ALDA Homework 1 – 2018 :

Submitted by :
Srinivasan Balan – Unity id : sbalan
Harika Malapaka – Unity Id : hsmalapa

1. (13 points) [Song Ju] Classify the following attributes as binary, discrete, or continuous. Also classify them as nominal, ordinal, interval, or ratio. Some cases may have more than one interpretation, so briefly justify your answer if you think there may be some ambiguity.

(a) (1 point) Hair color (Black, Blonde, Red)
Discrete, Nominal

(b) (1 point) Level of agreement (yes, maybe, no)
Discrete, Nominal

(c) (1 point) Income earned in a week
Continuous, Ratio

(d) (1 point) Celsius temperature
Continuous, Interval

(e) (1 point) Genotype (Bb, bb, BB, bB)
Discrete, Nominal

(f) (1 point) ISBN numbers for books.
Discrete, Nominal

(g) (1 point) Time in terms of AM or PM
If we consider the time in AM or PM, the solution is Binary, interval
Else, if we consider all the time say ex.) 4:30:45 pm which shows all hours, minutes and seconds, then the solution is Continuous, interval.

(h) (1 point) Waiting number for restaurant
Discrete, ordinal

(i) (1 point) Years of work experience
There is little ambiguity here:

If we consider there are chances that we may represent no. of years like 4.5, then **Continuous, Ratio**

We assumed the number of years of work experience is defined as a discrete year and even if we consider the months as additional feature, we will get 12 equal months as discrete lower and upper bound. In general sense we do not do that – so it can be **Discrete, Ratio** as well.

(j) (1 point) Categorization of clothing (hat, shirt, pants, shoes)
Discrete, Nominal

(k) (1 point) Angles as measured in degrees between 0 and 360
Continuous, Ratio

(l) (1 point) Ratings of movies (G, PG, R)
There is again little ambiguity here. Although the question says ranking, we were not given a specific order m so it would be **Discrete, Nominal**.

If we think the G, PG and R are in order, then it would be **Discrete, Ordinal**. Ranking scale of G(Low), PG(medium), R(High)

(m) (1 point) Coat check number. (When you attend an event, you can often give your coat to someone who, in turn, gives you a numb that you can use to claim your coat when you leave.)
Discrete, nominal

2) (10 points) [Ruth Okoilu] Data Transformation.

(a) (6 points) What are the maximum and minimum values of tf0 ij and tf00 ij respectively? Please specify what cases the max and min value achieves.

a) The max and min values are :
For tf(i,j)' :
Max case: where the no of documents in which the 'i' term occurs df(i) is 0.
When df(i) is zero , the IDF is log(m/df(i)) = log(m/0) = log(infinity)=infinity
tf(i,j)'= tf(i,j) * infinity = p * infinity = infinity

Min case: where the no of documents in which the 'i' terms occurs – df(i) is same as no of documents in corpus – i.e m.
When df(i) is m, IDF is log(m/df(i)) = log(m/m) = log(1)=0

tf(i,j) = tf(i,j)* IDF = p * 0 =0.

For tf(i,j)'' :

Max case: where the no of documents in which the 'i' term occurs (df(i)) is 0.
When df(i) is zero, IDF is log(($\sum^{k=m}_{k=i}$ (d_k)) / ($\sum^{k=df(i)}_{k=l}$ (d_k)))
= log(($\sum^{k=m}_{k=i}$ (d_k)) / ($\sum^{k=0}_{k=l}$ (d_k)))
= log(($\sum^{k=m}_{k=i}$ (d_k)) / 0)
= log (infinity)
= infinity

Min case: where the no of documents in which the 'i' terms occurs – df(i) is same as no of documents in corpus – i.e m.

When df(i) is m, IDF is log(($\sum^{k=m}_{k=i}$ (d_k)) / ($\sum^{k=df(i)}_{k=l}$ (d_k)))

= log(($\sum^{k=m}_{k=i}$ (d_k)) / ($\sum^{k=m}_{k=l}$ (d_k)))
= log(1)
=0

(b) (4 points) Briefly explain the purpose for using tf' (ij) and tf''(ij) respectively in the context of natural language processing and also explain what is the main difference between tf' and tf''

b)

- The difference between tf(i,j)' and tf(i,j)'' is that in the usage of tf-idf in NLP, there might be two kinds of applications of tf-idf.

- In both of the mentioned formulas, the change is only in IDF term – which specifies whether the term is more common or more specific across the collection.
- When text mining of Similar subject documents there might be terms which are important but occurring many times.
- If mining is on similar content documents, we can proceed with tf’.
- If we are dealing with random content documents, there might be a small document, but the presence of a term is very crucial for that document.
- There might be bias towards longer documents in this case.
- If we normalize the IDF, we can avoid the bias.

Example:

Say the word ‘hormone’ occurs in 2 documents out of 5 documents.
And the document lengths are:
Doc 1 -- 40
Doc 2 – 30
Doc 3 – 20
Doc 4—100
Doc 5 – 15

Say the word has occurred in Doc 3 and Doc 5
Since the document length is small compared to other documents in corpus, implies that word (hormone) is very important and it specifies the rarity. The below calculations explain the percentage of term frequency.

Let’s see whether tf(i,j)’ and tf(i,j)’’ formulas gives more importance – i.e more value of IDF
Tf(i,j)’ = p * IDF
IDF = log(5/2) = 0.397
Tf(I,j)’ = p*0.397

Tf(I,j)’’ = p * IDF
Where IDF = log((40+30+20+100+15) / (20+15))
= log(205/35)
= 0.76
Tf(I,j)” = p* 0.76

Thus, when the researcher wants to study a critical word of interest say “cybercrime”, it doesn’t matter how many documents it has the word cybercrime. All the documents are important. Hence Tf(I,j)” is very useful and helpful. On the other hand, if we are interested in a common word of study say “chemical”, we don’t need to capture the number of words in each corpus. We can use Tf(I,j)’ to capture the term frequency.

3. (8 points) [Xi Yang] Answer the following questions:

(a) (4 points) A healthcare dataset contains 523,000 patients. Among these patients, 26,150 patients have albinism and the remaining 496,850 patients have normal skin. Suppose we will sample 1,000 patients from the dataset to conduct albinotic analysis, which sampling method should be selected to apply in this situation: simple random sampling or stratified sampling, and why? With the selected sampling method, how many albinotic and normal skin patients will be sampled, respectively?

a) Since the dataset has 2 different groups, if we use Simple Random Sampling, the probability that any sample selected is same. In that case, there are chances that many data points come from one group. The sampling might not be uniform from the groups. So, if we perform any analysis on such data, the results may be in-accurate. Simple random sampling doesn't represent a good subset (sample) if different objects contain different frequencies.

- If we opt stratified sampling, it divides the group into strata and draws specified number (samples) from each group.
- The specific number maybe either proportional to entire dataset or a constant number from each group.
- Assuming a proportion stratified method, the no of samples from each group can be calculated using the formula :

(Sample size / population size) * strata size

So, from strata 1: no of samples from albinotic patients group is

(1000/523000) * 26150 = 50

From strata 2: no of samples from normal skin patients is

(1000/523000) * 496850 = 950

Therefore, number of samples are 50 and 950 from albinotic and normal skin patients respectively.

(b) (4 points) Consider the following scenario, a patient's systolic blood pressure (SBP) is recorded to be 250. When SBP is higher than 180, a patient is considered to have hypertensive crisis and need to seek the emergency care. For this given scenario, is the recorded data noise or outlier? And why? (no point will be given if you do not give a justification).

It's not that easy to categorize data as either noise or outlier because we do not know the size of the dataset and how it has been recorded. We also don't know the mean and the standard deviation of the spread of data.
Note: we have assumed that there is no problem with recording, since nothing has been specified in the question. Thus, there shouldn't be any kind of Noise in the data.

There might be 2 emerging cases to this task by categorizing this:

1. Outlier:

If we consider the fact that the dataset is genuine, we would have only few records of SBD value greater than 250 since that's high value and those patients need emergency care. This implies that the quartile deviation of the dataset is falling more than the 90th percentile. In other words, when the data point crosses the whisker of the distribution, the 250 point is considered as an outlier. On a general basis, there won't be many patients will such high SBP value. Q-Q plot and Box-plot will be helpful to visualize the dataset and identify the outlier easily.

2. Cannot be Determined

Since we do not know how many data points are having SBP value more than 250, we cannot really say this is an outlier. There might be the case that interquartile deviation (75th and 25th percentile) or more of the data points have the SBP value 250 or more.
In this case, we cannot call it as outlier, since the definition of outlier is when the range of the value is not abiding with the rest of the data points which is above the whisker points. Depending upon the dataset and the spread of data, it is easily characterized by the data whether it is an outlier or not.

4. (15 points) [Song Ju] Write your code in Matlab, R or Python to perform the following tasks, please report your outputs and key codes in the document file, and also include your code file (end with .m, .r or .py) in the .zip file.

(a) (1 point) Generate a 5*5 identity matrix A.

```
[[ 1.  0.  0.  0.  0.]
 [ 0.  1.  0.  0.  0.]
 [ 0.  0.  1.  0.  0.]
 [ 0.  0.  0.  1.  0.]
 [ 0.  0.  0.  0.  1.]]
```

(b) (1 point) Change all elements in the 2nd column of A to 3.

```
[[ 1.  3.  0.  0.  0.]
 [ 0.  3.  0.  0.  0.]
 [ 0.  3.  1.  0.  0.]
 [ 0.  3.  0.  1.  0.]
 [ 0.  3.  0.  0.  1.]]
```

(c) (1 point) Sum of all elements in the matrix (use a ”for/while loop”).

```
The sum of all elements of matrix is :  19.0
```

(d) (1 point) Transpose the matrix A (A = AT)

```
[[ 1.  0.  0.  0.  0.]
 [ 3.  3.  3.  3.  3.]
 [ 0.  0.  1.  0.  0.]
 [ 0.  0.  0.  1.  0.]
 [ 0.  0.  0.  0.  1.]]
```

(e) (2 points) Calculate sum of the 3rd row, and the diagonal in the matrix A.

```
The sum of the diagonal elements:  7.0
The sum of row 3 elements:  1.0
```

(f) (1 point) Generate a 5*5 matrix B following Gaussian Distribution with mean 5 and variance 3.

```
[[ 5.00170978  8.46154942  7.3055585   7.39077179  0.84517057]
 [ 3.96881521  7.90193676  3.79792795  6.79216631  4.40738773]
 [ 5.08755031  4.5194446   2.22832918  7.42762949  2.47472597]
 [ 4.95051884  3.66541729  5.7419155   5.41687799  3.57649978]
 [ 5.8943523   3.05552078  4.61295981  4.14252149  4.63152475]]
```

(g) (2 points) From B, using matrix operations to get a new matrix C such that, the first row of C is equal to the first row of B times the second row of B, the second row of C is equal to the sum of the 3rd and 4th row of B minus the 5th row of B.

```
[[ 19.85086182  66.86262846  27.74598479  50.19935121   3.72499441]
 [  4.14371684   5.12934112   3.35728486   8.70198599   1.419701  ]]
```

(h) (2 points) From C, using one matrix operation to get a new matrix D such that,the first column of D is equal to the first column of C times 2, the second column of D is equal to the second column of C times 3 and so on.

```
[[  39.70172364  200.58788539  110.98393917  250.99675603   22.34996648]
 [   8.28743368   15.38802335   13.42913945   43.50992996    8.51820601]]
```

(i) (2 points) X = [2,4,6,8]T, Y = [6,5,4,3]T, Z = [1,3,5,7]T. Compute the covariance matrix of X, Y and Z.

```
[[ 6.66666667 -3.33333333  6.66666667]
 [-3.33333333  1.66666667 -3.33333333]
 [ 6.66666667 -3.33333333  6.66666667]]
```

(j) (2 points) Verify the equation: ¯ x2 = (¯ x2 + σ2(x)), using x = [2,4,6,8,10,12,14,16,18,20]T. σ(x) is the standard deviation.

True - equation verified: This is true only when we consider population standard deviation. Refer the program file for illustration.
Otherwise it is not true.

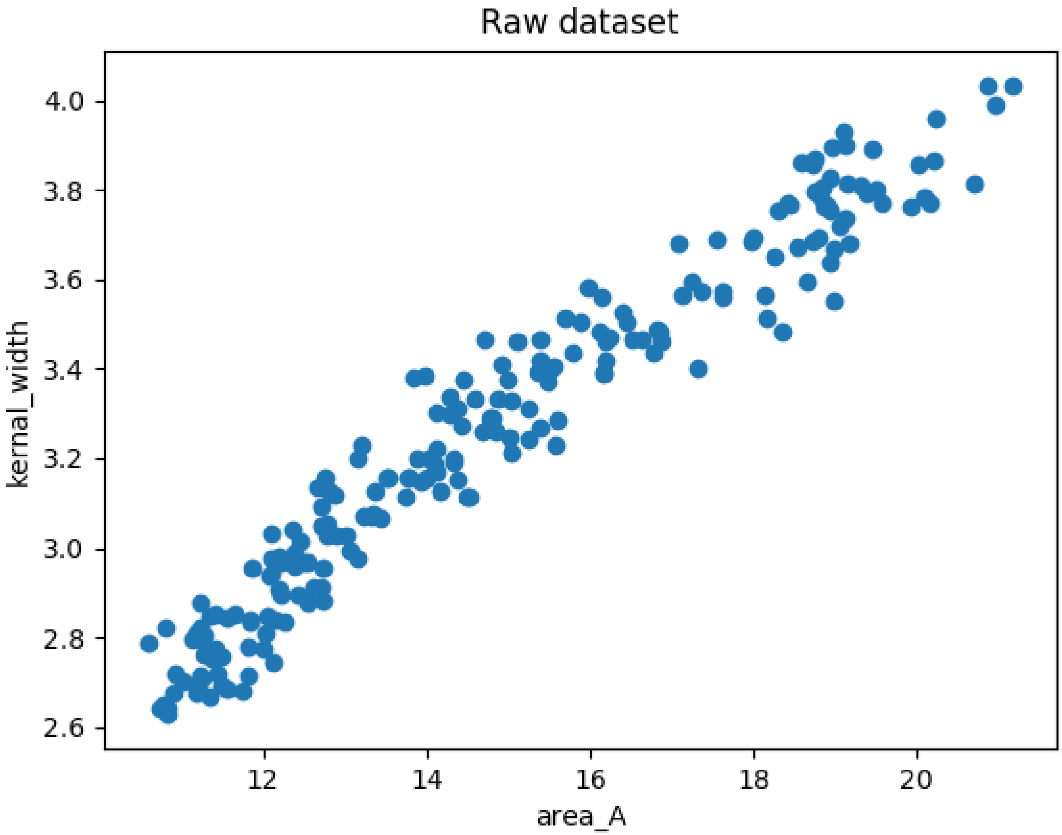
5. (33 points) [Ruth Okoilu] For this exercise, use the provided ‘seeds.csv’ file, which contains a list of 210 data instances. The examined group comprised kernels belonging to three different varieties of wheat: Kama, Rosa and Canadian, 70 elements each, randomly selected for the experiment. (Source: https://archive.ics.uci.edu/ml/datasets/seeds) There are 8 columns representing: 1) area A, 2)perimeter P, 3) compactness, 4) length of kernel, 5) width of kernel, 6) asymmetry coefficient, 7) length of kernel, and 8) groove Class (Type of wheat). For the purpose of this exercise, you consider two features, ‘area A’ and ‘kernel width’ (columns 1 & 5) of the provided ‘seeds.csv’ dataset. Write your codes in Matlab, R or Python to perform the following tasks, please report your outputs and key codes in the document file, and also include your code file (end with .m, .r or .py) in the .zip file.

