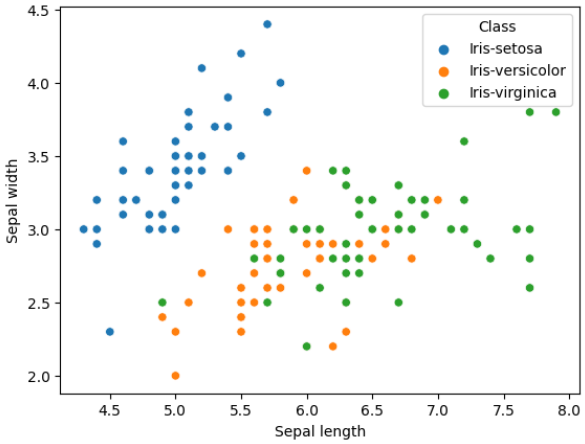
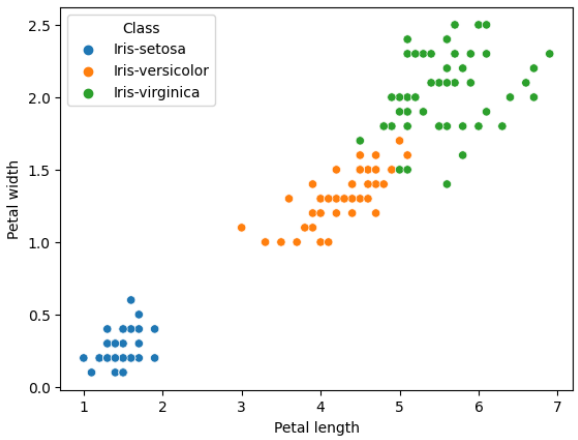
PRML LAB REPORT 4

MANISH(B21CS044)

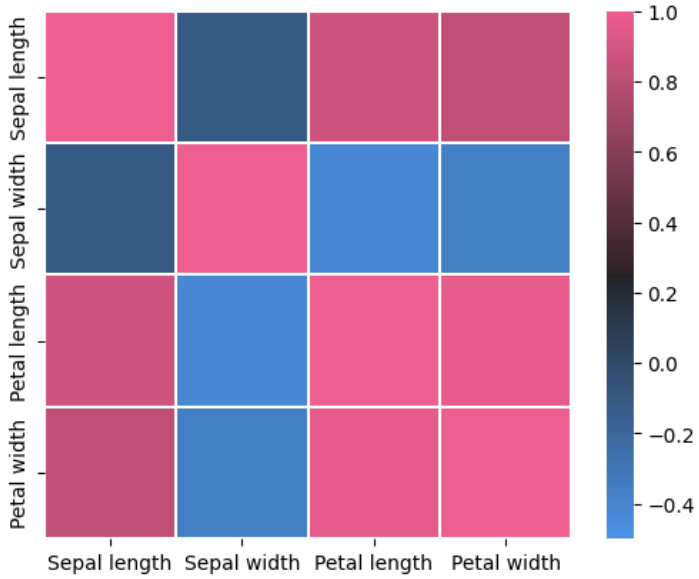
QUESTION 1

In this part first we preprocess the data so in this we spilled the data into 70:30 ratio in train and test data. And for proper visualization we plotted the scatter plots for better visualization of data.





After that we have plotted a heatmap to check correlation between different features.



From this heat map we can clearly see that features Petal length and Petal width are not independent of each other.

1. Implement a **Gaussian Bayes Classifier class** from scratch.(You are not allowed to use the inbuilt scikit function, you are only allowed to use numpy and pandas). The classifier class must have 3 variants defined using its constructor, for each of the cases given below. [10 marks]

Here we have the three possibilities implemented via class, with the type declared in the function Object() { [native code] } as type1, type 2, and type 3 respectively.

