Pattern Recognition & Machine Learning

Lab-4 :- *“Boosting & Bayes Classification”*

Objectives

Following were the portion-wise objectives of the whole Lab-4 :-

1. Take Decision Tree as base-learner/ weak learner and prepare AdaBoost Classifier Models with different numbers of Trees. Compare their performance in respect to Iris Dataset and perform other relevant tasks too.

2. With the help of Iris Dataset again, perform Naive Bayes Classification, use Logarithmic Discriminant Function to compute accuracy and calculate Bayes Risk associated with the Classification Process. Perform other relevant tasks too.

3. Plot different Graphs for different Probabilities and Visualise the insights of Bayes Classification through Prior,Likelihood,Evidence and calculate Posterior Probabilities for 3 different Datasets, with different number of features and classes accordingly.

Datasets

The following were the Datasets, used to fulfill the requirements of Lab-4 :-

1. Iris Dataset :- dataset (50 Samples of 3 different Iris Species(Total 150 Samples))

2. Car Dataset :- DATASET 1 (If Cost > 550, Label 1 otherwise Label 0)

3. “C1 and C2” Dataset :- c1 and c2

Dependencies

Following were the Dependencies in order to fulfill the requirements of Lab-4 :-

1. Numpy
2. Pandas
3. Sklearn
4. Seaborn
5. Matplotlib
6. Scipy

Preprocessing Methods

In the Scenario of Iris Dataset :-

1. Ordinal Encoder was used, which assigns the float value of a Categorical String-Type Label, according to its relative Alphabetical Order. Resultant Encodings were {“Iris-Setosa” : 0.0 , “Iris-Versicolor” : 1.0 , “Iris-Virginica” : 2.0}

2. For Visualisation and Posterior Probability Calculation purposes, the Continuous Features were given Quantile Ranges with their corresponding labels and converted to Discrete Values using pd.cut, as follows :-

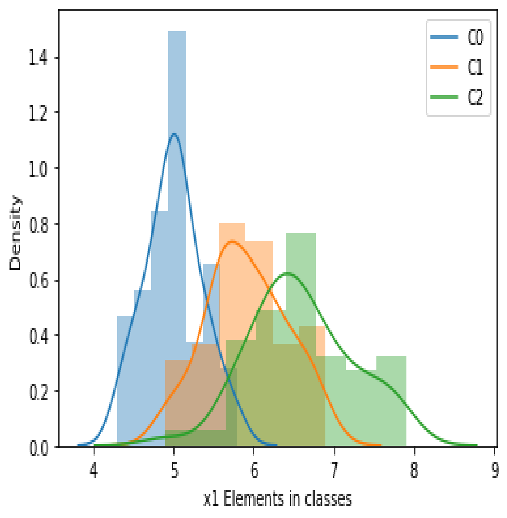
Labels	Sepal Length (cm)	Sepal Width (cm)	Petal Length (cm)	Petal Width (cm)
1	[4.3,4.9)	[2.0,2.4)	[1.0,1.983)	[0.1,0.5)
2	[4.9,5.5)	[2.4,2.8)	[1.983,2.967)	[0.5,0.9)
3	[5.5,6.1)	[2.8,3.2)	[2.967,3.95)	[0.9,1.3)
4	[6.1,6.7)	[3.2,3.6)	[3.95,4.933)	[1.3,1.7)
5	[6.7,7.3)	[3.6,4.0)	[4.933,5.917)	[1.7,2.1)
6	[7.3,7.904)	[4.0,4.402)	[5.917,6.906)	[2.1,2.502)

Note :-

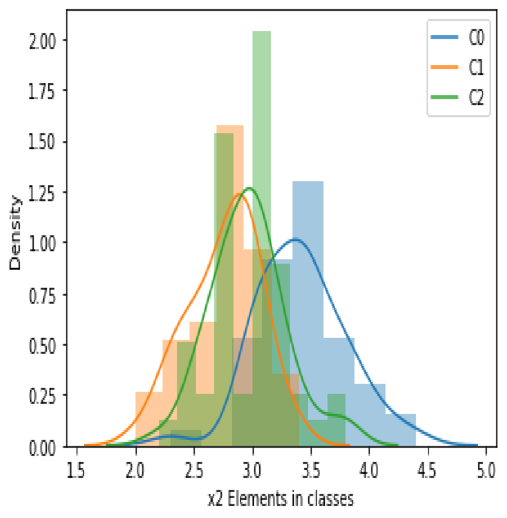
Ordinal Encoder was from “Sklearn” Dependency , whereas Cut from “Pandas” Dependency.

