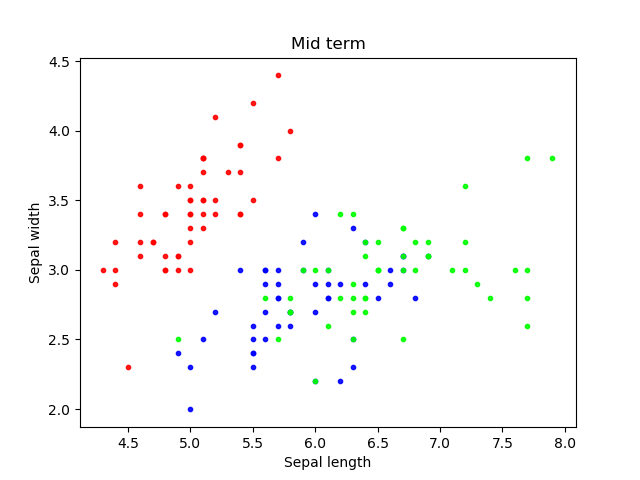
# Answer to the question number 1

In order to reduce over-fitting and nose of dataset we split the dataset into training and test data. The training data is used to train the model and the test data is used to evaluate the data. In case of random use of training and test data, we can use validation to minimize over fitting.

# Answer to the question number 2

## Visualization for sepal length and width

**Results:**



**Code:**

import xlrd

import numpy as np

import matplotlib.pyplot as plt

file\_location = "Book1.xlsx"

workbook = xlrd.open\_workbook(file\_location)

first\_sheet = workbook.sheet\_by\_index(0)

x1 = [first\_sheet.cell\_value(i, 1) for i in range(50)]

y1 = [first\_sheet.cell\_value(i, 2) for i in range(50)]

x2 = [first\_sheet.cell\_value(i, 1) for i in range(51, 100)]

y2 = [first\_sheet.cell\_value(i, 2) for i in range(51, 100)]

x3 = [first\_sheet.cell\_value(i, 1) for i in range(101, 150)]

y3 = [first\_sheet.cell\_value(i, 2) for i in range(101, 150)]

colors1 = (1,0,0)

colors2 = (0,0,1)

colors3 = (0,1,0)

area = np.pi\*3

# Plot

plt.scatter(x1, y1, s=area, c=colors1, alpha=0.9)

plt.scatter(x2, y2, s=area, c=colors2, alpha=0.9)

plt.scatter(x3, y3, s=area, c=colors3, alpha=0.9)

plt.title('Mid term')

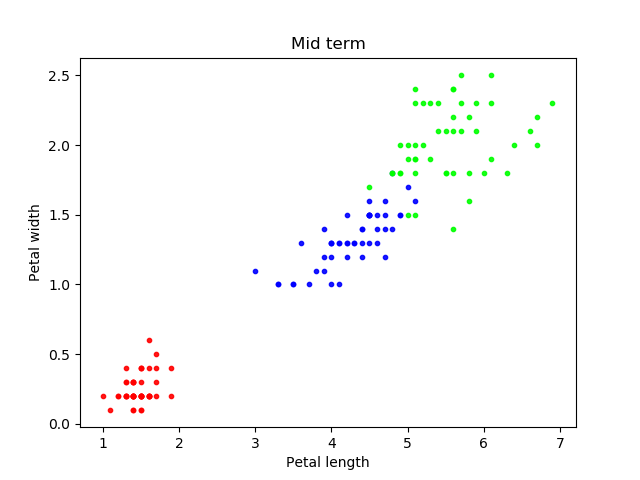
plt.xlabel('Sepal length')

plt.ylabel('Sepal width')

plt.show()