Mean column() returns the mean of a column, **mean all features()** returns the mean of all features and a mean matrix, and so on are all methods I declared in the GaussianNB class to achieve this. You can use the functions get mean class to determine the mean of all features pertaining to a single class, **covariance class()** to determine the covariance matrix of all classes, **calculate prior()** to determine the prior of all classes, **train()** to train our model, **discriminant()** to determine the discriminant between classes pertaining to a row, **predict()** to determine the class of a given row, **test()** to provide accuracy of and Y pred for all test data set, and **plot decision regions()** return plot of decision boundary between the features.

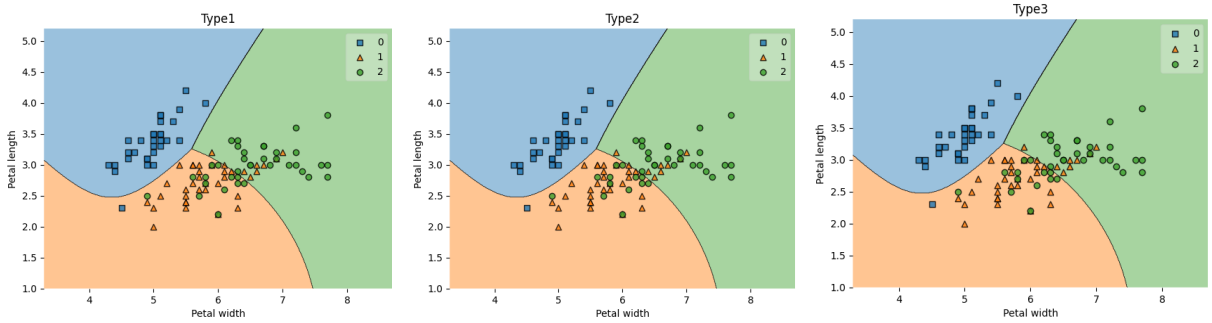
2. The Gaussian Bayes Classifier class should also have the following function:
    a. **Train**: Takes x,y (training data) as input and trains the model.
    b. **Test**: Takes testing data, testing labels as input, and outputs the predictions for every instance in the testing data, and also the accuracy.
    c. **Predict**: Takes a single data point as input, and outputs the predicted class.
    d. **Plot decision boundary**: Takes input the training data points, and their labels, and plots the decision boundary of the model with the data points superimposed on it. (Consider only two features while plotting the decision boundary) [10 marks]

The implementation of this part has been described in the above part
Train function will train the model for dataset
Test function will give accuracy and Y_pred
Predict function will give the predicted class for a row
Plot decision regions will plot the decision boundary between two features

3. Train the Bayes model on the training dataset and plot the decision boundary for each case implemented in Q1. Comment on the decision boundaries obtained in all the 3 cases. Compare the three models and report how well they perform on the dataset. [15 marks]

Here is the accuracy of the All types of GaussianNB on the above dataset.
Type1 : 0.9333333333333333
Type 2 : 9111111111111111
Type3 : 1.0
This is because the output will be better if we make fewer assumptions about the covariance matrix. In Type 1, we assume that features are statistically independent, but in Type 3, we don't make that assumption. Type 3 does better in this case because we can see from the heatmap that the features are not independent.

Here are the limits for choosing Petal length and Petal width in all 3 models.



From these three Decision Boundary plots we can see that Since the Type 1 model

assumes that the features are statistically independent and Type 2 model assumes it to be arbitrarily independent so these can not separate the features in since the features are not independent as we have discussed earlier also.But the Type 3 don't take features to be independent that's why it separates it better.