3. Class-Wise Continuous Histogram Plots for each Feature, were as follows :-

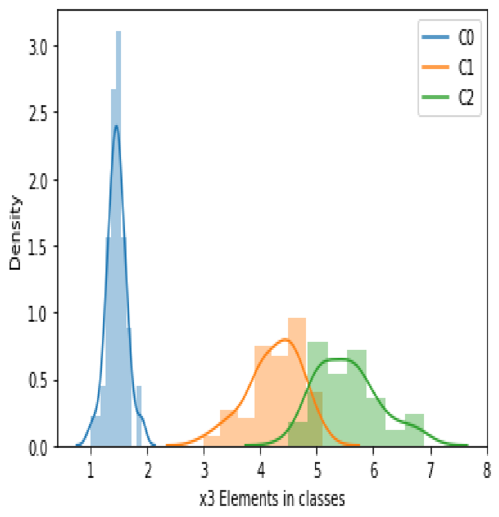
X1 Feature Histogram Plot



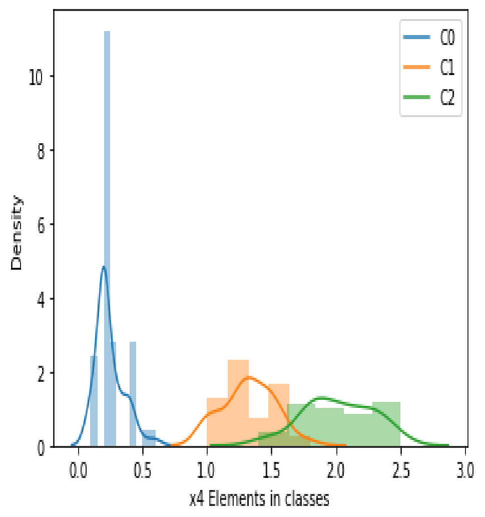
X2 Feature Histogram Plot



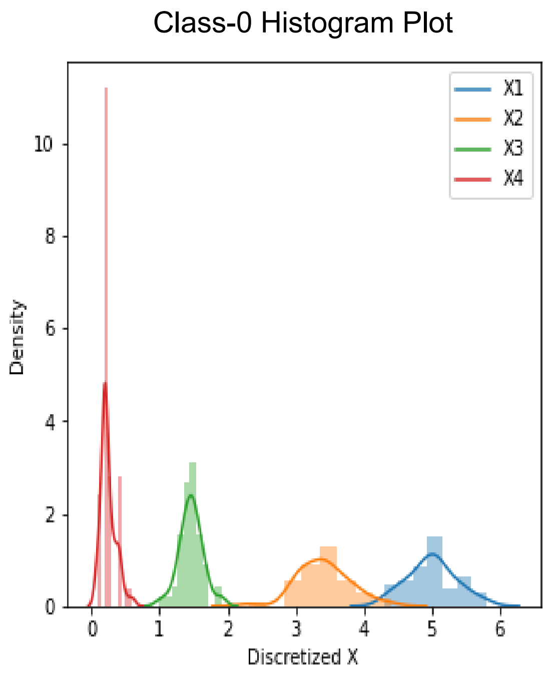
X3 Feature Histogram Plot

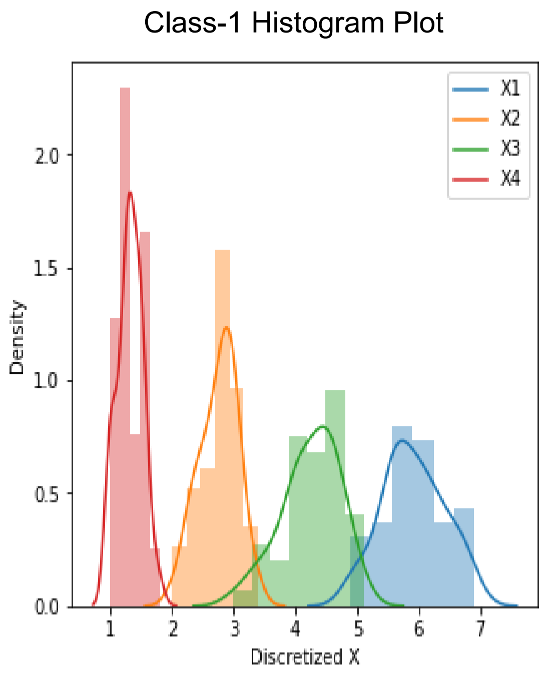


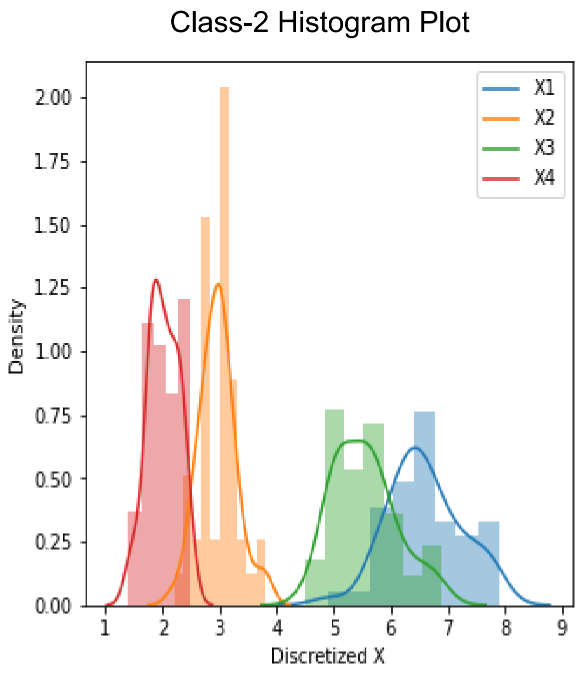
X4 Feature Histogram Plot



4. Feature-Wise Continuous Histogram Plots for each class, were as follows :-

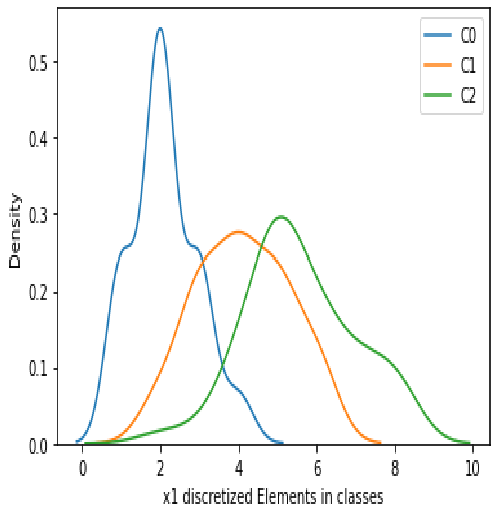




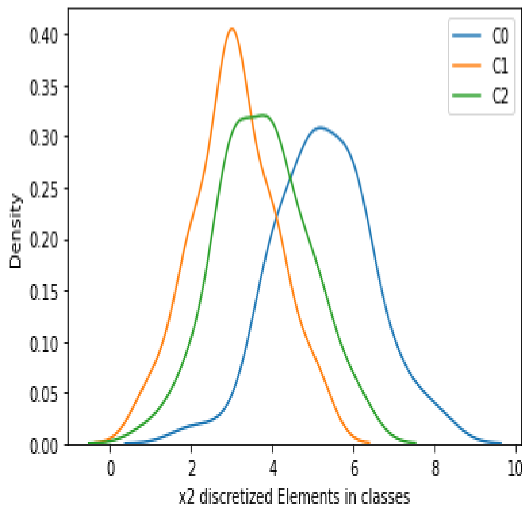


5. Class-Wise Discrete Histogram Plots for each Feature, were as follows :-

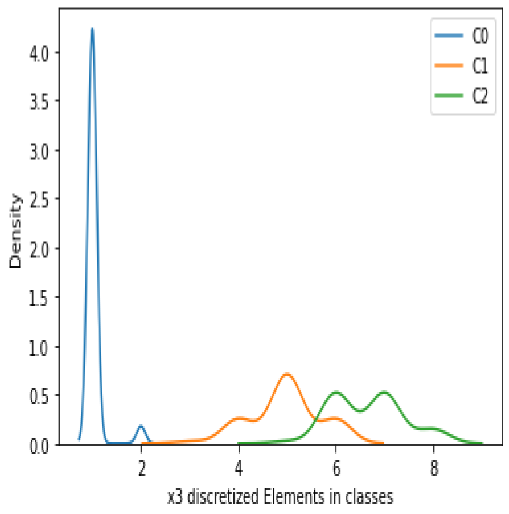
X1 Feature Histogram Plot



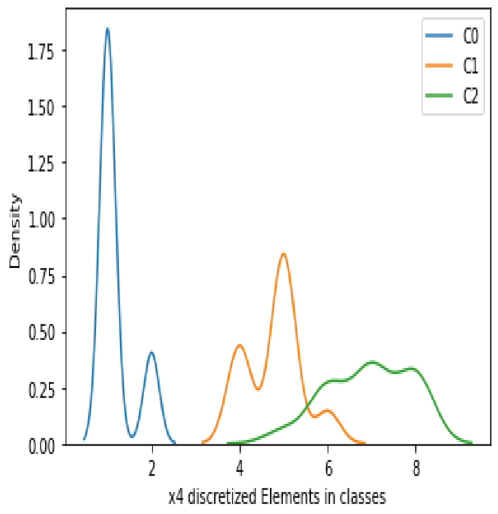
X2 Feature Histogram Plot



X3 Feature Histogram Plot



X4 Feature Histogram Plot



In the Scenario of Car Dataset :-

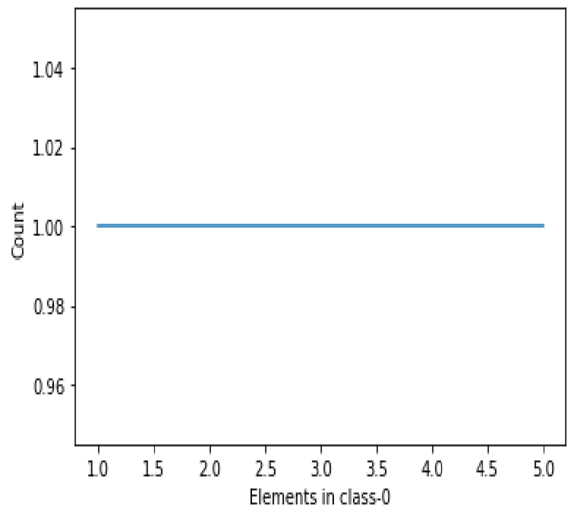
1. If Cost > 550, label was assigned as 1, otherwise 0.

2. Unique Discrete Values of Height were found out alongside their counts, as follows :-

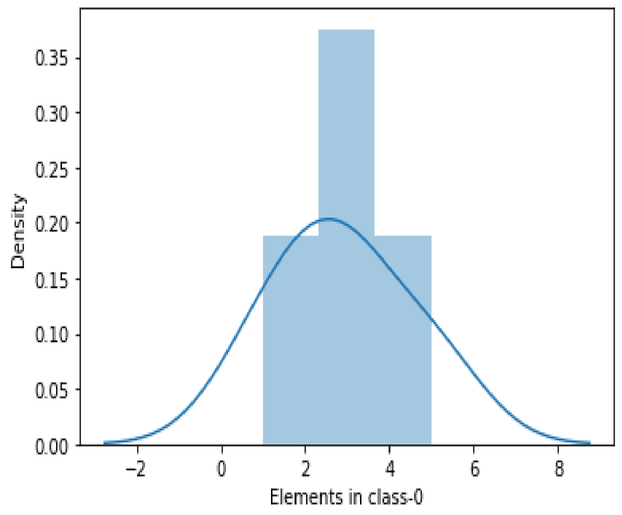
Height	Class “0” Count	Class “1” Count
1	1	5
1.5	0	5
2	0	4
2.5	1	2
3	1	6
4	0	1
5	1	0