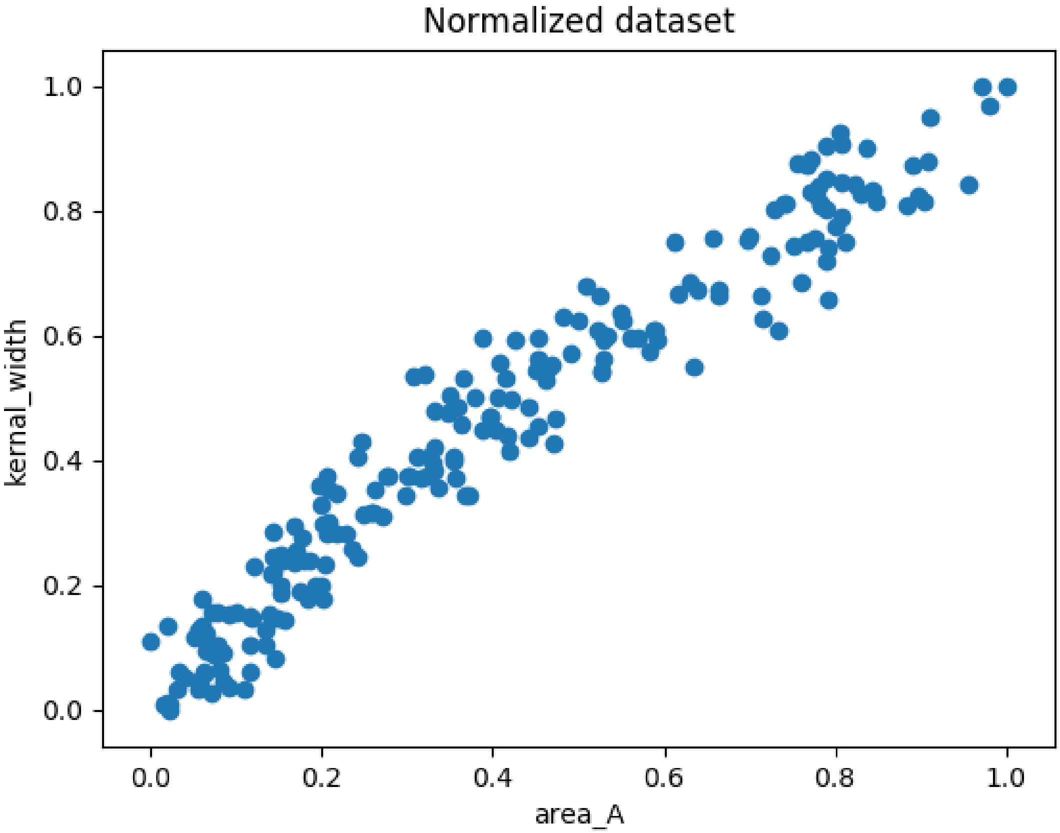
(a) (3 points) Load the file and read ‘area A’ and ‘kernel width’ columns and save them as the original raw dataset. Apply normalization (transformed data $z \in [0,1]$) to the raw dataset to get the normalized dataset and apply the standardization to the raw dataset to get the standardized dataset. Show the range of the two features in each dataset.

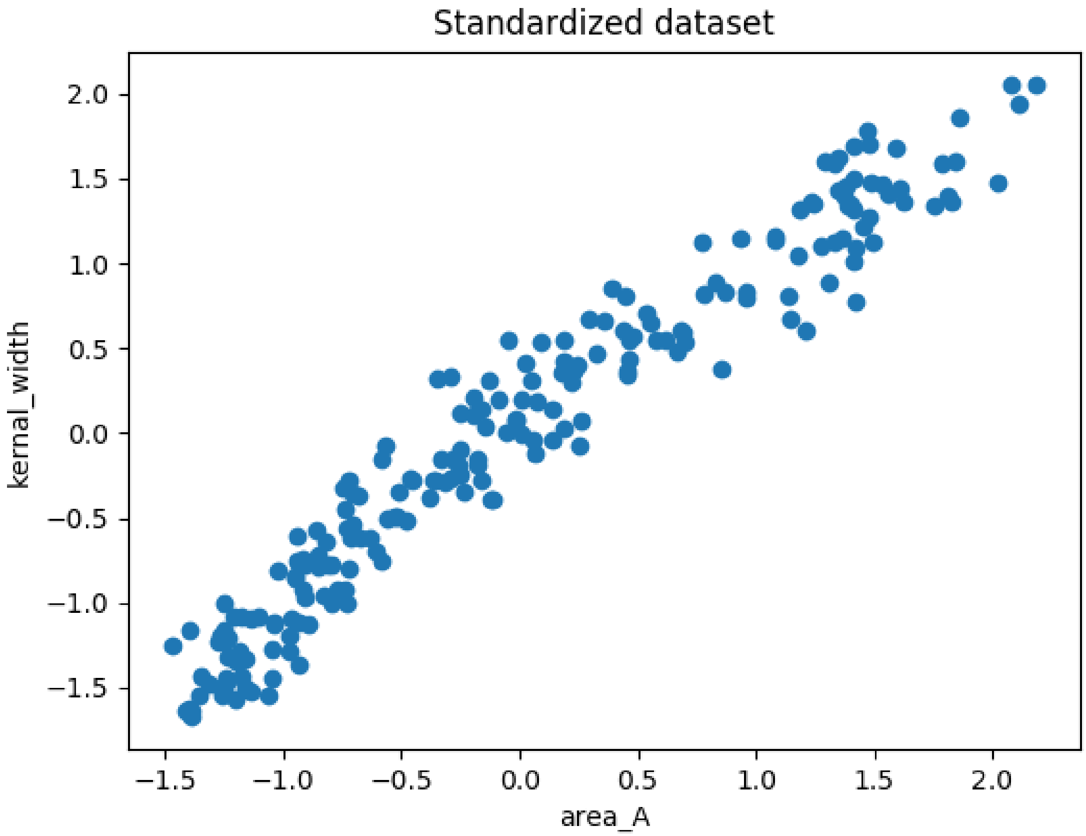
```
Range for raw_dataset
area_A          10.590
kernal_width     1.403
dtype: float64

 Range for Standardized dataset :
area_A          3.648248
kernal_width    3.723322
dtype: float64

 Range for Normalized dataset :
area_A          1.0
kernal_width    1.0
dtype: float64
```

(b) (30 points) Perform the following operations on the raw, normalized and standardized datasets respectively.

i. (3 points) Make a 2D plot of the values and label the axes (area A should be x-axis and kernel width should be y-axis). Compare the three plots.




Normalized dataset
kernal_width
area_A
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0



Comparison of the three plots :

- In *Raw dataset*, the range of the 2 variables (X and Y axis) – area and kernel width are different.
- One ranges from 0 to 20 and the other from 0 to 4.
- So if we want to make any assumptions, it' hard since they both are not on same scale.

- In the *Normalized dataset*, the range of the 2 variable is same i.e. 0 to 1.
- Here, the scales are same, so comparisons between the 2 variables is doable. This will be helpful when the data magnitude is big and we need to show the percentage of variation explained by x with y. i.e. visualizing using a scatter plot with scaled values.

- In the *Standardized dataset*, The range for the 2 variables is same, but they range from -1.5 to 2. In this case also, comparisons are easy. It explains the dataset in Z-scores and each standard deviation from the mean will show the spread of the dispersion.

ii. (3 points) Compute the mean of area A and kernel width values. Consider this point as P.

P_raw: [14.847523809523816, 3.258604761904762]
P_normalized: [0.4020324654885564, 0.44804330855649466]
P_standardized: [-5.3925118338936174e-17, -3.1720657846433045e-16]

iii. (9 points) Compute the distance between P and the 210 data points using the following distance measures: 1) Euclidean distance, 2) Mahalanobis distance, 3) City block metric, 4) Minkowski metric (for r=3), 5) Chebyshev distance, 6) Cosine distance and 7) Canberra distance.

There are 7 measure of distance applied on 3 kinds of datasets and each dataset has 210 points
The result is too long, and it can be viewed as part of program output.

iv. (3 points) For each distance measure, identify the 10 points from the dataset that are the closest to the point P from (ii). (You are allowed to use any package functions to calculate the distances.)

Raw dataset sorting

Euclidean distance

```
  point_x point_y       euc
49   14.86   3.258  0.0124908
38    14.8   3.288  0.0558802
48   14.79   3.291  0.0660185
1    14.88   3.333  0.0811748
6    14.69   3.259   0.157524
24   15.01   3.245   0.163045
57   14.92   3.412   0.169655
47   14.99   3.377   0.185248
55   15.03   3.212   0.188334
34   15.05   3.328   0.214038
```

Mahalanobis distance :

```
  point_x point_y      maha
49   14.86   3.258  0.0972761
38    14.8   3.288   0.150155
48   14.79   3.291   0.207978
18    14.7   3.466   0.438574
```

```
1    14.88   3.333   0.873564
55   15.03   3.212    1.10586
24   15.01   3.245    1.21825
6    14.69   3.259    1.28765
15   14.59   3.333    1.51813
57   14.92   3.412    1.84215
```

City block distance :

```
  point_x point_y cityblock
49   14.86   3.258  0.013081
38    14.8   3.288  0.076919
48   14.79   3.291  0.089919
1    14.88   3.333  0.106871
6    14.69   3.259  0.157919
24   15.01   3.245  0.176081
57   14.92   3.412  0.225871
55   15.03   3.212  0.229081
47   14.99   3.377  0.260871
34   15.05   3.328  0.271871
```

Minskowki distance:

```
  point_x point_y    minskow
49   14.86   3.258  0.0124767
38    14.8   3.288  0.0510105
48   14.79   3.291  0.0607627
1    14.88   3.333  0.0764035
6    14.69   3.259   0.157524
57   14.92   3.412   0.158609
24   15.01   3.245   0.162508
47   14.99   3.377   0.165727
55   15.03   3.212   0.183484
34   15.05   3.328   0.205158
```

Chebishev distance:

```
  point_x point_y       cheb
49   14.86   3.258  0.0124762
38    14.8   3.288  0.0475238
48   14.79   3.291  0.0575238
1    14.88   3.333  0.0743952
47   14.99   3.377   0.142476
57   14.92   3.412   0.153395
6    14.69   3.259   0.157524
24   15.01   3.245   0.162476
55   15.03   3.212   0.182476
```

```
34   15.05   3.328   0.202476
```

Cosine distance:

```
   point_x point_y       cosine
68    14.37   3.153  1.41763e-09
49    14.86   3.258  2.30338e-08
43     15.5   3.396  6.38334e-08
134   15.56   3.408  9.14085e-08
4    16.14   3.562   6.7999e-07
22    15.88   3.507  8.56893e-07
46    15.36   3.393  9.26412e-07
20    14.16   3.129  1.02786e-06
34    15.05   3.328  1.25065e-06
31    15.49   3.371  1.55371e-06
```

Canberra distance:

```
  point_x point_y     canberra
49   14.86   3.258  0.000512771
6    14.69   3.259   0.00539365
38    14.8   3.288   0.00609311
48   14.79   3.291   0.00688705
24   15.01   3.245    0.0075336
1    14.88   3.333    0.0123788
55   15.03   3.212      0.01331
10   15.26   3.242    0.0162544
50   14.43   3.272     0.016312
34   15.05   3.328    0.0173082
```

Normalized dataset sorting

Euclidean distance

```
     point_x   point_y        euc
49   0.403211  0.447612  0.00125449
6    0.387158  0.448325  0.0148774
24   0.417375  0.438346   0.0181499
38   0.397545  0.468995   0.0214269
48   0.396601  0.471133   0.0237203
55   0.419263  0.414825   0.0374211
50   0.362606  0.457591   0.0405658
10   0.440982  0.436208    0.040708
132  0.452314   0.45474    0.050725
34   0.421152  0.497505   0.0530288
```

Mahalanobis distance :

```
      point_x   point_y         maha
49   0.403211  0.447612  2.47958e-05
103  0.811143  0.749109  0.000104177
24   0.417375  0.438346  0.000355363
6    0.387158  0.448325   0.00047176
50   0.362606  0.457591   0.00064146
52   0.368272  0.344262  0.000663547
55   0.419263  0.414825  0.000664847
159 0.0849858 0.0463293  0.000665594
78    0.78848  0.719173   0.00071439
39   0.348442  0.476123  0.000791561
```

City block distance :

```
     point_x   point_y   cityblock
49  0.403211  0.447612  0.00160916
6   0.387158  0.448325   0.0151565
24  0.417375  0.438346   0.0250393
38  0.397545  0.468995   0.0254393
48  0.396601  0.471133   0.0285219
50  0.362606  0.457591   0.0489738
55  0.419263  0.414825   0.0504489
10  0.440982  0.436208   0.0507848
1   0.405099  0.501069   0.0560925
132 0.452314   0.45474   0.0569776
```

Minskowki distance:

```
     point_x   point_y     minskow
49  0.403211  0.447612  0.00119704
6   0.387158  0.448325   0.0148748
24  0.417375  0.438346    0.016538
38  0.397545  0.468995   0.0210201
48  0.396601  0.471133   0.0231897
55  0.419263  0.414825   0.0346966
10  0.440982  0.436208   0.0393105
50  0.362606  0.457591    0.039612
0   0.440982  0.486101   0.0485182
132 0.452314   0.45474   0.0503206
```

Chebishev distance:

```
    point_x   point_y        cheb
49 0.403211  0.447612  0.00117811
6  0.387158  0.448325   0.0148748
24 0.417375  0.438346   0.0153424
38 0.397545  0.468995   0.0209517
```

```
48  0.396601  0.471133    0.02309
55  0.419263  0.414825   0.0332179
10  0.440982  0.436208   0.0389496
0   0.440982  0.486101   0.0389496
50  0.362606  0.457591   0.0394262
5   0.357885  0.486101   0.0441477
```

