627274 2.82385818 3.95662851 1.65206875 3.30729587
  2.66158934 2.5028872  3.35334408 4.44446193 1.74422351 1.18741109
  3.98896515 2.64790777 2.34733079 1.86522745 3.20106107 1.70468897
  2.3159741  3.23495345 2.90038945 2.30952506 3.20971014 2.18604104
  3.0292573  3.43224772 1.95471861 3.10186356 3.66417623 2.64793472
  5.84175405 2.49369994 2.31879733 2.53080537 4.2468546  2.6045059
  1.69075911 1.66797368 2.45103955 2.20183801 2.02783623 3.32807302
  4.94741107 3.58271412 2.78430289 1.90147377 2.02256462 2.24897404
  1.31235984 1.96353113 3.03422587 4.17305139 4.26990293 3.59711889
  3.81087155 2.41694157 2.64325584 1.5627539  2.05948155 1.67859964
  3.60241914 5.12771755 4.94494442 5.0157077  5.68111773 4.39215436
  5.04137367 4.69187521 8.00780499 6.806674   3.80167235 5.40439555
  6.52832806 7.745899   6.0256254  6.23139711 4.87673631 5.17691642
  6.26623228 8.93721994 8.28064675 6.70755076 5.45016803 5.57578509
  6.88820895 6.49122    4.34590313 6.23672961 5.67386902 5.06485939
  5.20487569 4.16192522 4.26600394 6.57002894 6.10471727 5.81562287
  5.24979465 5.45469526 4.8013923  6.92151523 4.84348503 4.79259986
  6.32105099 5.43572095 7.91177703 8.27802234 5.59082491 6.05123006
  5.87952821 5.74814819 6.80638702 8.20973082 5.16825037 4.20970966
  4.97739302 3.00284631 5.91149812 6.08842328 4.35959407 6.65669026
  5.67370592 4.81897662 5.50868584 4.12697145 4.22986032 4.59090256
  3.14080818 4.53739796 2.96464151 2.90368955 3.90680662 4.94445801
  6.45275698 5.4398727  5.4518839  4.90775313 6.60867573 4.05113864
```

B.Implement Nearest Neighbour classification for your test set:

I have implemented the nearest neighbour from the data and error terms after running the program is given is given below:



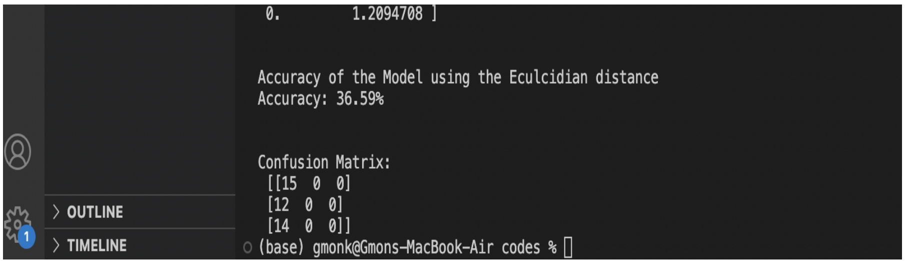
```python
predicted classes, error terms = nearest neighbor classification(df exemplars, df unkno
```

```
 [ 0  0 14]]
(base) gmonk@Gmons-MacBook-Air codes % /usr/local/bin/python3 /Users/gmonk/Desktop/PM2/codes/1b.py
--------error terms------------


[0.66671658 0.28493018 0.49502655 0.         0.38059427 0.50524426
 0.         1.1401194  0.84694874 0.92544294 0.         1.11476708
 0.58481453 0.51314315 0.827558   1.56159573 0.4642468  0.
 0.         0.59118719 0.3194029  0.         0.50096228 0.59118623
 0.53215282 0.83341001 0.         0.17901064 1.08598031 1.29224927
 0.         0.50314889 0.         0.92478389 0.         0.
 0.         0.67702954 0.62729161 0.57279948 1.0272522  0.38712749
 0.31545053 0.         0.30777427 0.46700981 0.95173591 0.41679694
 0.         0.         0.60283218 0.90868928 0.74617631 0.45174715
 0.67536749 0.6404851  0.72672155 0.84616304 1.23627981 0.81256264
 0.62064753 0.86853134 0.48305823 0.60220511 0.20759357 0.58431498
 0.66533814 0.         1.07503163 0.72984176 0.         0.98294135
 1.31780053 0.47402084 1.4656808  1.39504767 0.35087133 1.16027885
 1.51865047 1.87460338 0.         0.53638577 0.50722678 0.80479689
 0.         2.3566529  2.04655843 0.80414793 0.         0.212035
 1.08463513 1.393335   0.92296438 0.22472659 0.         0.75036664
 0.48498475 1.32968554 0.2249778  1.05676405 0.         0.35202903
 0.46449888 0.         0.40628966 0.         0.53682526 0.