3. Count and Density Visualisation Plots for Class-0 and Class-1 were as follows :-

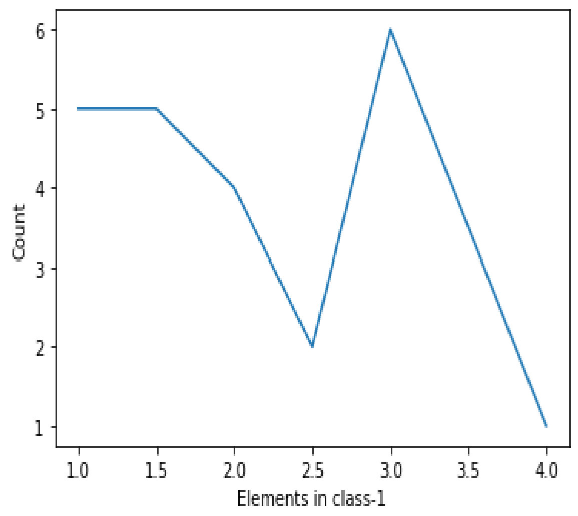
Count Plot for Class-0



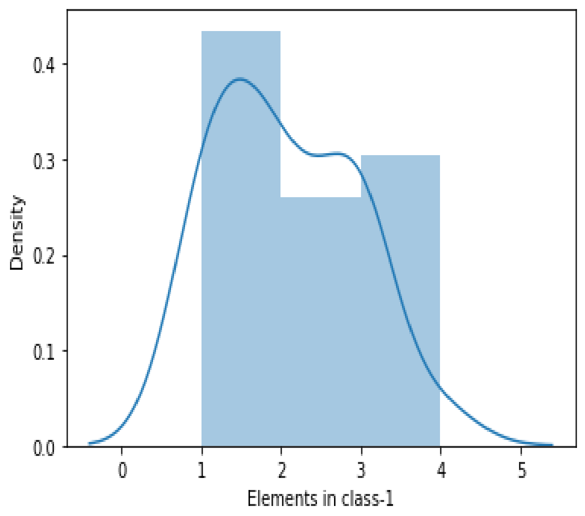
Density Plot for Class-0



Count Plot for Class-1



Density Plot for Class-1



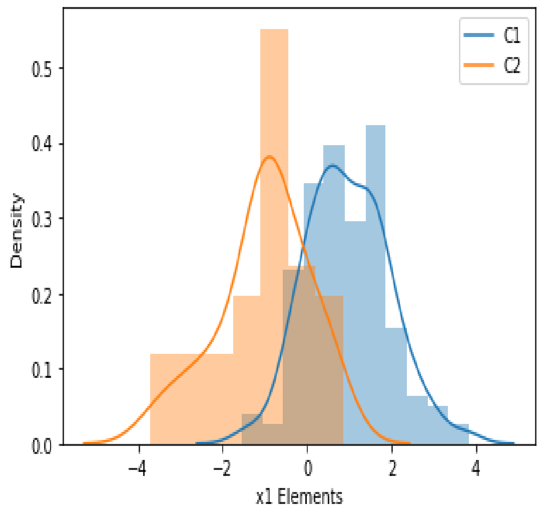
In the Scenario of “c1 and c2” Dataset :-

1. Different Datasets were already provided with their corresponding Feature Values, according to c1 and c2 classes.

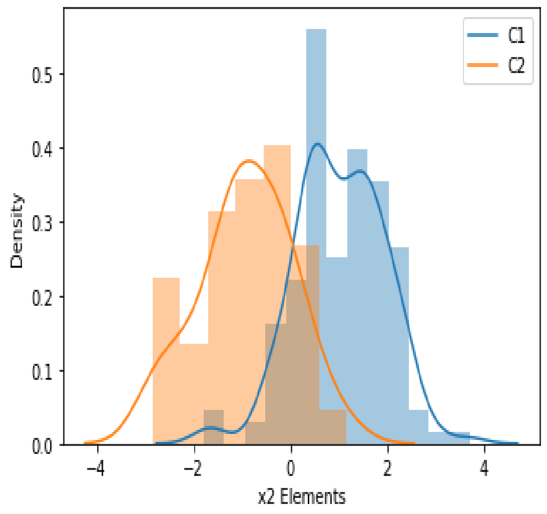
Label	X1 Feature Range	X2 Feature Range
1	[-3.702,-2.76)	[-2.846,-2.026)
2	[-2.76,-1.818)	[-2.026,-1.205)
3	[-1.818,-0.876)	[-1.205,-0.384)
4	[-0.876,0.0659)	[-0.384,0.437)
5	[0.0659,1.008)	[0.437,1.258)
6	[1.008,1.95)	[1.258,2.079)
7	[1.95,2.892)	[2.079,2.899)
8	[2.892,3.842)	[2.899,3.727)

2. Class-Wise Continuous Histogram Plots for x1 and x2 elements, were as follows :-

Histogram Plot wrt. X1 elements

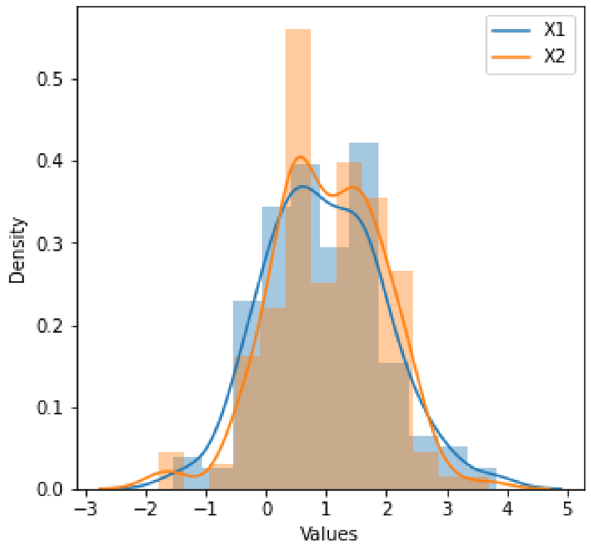


Histogram Plot wrt. X2 elements

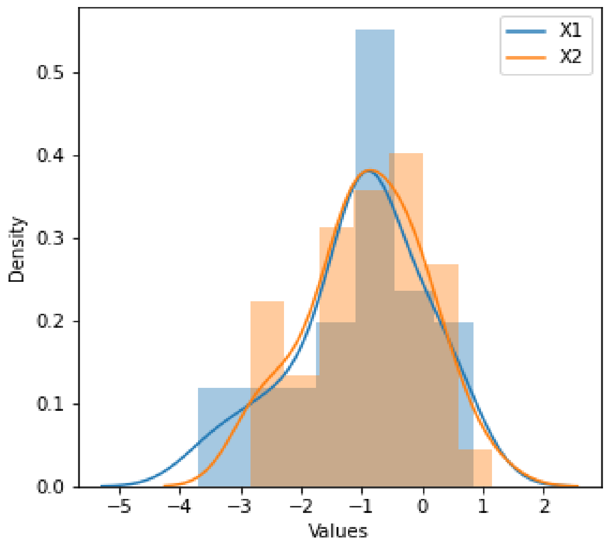


3. Feature-Wise Continuous Histogram Plots for c1 and c2 classes, were as follows :-

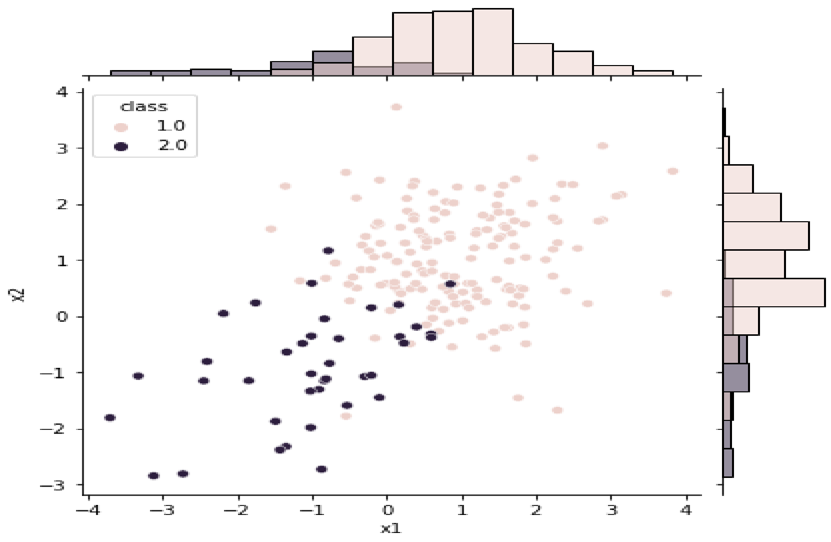
Histogram Plot wrt. C1 classes



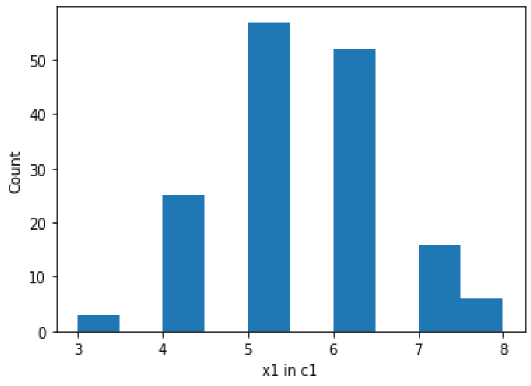
Histogram Plot wrt. C2 classes

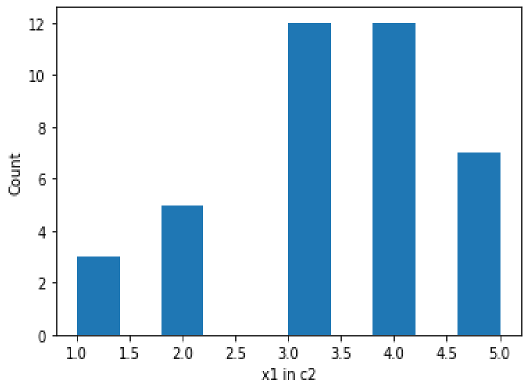


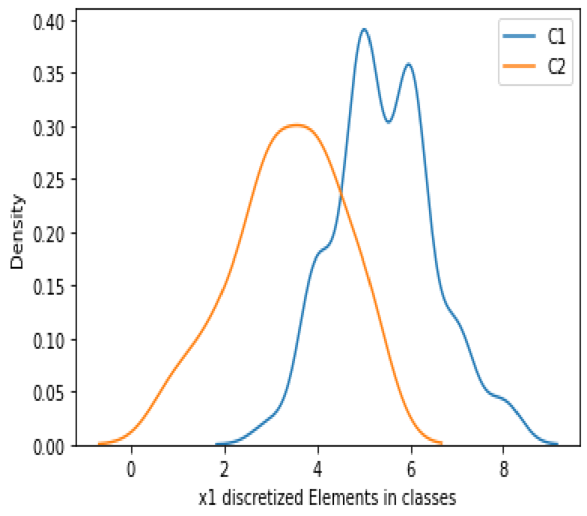
4. Joint Scatter and Histogram Plot for “C1 and C2 Dataset”, was as follows :-