Cosine distance:

```
     point_x   point_y      cosine
49   0.403211  0.447612  1.86894e-06
140  0.234183  0.259444  4.33199e-06
162  0.137866  0.154669  5.45275e-06
36   0.529745   0.59444  5.82258e-06
112  0.806421  0.906629  9.49235e-06
0    0.440982  0.486101  1.48149e-05
139  0.532578  0.600143  1.51509e-05
66   0.354108  0.399145  1.59397e-05
130  0.728045  0.801853  1.72083e-05
9    0.552408  0.623664  2.07436e-05
```

Canberra distance:

```
     point_x   point_y    canberra
49   0.403211  0.447612  0.00194432
6    0.387158  0.448325   0.0191624
38   0.397545  0.468995   0.0284596
24   0.417375  0.438346   0.0296636
48   0.396601  0.471133   0.0319218
55   0.419263  0.414825   0.0594773
10   0.440982  0.436208   0.0595872
1    0.405099  0.501069   0.0596683
50   0.362606  0.457591   0.0621043
132  0.452314   0.45474   0.0662709
```

Standardized Dataset sorting

Euclidean distance

```
       point_x     point_y    euc
49   0.00429804 -0.00160493  0.005
6    -0.0542668  0.00104889  0.054
24    0.0559729  -0.0361047  0.067
38   -0.0163719   0.0780099   0.08
48   -0.0198169   0.0859714  0.088
55    0.0628629   -0.123681  0.139
50    -0.143837   0.0355487  0.148
```

```
10     0.142098  -0.0440662  0.149
132    0.183438   0.0249334  0.185
34    0.0697529    0.184163  0.197
```

Mahalanobis distance :

```
       point_x     point_y         maha
49   0.00429804 -0.00160493  6.69905e-06
0      0.142098    0.141702  3.96731e-05
6    -0.0542668  0.00104889  4.10266e-05
14    -0.381541   -0.383756  0.000601078
66    -0.174842   -0.182065  0.000929975
53    -0.178287   -0.158181   0.00224884
21    -0.254076    -0.24045    0.0023169
24    0.0559729  -0.0361047   0.00235073
132    0.183438   0.0249334   0.00279452
67    -0.288526   -0.266988   0.00406624
```

City block distance :

```
       point_x     point_y cityblock
49   0.00429804 -0.00160493     0.006
6    -0.0542668  0.00104889     0.055
24    0.0559729  -0.0361047     0.092
38   -0.0163719   0.0780099     0.094
48   -0.0198169   0.0859714     0.106
50    -0.143837   0.0355487     0.179
10     0.142098  -0.0440662     0.186
55    0.0628629   -0.123681     0.187
132    0.183438   0.0249334     0.208
1      0.011188    0.197432     0.209
```

Minskowki distance:

```
       point_x     point_y minskow
49   0.00429804 -0.00160493   0.004
6    -0.0542668  0.00104889   0.054
24    0.0559729  -0.0361047   0.061
38   -0.0163719   0.0780099   0.078
48   -0.0198169   0.0859714   0.086
55    0.0628629   -0.123681   0.129
10     0.142098  -0.0440662   0.143
50    -0.143837   0.0355487   0.145
0      0.142098    0.141702   0.179
132    0.183438   0.0249334   0.184
```

Chebishev distance:

```
      point_x     point_y   cheb
49  0.00429804 -0.00160493  0.004
6   -0.0542668  0.00104889  0.054
24   0.0559729  -0.0361047  0.056
38  -0.0163719   0.0780099  0.078
48  -0.0198169   0.0859714  0.086
55   0.0628629   -0.123681  0.124
10    0.142098  -0.0440662  0.142
0     0.142098    0.141702  0.142
50   -0.143837   0.0355487  0.144
5    -0.161062    0.141702  0.161
```

Cosine distance:

```
     point_x   point_y cosine
54  -0.112832  -0.38641  0.007
52  -0.123167  -0.38641   0.01
68  -0.164507 -0.280257  0.065
157 -0.936185  -1.36302  0.092
20  -0.236851 -0.343949  0.093
158  -1.06709  -1.54083  0.094
186  -1.04642  -1.43998  0.104
69  -0.729486 -0.999445  0.105
185  -1.13255  -1.52756  0.108
151 -0.977525  -1.28075  0.115
```

Canberra distance:

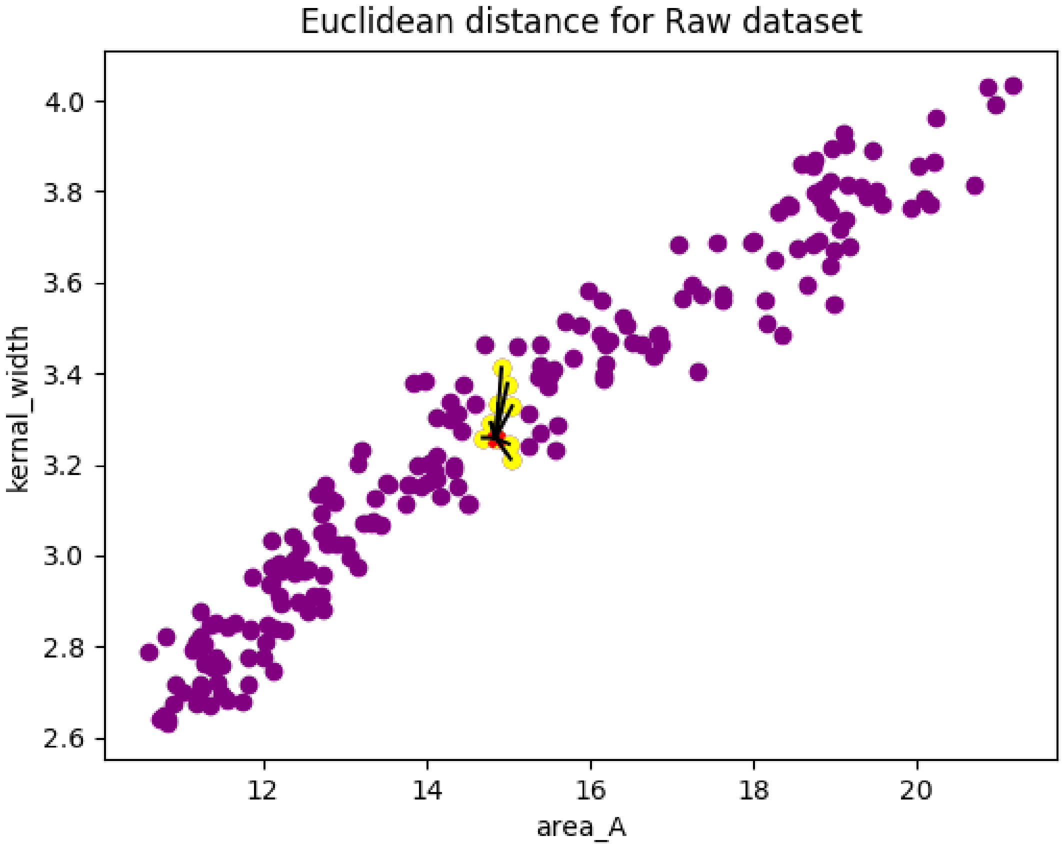
```
     point_x    point_y canberra
49   0.00429804 -0.00160493       2
24    0.0559729  -0.0361047       2
10     0.142098  -0.0440662       2
207   -0.567571  -0.0706045       2
137    0.248893  -0.0732583       2
28    -0.254076  -0.0997966       2
38   -0.0163719   0.0780099       2
48   -0.0198169   0.0859714       2
55    0.0628629   -0.123681       2
53    -0.178287   -0.158181       2
```

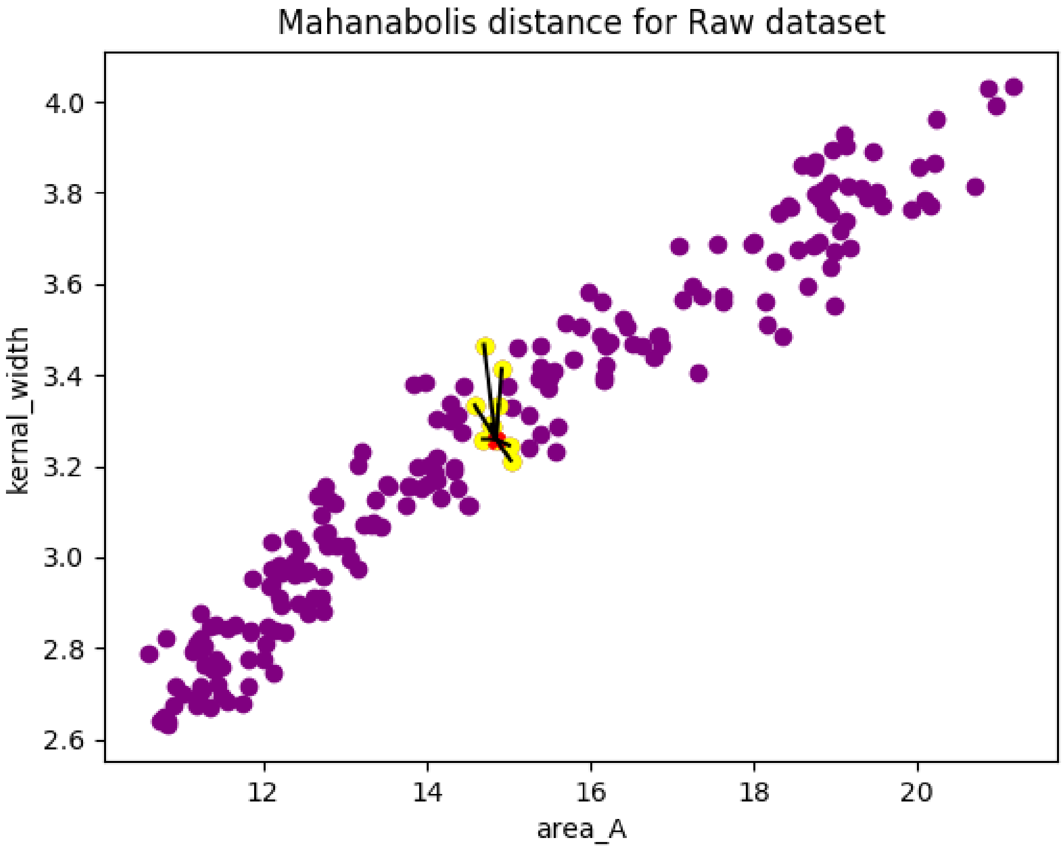
v. (6 points) Create plots, one for each distance measure. Place an 'X' for P and mark the 10 closest points. To mark them, you could place a circle or draw the

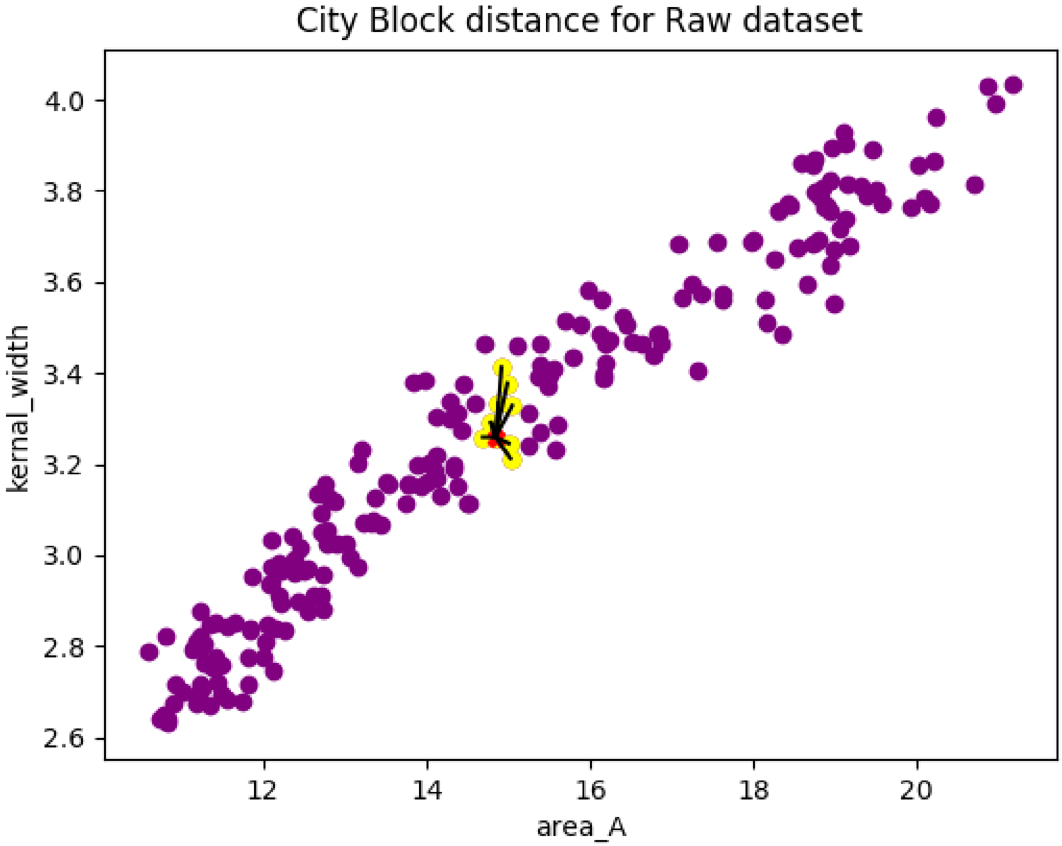
line between these closest neighbors and the points 'X'. Make sure the points can be uniquely identified.