 0.4871273  0.74489661 0.         1.82203828 0.38075306 1.21936774
 0.3044845  0.54634216 0.37298232 1.40792516 0.71307504 1.28295244
 0.48966876 0.68282135 1.00968451 0.94541242 0.         0.80526353
 1.45249896 0.65966185 0.24799639 1.09908382 1.28848322 0.97526509
 0.50824065 0.65888652 1.0568075  0.89476386 1.11259596 1.40894654
```

The python program to run this is 1b.py

C.Evaluate your classifier:

I have tired to evaluate the model using the euclidean distance and the accuracy of the model is 36.5% and the output of the confusion he confusion matrix i got after running this is t [[15 0 0], [12 0 0], [14 0 0]] indicates that there are no true positive predictions made by the nearest neighbour classifier. This suggests that the classifier is not performing well and needs further investigation. The output image of the file is given below and program to get the accuracy and confusion matrix is ‘1b.py’

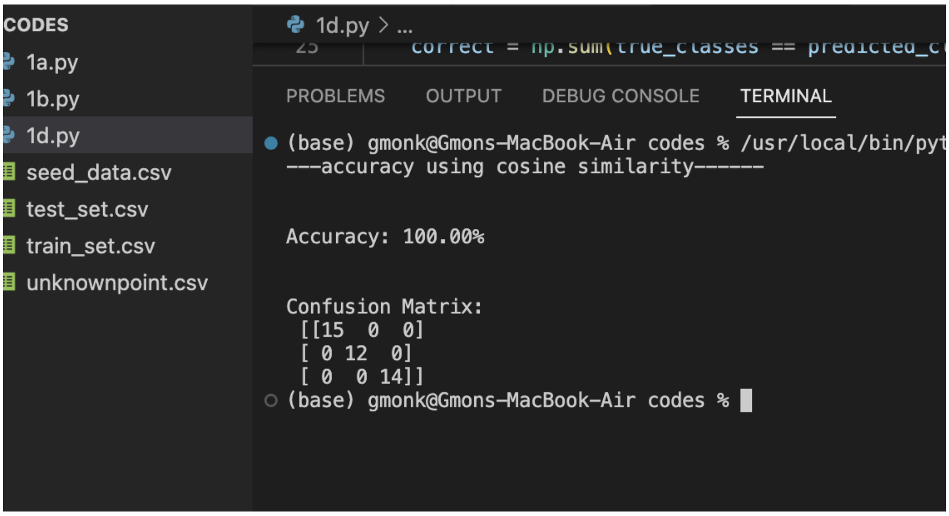


```
 0.         1.2094708 ]


Accuracy of the Model using the Eculcidian distance
Accuracy: 36.59%


Confusion Matrix:
 [[15  0  0]
 [12  0  0]
 [14  0  0]]
(base) gmonk@Gmons-MacBook-Air codes % 
```

D.Experiment with your classifier:

In order to improve the accuracy, I have tried to change the distance metric. I have tried the manhattan distance, it is giving me the same accuracy and confusion matrix as the euclidean matrix.

In order to get more accuracy, I have tried to use the more complex distance metric called the cosine similarity. I was able to get 100% accuracy and the confusion matrix found out as below:



```
25        correct = np.sum(true_classes == predicted_c
```

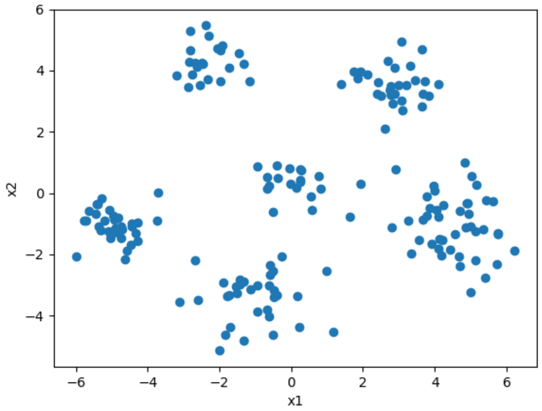
```
(base) gmonk@Gmons-MacBook-Air codes % /usr/local/bin/pyt
---accuracy using cosine similarity------


Accuracy: 100.00%


Confusion Matrix:
 [[15  0  0]
 [ 0 12  0]
 [ 0  0 14]]
(base) gmonk@Gmons-MacBook-Air codes % 
```

Thus I was able to change the accuracy of the model from 36.5% to 100% by changing my distance metric and classifier with a better true positive prediction.