Perform 5 fold cross validation on the training dataset and report the accuracies on each validation set as well as comment on the generalizability of each model. [10 marks]

Here we have used 5 fold cross validation on the all types of model and here are the Accuracy and mean accuracy of all Types
Type 1:
[0.76190476 0.71428571 0.76190476 0.71428571 0.9047619]

0.7714285714285715

Type 2:
[0.76190476 0.61904762 0.9047619 0.80952381 0.80952381]

0.7809523809523808

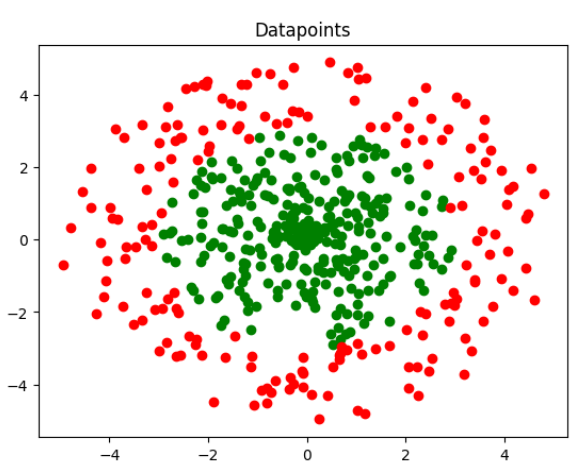
Type 3:
[0.85714286 0.80952381 0.9047619 0.76190476 0.61904762]

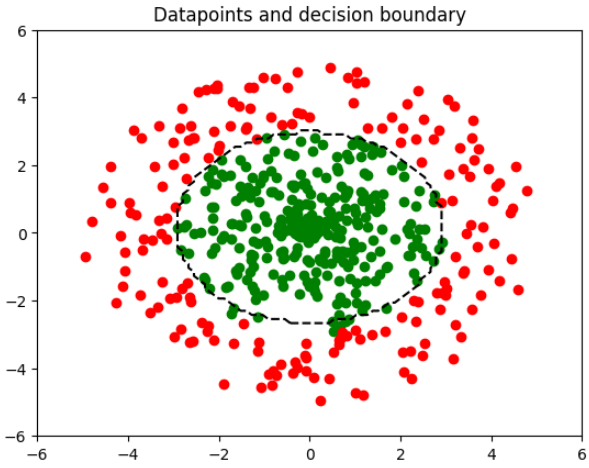
0.7904761904761904

5. Create a synthetic dataset which has 2 features, and data is generated from a circular distribution: $x^2 + y^2 = 25$. The data has 2 classes, points which have distance <=3 have class=1 and points having distance>3 and distance<= 5 have class=2. {Note that the classes are thus, not linearly separable)
   Train the implemented Gaussian Bayes Classifier(case 3) on such a synthetic dataset, and plot the decision boundary for the same. [15 marks]

In this we have sampled 500 points from the curve given and made it a DataFrame and now trained the model for points having distance<=3 class 1 and distance>3 and distance<= 5 class=2
We have plotted the decision boundary of the data points on the data points itself.

Here are the plots for the same:





QUESTION 2:

1. Calculate the covariance matrix of the sample X , say Σ_s. Find the eigenvectors and eigenvalue of Σ_s and plot it superimposed on the datapoints X. [5 marks]

In This part we have calculated the covariance_matrix, eigenvalues and eigenvectors for the data points sampled from the given Covariance matrix. Here are the values for the same:

Covariance matrix of dataset = [[1.60125141 0.70142455]