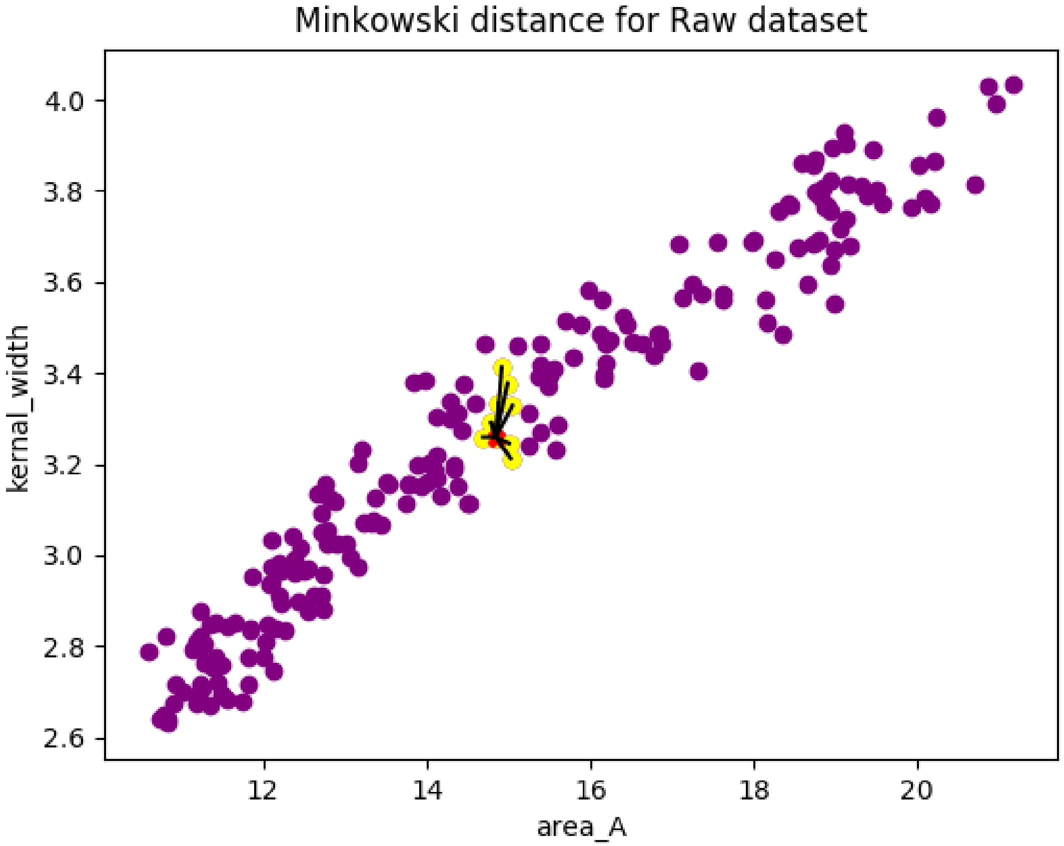
All the graphs are stored in the folder 21 graphs which is zipped along with this document.

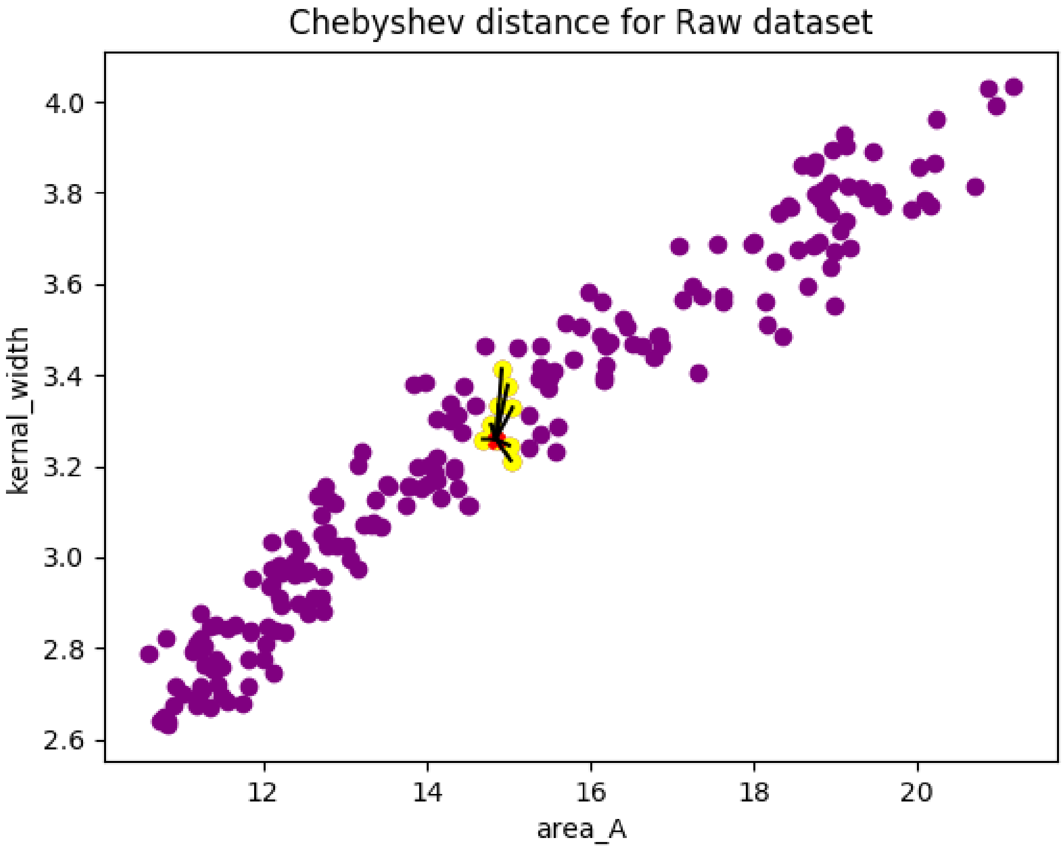
Graphs for Raw Data :

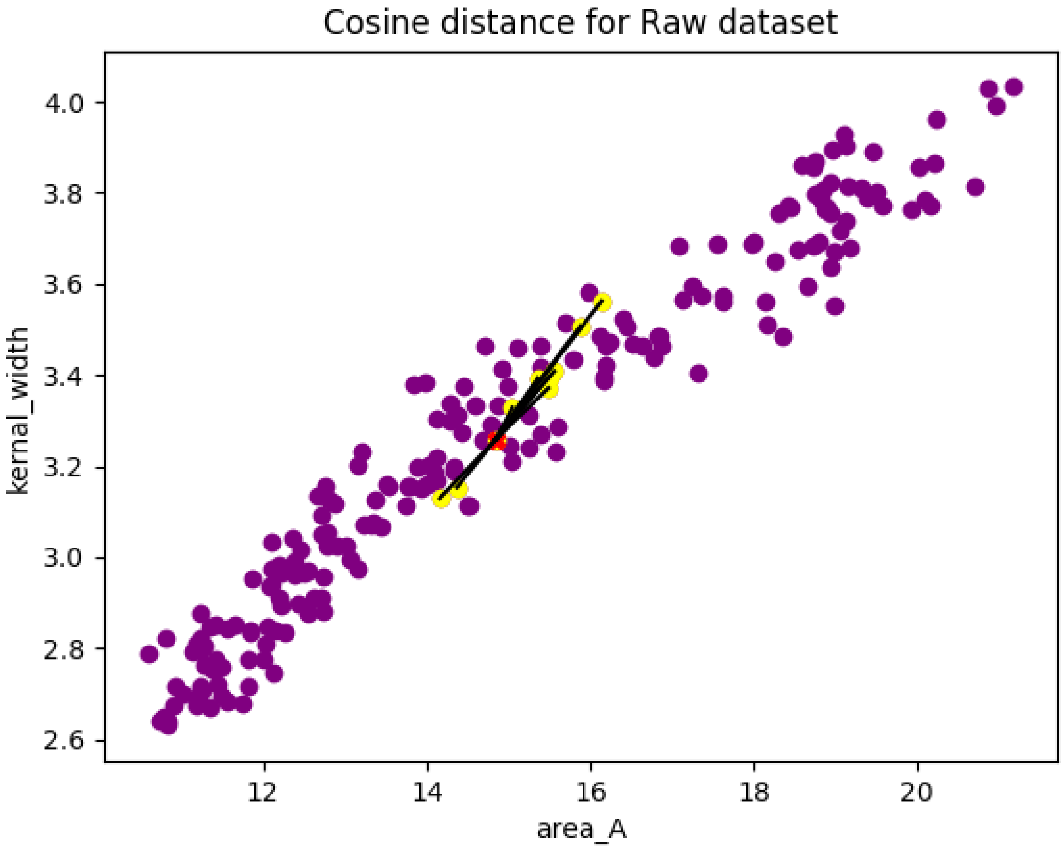


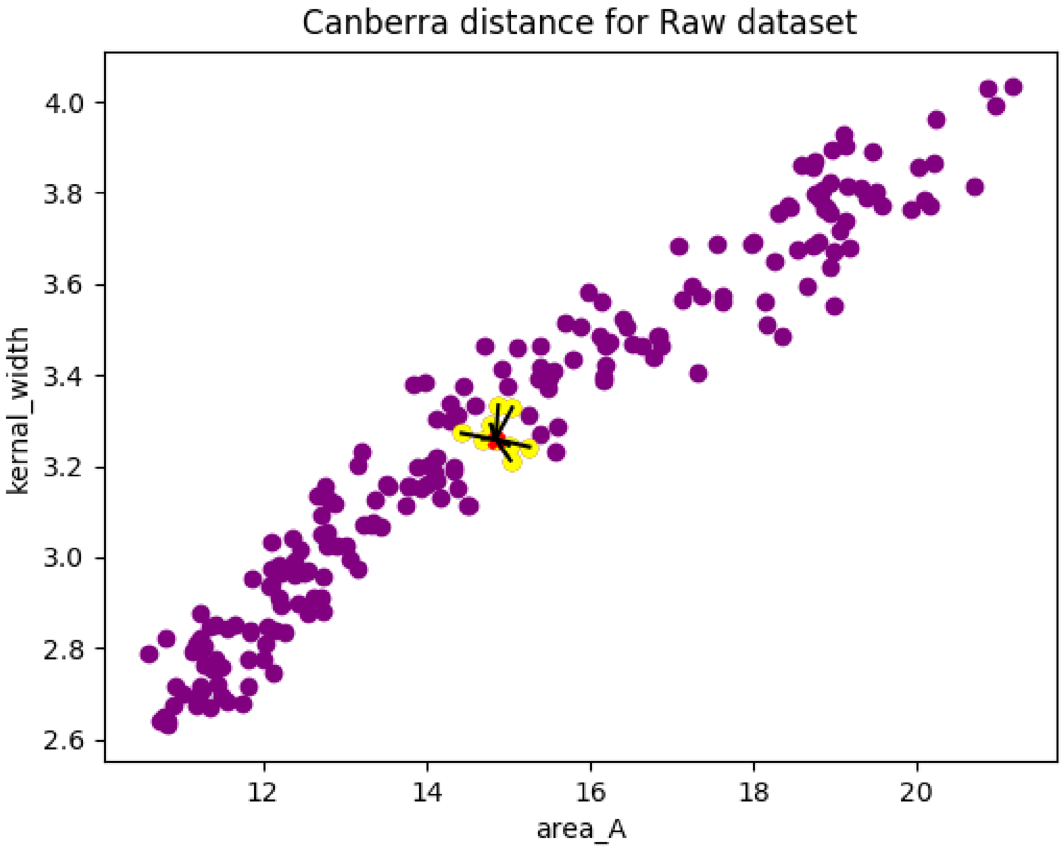

Mahanabolis distance for Raw dataset
kernal_width
4.0
3.8
3.6
3.4
3.2
3.0
2.8
2.6
12
14
16
18
20
area_A


City Block distance for Raw dataset
kernal_width
area_A
4.0
3.8
3.6
3.4
3.2
3.0
2.8
2.6
12
14
16
18
20

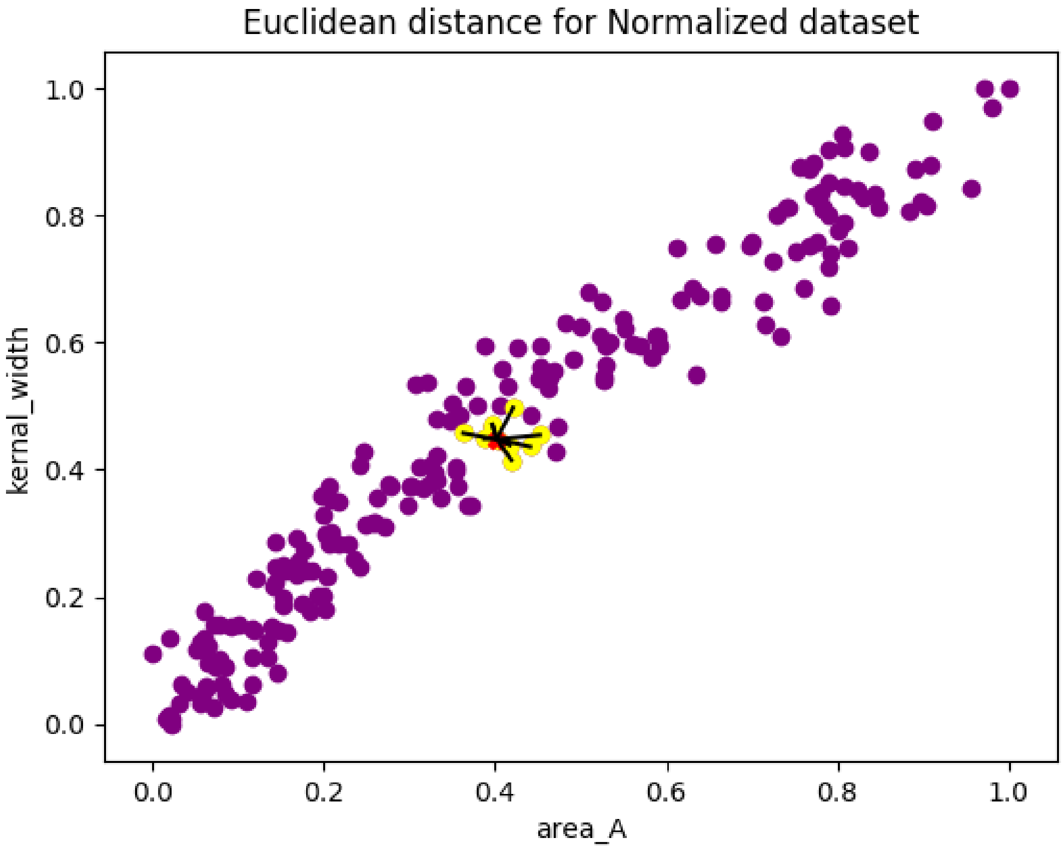

Minkowski distance for Raw dataset
kernal_width
4.0
3.8
3.6
3.4
3.2
3.0
2.8
2.6
12
14
16
18
20
area_A