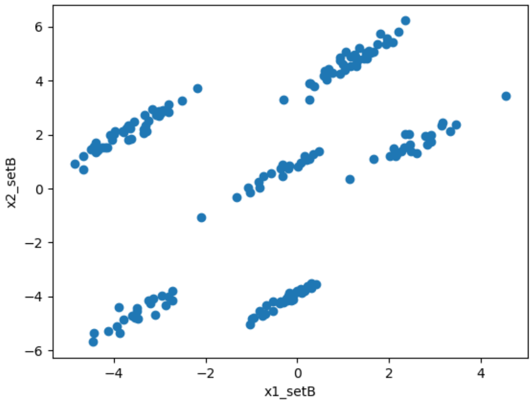
2.Experiment with clustering and PCA on two structured data sets

A) Download and plot the example data:

The scatter plot for the first set A is shown below. I could see 6 slightly circular clusters naturally occurring from the data. The scatter plot for the Set A csv file is shown below:



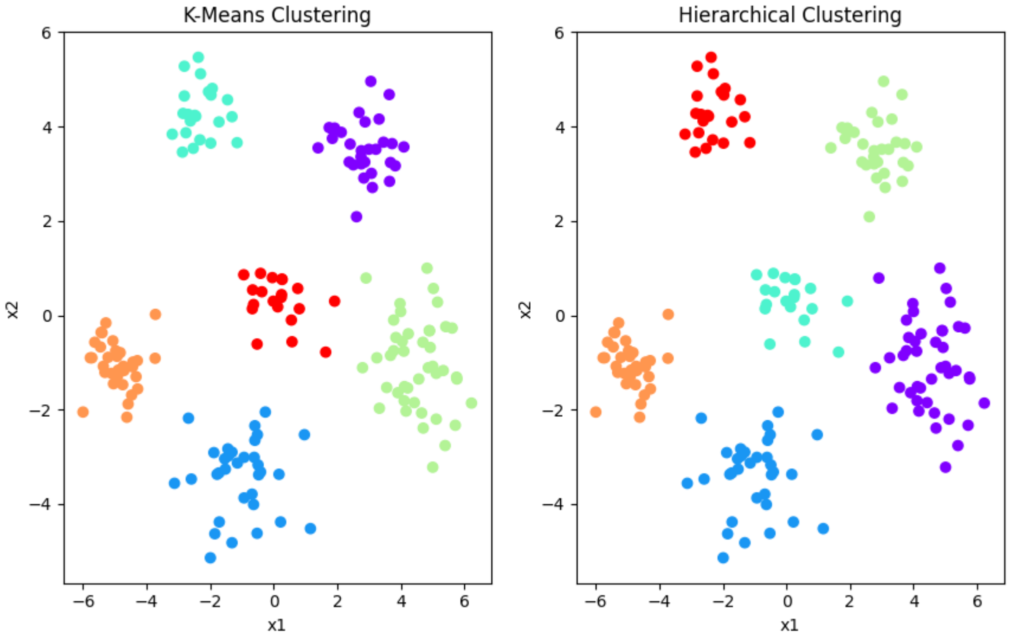
The scatter plot for the Set B csv file is also 6 clusters and it more like a straight line clusters and the the scatter plot is given below for Set B:



The python file to run this is ‘2a.py’

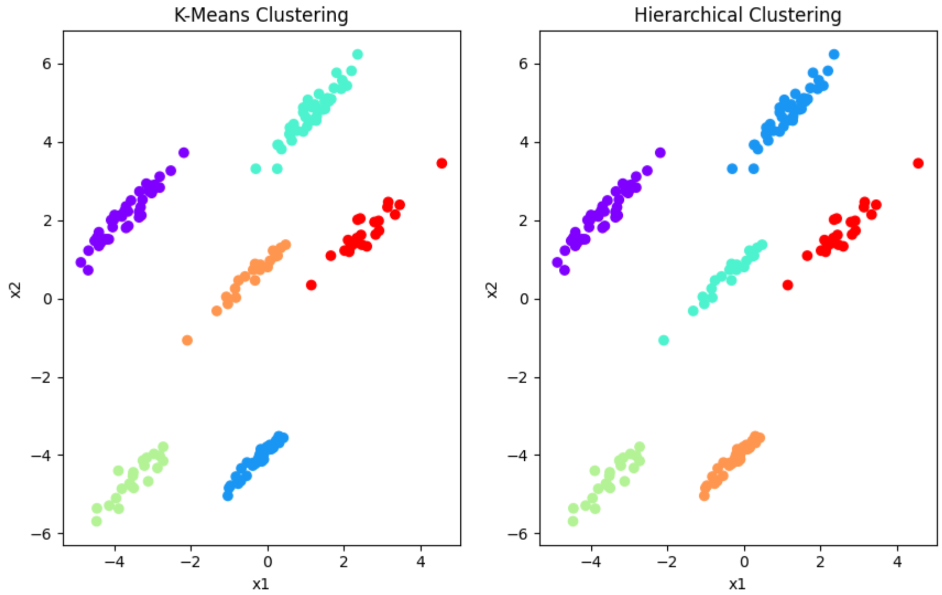
B.Cluster the example data:

For the setA: using the k means clustering plot diagram(k=6) and hierarchical clustering plot diagram is given below:



From the plot diagram for the data setA. I could see that both of them have produced the same shape and number of clusters using the different clustering algorithms.

