Report

HW1

Yasaman Mirmohammad | Data Mining | Fall_2018

Task 1:

Introduction

The ***Iris* flower data** is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper *The use of multiple measurements in taxonomic problems* as an example of linear discriminant analysis . It is sometimes called **Anderson's *Iris* data set** because Edgar Anderson collected the data to quantify the morphologic variation of *Iris*flowers of three related species. Two of the three species were collected in the Gaspé Peninsula "all from the same pasture, and picked on the same day and measured at the same time by the same person with the same apparatus".

- Three flower types (classes):
  - Setosa
  - Virginica
  - Versicolour
- Four (non-class) attributes
  - Sepal width and length
  - Petal width and length

IRIS dataset

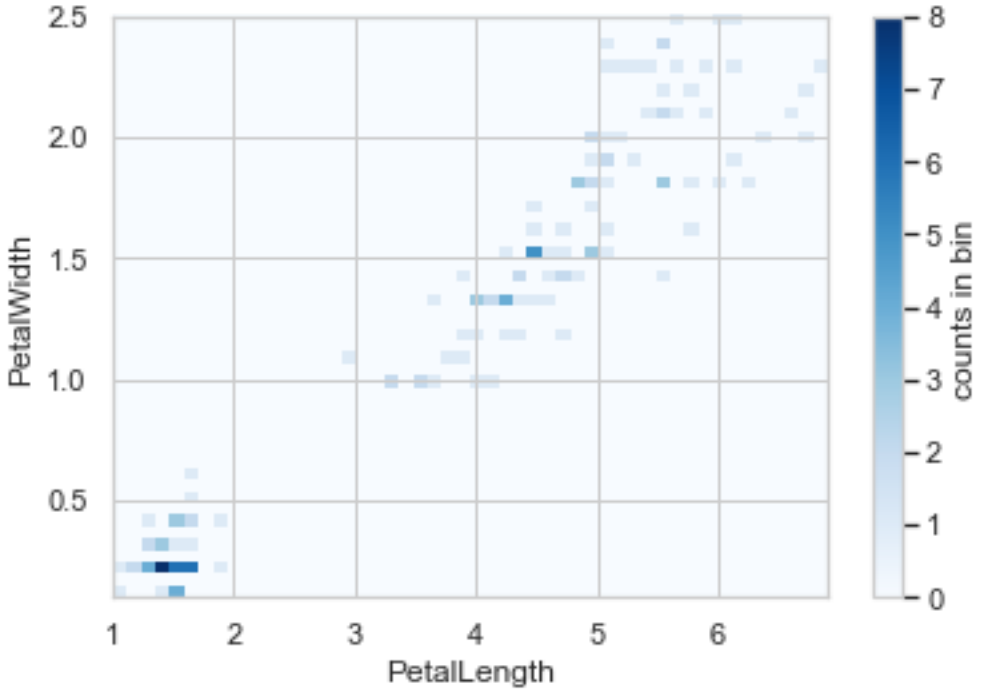
Iris Versicolor

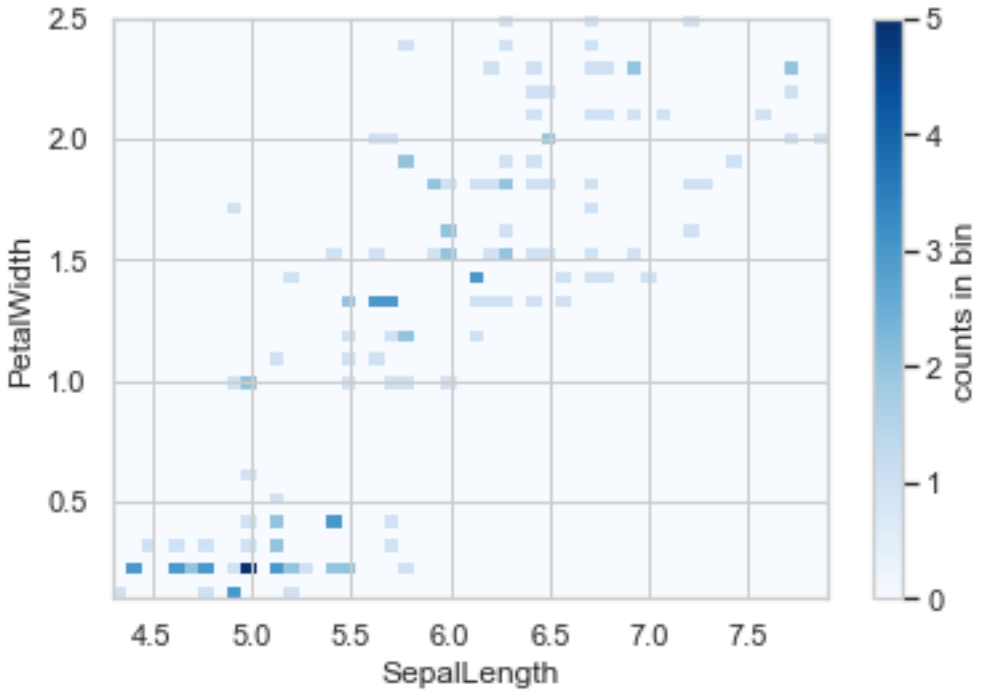
Iris Setosa

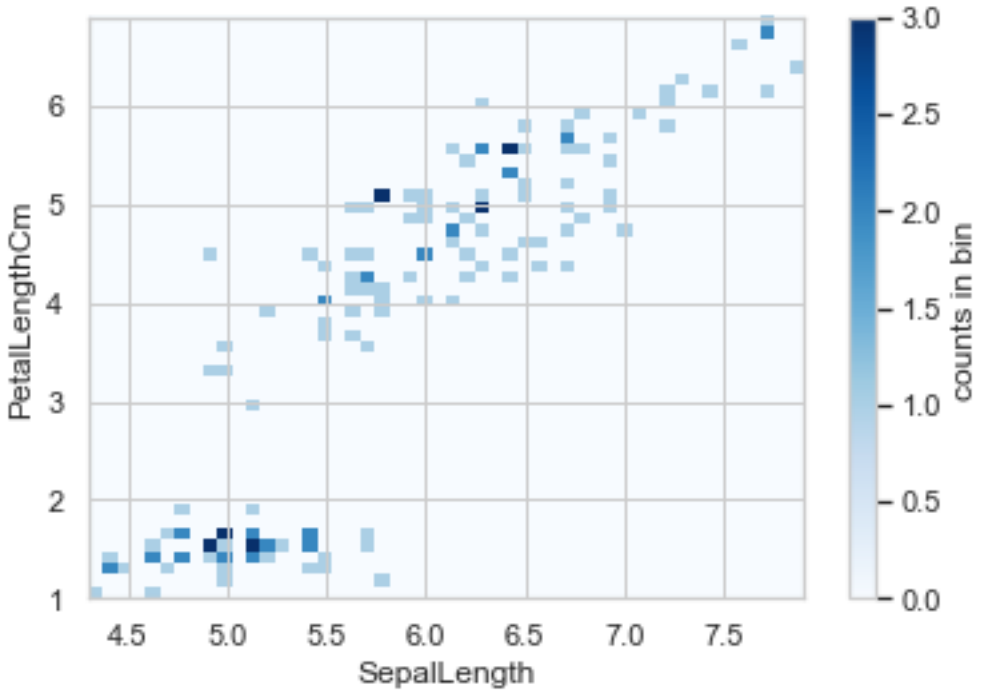
Iris Virginica

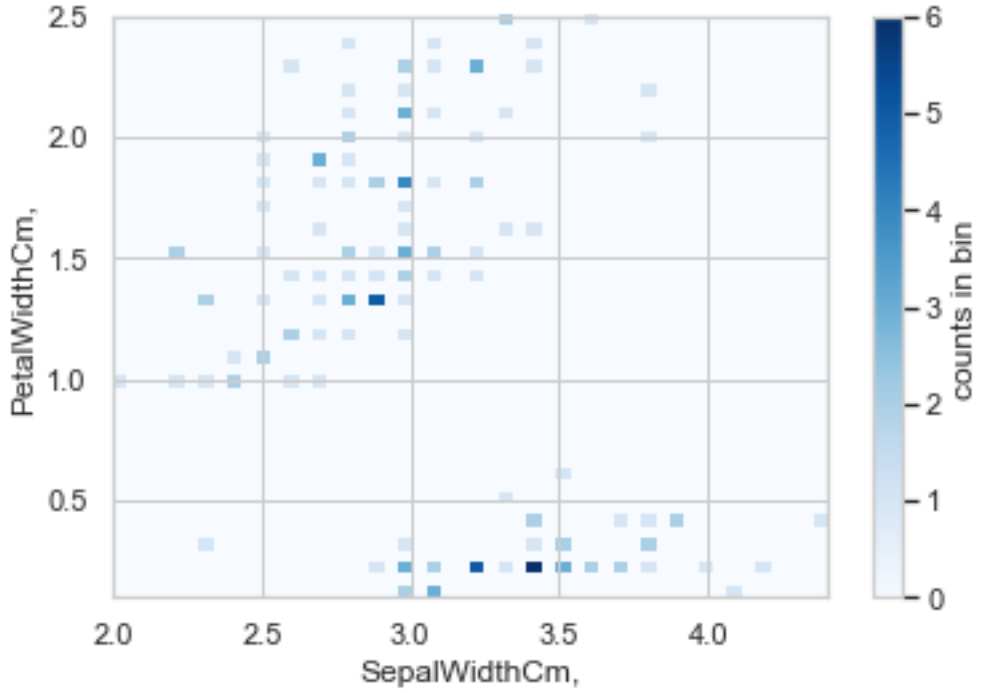
A)Plot the histograms:

2 dimenional Histogram:



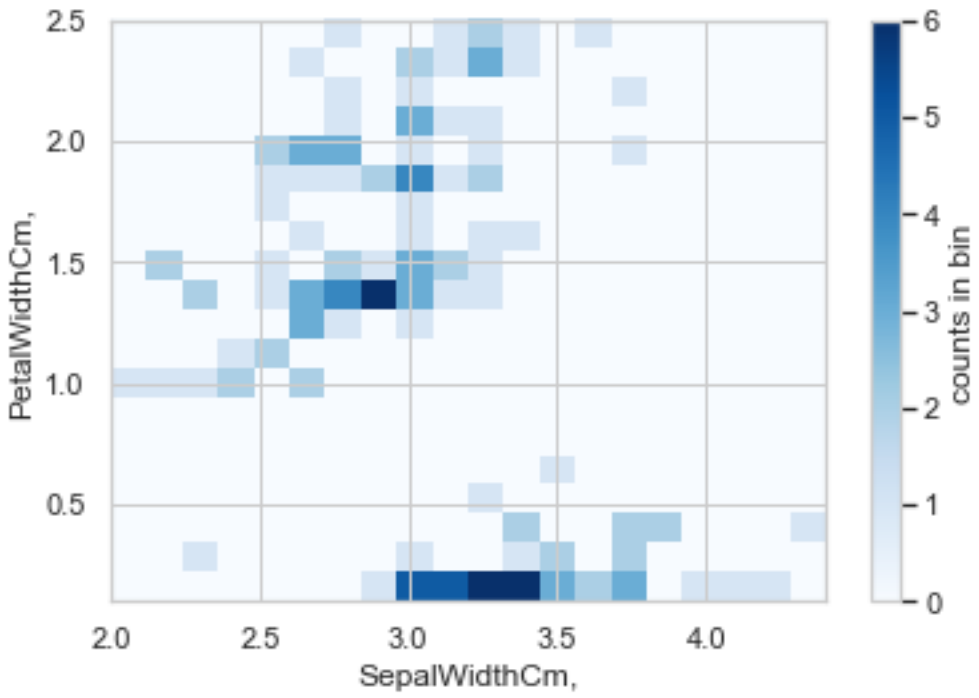



PetalLengthCm
SepalLength
counts in bin
1
2
3
4
5
6
4.5
5.0
5.5
6.0
6.5
7.0
7.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0

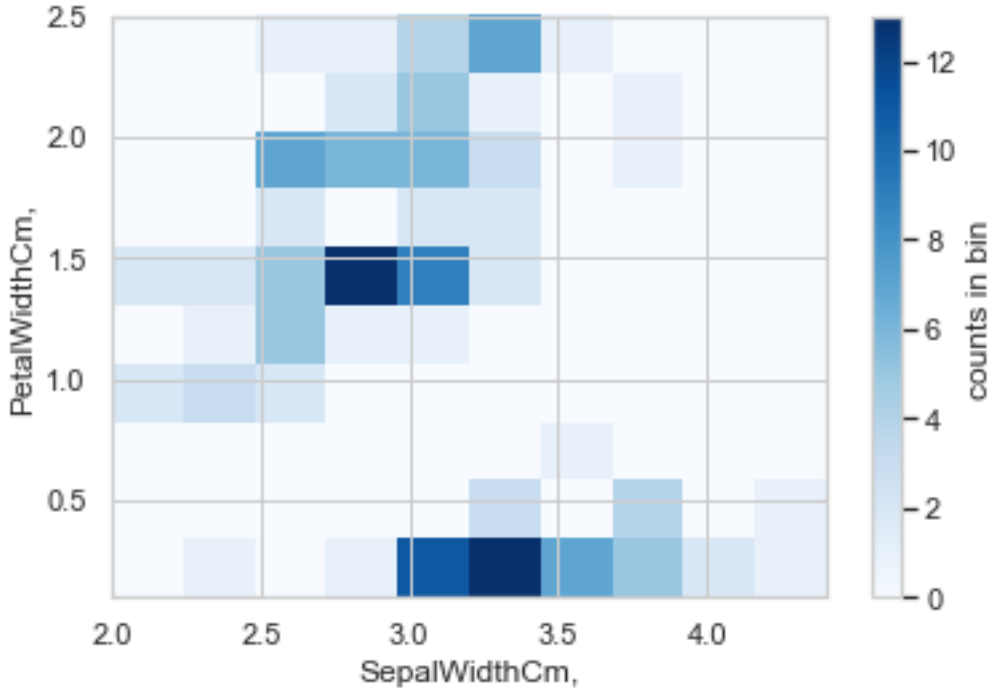

PetalWidthCm,
SepalWidthCm,
counts in bin
0.5
1.0
1.5
2.0
2.5
2.0
2.5
3.0
3.5
4.0
0
1
2
3
4
5
6