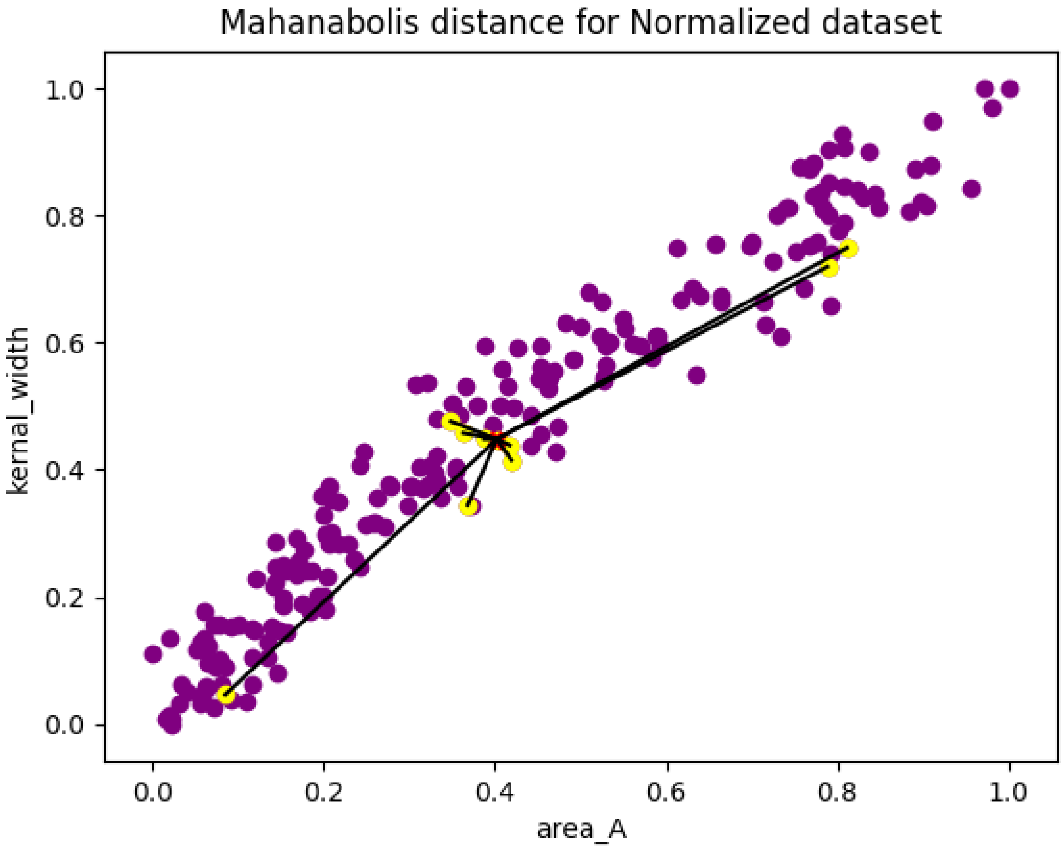

Chebyshev distance for Raw dataset
kernal_width
4.0
3.8
3.6
3.4
3.2
3.0
2.8
2.6
12
14
16
18
20
area_A

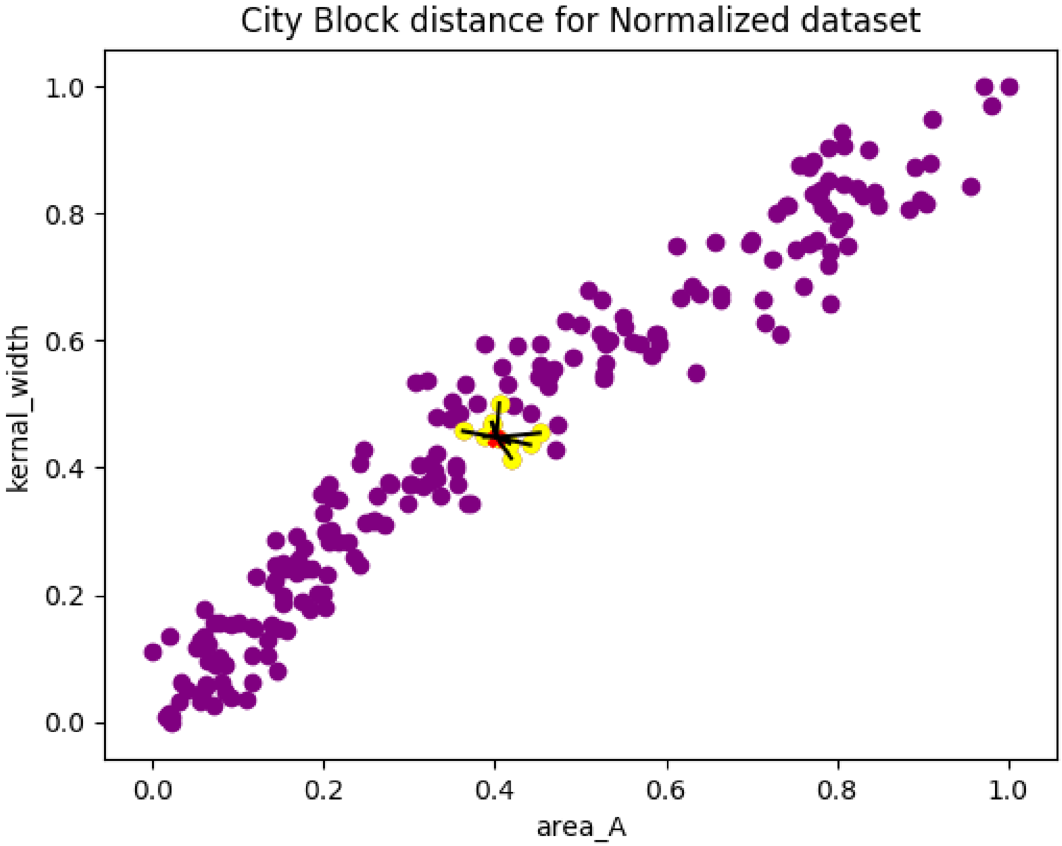

Cosine distance for Raw dataset
kernal_width
4.0
3.8
3.6
3.4
3.2
3.0
2.8
2.6
12
14
16
18
20
area_A

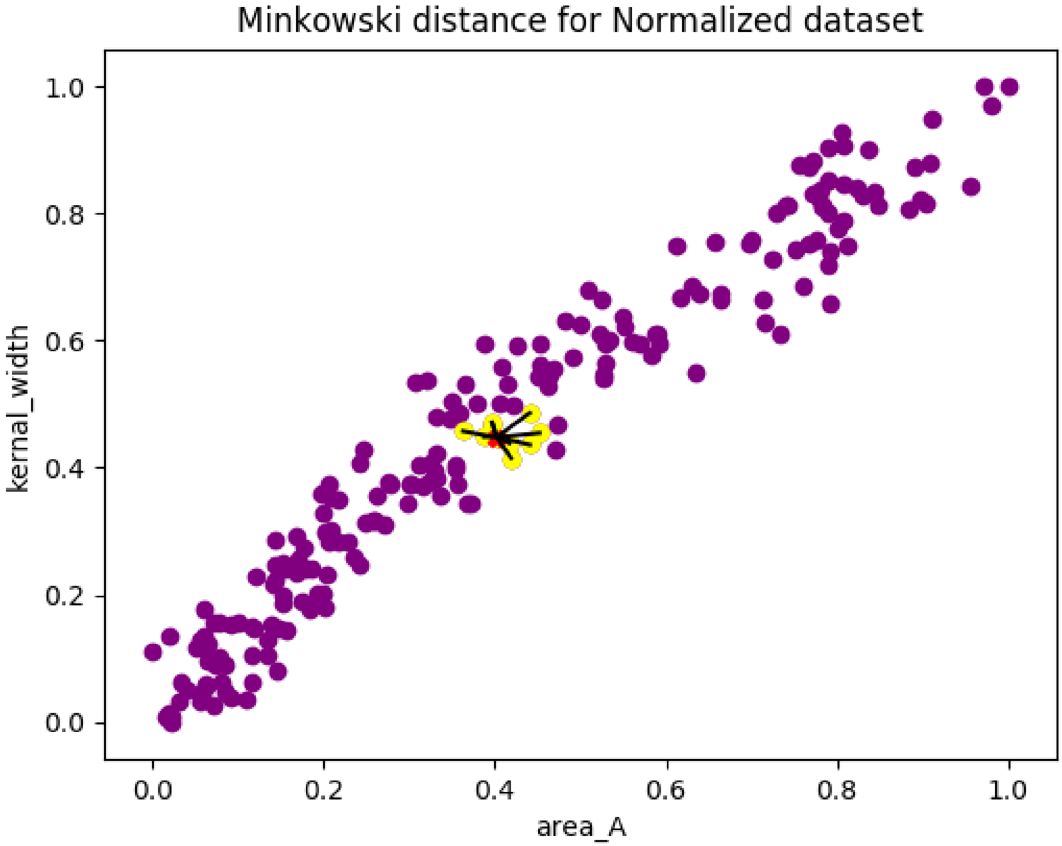


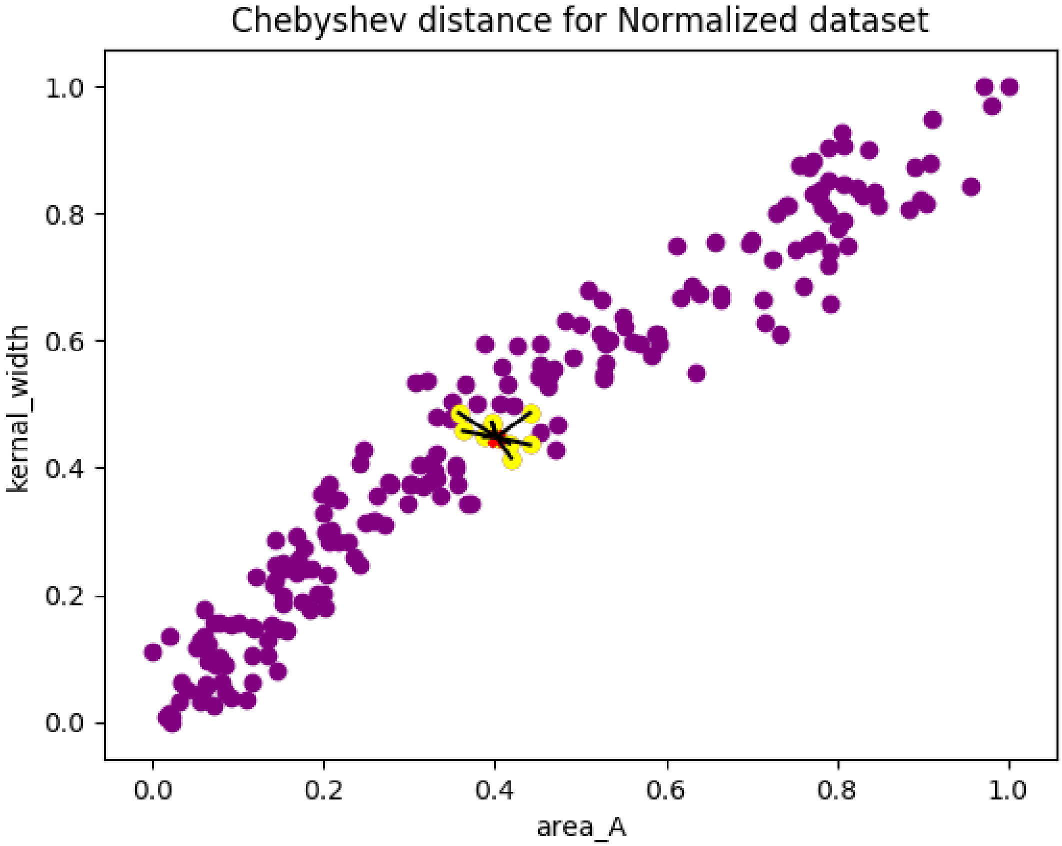
Graphs for Normalized Dataset :

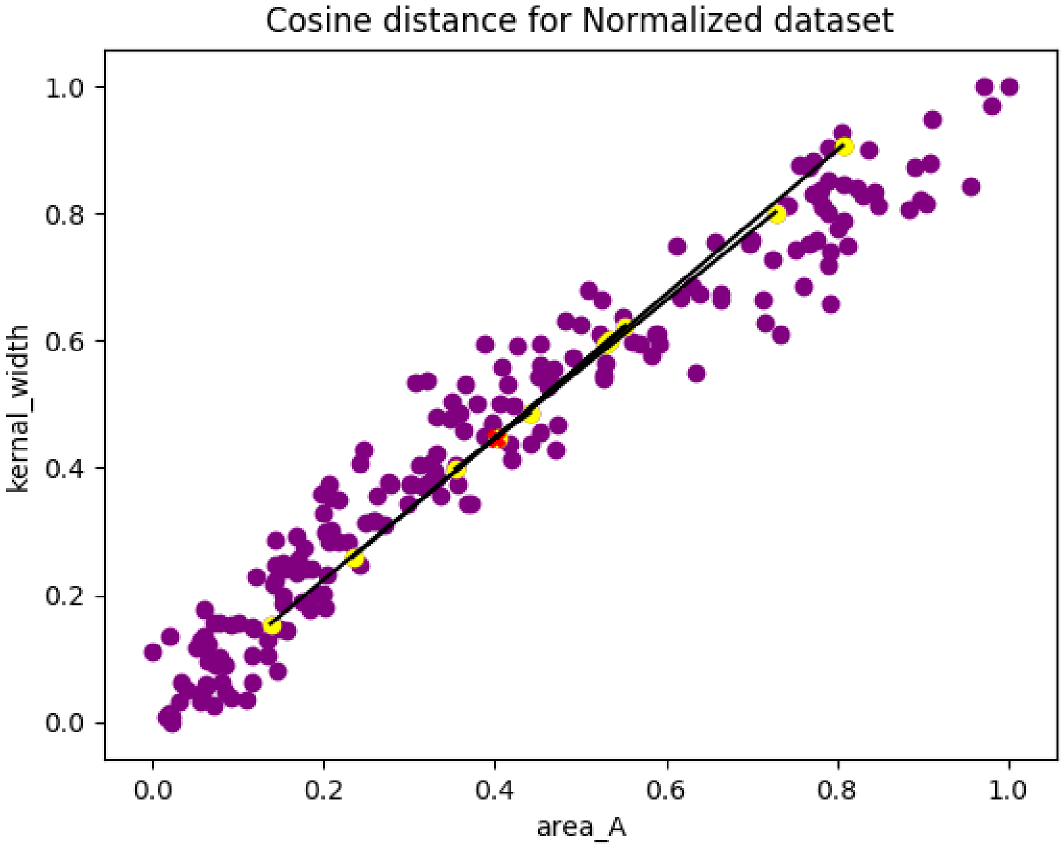

Euclidean distance for Normalized dataset
1.0
0.8
0.6
0.4
0.2
0.0
kernal_width
0.0
0.2
0.4
0.6
0.8
1.0
area_A