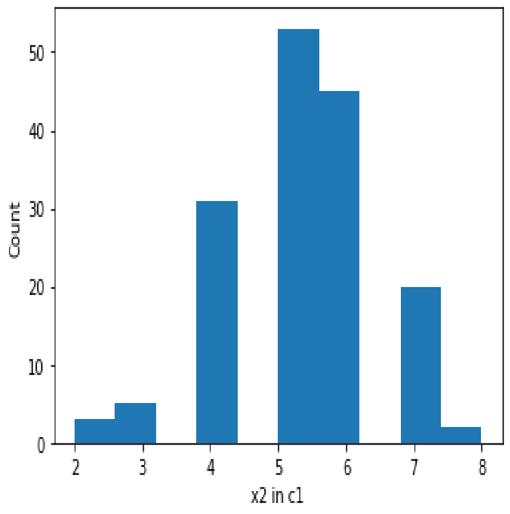


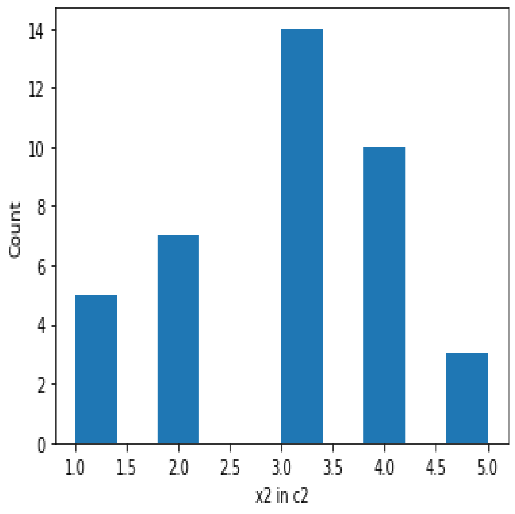
5. Discrete Histogram Plots and Line Plots for Class-wise Feature distribution, were as follows :-

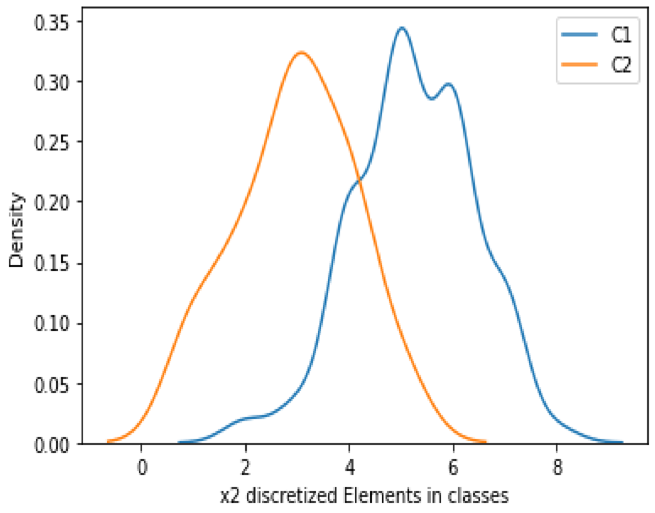






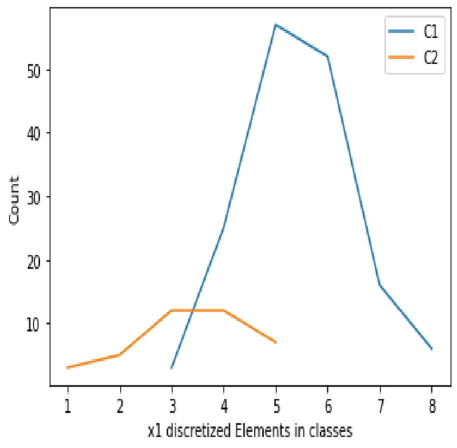




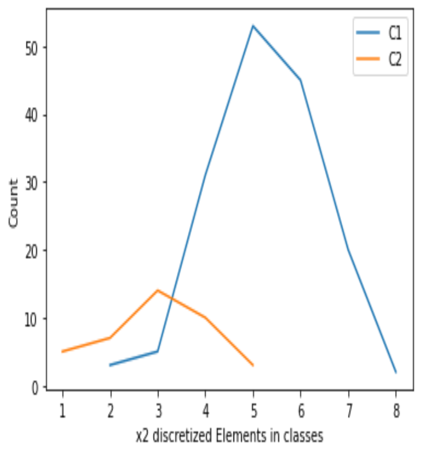


6. Class-Wise Line Plots for Unique Discrete Labels of each Feature, were as follows :-

Class-Wise X1 Line Plot



Class-Wise X2 Line Plot



Boosting with Decision Tree

Procedures

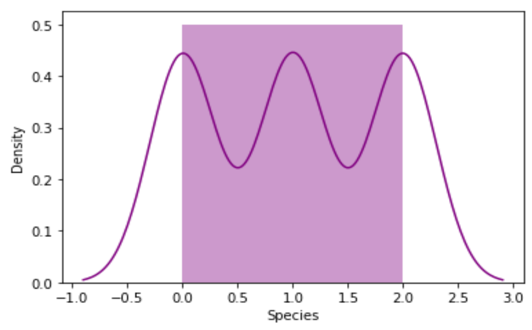
1. Visualize the Target Variable and distribution of each sample for each Feature amongst 4 features.

2. Choose Decision Tree as Weak Learner, tune its parameter and train the boosting classification model, using it.

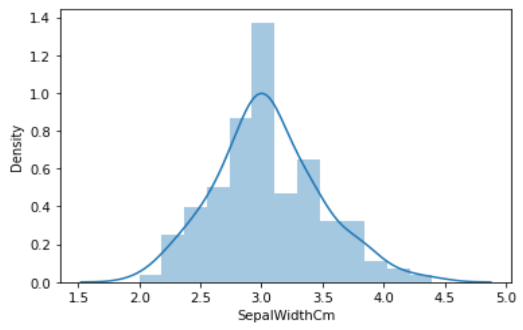
3. Prepare AdaBoost Model with Decision Tree as Weak Learner, vary its parameters and compare their performance.

Visualisations

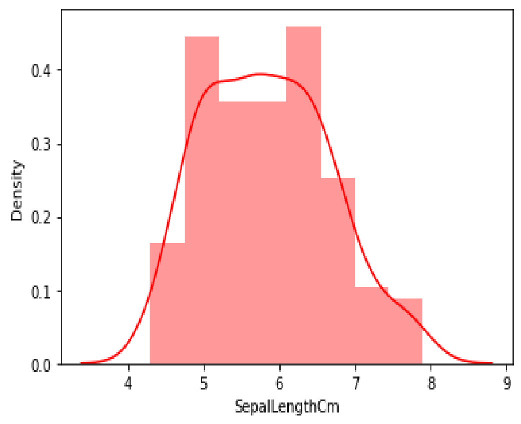
Target Variable



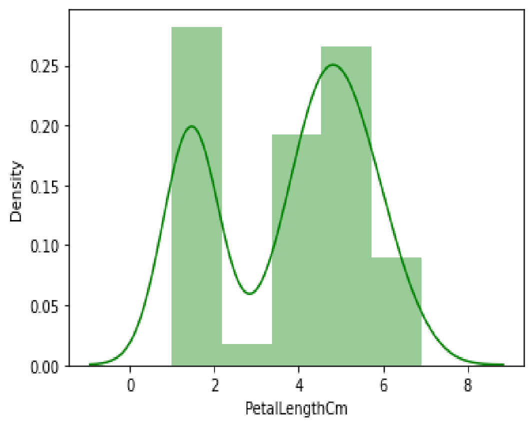
Sepal Width (cm)



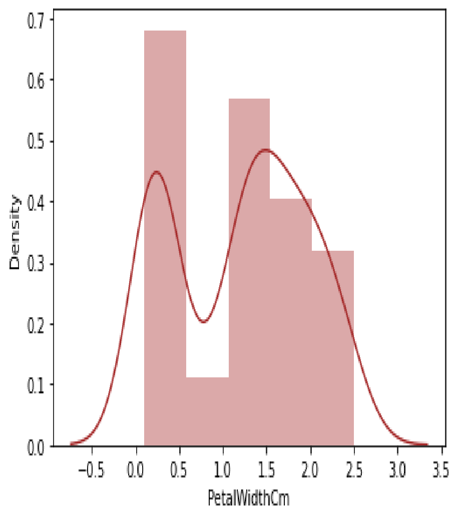
Sepal Length (cm)



Petal Length (cm)



Petal Width (cm)



Stratified K-Fold Split

Unlike the classic method of Train_Test_Split where we divide the Training and Test portions according to mentioned split ratio and random state parameter, Stratified K-Fold Split allows the user to prevent Biasness of Multi-Class Dataset, as it constitute one more parameter, i.e number of different Folds/Combinations for train-test splits. The Model/Estimator having this type of Training-Test Split, have more varieties/options to try randomized input, which will lead to lesser Biasness towards a particular Input Class, amongst other classes.

