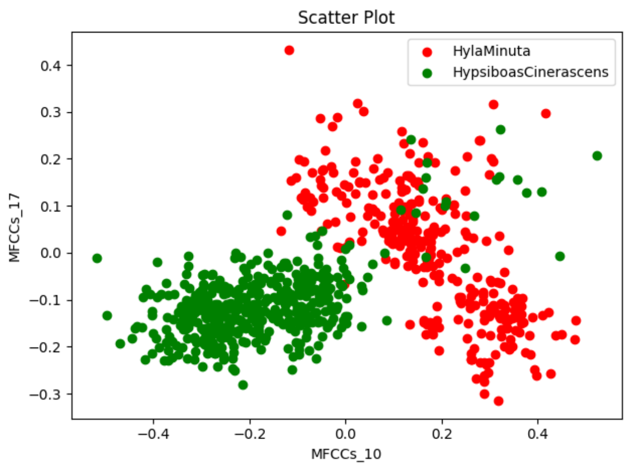
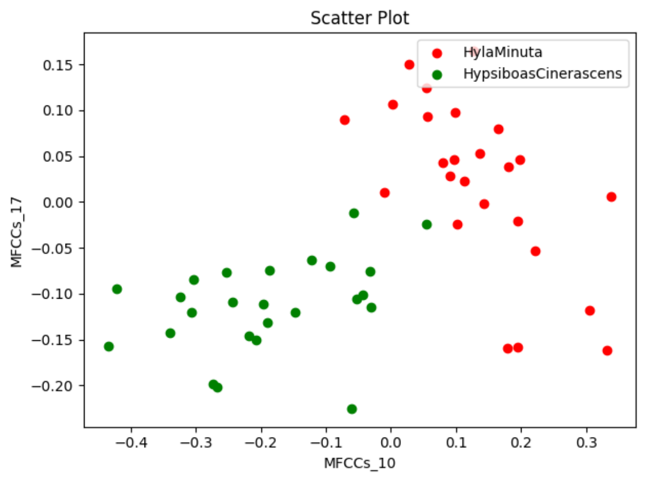
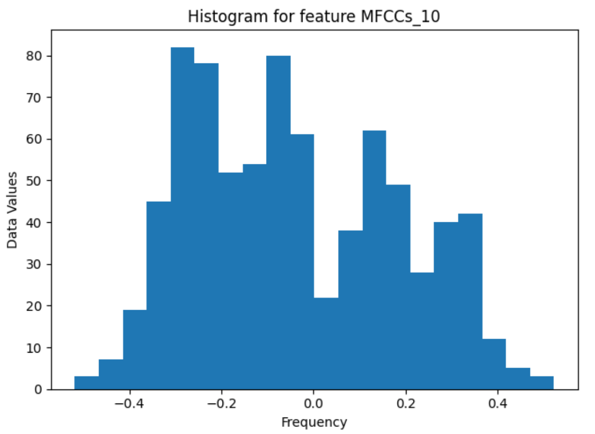
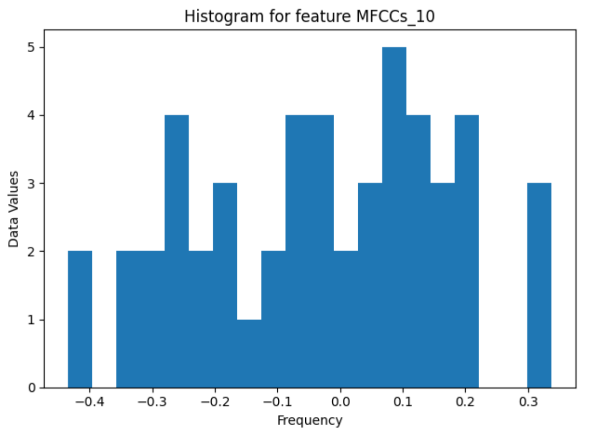
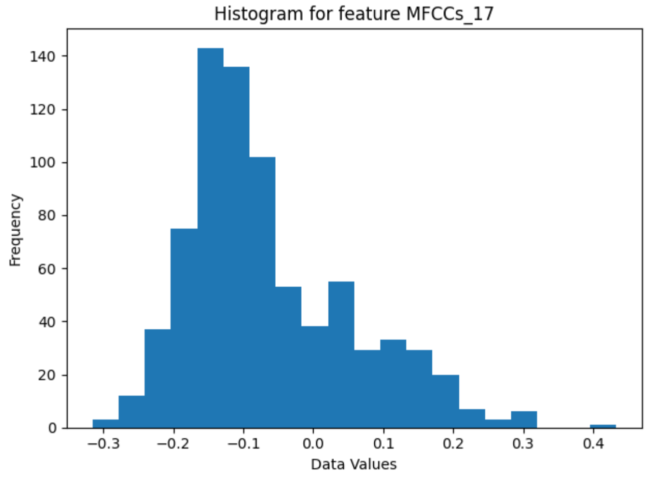
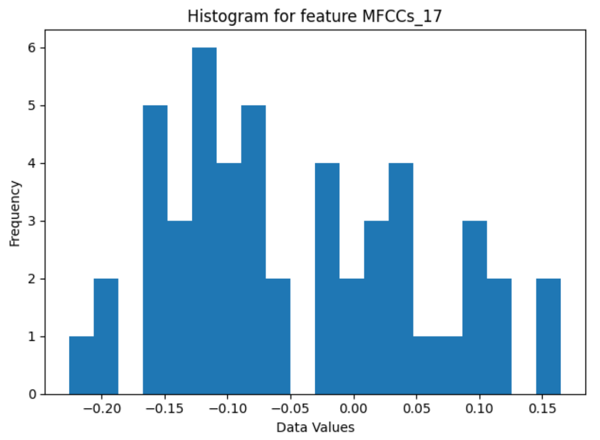
Introduction to Machine Learning