For set B:using the k means clustering plot diagram(k=6) and hierarchical clustering plot diagram is given below:

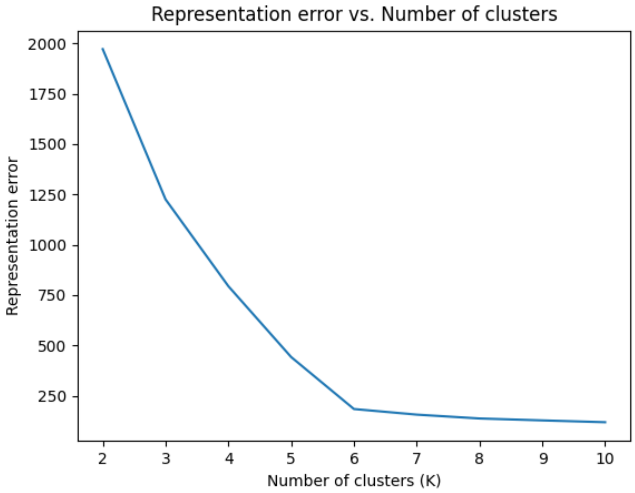


From the plot diagram for the data setB. I could see that both of them have produced the same shape and number of clusters using the different clustering algorithms also.

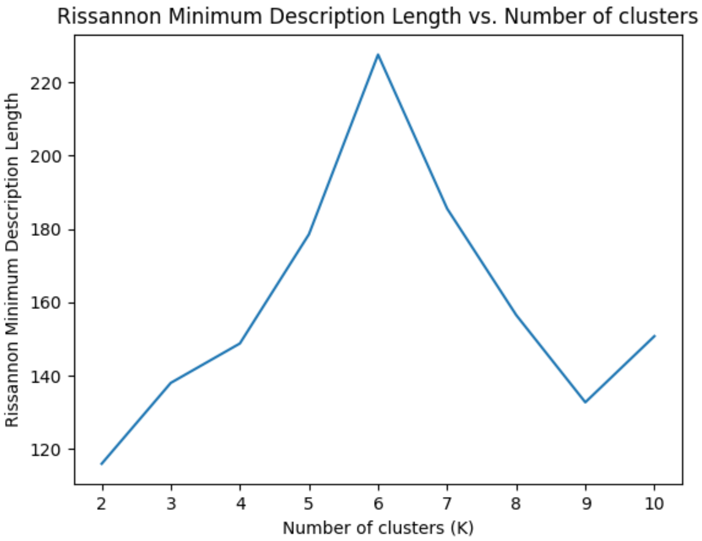
PS: python program to run this is ‘2b.py’

C.Compare using different numbers of clusters:

I have tried to plot the representation error with the number of clusters(k) and I could find that the error is decreasing as the number of the clusters are increasing. The plot using the setA a is given below:



Representation error almost got a stable from k=6 onwards. In order to find the cluster quality, I have used the Rissannon Minimum Description Length and the plot for RMDL to number of clusters is given below:

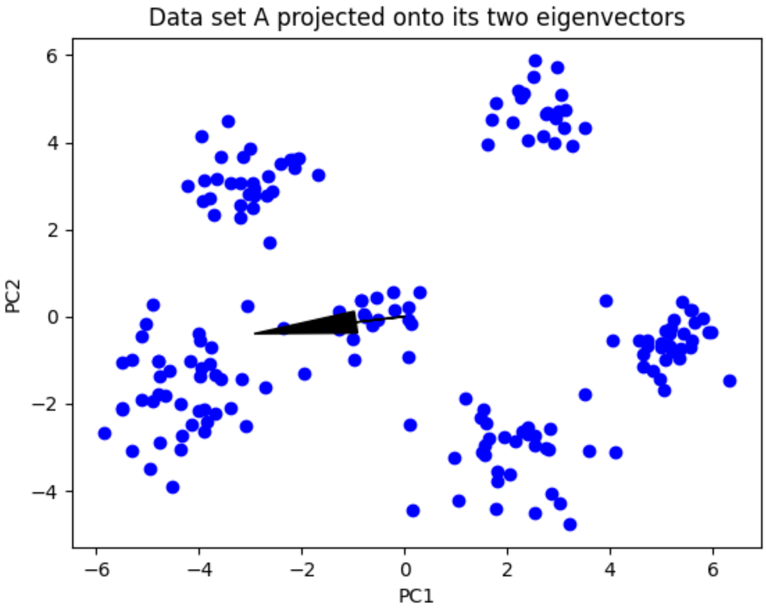


From the graph, I could see that k=6 has the best cluster quality as per the Rissannon Minimum Description Length. Thus for the Set A dataset, we could say that the ideal state of the data will be having 6 clusters.