Decision Tree as Weak Learner

After using Stratified K-Fold Split, the Training Data portion was used for Decision Tree with Max Depth = 2. Following, were some details, obtained from Weak Learner, as follows :-

1. Cross-Validation Scores - [0.91667 , 0.95833 , 0.95833 , 0.95833 , 0.91667]
2. Mean Score - 0.94166
3. Standard Deviation of Scores - 0.02041
4. Variance of Scores - 0.00041
5. Test Accuracy - 96.66667 %

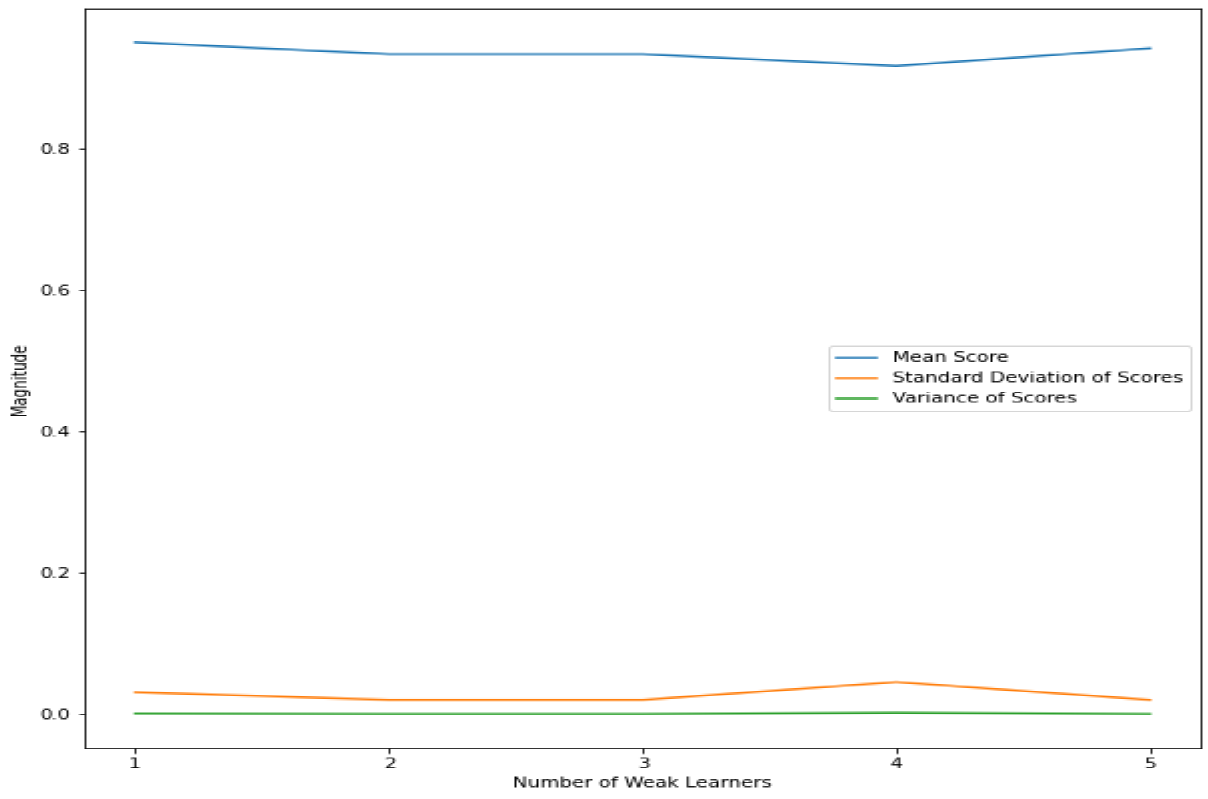
AdaBoost with Varying Number of Trees

Adaboost with Number of Trees (Weak Learners) from 1 to 5, were made and details were drawn out as follows :-

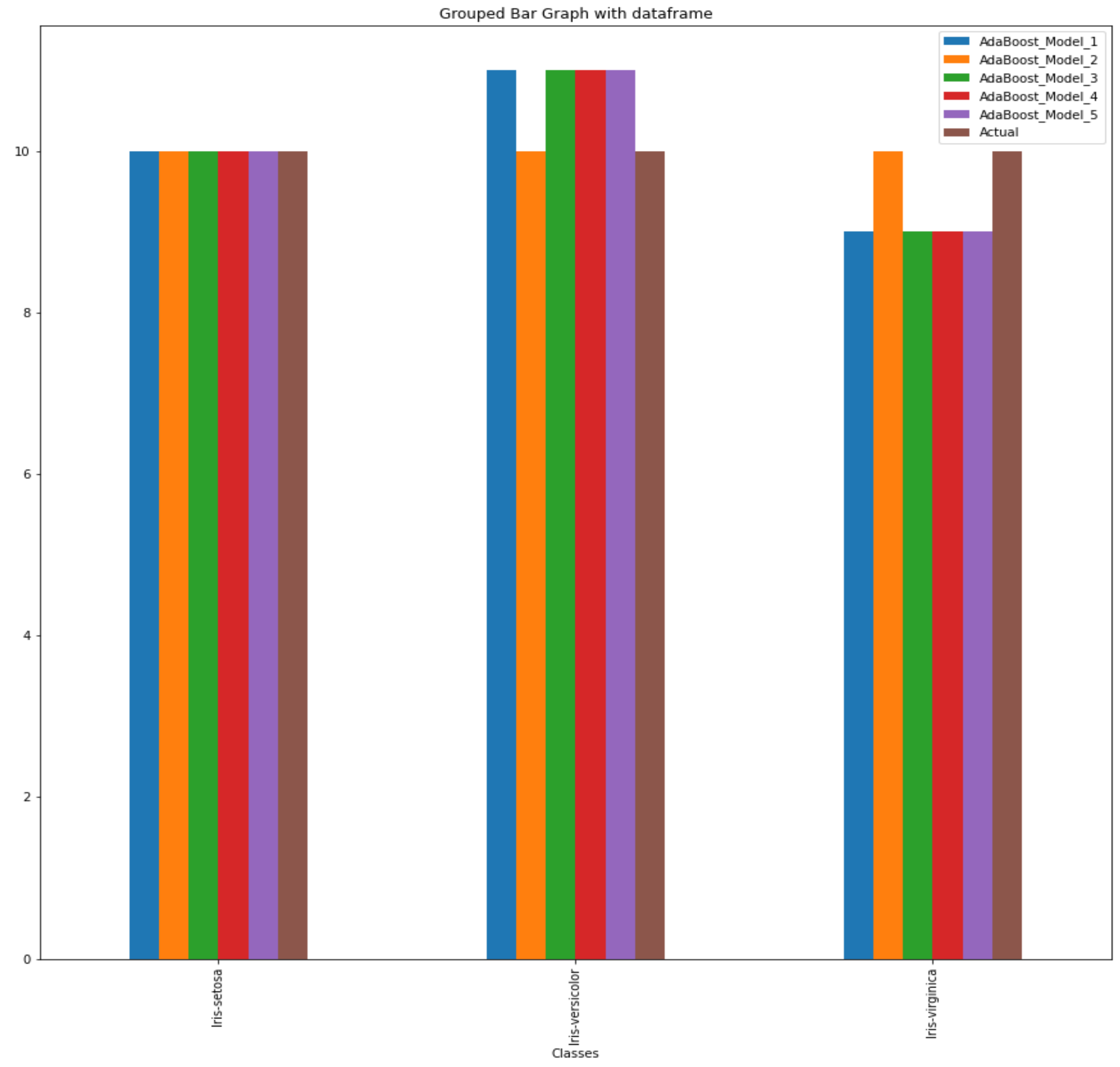
No.of Trees	Mean CV Scores	Std. of CV Scores	Var. of CV Scores	Test Accuracy
1	0.95	0.0311	0.0009	96.667%
2	0.9333	0.0204	0.0004	100%
3	0.9333	0.0204	0.0004	96.667%
4	0.9166	0.0456	0.0002	96.667%
5	0.9416	0.0204	0.0004	96.667%

To make things more simple, some of the useful plots , were as follows :-

Line Plot for Cross-Validation Score Analysis of each Adaboost Model



Class-Wise Grouped Bar Plots for Prediction analysis of each model, in respect to Actual



Bayes Classification

Procedures

1. Estimate the accuracy of different Naive Bayes algorithms, use 5-fold cross validation on the data set and plot the ROC AUC curve for different values of parameters.