Assignment 1

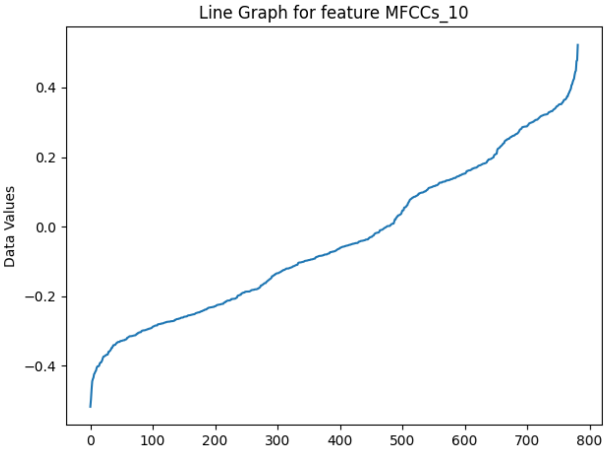
Divesh Badod

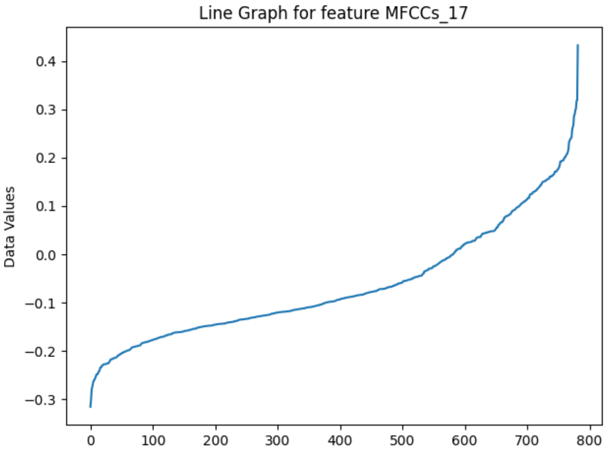
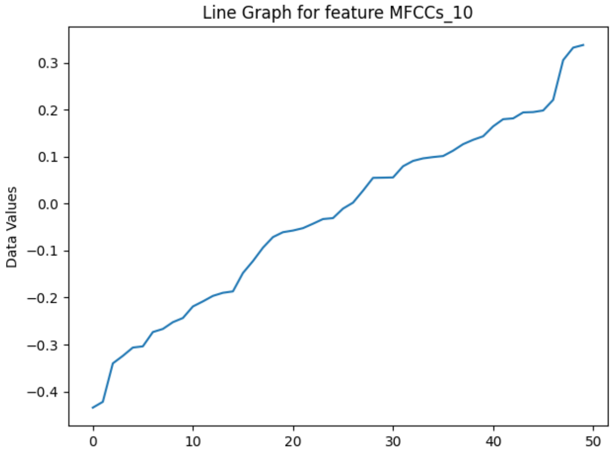
Frogs.csv	*Frogs-subsample.csv*
	