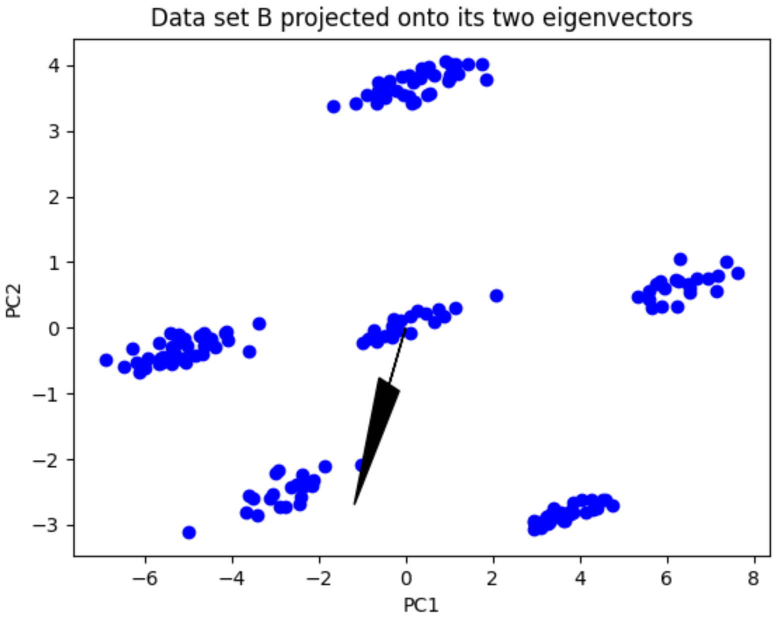
PS : Python code for running this ‘2c.py’

D:Apply PCA to the example data:

PCA is applied on the set A and the plot of the graph is given below the arrow head is downward. This implies that there is negative correlation with the features.



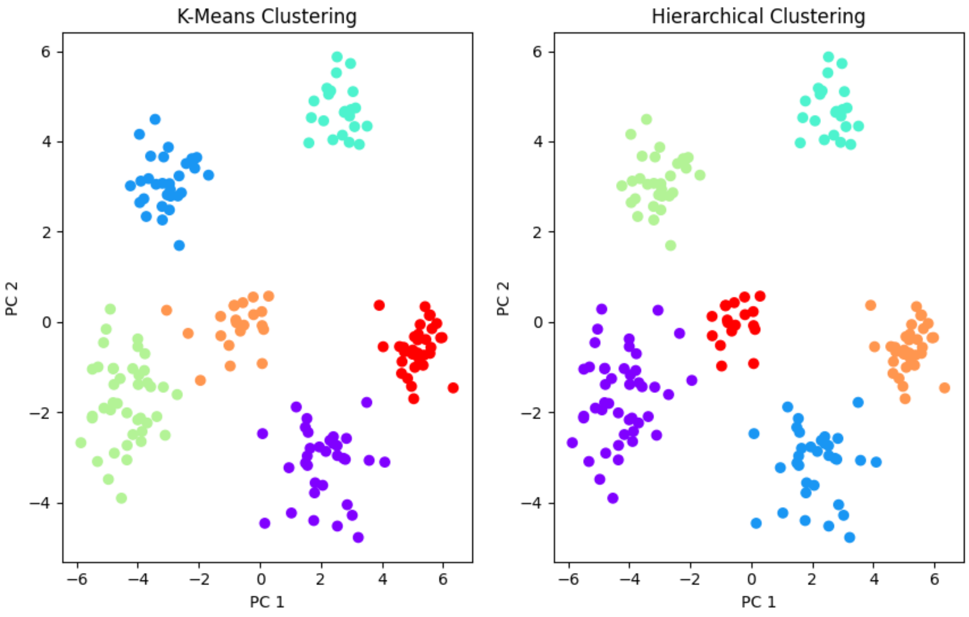
For the set B the plot of the graph is given below and the arrow is pointing to the PC1 axis direction that there is a positive correlation with the features.