2. Use a linear discriminant function to calculate the accuracy on the classification task with 80% training and 20% testing data split.

3. Calculate the Bayes risk, corresponding to each Naive Bayes Algorithm.

   Given: $\lambda = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

Different Naive Bayes Algorithms

4 Different kinds of Naive Bayes Algorithms were used, as following :-

1. Gaussian Naive Bayes

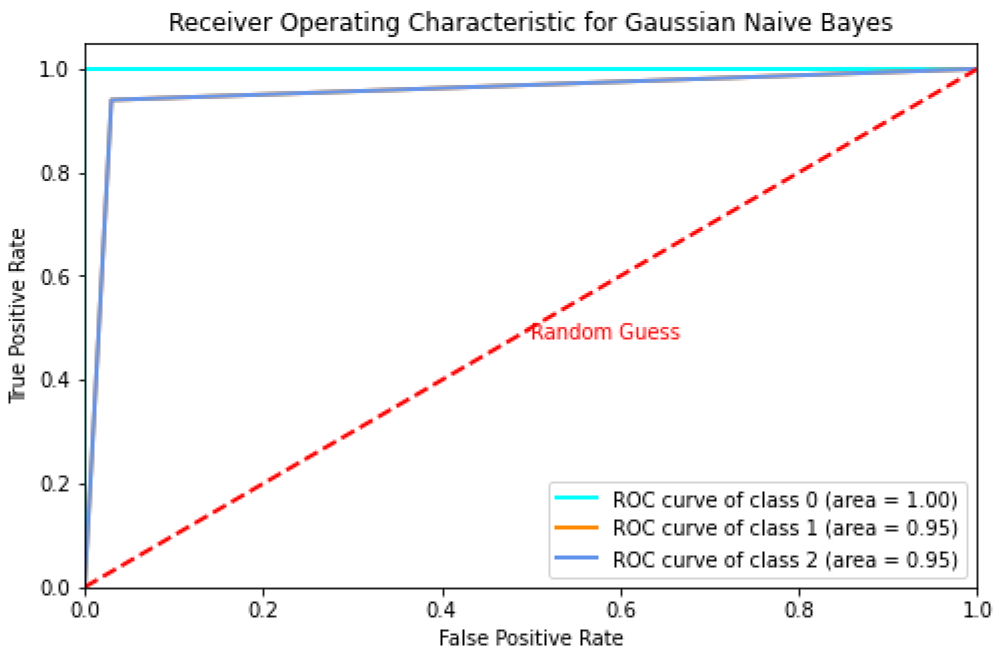
2. Multinomial Naive Bayes

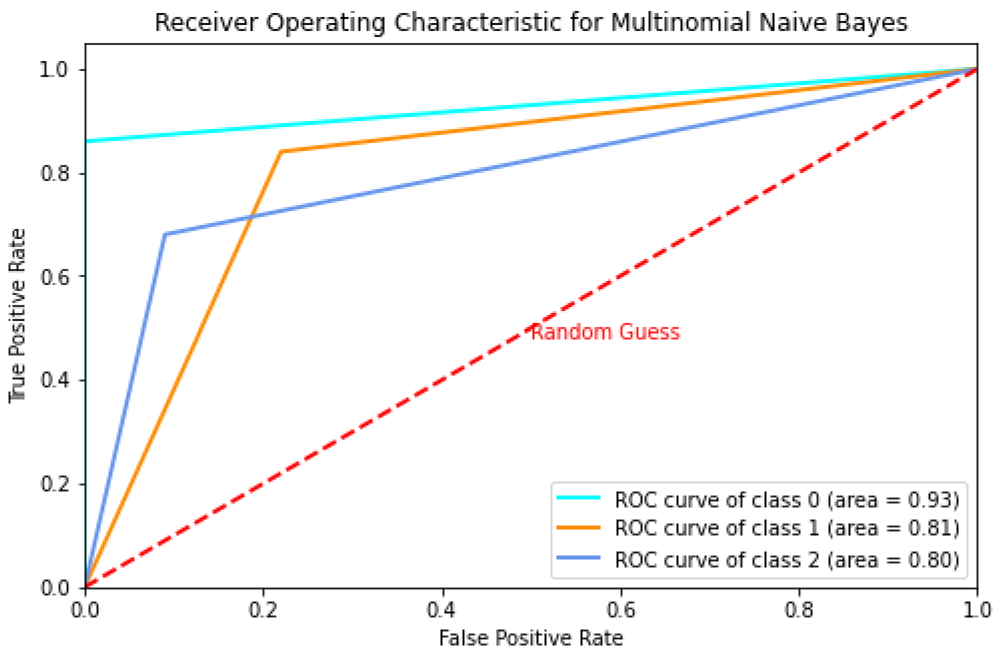
3. Complement Naive Bayes

4. Bernoulli Naive Bayes

Also, for accuracy check on Classification Task, Logarithmic Discriminant Function was used with 8:2 split ratio for training-test data points.

ROC-AUC Curves for Multinomial and Gaussian Naive Bayes Models, were as follows :-

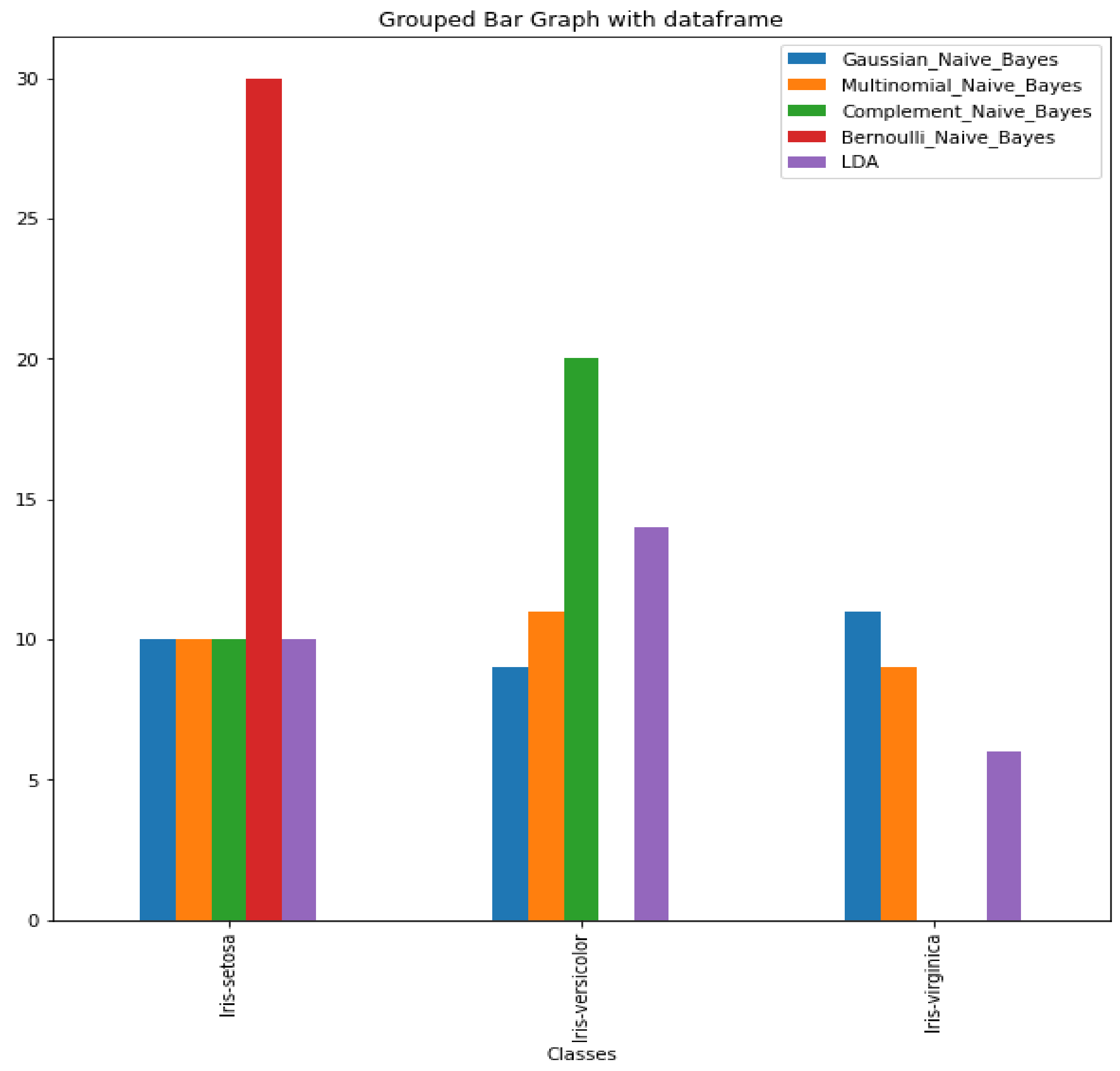




Some of the Details regarding these Algorithms were drawn, as follows :-

Naive_Bayes Model	Mean CV score	Std. of CV scores	Var. of CV scores	Test Accuracy
Gaussian	0.95	0.0485	0.0023	96.667%
Multinomial	0.95833	0.0456	0.0020	96.667%
Complement	0.66667	0	0	66.667%
Bernoulli	0.33334	0	0	33.334%
LDA	0.96668	0.0311	0.0009	100%