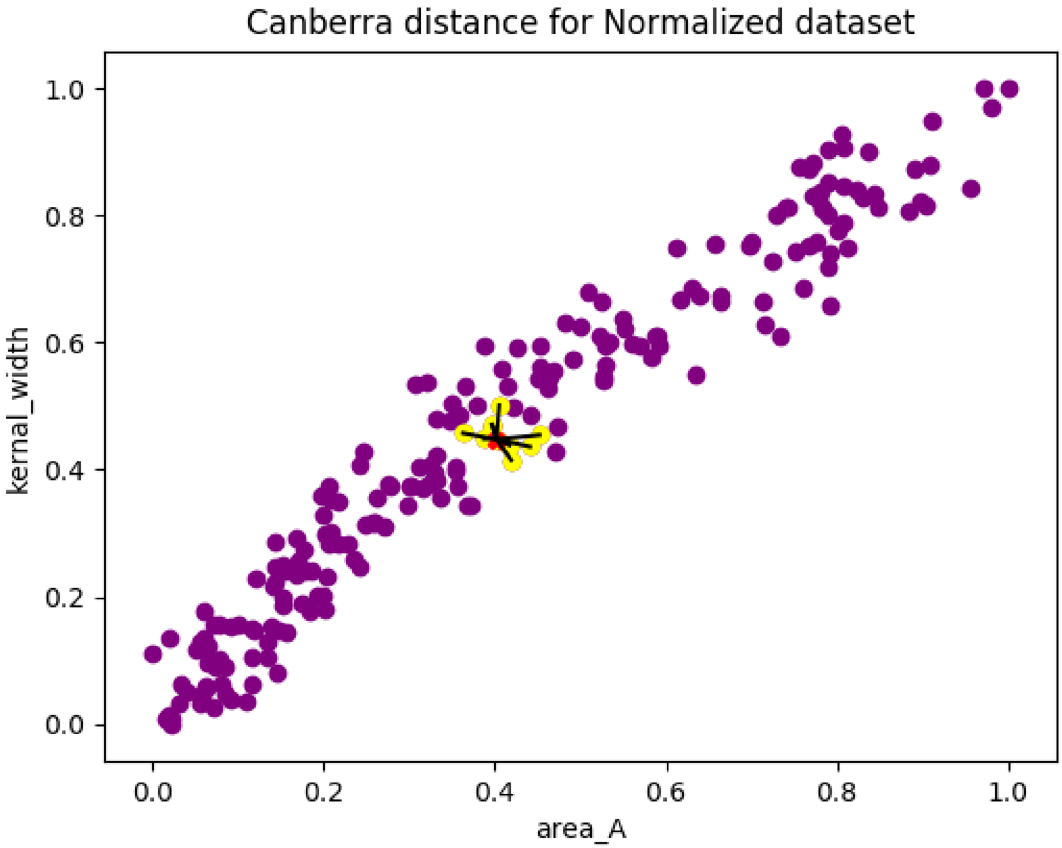

Mahanabolis distance for Normalized dataset
kernal_width
area_A
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0


City Block distance for Normalized dataset
kernal_width
area_A
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0

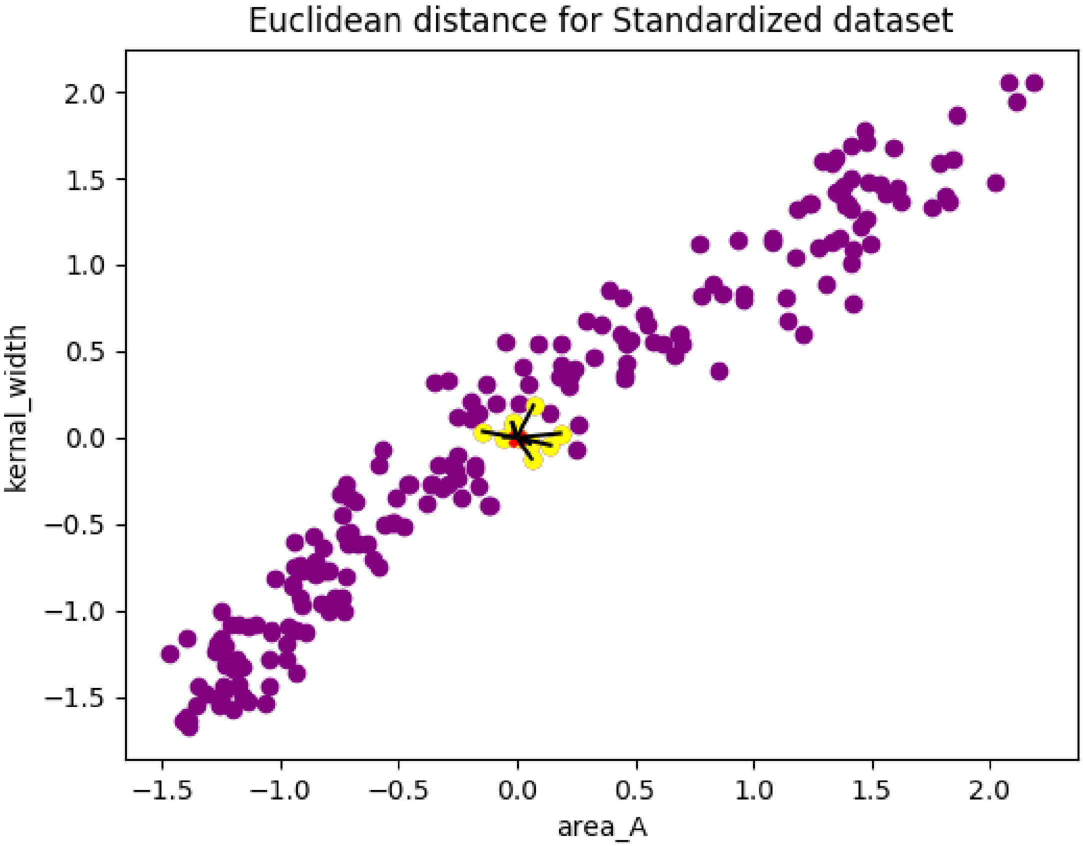

Minkowski distance for Normalized dataset
kernal_width
area_A
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0

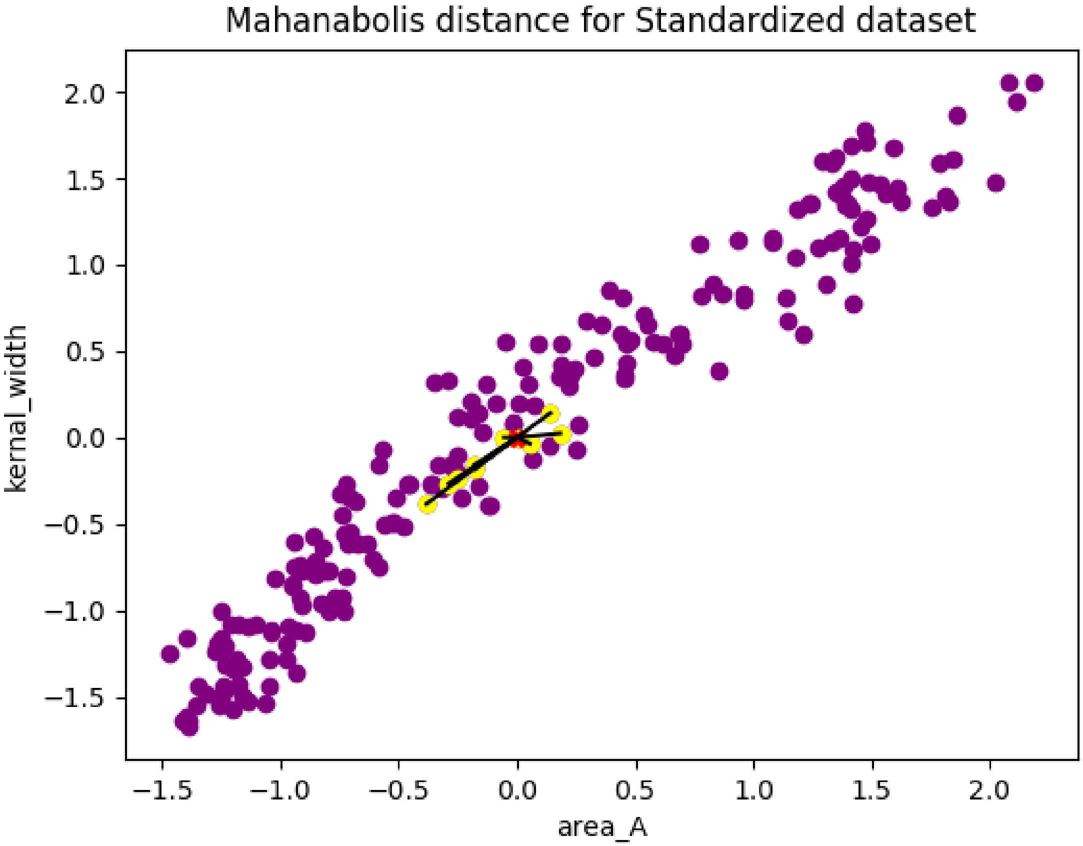

Chebyshev distance for Normalized dataset
1.0
0.8
0.6
0.4
0.2
0.0
kernal_width
0.0
0.2
0.4
0.6
0.8
1.0
area_A

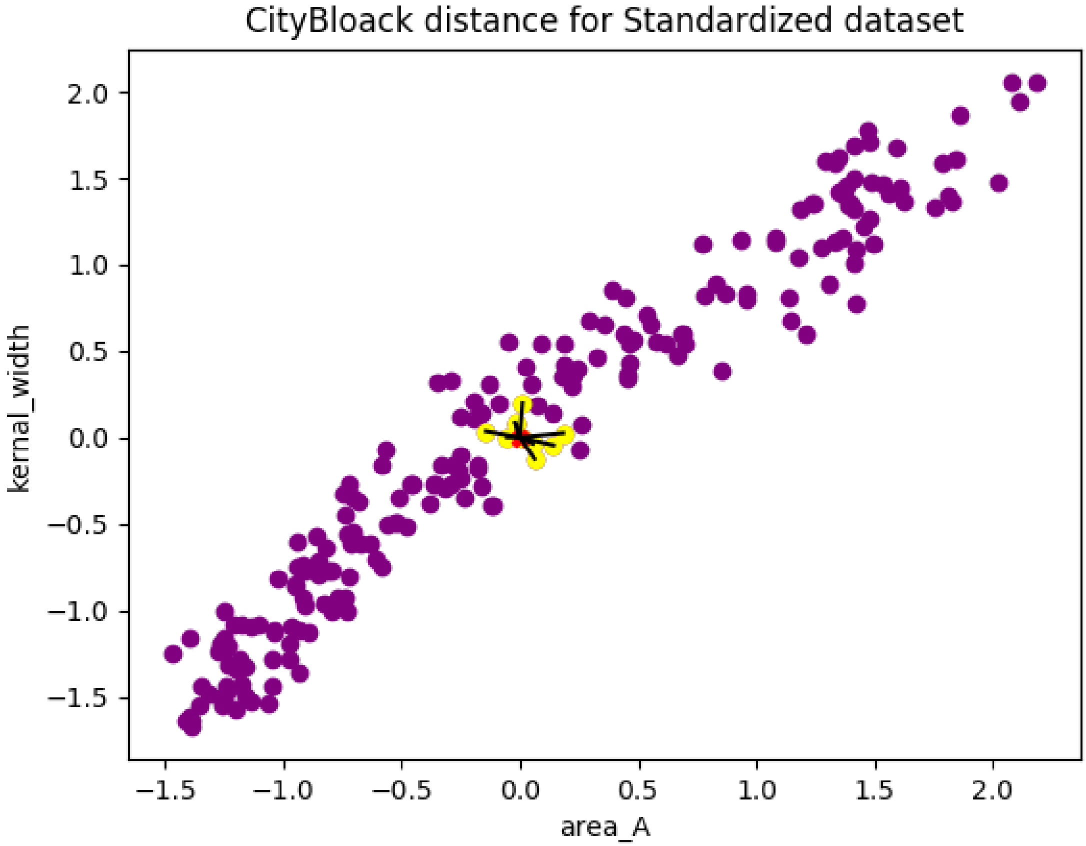

Cosine distance for Normalized dataset
kernal_width
area_A
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0