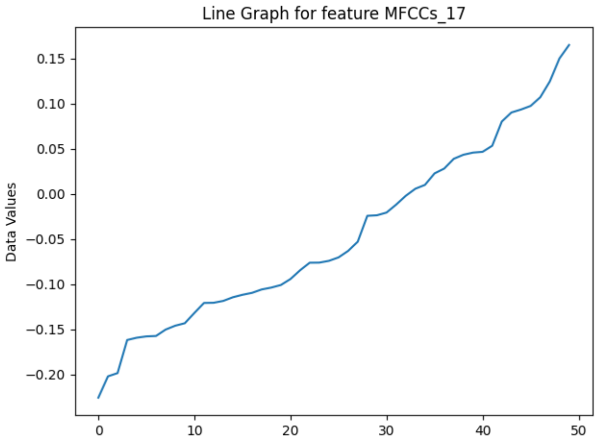
Scatter plots are mathematical diagram which displays typically two variables of a dataset. In our case the scatter plot diagram displays the two classes which are classified in the datasets named HylaMinuta and HypsiboasCinerascens. These plots help us determine the variable dependencies and independencies. In machine learning scatter plots are usually used while training a model usually a binary classifier to build a decision boundary separating the two classes. In this case, later we will be portraying the decision boundary used to classify between these two classes.

Frogs.csv	*Frogs-subsample.csv*
	
	

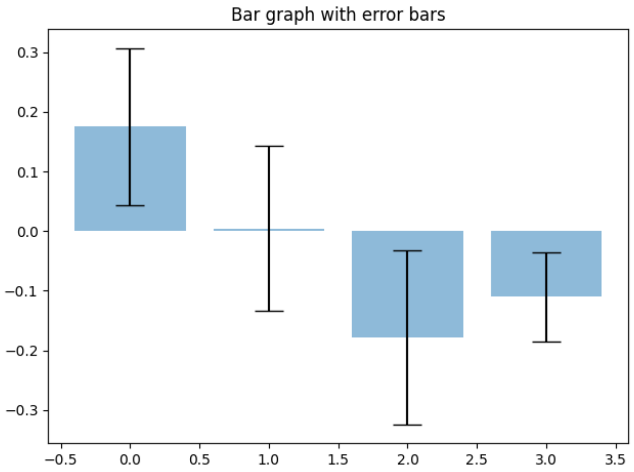
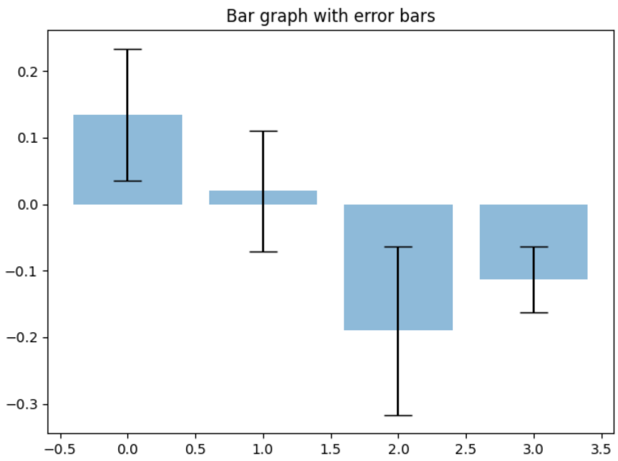
Histograms are used to show the frequency distributions of data values on the datasets. For machine learning, it is important to learn about the frequency distribution to prepare a model with a less steep learning curve hence histograms play an important role in ML. (Note: A small mistake of labelling the axes in MFCCs_10 distributions X-axis should be data values and Y-axis should be frequency similar to the labels in MFCCs_17 distributions)