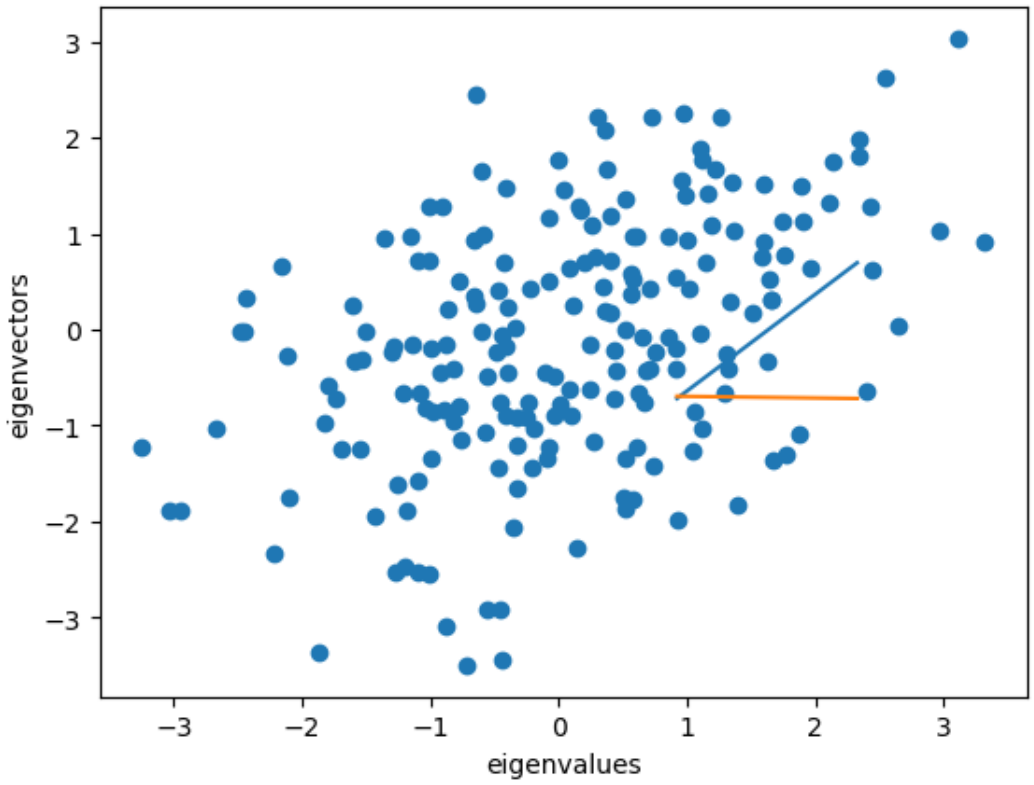
[0.70142455 1.64734089]]

Eigenvalues of Cov_mat = [0.92249314 2.32609916]

Eigenvectors of Cov_mat = [[-0.71862246 -0.69540043]

[0.69540043 -0.71862246]]

Here is plot of eigenvectors superimposed over data points:



2. Perform the transformation $Y = \Sigma_s^{-1/2} X$ on the datapoints X. Calculate the covariance matrix of transformed datapoints Y , say Σ_Y. Comment on the obtained covariance matrix and infer what was the purpose of the transformation.[10 marks]

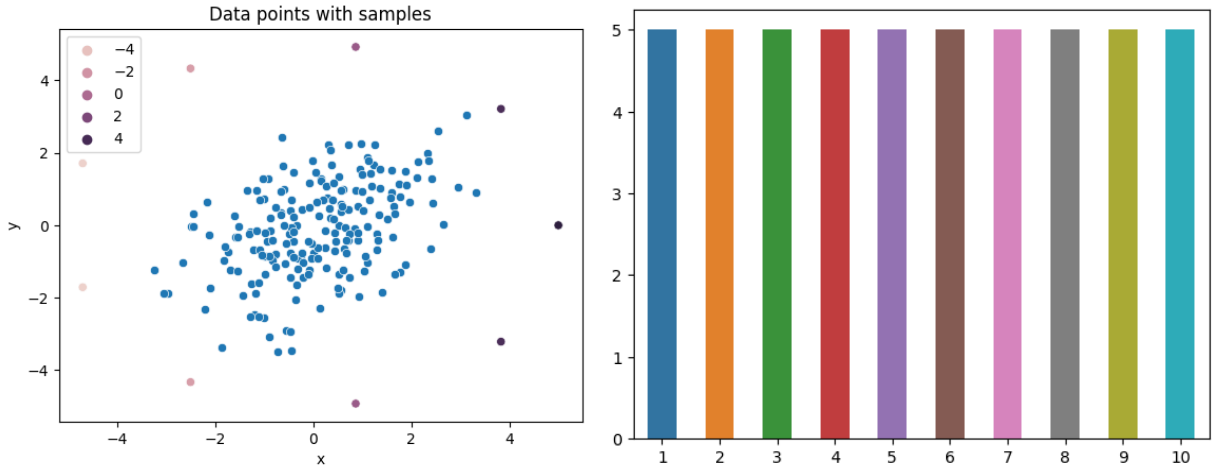
Here is the Cov_mat after transformation:
[[1.00000000e+00 1.65520973e-16]