If we change number of the bins, we will get result like ths(previous results are with bin=50):

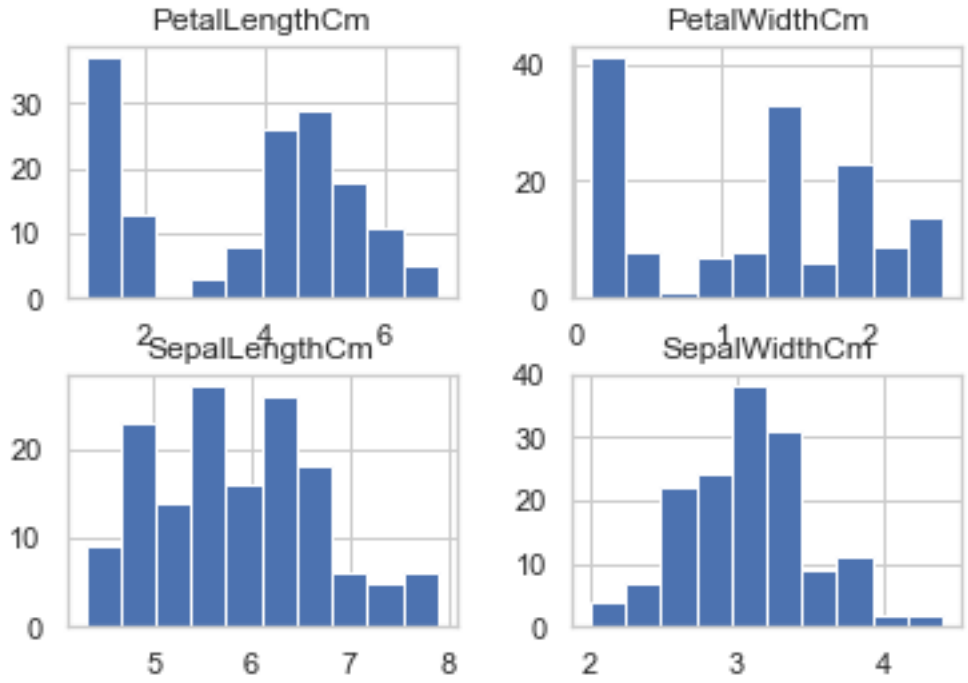
Bin=20:



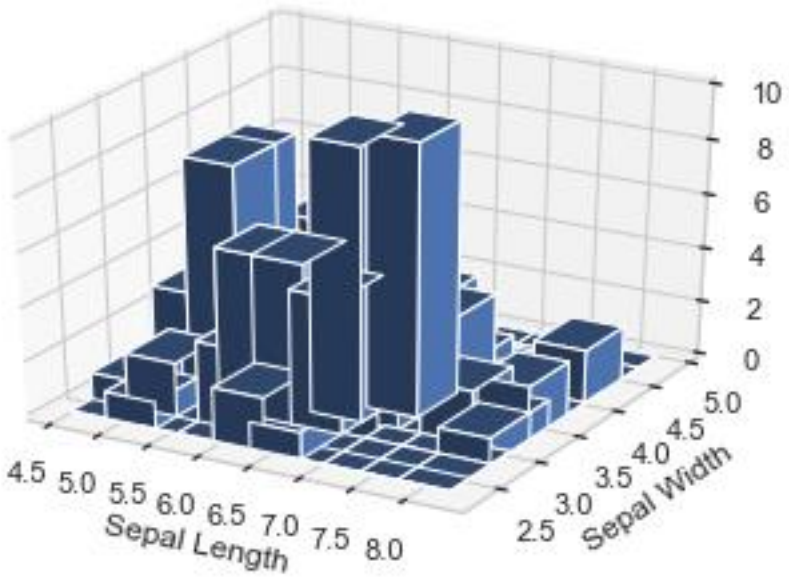
Bin=10:



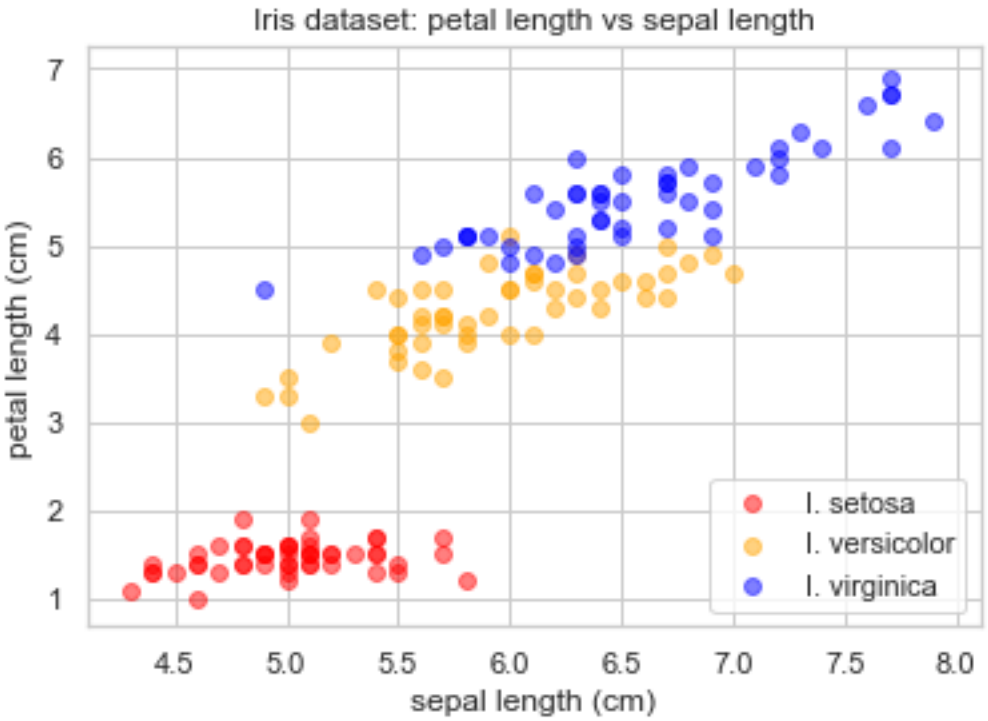
1 dimensional:

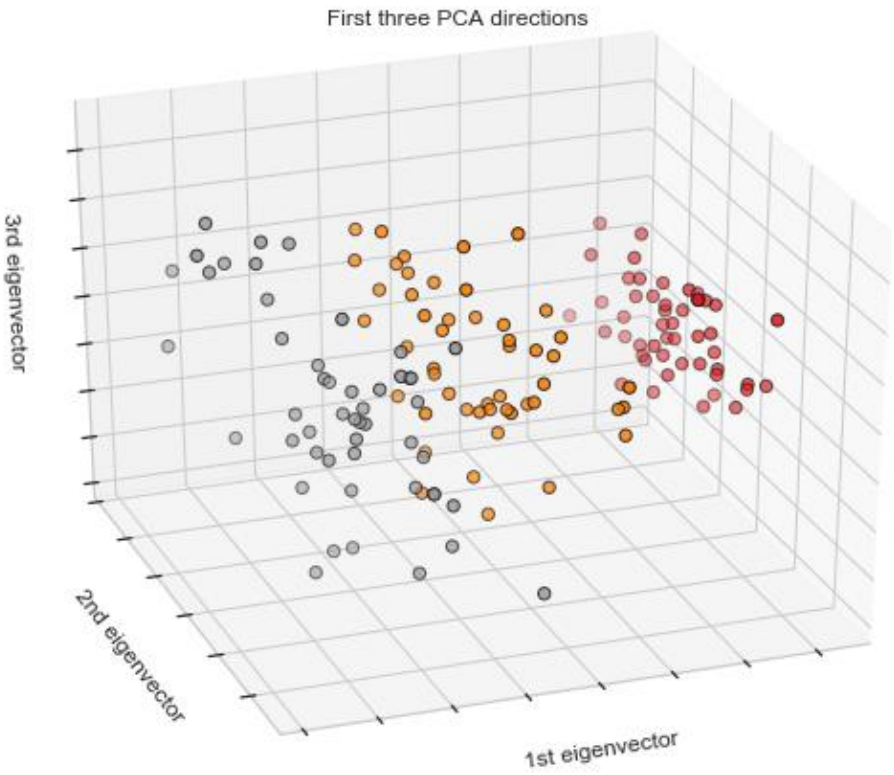


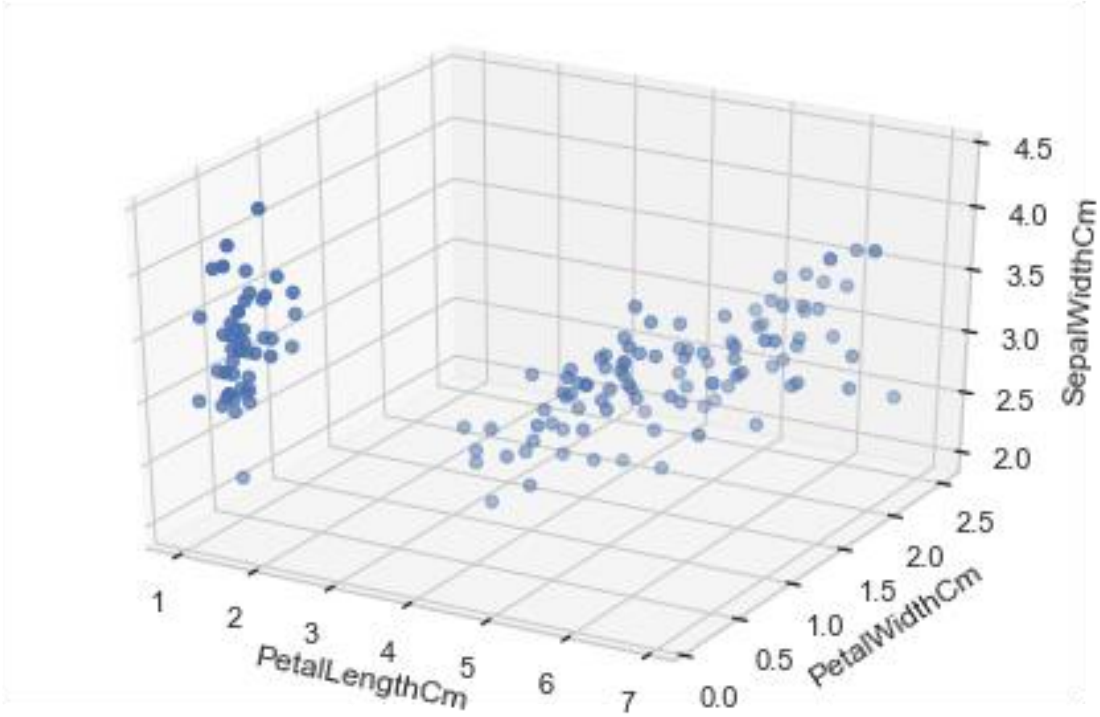
B) 3D Histogram

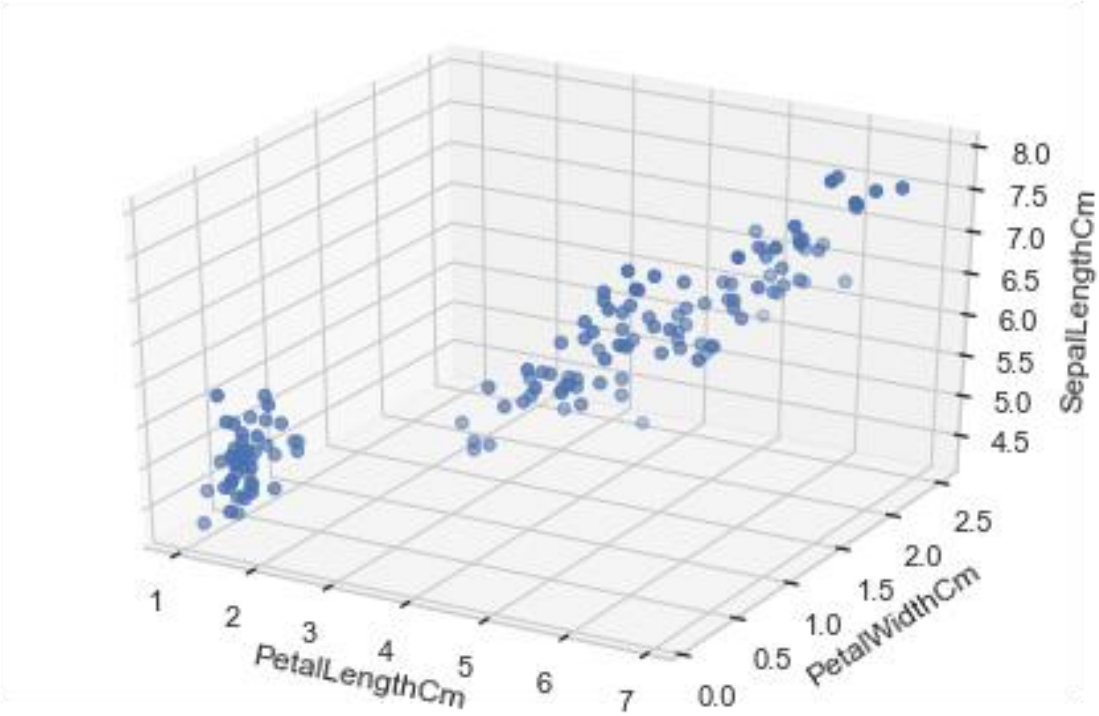


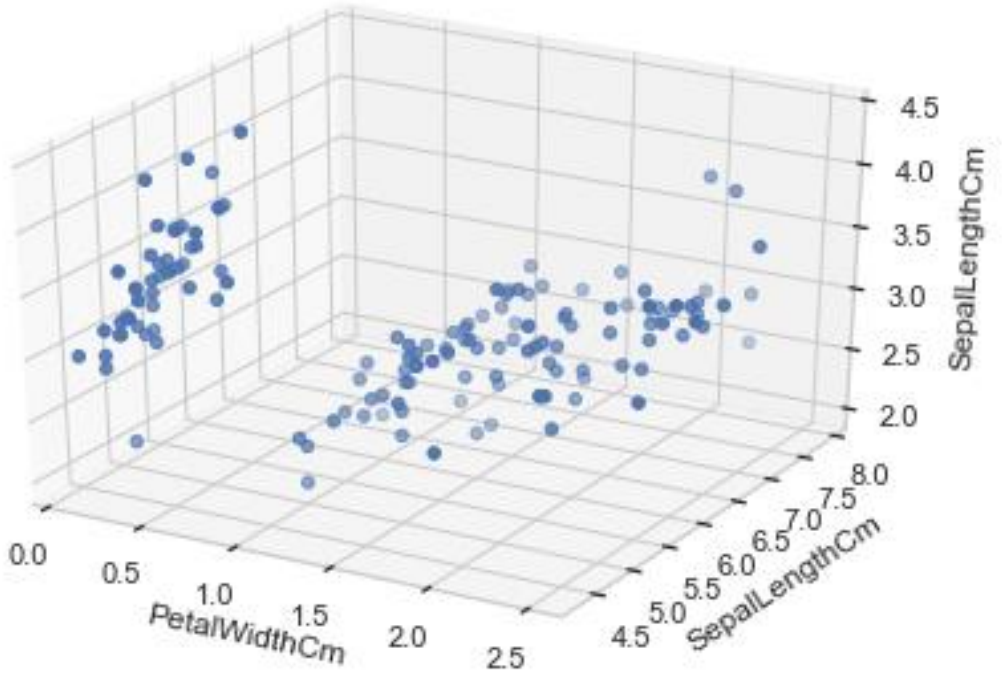
C)Scatter Data(next page):