## Visualization based on Petal length and width

**Results:**



**Code:**

import xlrd

import numpy as np

import matplotlib.pyplot as plt

file\_location = "Book1.xlsx"

workbook = xlrd.open\_workbook(file\_location)

first\_sheet = workbook.sheet\_by\_index(0)

x1 = [first\_sheet.cell\_value(i, 3) for i in range(50)]

y1 = [first\_sheet.cell\_value(i, 4) for i in range(50)]

x2 = [first\_sheet.cell\_value(i, 3) for i in range(51, 100)]

y2 = [first\_sheet.cell\_value(i, 4) for i in range(51, 100)]

x3 = [first\_sheet.cell\_value(i, 3) for i in range(101, 150)]

y3 = [first\_sheet.cell\_value(i, 4) for i in range(101, 150)]

colors1 = (1,0,0)

colors2 = (0,0,1)

colors3 = (0,1,0)

area = np.pi\*3

# Plot

plt.scatter(x1, y1, s=area, c=colors1, alpha=0.9)

plt.scatter(x2, y2, s=area, c=colors2, alpha=0.9)

plt.scatter(x3, y3, s=area, c=colors3, alpha=0.9)

plt.title('Mid term')

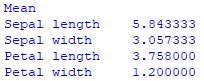
plt.xlabel('Petal length')

plt.ylabel('Petal width')

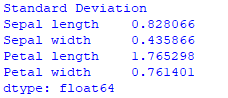
plt.show()

## Answer to the question number 3

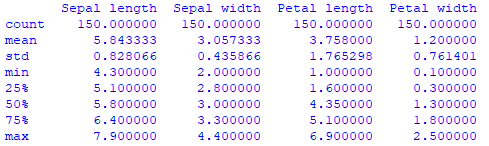
b)



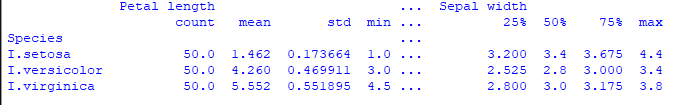
c)



d)



a) and e)



## Answer to the question no 4

a) Gaussian Probability:

[[1.00000000e+000 3.19496100e-015 3.21887730e-033]

[2.95309505e-237 1.15865992e-001 8.84134008e-001]

[6.49883737e-049 1.00000000e+000 7.56829143e-011]

[1.00000000e+000 1.44947089e-015 2.42078642e-032]

[1.06435239e-166 9.99795311e-001 2.04688707e-004]

[0.00000000e+000 1.47974598e-012 1.00000000e+000]

[3.22839040e-129 9.99994573e-001 5.42717052e-006]

[1.00000000e+000 6.37211928e-016 6.73764302e-035]

[0.00000000e+000 1.88224228e-007 9.99999812e-001]

[0.00000000e+000 1.52765707e-008 9.99999985e-001]

[0.00000000e+000 4.38584077e-007 9.99999561e-001]

[1.18878815e-317 2.62142567e-005 9.99973786e-001]

[1.00000000e+000 2.37206966e-016 2.04131715e-032]

[1.00000000e+000 2.84800290e-015 1.91343022e-032]

[6.34925191e-257 3.52669861e-004 9.99647330e-001]

[0.00000000e+000 6.13373355e-007 9.99999387e-001]

[1.00000000e+000 3.40245329e-013 4.64487483e-032]

[5.47348582e-313 2.65556462e-007 9.99999734e-001]

[1.00000000e+000 4.28047069e-015 2.43405341e-033]

[2.59578737e-243 6.72190858e-002 9.32780914e-001]

[2.35060869e-299 4.94816059e-006 9.99995052e-001]

[1.00000000e+000 2.68698536e-016 2.29478666e-034]

[1.00000000e+000 6.56681456e-017 8.52338216e-034]

[3.08137057e-306 1.40663966e-003 9.98593360e-001]

[1.00000000e+000 2.87161007e-015 1.18195088e-031]

[1.00000000e+000 1.01850618e-015 3.44667456e-034]

[1.00000000e+000 1.42548794e-012 1.17156345e-026]

[1.46147985e-115 9.99999825e-001 1.74621074e-007]

[0.00000000e+000 6.30303396e-009 9.99999994e-001]