Grouped Bar Plots for Predictions according to Classes and Models , were as follows :-



Bayes Risk

Given: λ= $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

Bayes Risk for a particular Discrete Point x is as follows :-

$$R(\alpha_i|\mathbf{x}) = \sum_{j=1}^{c} \lambda(\alpha_i|\omega_j)P(\omega_j|\mathbf{x})$$

In this Lab-4, Train-Test Split ratio was 8:2, which led to 30 Dataset Samples in Test Portion. Just for Comparison, another Lambda Matrix was taken from Zero-One Loss Method, where Diagonal Elements becomes 0 and other elements remain 1. Total Bayes Risk Values for Whole Test Samples, were drawn as follows :-

Naive_Bayes Models	Zero-One Loss Method	Custom Lambda Matrix
Gaussian	59.9999	451.6251
Multinomial	59.9999	457.7253
Complement	60.0000	448.4974
Bernoulli	60	450
LDA	60	438.0873

Bayes Visualisations

Formula for calculating Posterior Probabilities, os as follows :-

$$\underline{posterior} = \frac{likelihood \times prior}{evidence}.$$

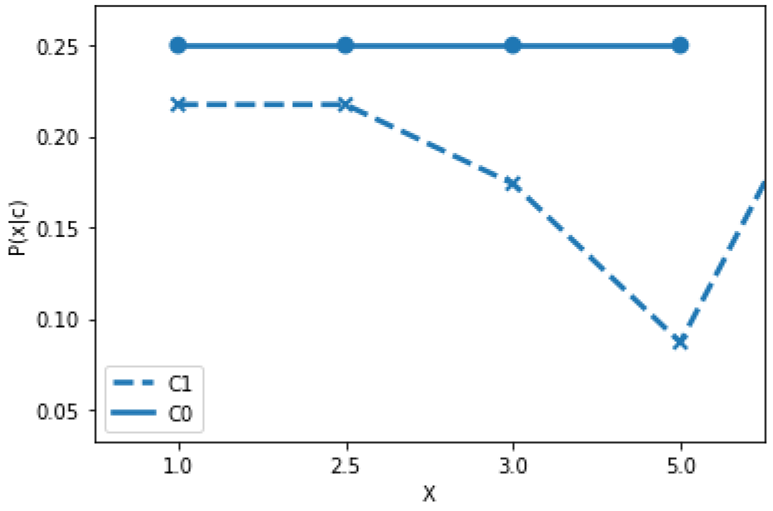
Where :-

Likelihood is P(X | Wi) , Prior is P(Wi) and Evidence is as follows :-

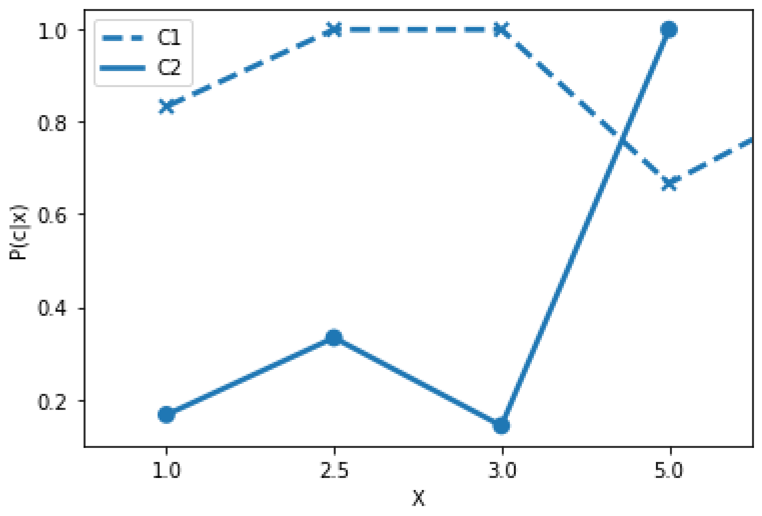
$$p(\mathbf{x}) = \sum_{j=1}^{c} p(\mathbf{x}|\omega_j)P(\omega_j).$$