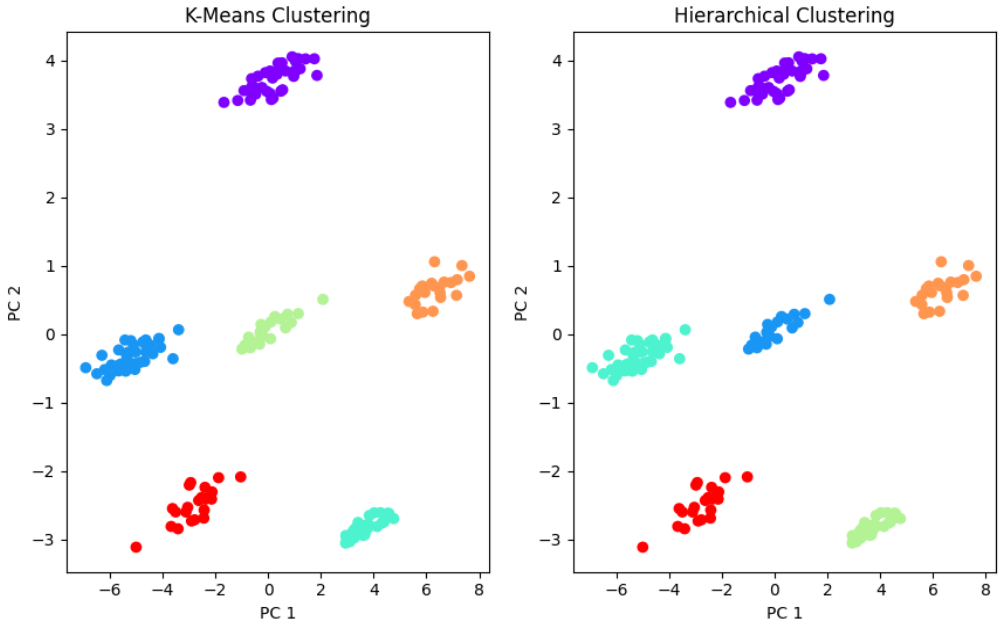


PS: python file to run this program is ‘2d.py’

E.Recluster using the projected data:

There is some difference in the Projected Data plot compared it is less scattered and the data are easier to interpret because of the lesser dimensionality. Scatter plot for Set A and Set B are given below:

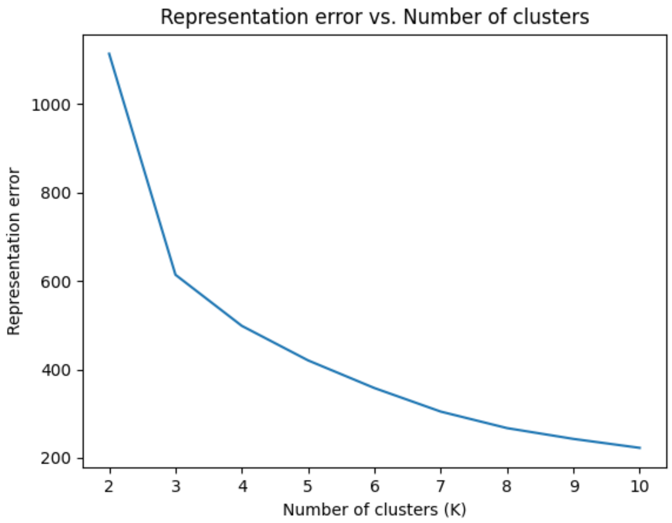




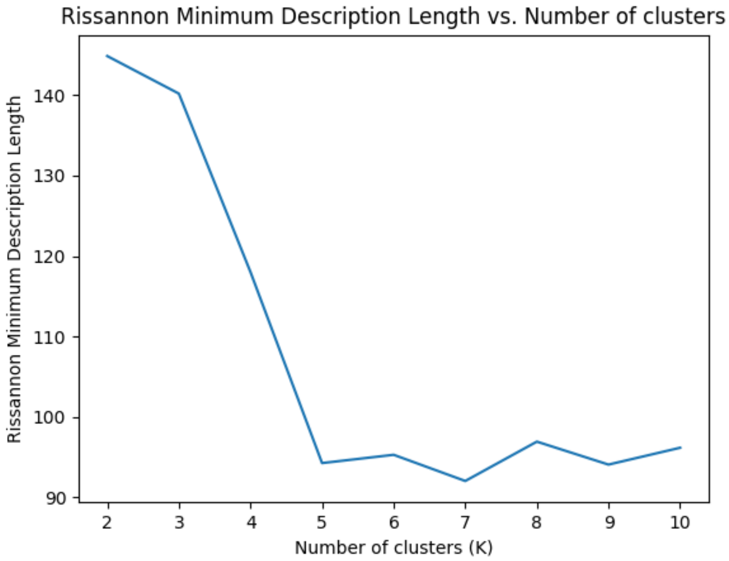
PS: python code for running this is ‘2e.py’

3.Apply K-Means Clustering to your data set:

The data used for the project belongs to the seed data csv containing 7 attributes and the dependent variable is the wheat type, we are trying to classify 3 types of wheat types. The representation error vs number of cluster of the dataset is given:



If we check, we could see a variation for the cluster number 3 and plot which represent the cluster quality(used Rissannon Minimum Description Length) is given as below:



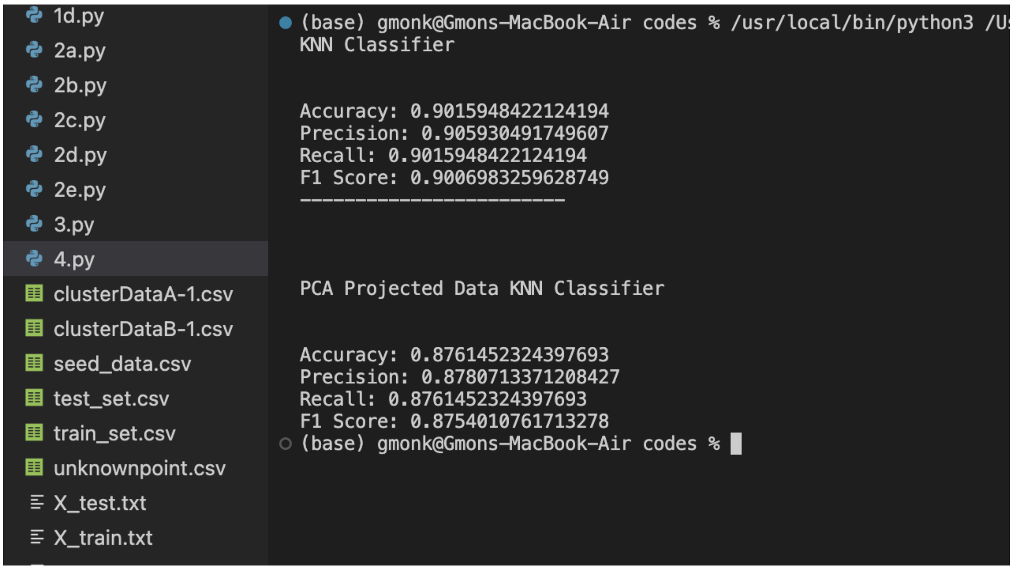
From this graph, we can conclude that, best cluster value(k value) for my dataset will be nearly equal to 3.

PS: Python program to run this program is ‘3.py’

4.Use K-Nearest Neighbour and PCA to classify activity from phone measurements:

Comparing the KNN classifier and the PCA projected data KNN classifier, it can be seen that the raw data KNN classifier has a higher accuracy (0.9016) compared to the PCA projected data KNN classifier (0.8761). The precision and recall scores are also higher for the raw data KNN classifier. The F1 score, which is a weighted average of precision and recall, is also higher for the raw data KNN classifier (0.9007) compared to the PCA projected data KNN classifier (0.8754).

These results suggest that the raw data KNN classifier is performing better than the PCA projected data KNN classifier. The output of the comparison is given below on the data.