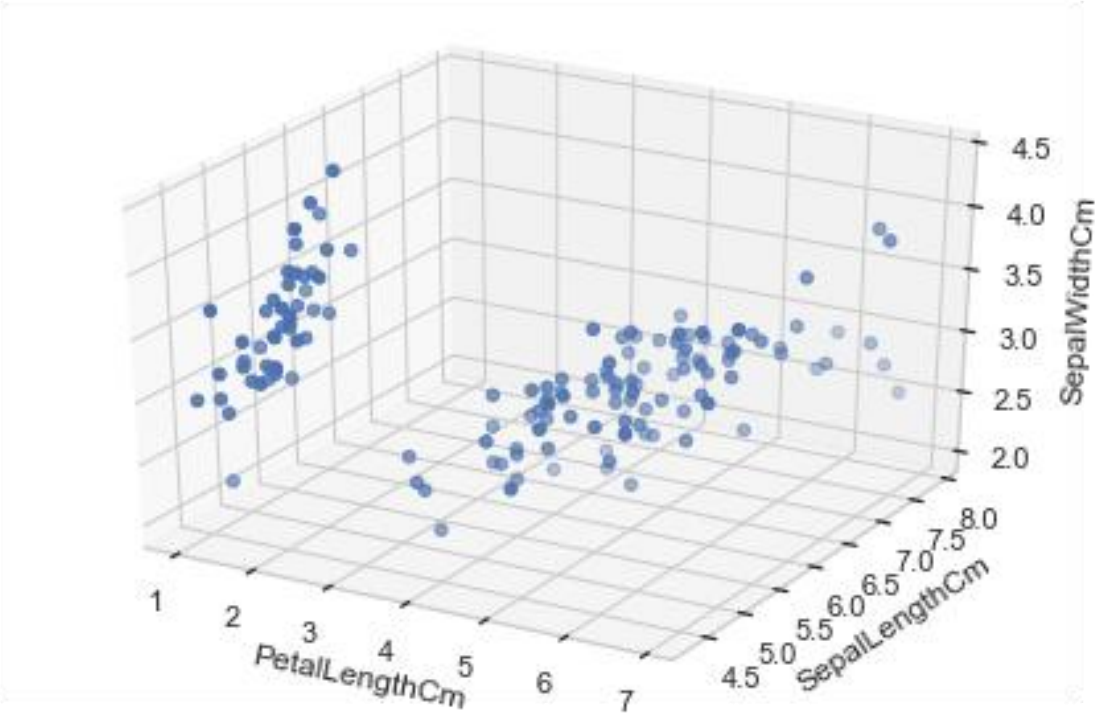


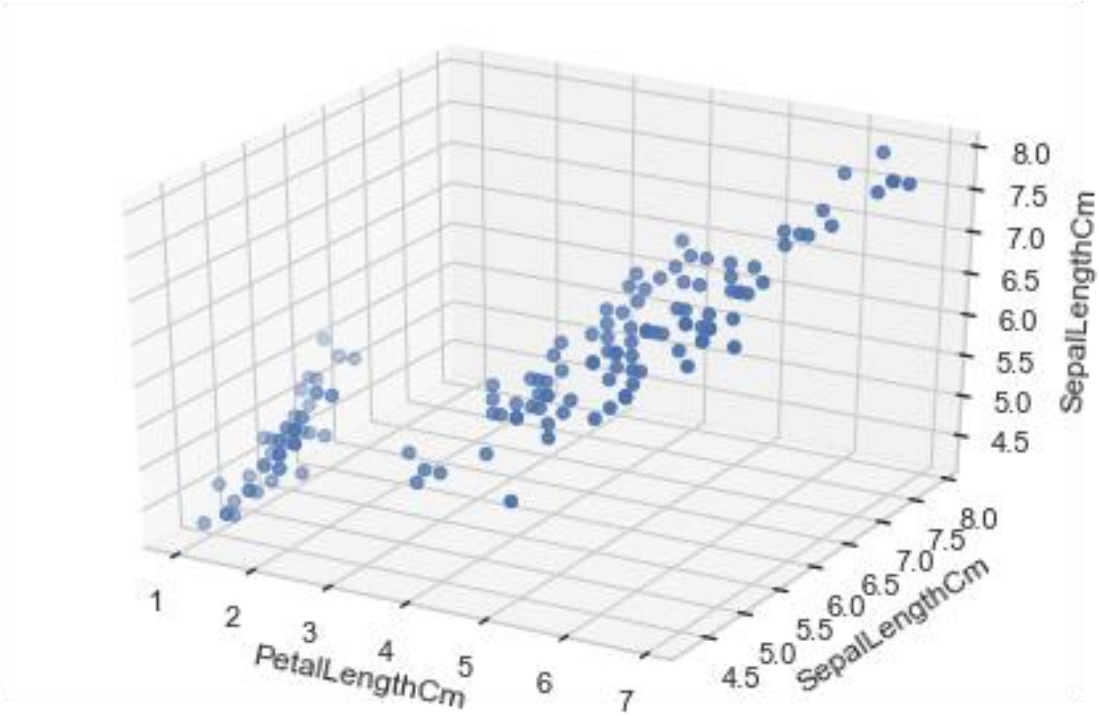

First three PCA directions
3rd eigenvector
2nd eigenvector
1st eigenvector

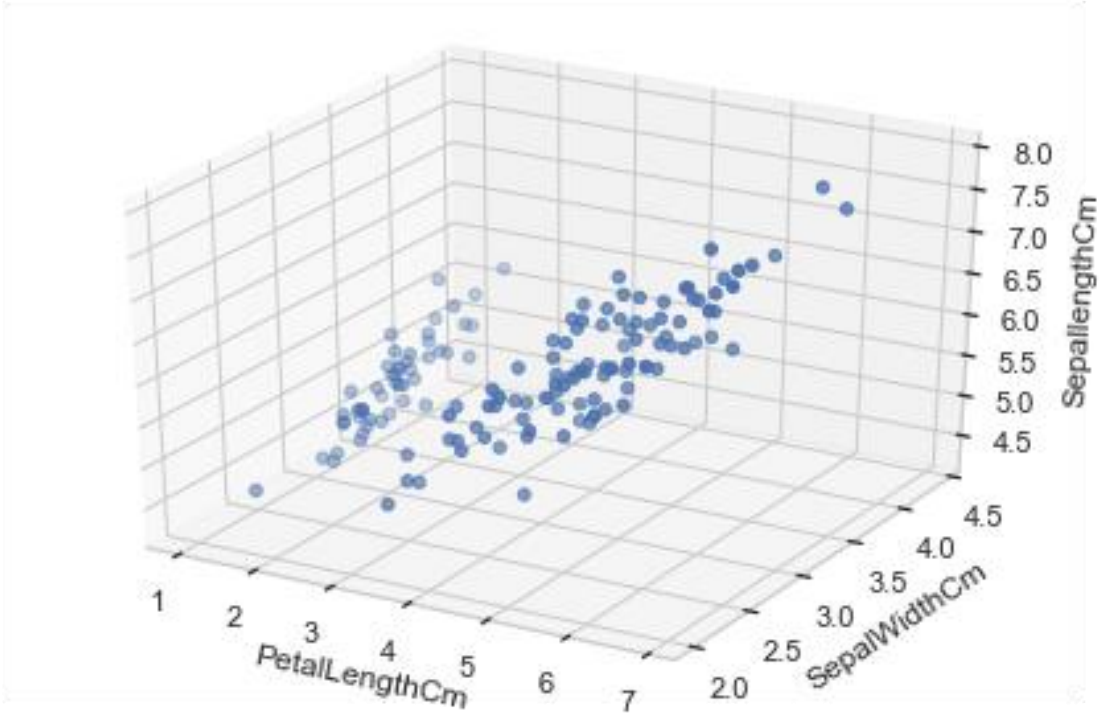

SepalWidthCm
4.5
4.0
3.5
3.0
2.5
2.0
PetalLengthCm
1
2
3
4
5
6
7
PetalWidthCm
0.0
0.5
1.0
1.5
2.0
2.5


8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
SepalLengthCm
2.5
2.0
1.5
1.0
0.5
0.0
PetalWidthCm
1
2
3
4
5
6
7
PetalLengthCm


4.5
4.0
3.5
3.0
2.5
2.0
SepalLengthCm
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
SepalLengthCm
0.0
0.5
1.0
1.5
2.0
2.5
PetalWidthCm


4.5
4.0
3.5
3.0
2.5
2.0
SepalWidthCm
1
2
3
4
5
6
7
PetalLengthCm
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
SepalLengthCm


8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
SepalLengthCm
1
2
3
4
5
6
7
PetalLengthCm
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
SepalLengthCm



D) Mean and Variance:

	mean	Var
SepalLengthCm	5.843333	0.685694
SepalWidthCm	3.054000	0.188004
PetalLengthCm	3.758667	3.113179
PetalWidthCm	1.198667	0.582414

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

E)Covariance:

On features:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
SepalLengthCm	0.685694	-0.039268	1.273682	0.516904
SepalWidthCm	-0.039268	0.188004	-0.321713	-0.117981
PetalLengthCm	1.273682	-0.321713	3.113179	1.296387
PetalWidthCm	0.516904	-0.117981	1.296387	0.582414

For calculating covariance on classes, we have to use Mean_Covariance:

Mean vector:

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$$

$$\mathrm{COV} = \frac{\sum_{i=1}^{n}(X_i - \bar{x})(Y_i - \bar{y})}{n-1}$$

Setosa:

```
mean_vector:
[5.005999999999999, 3.4180000000000006, 1.464, 0.2439999999999999]
get_covariance_matrix:
[[ 0.12424897  0.10029795  0.01613878  0.01054694]
 [ 0.10029795  0.14517959  0.01168164  0.01143674]
 [ 0.01613878  0.01168164  0.03010613  0.00569796]
 [ 0.01054694  0.01143674  0.00569796  0.01149388]]
```

‘

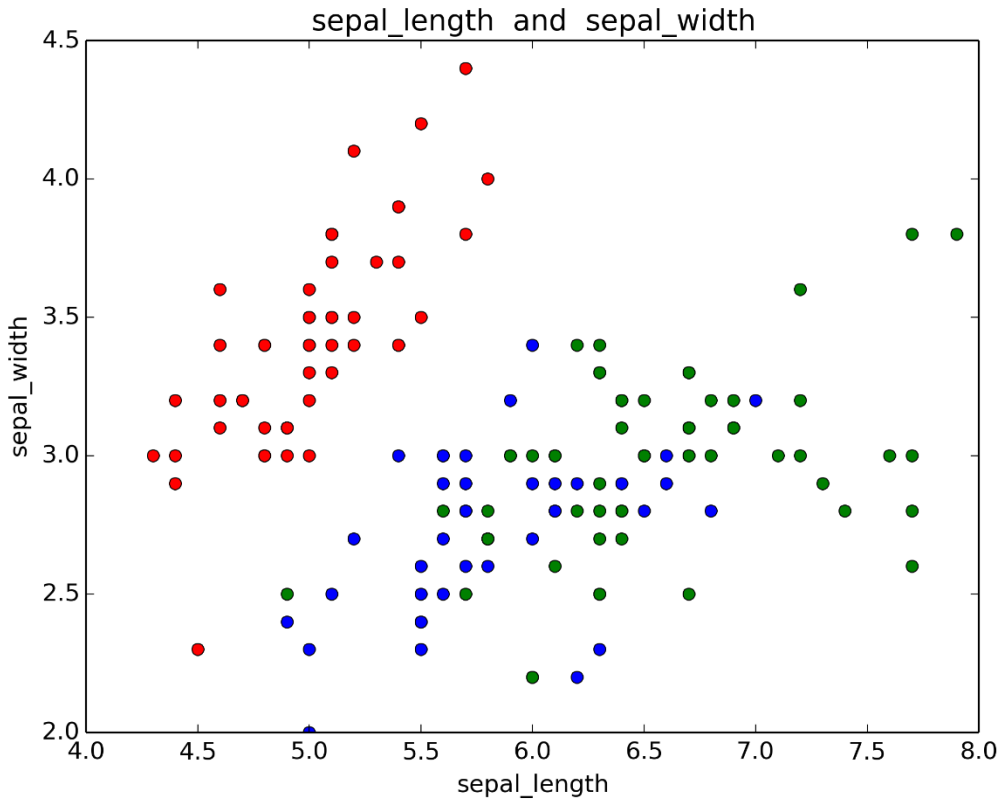
Iris_versicolor:

```
mean_vector:
[5.936, 2.7700000000000005, 4.26, 1.3259999999999998]
get_covariance_matrix:
[[ 0.26643266  0.08518367  0.18289797  0.05577959]
 [ 0.08518367  0.09846939  0.08265305  0.04120408]
 [ 0.18289797  0.08265305  0.22081632  0.07310204]
 [ 0.05577959  0.04120408  0.07310204  0.03910612]]
```