In the scenario of Car-Dataset :-

1. Likelihood Plot -

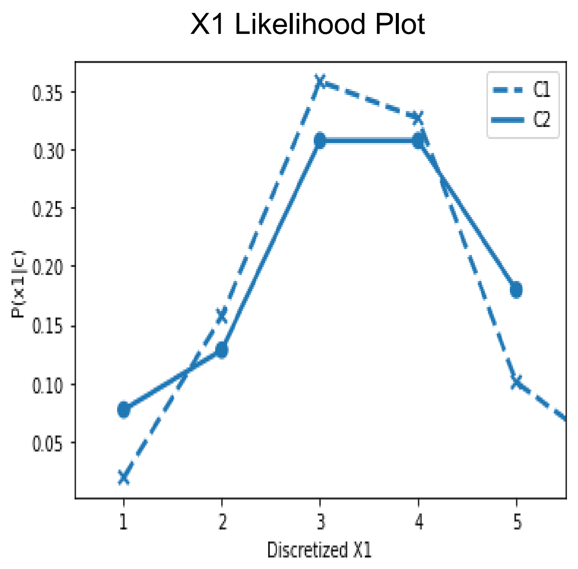


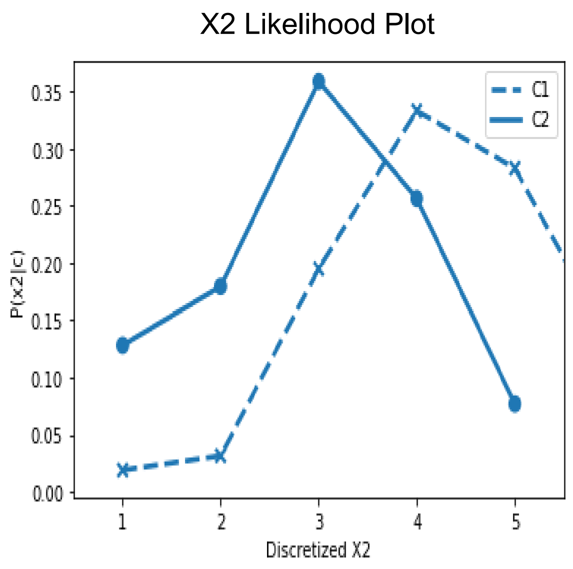
2. Posterior Plot -



In the Scenario of “C1 and C2” Dataset :-

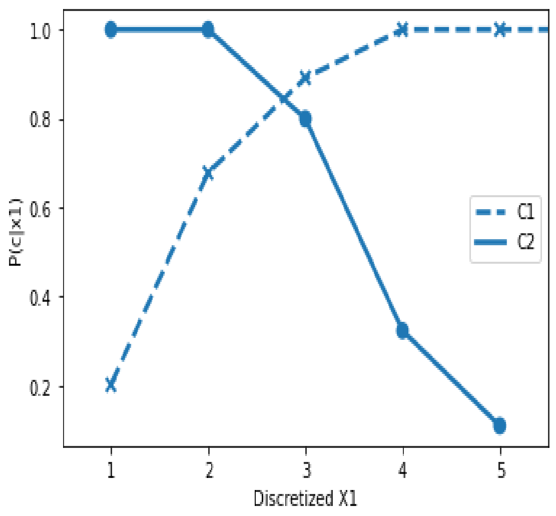
1. Likelihood Plots -



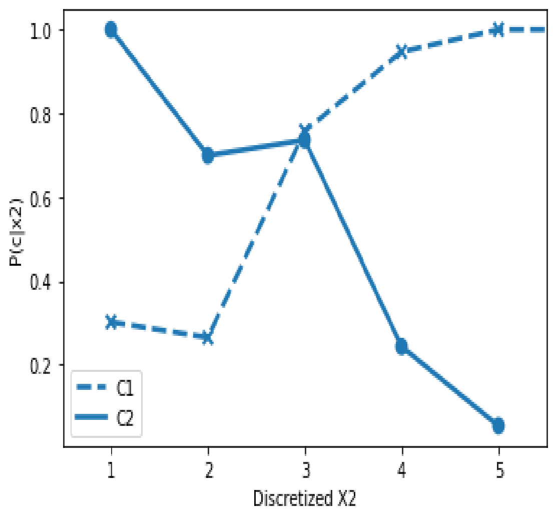


2. Posterior Plots -

X1 Posterior Plot



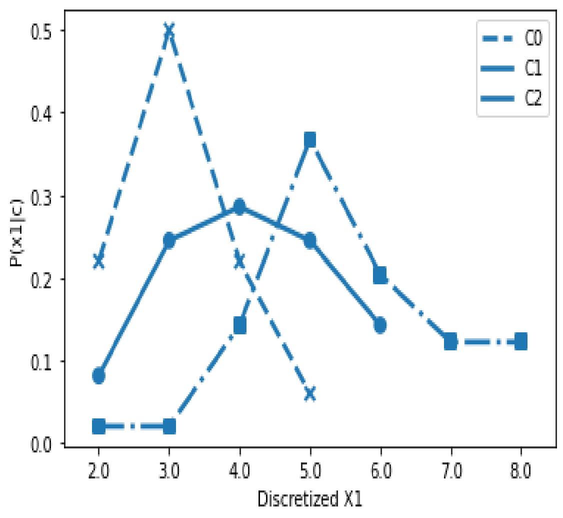
X2 Posterior Plot



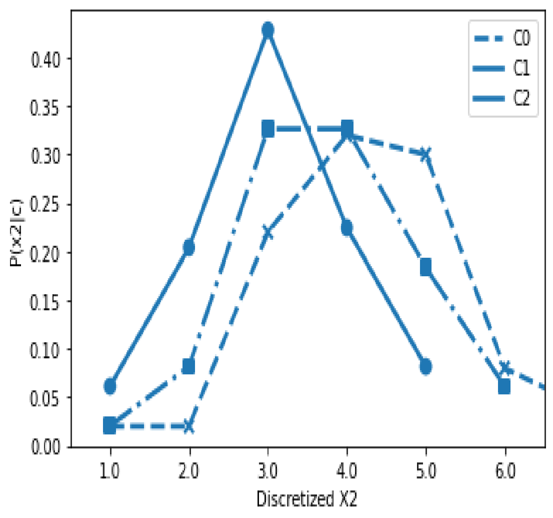
In the Scenario of “Iris” Dataset :-

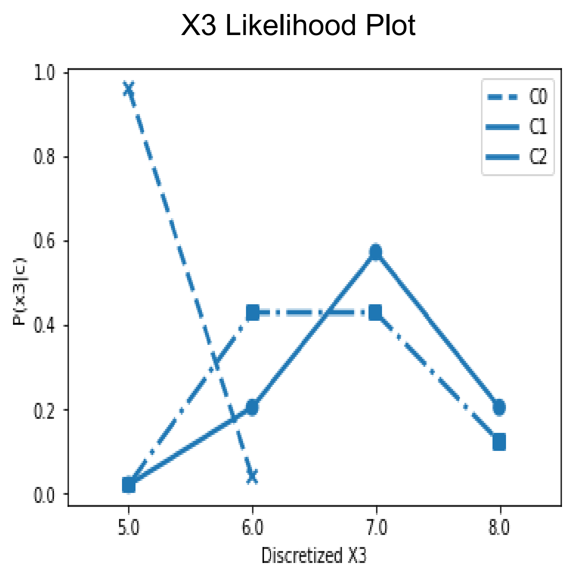
1. Likelihood Plots :-

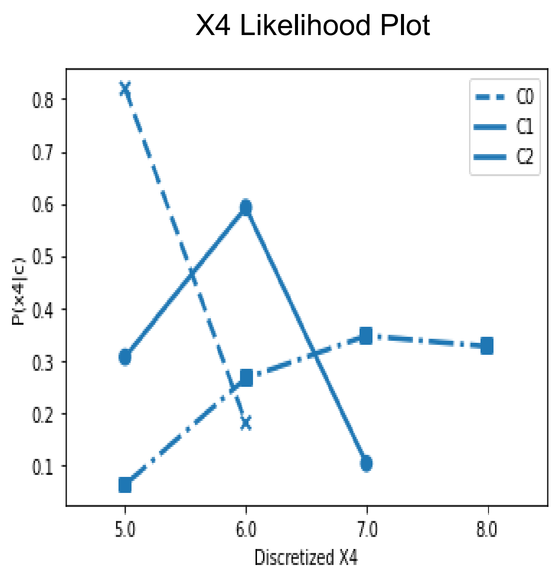
X1 Likelihood Plot



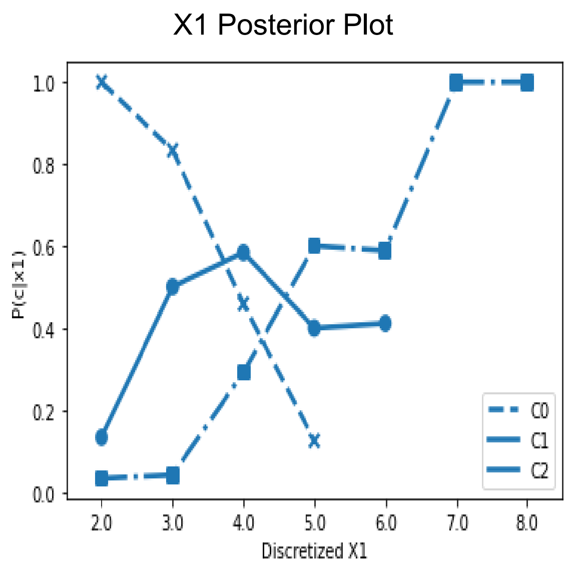
X2 Likelihood Plot

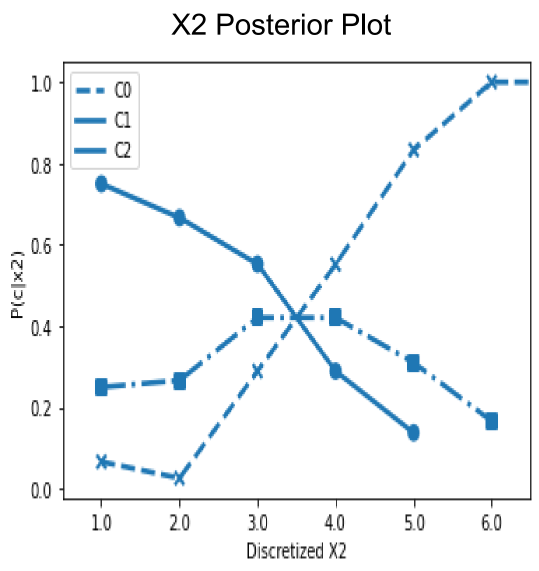




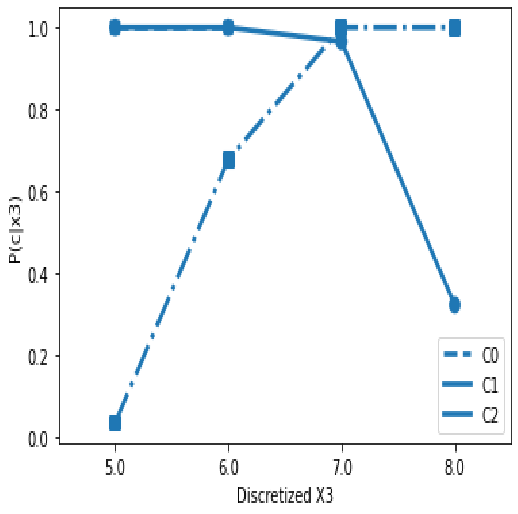


2. Posterior Plots :-

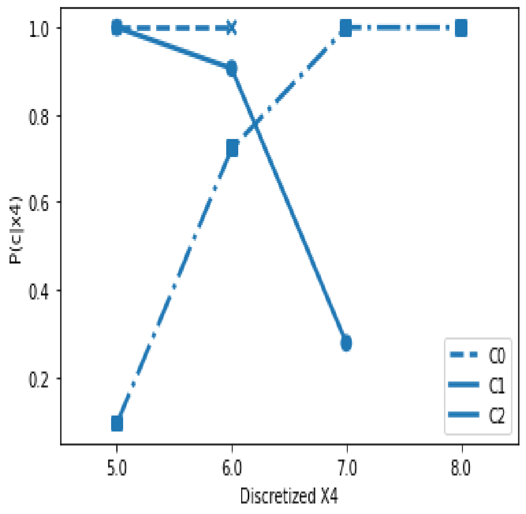




X3 Posterior Plot



X4 Posterior Plot



Conclusion

1. In Order to deal with Continuous Dataset for Naive Bayesian Algorithms, convert them to Discrete Bins and assign Labels altogether.

2. Although Decision Tree with Max_Depth = 2 was a weak learner, it gave an accuracy of 96.67% on Iris Dataset.

3. Bayesian Risk of Whole Test Dataset for given lambda matrix was maximum for Multinomial Naive Bayes and least for LDA Model.

4. Amongst Naive Bayesian Models, Bernoulli Naive Bayes Model performed worst for Iris Dataset, with 33.34% accuracy. The Reason might be because its purpose is in Binary Classification only, not in Multi-Class Classification.