[1.65520973e-16 1.00000000e+00]]

So after transformation the matrix approaches the Identity matrix , So features tend to be independent from each other.

3. Uniformly sample 10 points on the curve $x^2 + y^2 = 25$. Let these set of points be called P. Plot points in P along with the datapoints in X. Make sure to give each point a different color [Hint :

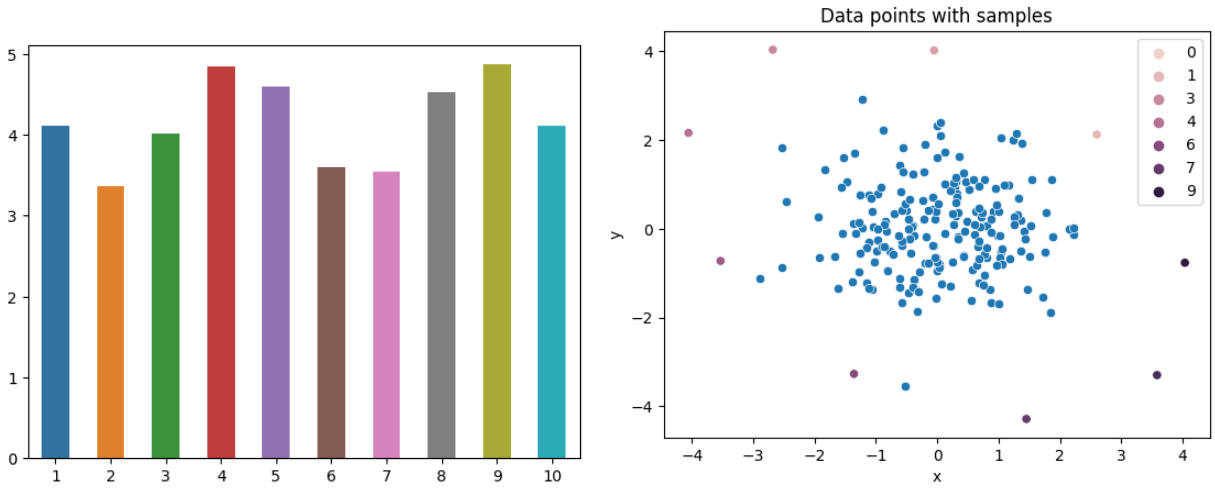
In this part we have sampled 10 data points and plotted it along with X data points.



It shows that the euclidean distance of all points is 5 since it is measured using mu for all points which is[0,0]. And all data points are inside the circle made by all data points.

4. Perform the transformation $Q = \Sigma_s^{-1/2} P$ on the datapoints P. Calculate the eucledian distance of transformed datapoints Q from μ and report it using barplot . Plot points in Q along with datapoints in Y. Make sure that the color of point before and transformation doesn't change. Comment on the difference in eucledian distance before and after transformations of the points in P. [15 marks]

In this part we have calculated the transformed data points named as Q and plotted it with Y.



From these plots we can see that as we have transformed both the data points X and sampled points so ultimately it follows the Mahalanobis decision Classifier and tends to be in the elliptical form and so euclidean distance of all points are less than 5 and all points comes in the initial circle. And the direction of the ellipse is followed by the eigenvectors as we have plotted above; it makes its shapes like the eigenvectors .