Iris_virginica:

```
mean_vector:
[6.587999999999998, 2.9739999999999998, 5.552, 2.026]
get_covariance_matrix:
[[ 0.40434278  0.09376325  0.30328976  0.04909387]
 [ 0.09376325  0.10400408  0.07137958  0.04762857]
 [ 0.30328976  0.07137958  0.30458773  0.04882448]
 [ 0.04909387  0.04762857  0.04882448  0.07543266]]
```

Considering the two features, sepal_length and sepal_width (mean_vector[0] and mean_vector[1]), we find Iris_setosa(Red) is far from the others. By contrast, Iris_versicolor(Blue) and Iris_virginica(Green) are near each other.



F)Correlation:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
SepalLengthCm	1.000000	-0.109369	0.871754	0.817954
SepalWidthCm	-0.109369	1.000000	-0.420516	-0.356544
PetalLengthCm	0.871754	-0.420516	1.000000	0.962757
PetalWidthCm	0.817954	-0.356544	0.962757	1.000000

Lets calculate correlation on each class:

Setosa-versicolor:

```
array([[1.       , 0.7638543],
       [0.7638543, 1.       ]])
```

Setosa_virginica:

```
array([[1.        , 0.61843828],
       [0.61843828, 1.        ]])
```

Virginica_versicolor:

```
array([[1.        , 0.97948861],
       [0.97948861, 1.        ]])
```

```
setosa_mean=setosa.mean()
versicolor_mean=versicolor.mean()
virginica_mean=virginica.mean()
```

```
np.corrcoef(setosa_mean,versicolor_mean)
```

```
array([[1.        , 0.7638543],
       [0.7638543, 1.       ]])
```

```
np.corrcoef(setosa_mean,virginica_mean)
```

```
array([[1.        , 0.61843828],
       [0.61843828, 1.        ]])
```

```
np.corrcoef(versicolor_mean,virginica_mean)
```

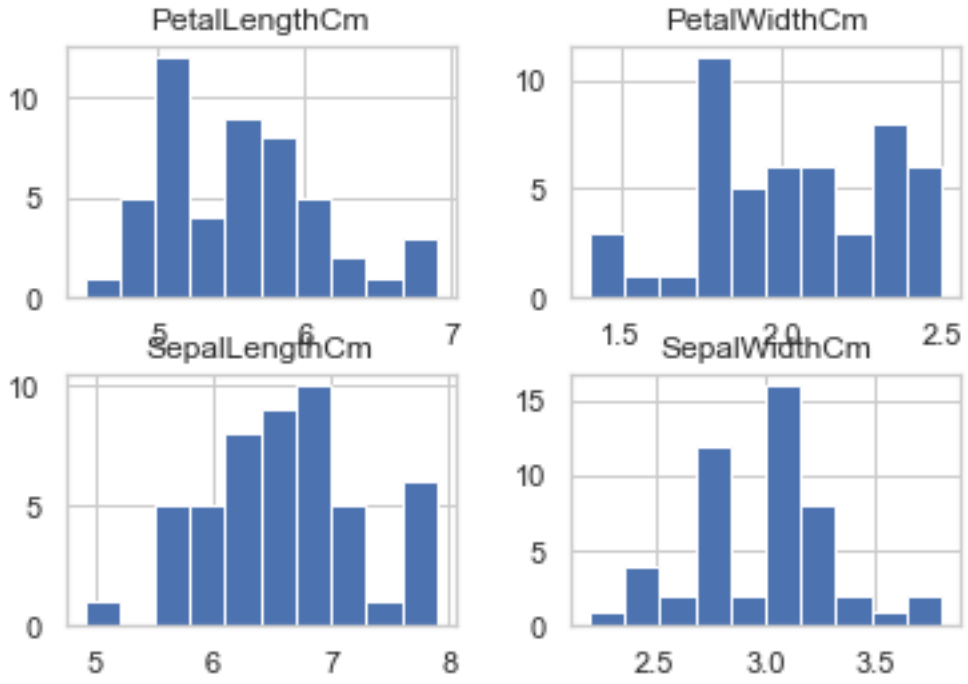
```
array([[1.        , 0.97948861],
       [0.97948861, 1.        ]])
```

The most important difference between Correlation Matrix and Covariance .Matrix is that:

- ***Correaltion Matrix main diagonal is 1.(each feature has correlation 1 with itself)***
- Correlation is when the change in one item may result in the change in the another item. On the other hand, covariance is when two items vary together.
- Correlation is concerned as A measure used to indicate the extent to which two random variables change in tandem is known as covariance. A measure used to represent how strongly two random variables are related known as correlation.
- Covariance is nothing but a measure of correlation. On the contrary, correlation refers to the scaled form of covariance.
- The value of correlation takes place between -1 and +1. Conversely, the value of covariance lies between $-\infty$ and $+\infty$.

- ❖ Covariance is affected by the change in scale, i.e. if all the value of one variable is multiplied by a constant and all the value of another variable are multiplied, by a similar or different constant, then the covariance is changed. As against this, correlation is not influenced by the change in scale.
- ❖ Correlation is dimensionless, i.e. it is a unit-free measure of the relationship between variables. Unlike covariance, where the value is obtained by the product of the units of the two variables.

G)Viriginica



Correlation and Covariance:

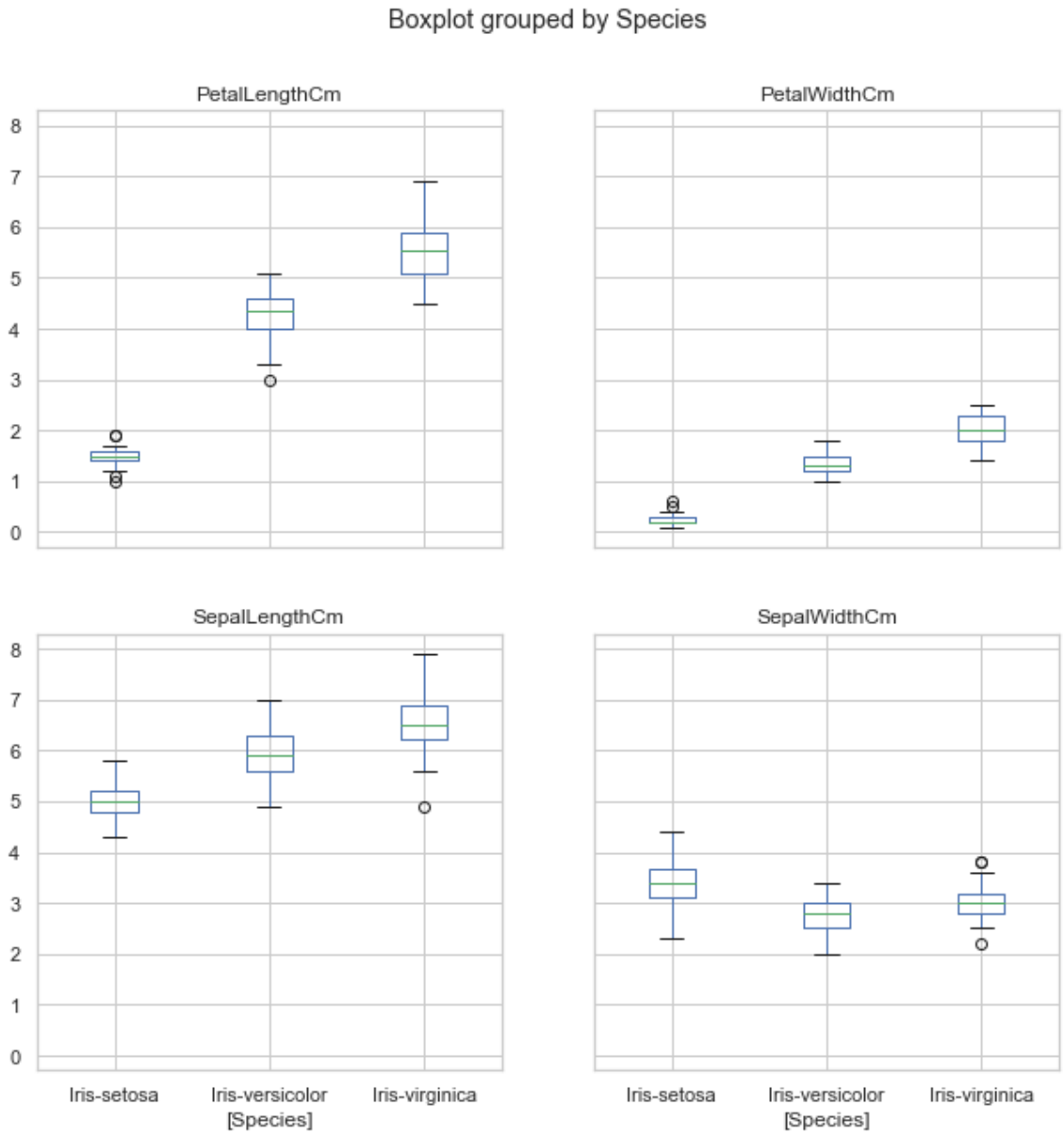
	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
SepalLengthCm	1.000000	0.457228	0.864225	0.281108
SepalWidthCm	0.457228	1.000000	0.401045	0.537728
PetalLengthCm	0.864225	0.401045	1.000000	0.322108
PetalWidthCm	0.281108	0.537728	0.322108	1.000000

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
SepalLengthCm	0.404343	0.093763	0.303290	0.049094
SepalWidthCm	0.093763	0.104004	0.071380	0.047629
PetalLengthCm	0.303290	0.071380	0.304588	0.048824
PetalWidthCm	0.049094	0.047629	0.048824	0.075433

Based on Correlation,”PetalLength” and “Sepal Length” are more similar.

H)Feature Selection:

I used boxplots to find out which feature seperates the classes better(I more informative)



Based On these Results, Best Features is PetalWidth

I tested this task with “SelectKBest” Function in python and the result was the same.

What is gradient descent ?

It is an optimization algorithm to find the minimum of a function. We start with a random point on the function and move in the **negative direction** of the **gradient of the function** to reach the **local/global minima**.

Cost fuction is SSE:

$$J(w)=12\sum_i(target_{(i)}-output_{(i)})^2 \qquad ;output_{(i)}\in R$$

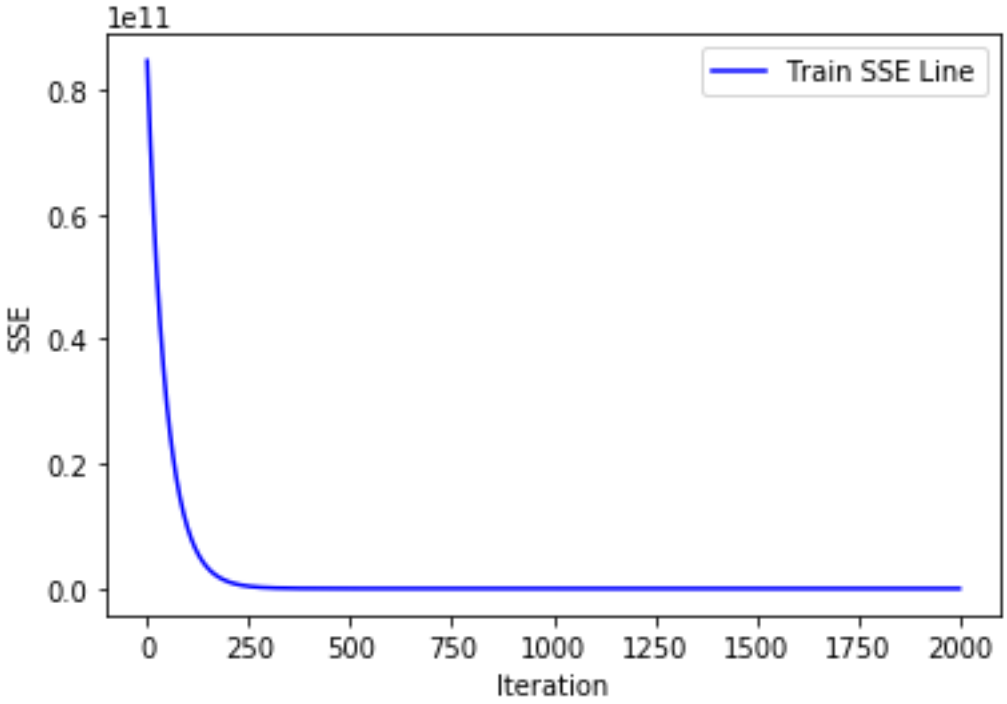
We implement Gradient Descent with the appropriate error anf plot it.

Iteration 0 | Cost: 84604307330.630157
Iteration 1 | Cost: 82781761231.728531
Iteration 2 | Cost: 80998484776.147720
Iteration 3 | Cost: 79253631835.952148
Iteration 4 | Cost: 77546374514.400330
Iteration 5 | Cost: 75875902753.124390
Iteration 6 | Cost: 74241423947.773407
Iteration 7 | Cost: 72642162571.938477
Iteration 8 | Cost: 71077359809.180817
Iteration 9 | Cost: 69546273192.988403
Iteration 10 | Cost: 68048176254.490433
Iteration 11 | Cost: 66582358177.762253

…..

Iteration 1999 | Cost: 13541479.738889

SSE Test = 6386268.633887
Train = 13539468.826632



Bonus Part(3d plot):