<table>
<tr><th>Frogs.csv</th><th>Frogs-subsample.csv</th></tr>
<tr><td>


</td><td>

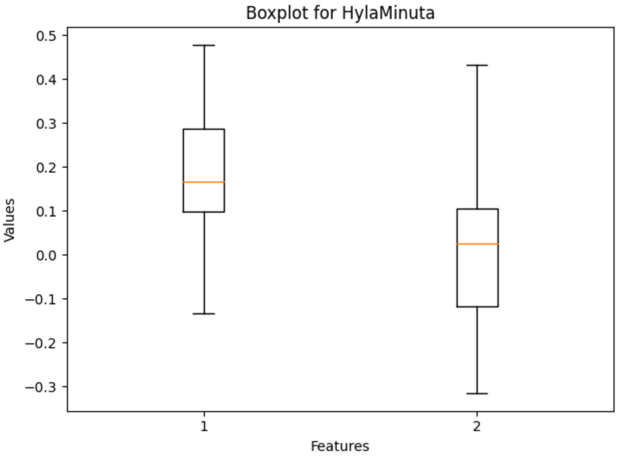

</td></tr>
</table>

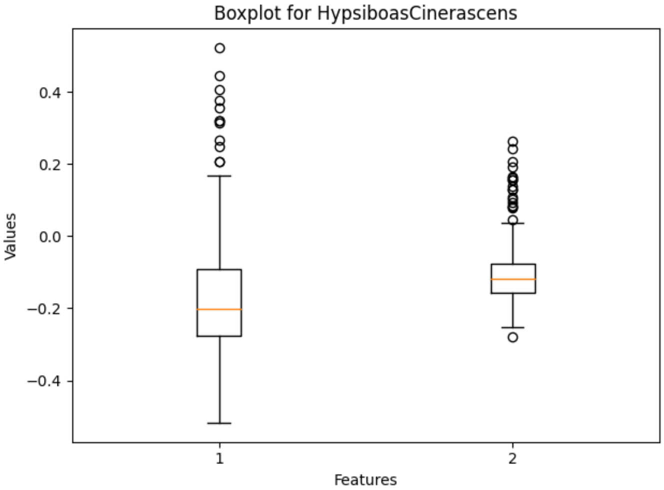
Line graphs is a graphical display of data that changes continuously over time. Since our data set is sorted before making a line graph it shows an increase in value as you move through the data. This graph is mostly used to depict the range and the continuous change in the values in the dataset.

<table>
<tr><th>Frogs.csv</th><th>Frogs-subsample.csv</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

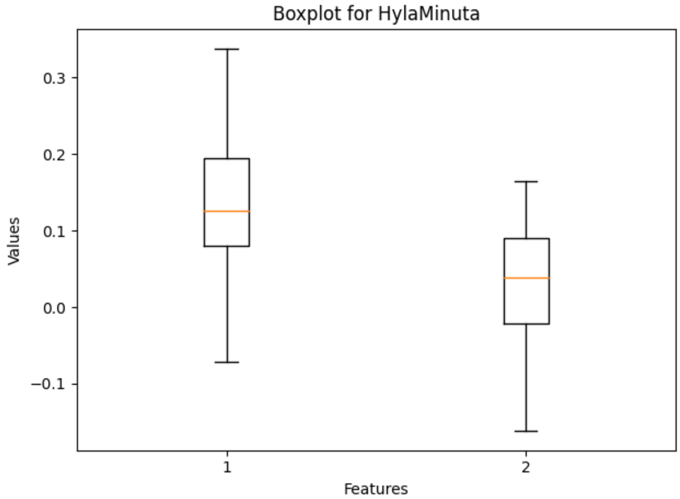
Error bar is a line that passes through a value, in this case, a range of values to represent the uncertainty or the variation of the corresponding values.