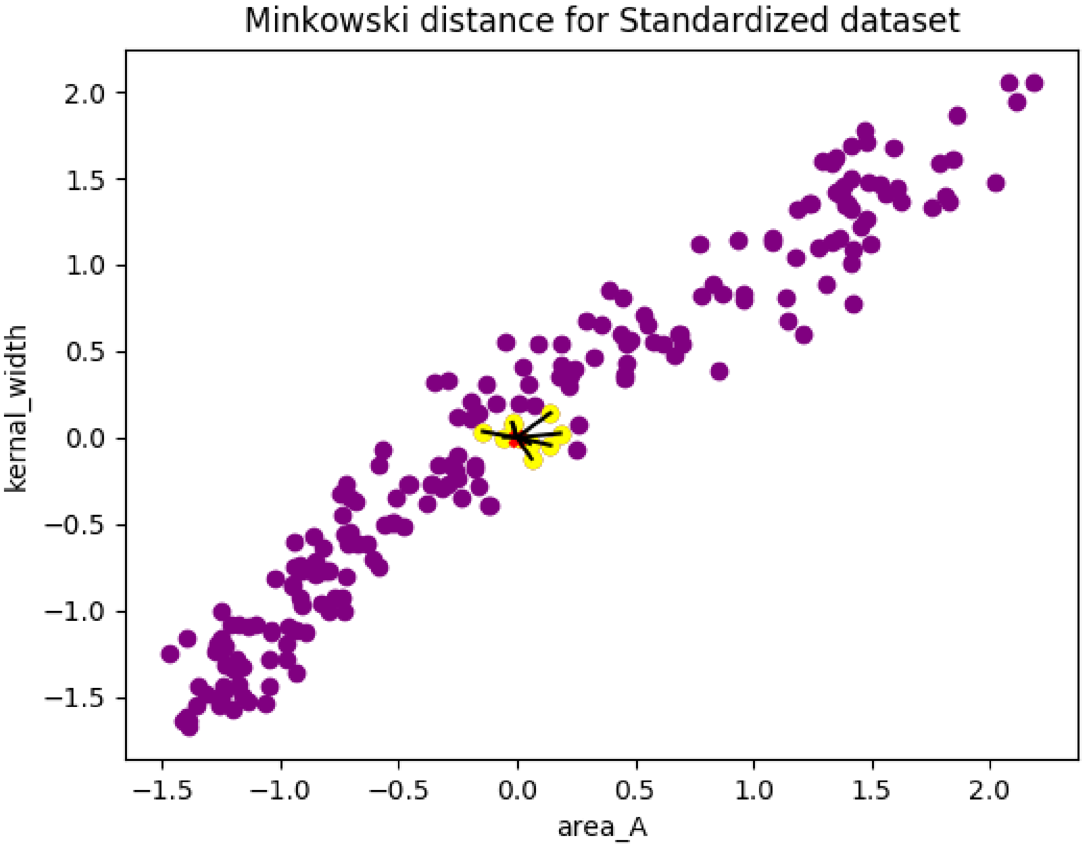


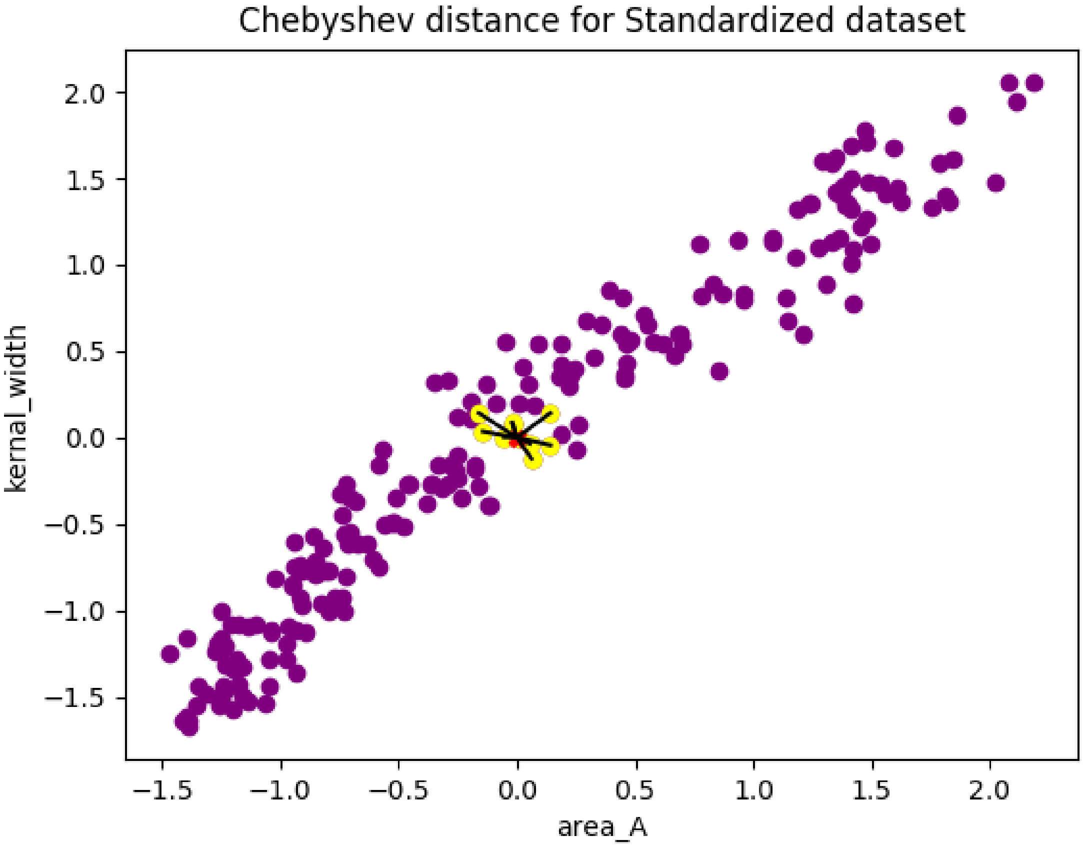
Graphs for Standardized dataset :

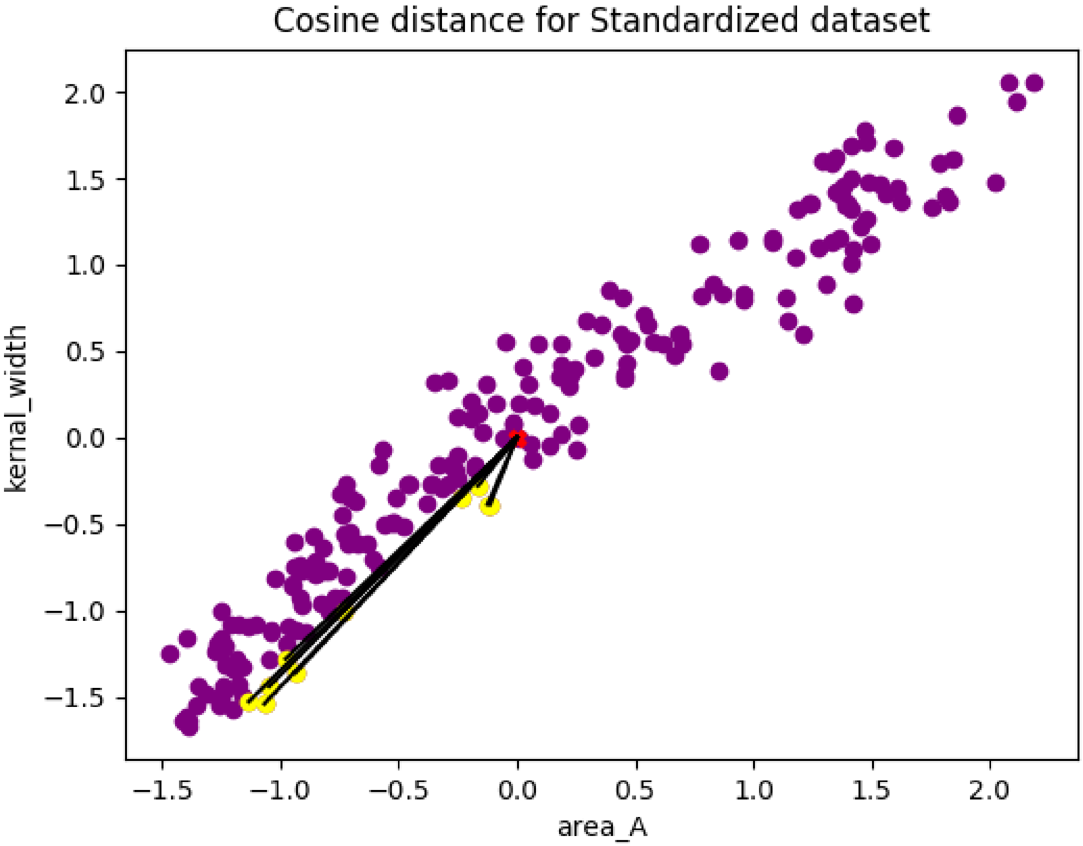

Euclidean distance for Standardized dataset
kernal_width
area_A
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

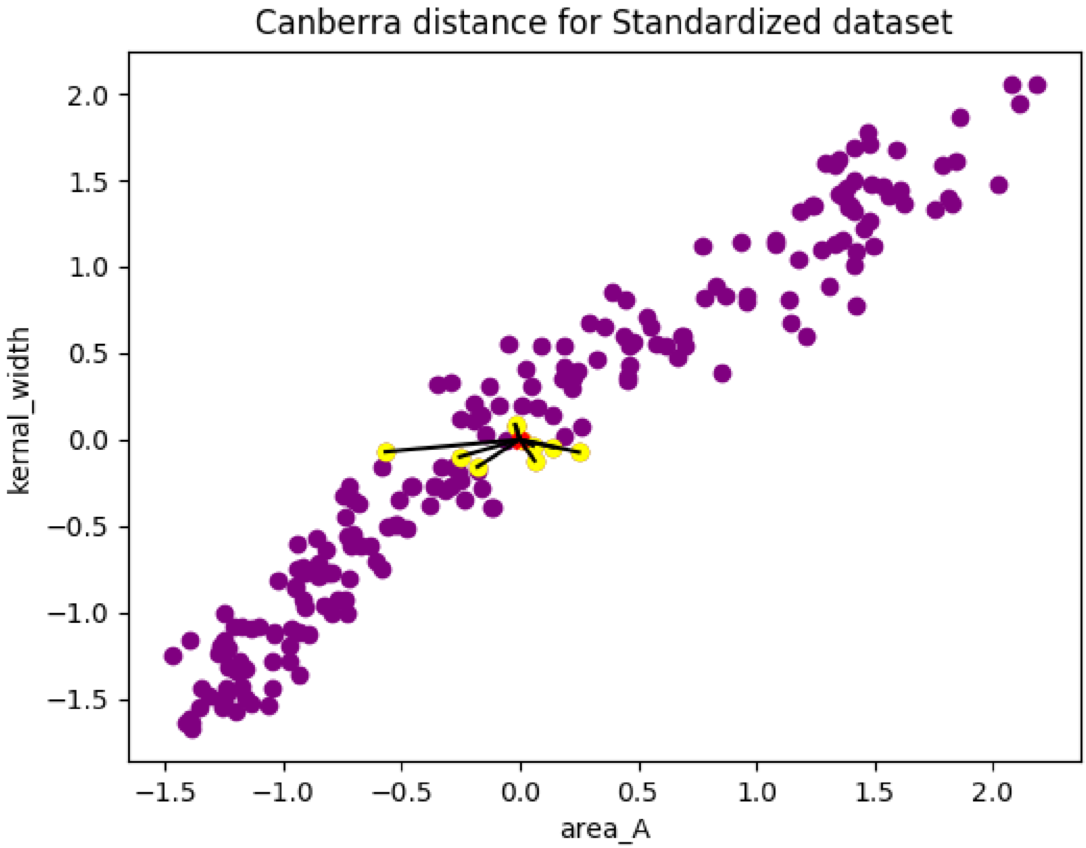

Mahanabolis distance for Standardized dataset
kernal_width
area_A
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


CityBloack distance for Standardized dataset
kernal_width
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
area_A


Minkowski distance for Standardized dataset
kernal_width
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
area_A


Chebyshev distance for Standardized dataset
kernal_width
area_A
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


Cosine distance for Standardized dataset
kernal_width
area_A
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0



vi. (3 points) Verify if the set of points is the same across all the distance measures. If there is any big difference, briefly explain why it is.

For the Raw dataset:

The set of points across all distance measures is almost same but here's the detailed description:

- We can see that cosine measure is a bit different from other measures as cosine considers the direction of the vectors.
- We can observe little difference in the order of points in Chebyshev and Canberra distance as well. But these two differ from the rest only little, while cosine distance has a lot of difference.

For the Normalized dataset:

- While considering 10 nearest points from P, all distance measure pick the same points except Cosine. And due to the magnitude and scale of angle vectors, the values are quite skewed and spread across the datapoints. The orientation angle (polar coordinates) is different from other distance measures which are measured in cartesian coordinates.

- Canberra's top 10 points are not similar with respect to each other. (The nearest points they choose are different)

For the Standardized dataset:

- It's almost the same as Normalized dataset, where the distance measures (except Cosine and Canberra) pick the same nearest points.
- In Canberra, almost for the nearest 10 points to P, the distance is 2 units.
- Cosine measure of standardized is not matching with the points that were picked by in Normalized dataset. This is because the angle of orientation is different for normalized and standardized dataset.

On a general note: Since Mahalanobis distance measures the standard deviation of P from distribution of 210 datasets and Cosine takes into consideration the angle of vectors and Canberra is the numerical distance between a pair of points around origin and it is very significant when the data points are close to (0,0).

vii. (3 points) Reason about your results and state the importance of data transformation in the dataset.

Reason for results:

1. Euclidean distance is the measure of finding the distances based on nearest point using a scale.
2. Whereas for cosine, the angle between the two points should be shortest.
3. We can measure the angle between 2 points by drawing straight lines from the point to the origin and measuring the angle between the 2 straight lines. The angle of orientation is critical, and it don't show the deviation and spread of the dataset. In fact, these 2 points may have long distance (scalar distance) compared to other points.
4. Cosine distance considers magnitude and direction where as other measures have only magnitude between the two points.
5. These being reasons, we can see that the graphs of all distance measures except cosine are similar. Since cosine is not just based on magnitude but includes direction too, the graph is not cluttered near the 'P' point which is mean of the dataset represented by 'x' in the graph. The yellow circles in the graph represent the 10 nearest points. If we observe carefully, for cosine they are spread and not very near to point 'P'.

Importance of data transformation:

- Data transformation is one of the important tasks in data mining.

- If we must compare two attributes in the same dataset whose range, scale and units are different, we can scale them (standardize or normalize or log transformation).
- This way comparison or analyzing one feature with respect to other becomes easy.
- Standardization wouldn't change the shape of the data and it retains the originality.
- To have the same range of values across all the attributes, we can normalize them.
- In the given assignment data, where we used 3 kinds of datasets, we could observe that the graphs of Normalized and Standardized are easier and clearer to interpret since the scale of both the variables is same.
- Both the variables range between 0 an1 for Normalized dataset and between 0 and 2 in standardized dataset. Whereas in raw dataset, the range for area is 0 to 20 and for kernel is 0 to 4.
- The data scientist should know the appropriate type of transformation needed before proceeding with the data analysis and modeling.

6. (21 points) [Xi Yang] In this question, please summarize and explore data in the provided file "hw1q6 data.csv", which comes from the Pima Indians Diabetes Database (https://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/). In this data file, each row indicates the data for a patient. The first 6 columns are features for patients, and the last column "Class" indicates if a patient has diabetes: 1 (diabetic) or 0 (nondiabetic). The specific meaning for each feature is as follows: 1. Glucose: Plasma glucose concentration a 2 hours in an oral glucose tolerance test. 2. BloodPressure: Diastolic blood pressure (mm Hg). 3. SkinThickness: Triceps skinfold thickness (mm). 4. BMI: Body mass index (weight in kg/(height in m)2). 5. DiabetesPedigreeFunction: Diabetes pedigree function. 6. Age: (years). Write code in Matlab, R or Python to perform the following tasks. Please report your outputs and key codes in the document file, and also include your code file (end with .m, .r or .py) in the .zip file.

(a) (1 point) How many diabetic and nondiabetic patients are in the dataset?

The no of diabetic patients are : 268 and no of non diabetic patients are : 500

(b) (2 points) There are missing values in the features which are marked as 0. What is the missing rate (%) for each feature?

The percentage of missing values in :

Glucose (Attribute 1) : 0.65
Blood Pressure (Attribute 2) : 4.55
Skin Thickness (Attribute 3) : 29.55
BMI (Attribute 4) : 1.43
Diabetic Pedigree Function (Attribute 5) : 0.0
Age (Attribute 6) : 0.0

(c) (4 points) Specify two methods for missing data handling and discuss their respective advantages and disadvantages.
Remove the patients (rows) in dataset with missing values, then answer the following questions based on the remaining data:

Two methods for handing missing data :

1) Remove the data point:

Here, we can just delete the row which has the missing values assuming removing such data points wouldn't bias the dataset so much.

Advantages :

- Simplicity of the method and availability of many packages in almost all programming languages
- If the data is missing completely at random, we can delete the row (data point) without effecting out analysis. This is the easiest way to handle missing and does not have any bias.
- If the sample size is large enough after deleting the missing values, the analysis would be proper and effective.
- If each attribute is missing on the same 10 data points, removing them in list wise deletion is easy and doesn't really produce bias.

Disadvantages :

- If the data is not missing at random, and the missing value is depending on other values , removing it will produce a bias in the dataset.
- If after deleting, the sample size shrinks, there might be problem because the statistical analysis is always tightly related to sample size and as such the results of analysis may be wrong.
- It's possible to have only a small percentage of observations missing overall, yet still lose a large part of the sample to list wise deletion.

2) Impute the values :

Here, we can determine the missing value either by finding out the mean of that attribute or finding the most common (mode) of that column and filling it. If the data spread is quite large, then we use median to fill the missing data depending upon the quantile deviations and interquartile deviations in the data set.

Advantages :

- This tries to leverage the bias that would be produced if we remove it completely.
- The dataset size remains the same.
- There are many ways to impute- so it's easy and almost every programming language has methods/packages to make it even more easy.
- Very advantageous if list wise deletion eliminates many data points.