[0.00000000e+000 2.73645799e-010 1.00000000e+000]

[1.00000000e+000 2.37496124e-016 1.93957429e-033]

[1.00000000e+000 1.33503535e-015 3.21001135e-034]

[1.00000000e+000 8.70128931e-016 4.63223886e-033]

[1.39524225e-152 9.99983479e-001 1.65209801e-005]

[1.42358462e-153 9.98866873e-001 1.13312750e-003]

[1.00000000e+000 1.04183746e-015 9.18377384e-034]

[9.99999996e-001 4.32275928e-009 3.99977770e-024]

[1.05484250e-139 9.99954654e-001 4.53462948e-005]

[1.00000000e+000 2.80926042e-012 6.71177501e-028]

[0.00000000e+000 3.27704140e-012 1.00000000e+000]

[1.04880000e-058 1.00000000e+000 3.01530163e-011]

[0.00000000e+000 2.17909083e-011 1.00000000e+000]

[7.06528858e-154 9.97571780e-001 2.42821950e-003]

[1.00000000e+000 6.31214672e-019 6.30789857e-035]

[1.46136117e-209 3.54391892e-001 6.45608108e-001]

[1.00000000e+000 1.19071996e-016 3.29834054e-032]

[0.00000000e+000 1.19152776e-009 9.99999999e-001]

[1.00000000e+000 5.47157848e-016 1.75001958e-031]

[1.00000000e+000 1.01272213e-014 2.32649721e-032]

[0.00000000e+000 1.26648483e-006 9.99998734e-001]

[1.00000000e+000 7.76647008e-016 1.99393364e-033]

[0.00000000e+000 1.06896840e-008 9.99999989e-001]

[3.00018447e-122 9.99997794e-001 2.20618311e-006]

[3.24983586e-104 9.99999872e-001 1.27767370e-007]

[5.49046543e-071 1.00000000e+000 4.03861793e-010]

[5.70637249e-304 7.68847004e-007 9.99999231e-001]

[5.23764622e-296 1.40467787e-006 9.99998595e-001]

[4.41900816e-201 9.94592929e-001 5.40707118e-003]

[7.95669862e-225 1.33957821e-001 8.66042179e-001]

[1.00000000e+000 9.48226392e-014 1.07242275e-031]

[2.30016947e-082 1.00000000e+000 3.65668438e-010]

[0.00000000e+000 1.47378634e-008 9.99999985e-001]

[5.61330462e-245 8.78191938e-002 9.12180806e-001]

[1.00000000e+000 1.59652736e-015 1.02276064e-032]

[1.14645526e-164 9.99620074e-001 3.79925592e-004]

[2.04071752e-121 9.99995737e-001 4.26286076e-006]

[8.50581405e-209 2.13001550e-001 7.86998450e-001]

[4.70203675e-148 9.99993077e-001 6.92341777e-006]

[1.00000000e+000 1.35318016e-015 1.40293321e-032]

[1.00000000e+000 4.19656583e-014 3.71638875e-029]

[1.00000000e+000 1.33503535e-015 3.21001135e-034]

[9.48649814e-292 4.26615357e-003 9.95733846e-001]

[3.74066926e-109 9.99999988e-001 1.16120615e-008]

[0.00000000e+000 2.23592942e-013 1.00000000e+000]

[1.00000000e+000 5.43833695e-016 6.53706890e-034]]

c) Prediction:

[2 1 0 2 0 2 0 1 1 1 1 1 1 1 1 0 1 1 0 0 2 1 0 0 2 0 0 1 1 0 2 1 0 2 2 1 0

1 1 1 2 0 2 0 0 1 2 2 1 2 1 2 1 1 2 1 1 2 1 2 1 0 2 1 1 1 1 2 0 0 2 1 0 0

1]

d) Accuracy:

0.9466666666666667

## Answer to the quest no 5

[[50 0 0]

[ 0 47 3]

[ 0 4 46]]