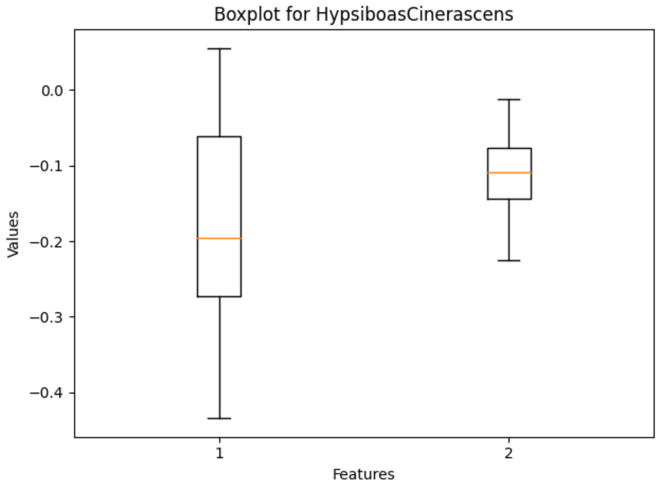
Frogs.csv





Frogs-subsample.csv



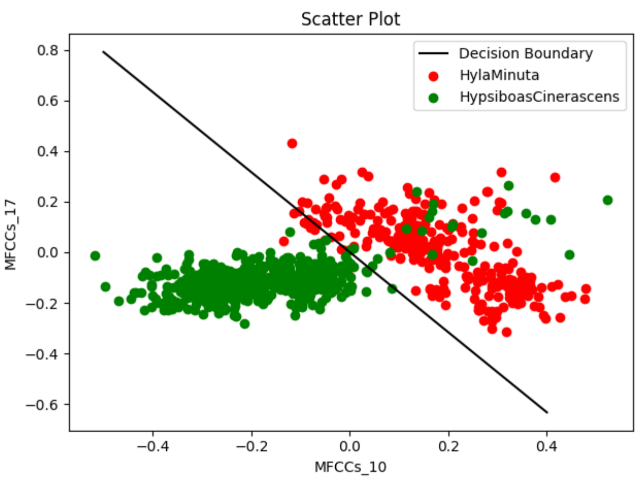


Box plots are used to show the shape of the distribution, its central value, and its variability. In a box plot, the ends of the box are the upper and lower quartiles, so the box spans the interquartile range. the median is marked by a vertical line inside the box. The dots show the outliers present in the data.(Note: The features 1 and 2 are MFCCs_10 and MFCCs_17 respectively)

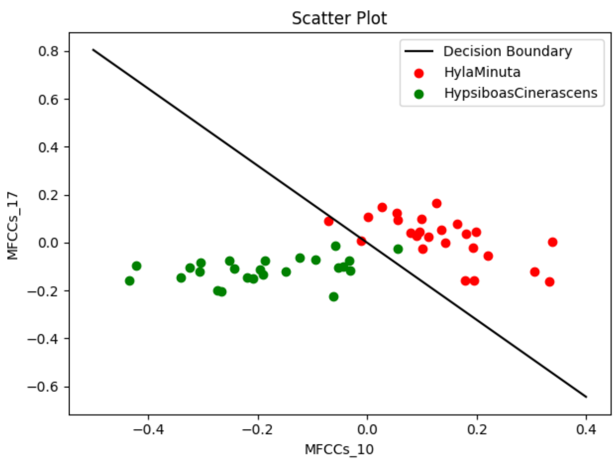
Frogs.csv(Class1:HylaMinuta, Class2: HypsiboasCinerascens)	*Frogs-subsample.csv(Class1:HylaMinuta, Class2: HypsiboasCinerascens)*
Mean MFCCs_10: -0.038321155240409216 Mean MFCCs_17: -0.06451480131585678 Mean MFCCs_10 in class 1: 0.175102095561 Mean MFCCs_17 in class 1: 0.004275941151 Mean MFCCs_10 in class 2: -0.17849320555 Mean MFCCs_17 in class 2: -0.10969516183 Covariance Matrix: [[0.0496062 0.00859857] [0.00859857 0.0140389]] Covariance Matrix class 1: [[0.01734669 -0.01235759] [-0.01235759 0.01912491]]	Mean MFCCs_10: -0.028011301779999997 Mean MFCCs_17: -0.04633025915999999 Mean MFCCs_10 in class 1: 0.13406434708 Mean MFCCs_17 in class 1: 0.02009333540000 Mean MFCCs_10 in class 2: -0.19008695063999 Mean MFCCs_17 in class 2: -0.11275385371999 Covariance Matrix: [[0.04005957 0.0092157] [0.0092157 0.00995278]] Covariance Matrix for class 1: [[0.01031672 -0.00597868] [-0.00597868 0.0085536]]