Disadvantages:

- The imputed value maybe not the best value – thus producing little amount of bias.
- Some imputation methods result in biased parameter estimates, such as means and correlations, unless the data are missing completely at random

- The resulting standard error after imputing will be small – because if we calculate it overall and do not recognize that certain data has been imputed and on that dataset will have more error since they itself are estimates.
- The noise maybe increased after imputation

(d) (1 point) How many diabetic and nondiabetic patients are in the remaining data?

The no of diabetic patients: 177 and no of non-diabetic patients: 355

(e) (3 points) Compute the mean, median, standard deviation, range, 25th percentiles, 50th percentiles, 75th percentiles for each feature.

For attribute Glucose, the mean, median, standard deviation, range, 25th percentile, 50th percentile and 75th percentile is shown below:

```
121.030075188   115.0   30.9700776884   143.0   98.75   115.0    141.25
```

For attribute Blood Pressure, the mean, median, standard deviation, range, 25th percentile, 50th percentile and 75th percentile is shown below:

```
71.5056390977    72.0    12.298    86.0    64.0    72.0     80.0
```

For attribute Skin Thickness, the mean, median, standard deviation, range, 25th percentile, 50th percentile and 75th percentile is shown below:

```
29.1823308271     29.0     10.5139822683     92.0     22.0     29.0     36.0
```

For attribute BMI, the mean, median, standard deviation, range, 25th percentile, 50th percentile and 75th percentile is shown below:

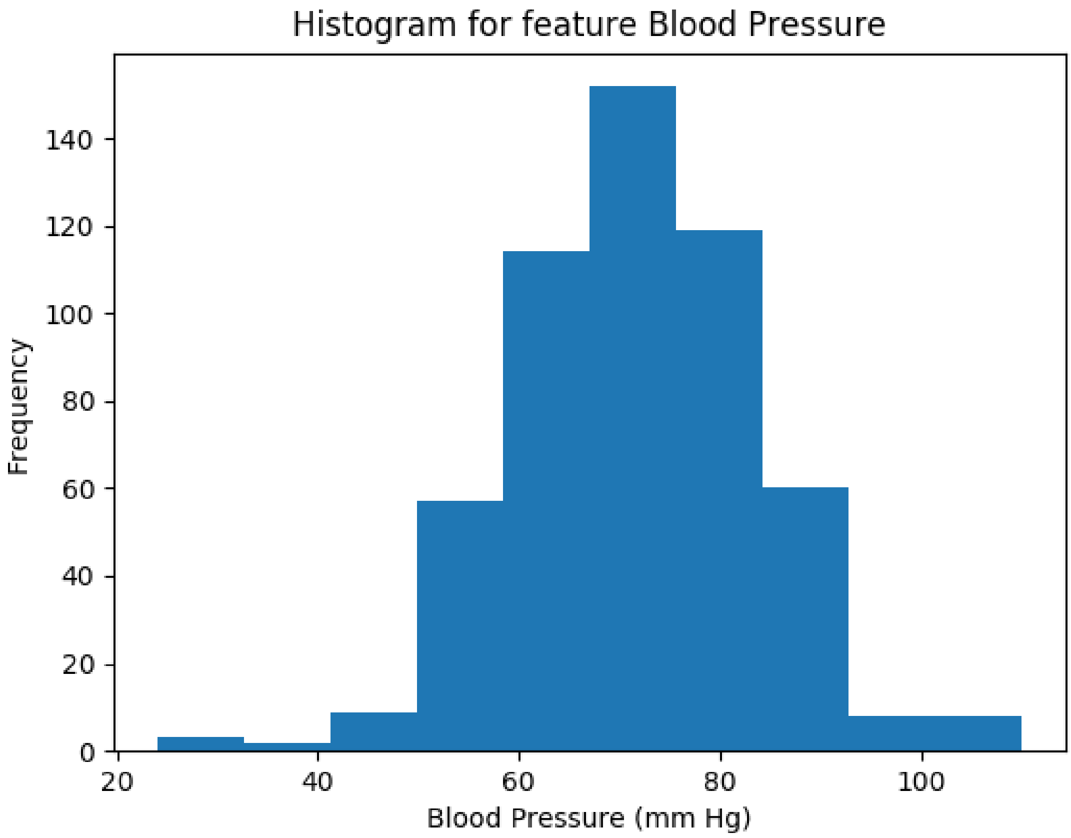
```
32.8902255639   32.8  6.87463863342    48.9     27.875     32.8     36.9
```

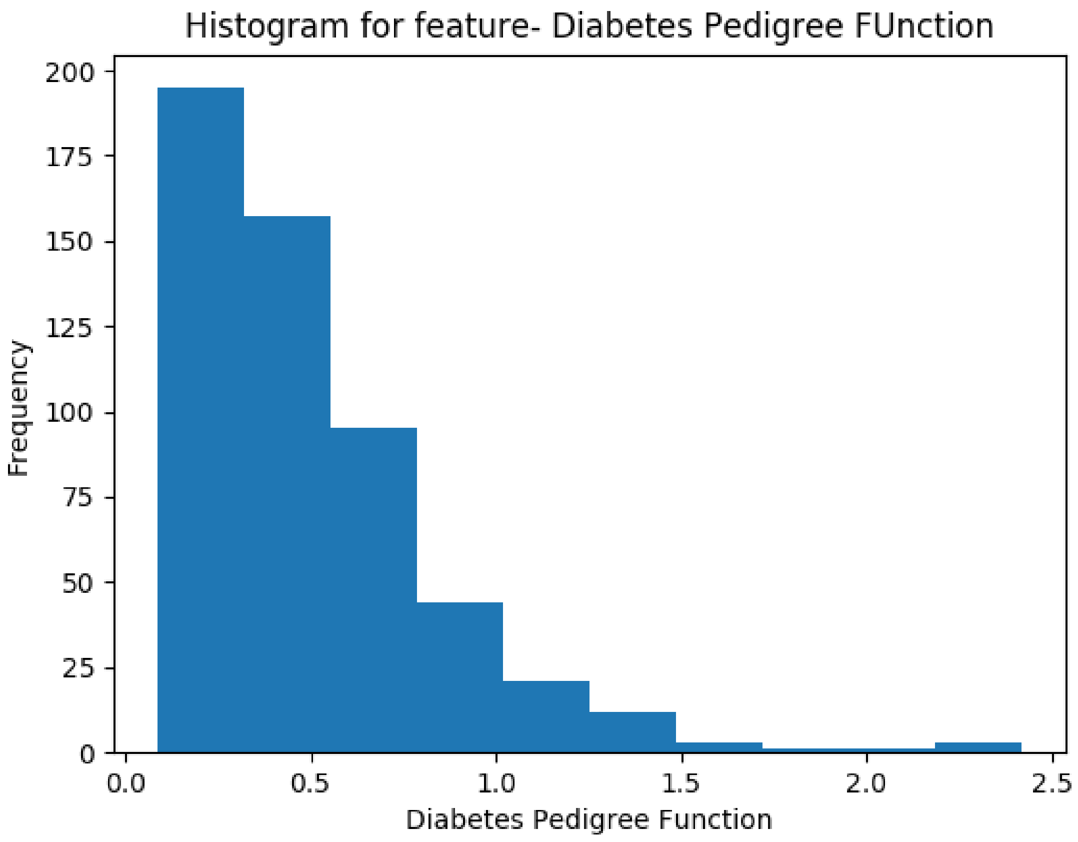
For attribute DiabetesPedigreeFunction, the mean, median, standard deviation, range, 25th percentile, 50th percentile and 75th percentile is shown below:

```
0.502966165414    0.416    0.344222277459    2.335    0.25875    0.416    0.6585
```

For attribute age, the mean, median, standard deviation, range, 25th percentile, 50th percentile and 75th percentile is shown below:
31.6146616541 28.0 10.7514648101 60.0 23.0 28.0 38.0

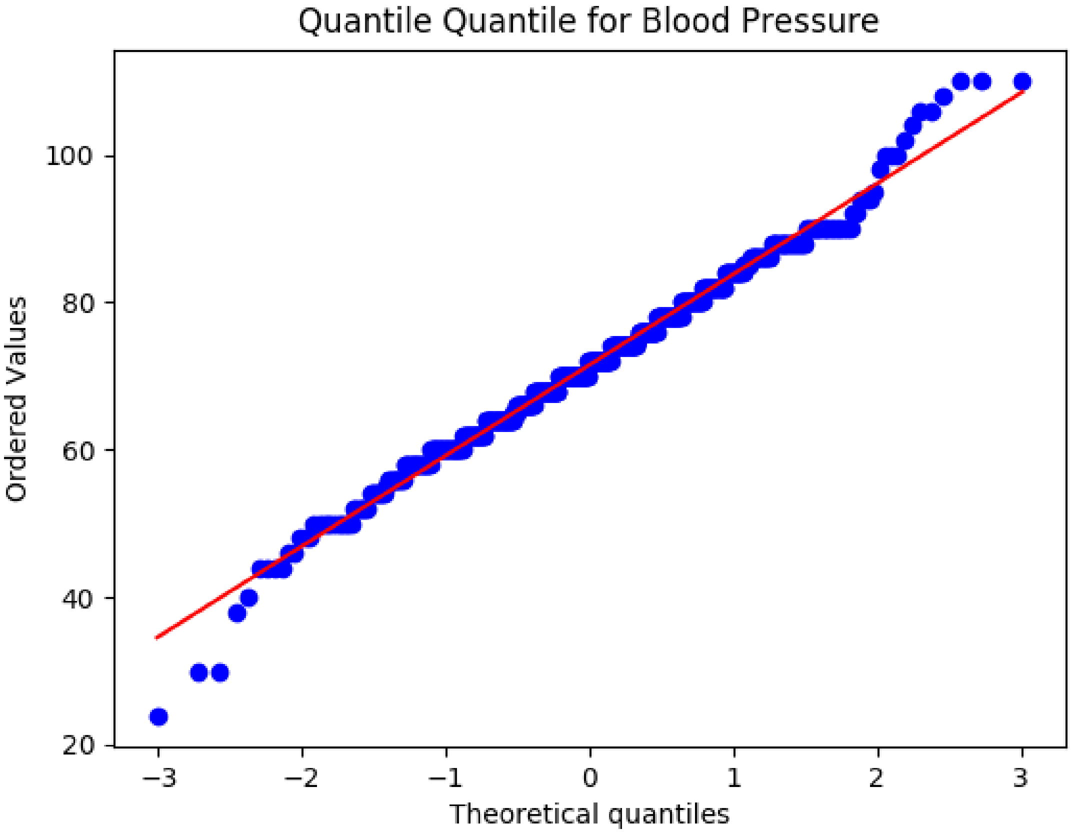
(f) (4 points) Create histogram plot using 10 bins for the two features BloodPressure and DiabetesPedigreeFunction, respectively.



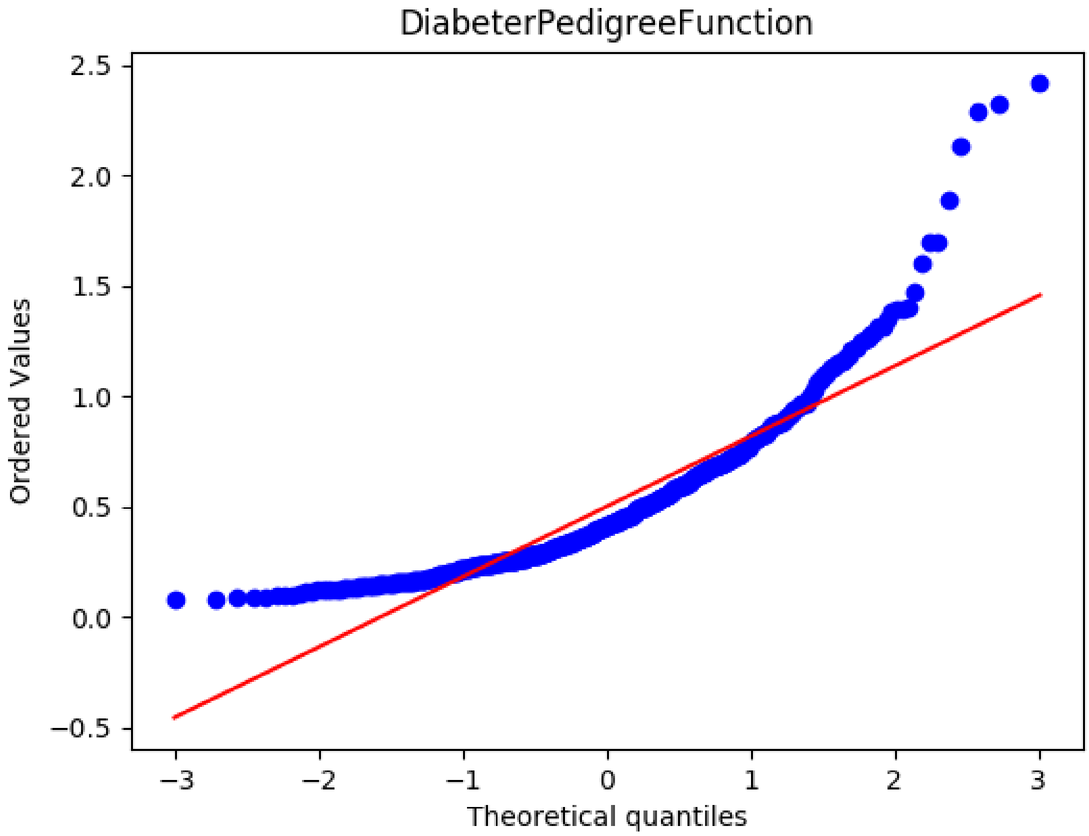


(g) (6 points) Quantile-quantile plot can be used for comparing the distribution of data against the normal distribution. Create quantile-quantile plot for the two features BloodPressure and DiabetesPedigreeFunction, respectively. Give a brief analysis for the two plots.

Quantile-quantile plots are useful to determine whether the variable is normally distributed or not. It is used to compare two probability distributions by plotting their quantiles among each other. To Visually check the normality of data, quantile-quantile plots are built and are better than histograms. In few cases, QQ plots may also be used to check whether 2 samples came from same distribution. As we can see in the graphs below, a 45-degree line is plotted which represents normal distribution. If our variable distribution is normal, it will lie along the normally distributed 45degree line. The more the distribution approaches the normal line, the more normal it is. Few advantages of Q-Q plot is as follows: the sample sizes do not need to be equal. Many distributional aspects can be tested simultaneously. For example, change in scale, symmetry, presence of outliers etc. can be detected in single plot.



The above plot clearly depicts the fat tails of the symmetric data set.



As we can see from the graph, the variable DiabetesPedigreeFunction is not normally distributed. We can see a curve which bypassed the straight line. Thus, we can say that this variable isn’t normally distributed. Whereas the variable blood pressure is normally distributed since most of the points abide by the 45-degree normal line. Clear conclusion is that the data is positively skewed.