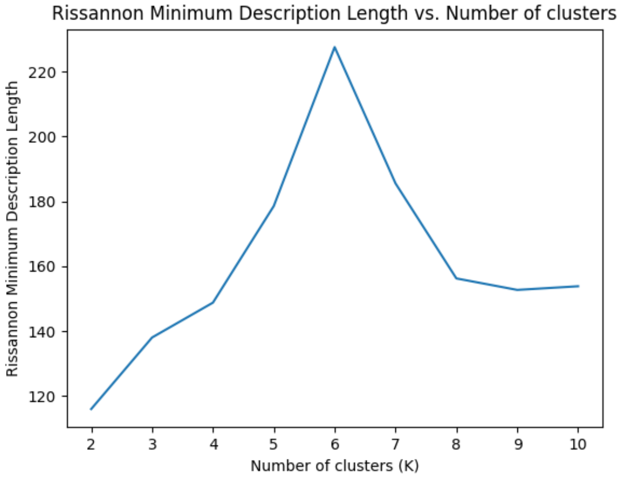


```
● (base) gmonk@Gmons-MacBook-Air codes % /usr/local/bin/python3 /U
  KNN Classifier


  Accuracy: 0.9015948422124194
  Precision: 0.905930491749607
  Recall: 0.9015948422124194
  F1 Score: 0.9006983259628749
  ------------------------



  PCA Projected Data KNN Classifier


  Accuracy: 0.8761452324397693
  Precision: 0.8780713371208427
  Recall: 0.8761452324397693
  F1 Score: 0.8754010761713278
○ (base) gmonk@Gmons-MacBook-Air codes % 
```

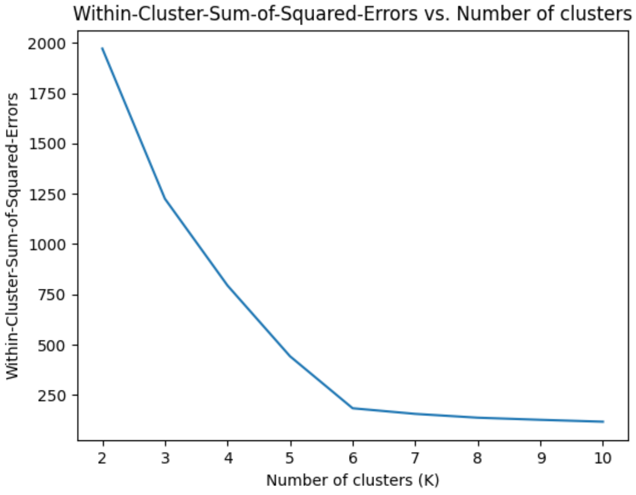
The reasons why the PCA projected data KNN classifier performed worse than the raw data KNN classifier could be that the PCA projection lost some important information in the data, leading to decreased performance. Another possibility is that the number of dimensions used in the PCA projection was not enough to explain the complex relationships in the data, leading to overfitting or underfitting

Extensions:

I tried to implement Within-Cluster-Sum-of-Squared-Errors (WCSS) also known as inertia, this metric measures the sum of squared distances between each data point and its closest cluster centroid. A smaller WCSS score indicates a better clustering solution. I tried to compare that with Rissannon Minimum Description Length on the dataset given for task 2. Tried to plot the number of clusters from (2 to 11) with the different cluster quality metric value. The plot for Rissannon Minimum Description Length is given below, for this highest value is the best and we could see that k=6 is a better value for it.



I tired to Plot WCSS, with the number of clusters and we could see that it is lower for when k=6, so it is better to take the k value for the above k mean clustering algorithm as 6



Summary (Reflection of what I learned from the project):

Some of the key points, I have learned from the projects are given below

➔ Implementing different distance metrics, which are used in the different algorithms of machine learning.

➔ Implementing nearest neighbour algorithm from the distance metrics that i have designed.

➔ Checking the accuracy of the classifies and analysing it with the confusion matrix

➔ Tried to improve the accuracy of the nearest neighbour classifier with different distance metrics and I was able to to improve the accuracy to 100% using cosine similarity distance metrics in the nearest neighbour.

➔ Tried to implement different clustering algorithms using sk-learn.

➔ learned to find the natural clusters in a scatter plot, before applying any cluster algorithm by plotting the data.

➔ Find out the variation of the representation error in the clustering algorithms with increase in the cluster algorithms.

➔ Find out the trend of applying principal component analysis(PCA) on the various cluster algorithms and plotting the eigenvector on scatter plot.

➔ I was able to find the variation of the PCA with nearest neighbour algorithms as well.

➔ I was able to learn different types of the cluster quality metrics, which can be used to compare different cluster quality in the algorithms.

➔ Learned about the representation error used in the clustering algorithms.

➔ As a part of the extension, I have tried to use different cluster quality metrics and tried to compare it.

Bibliography:

Some of the importance links and materials i have used for this projects are:

1. https://archive.ics.uci.edu/ml/datasets/seeds#

2. https://scikit-learn.org/stable/

3. Textbook by Muller and Guido

4. https://youtu.be/ukzFI9rgwfU