Covariance Matrix class 2: [[0.021206 0.00635566] [0.00635566 0.00557183]] Std MFCCs_10: 0.2225820435148893 Std MFCCs_17: 0.11841006605870075 Std MFCCs_10 in class 1: 0.13149422514638 Std MFCCs_17 in class 1: 0.13806960598792 Std MFCCs_10 in class 2: 0.14546844424478 Std MFCCs_17 in class 2: 0.07456558385200	Covariance Matrix for class 2: [[0.0167455 0.00236565] [0.00236565 0.00257479]] Std MFCCs_10: 0.19813728038317813 Std MFCCs_17: 0.09876093885552142 Std MFCCs_10 in class 1: 0.099519109234755 Std MFCCs_17 in class 1: 0.0906170949348600 Std MFCCs_10 in class 2: 0.1267899000142564 Std MFCCs_17 in class 2: 0.0497172039064157

Frogs.csv



Frogs-subsample.csv



The decision boundary is a hypersurface which partitions the underlying vector space into two sets or classes. A classifier is used to classify the values in the datasets on either side of the boundary. These decision boundaries in the scatter plots were made by the logistic regression model when trained on the dataset Frogs.csv and Frogs-subsample.csv. As you can see there are only two classes and the values can be classified into two sides of the decision boundary, we call this as a binary classifier as well. When training the dataset, the model uses activation function for distinguishing the values which are present in either of the classes a threshold is set as to which class this particular data value belongs to. Since here we have only two classes this threshold will classify the values depending on the output of the activation function. If the output is greater than the threshold then that dataset point or value will belong to class 2 and if the output is lesser than or equal to the threshold then that value will belong to class 1

Another method used in conjunction with the activation function here is called gradient descent. Before there is an input into the activation function the vector of raw inputs is multiplied by a vector of so-called weights which is randomly initialized essentially called a dot product. With that dot product passed through the activation function, the resulting output is then compared with the actual output and a loss calculated using a loss function. So, the main role of gradient descent is to minimize this loss function’s value and over several iterations, the weights are modified, and the loss function is minimized. The last obtained weights after going through this gradient process determine the position of the decision boundary in the vector plane. Hence after training the data with an activation function(sigmoid in this case) and using gradient descent to minimize the loss of the model we get the following decision boundaries for the respective datasets.

Date
Page

23

1 Defective = D
Shipped = S
Not defective = !D
Not shipped = !S

We know that

$P(!D) = .85$ Not defective probability

$P(D) = .15$ Defective probability

$P(!S/!D) = .1$ Not defective and not shipped probability

$P(S/!D) = .9$ Not defective and shipped "

$P(S/D) = .0\cancel{9}5$ Defective and shipped "

$P(!S/D) = .95$ Defective and Not shipped "

Using Bayes theorem to solve:-

$P(D/S) = (P(S/D)P(D)) / (P(S/D)P(D) + P(S/!D)P(!D))$

$= \frac{.05 \times .15}{(.05 \times .15 + .9 \times .85)}$

$= .0097$

SO, the ~~probability of~~ Product that is shipped has a probability of 0.0097 of being defective.

(2) Let A be generation of strings containing an even number of 1s

Let B be generation of strings ending with 1

For event A,
Total no. of strings generated = 16
Strings having even no. of 1s = 8
So, A = 8 $P(A) = 8/16 = 0.5$

For event B,
Total no. of strings generated = 16
Strings which end with 1 = 8
So, B = 8 $P(B) = 8/16 = 0.5$

For two events to be independent the below equation should satisfy:
$P(A) \times P(B) = P(A \& B)$

For event A&B,
Strings should have even no. of 1s and end with 1
Total no. of strings = 16
A&B = 4 (1111, 1001, 0101, 0011)
So, $P(A\&B) = 4/16 = 0.25$

Since $P(A) = 0.5$ $P(B) = 0.5$
$P(A) \times P(B) = 0.25$
$P(A \& B) = 0.25$
$P(A) \times P(B) = P(A\&B)$
The above demonstration proves that both the events are independent.

(3) In the given model
~~$p(X=1)=13$~~
Occurrence of $X=1$ is 13 and $X=0$ is 17
This is described as bernoulli distribution
the equation is given as

$$f(x_i/p) = p^{x}(1-p)^{1-x_i}$$

where i is the i^{th} coin flip
by solving this equation we get

$p = 0.43$ for heads
$p = 0.57$ for tails

The maximum likelihood function uses frequencies as estimates under a simple dataset of or model.