CSCI 335

HW 1

Prakhar Gupta pg9349

Q1

(refer to q1.py or q1.ipynb)

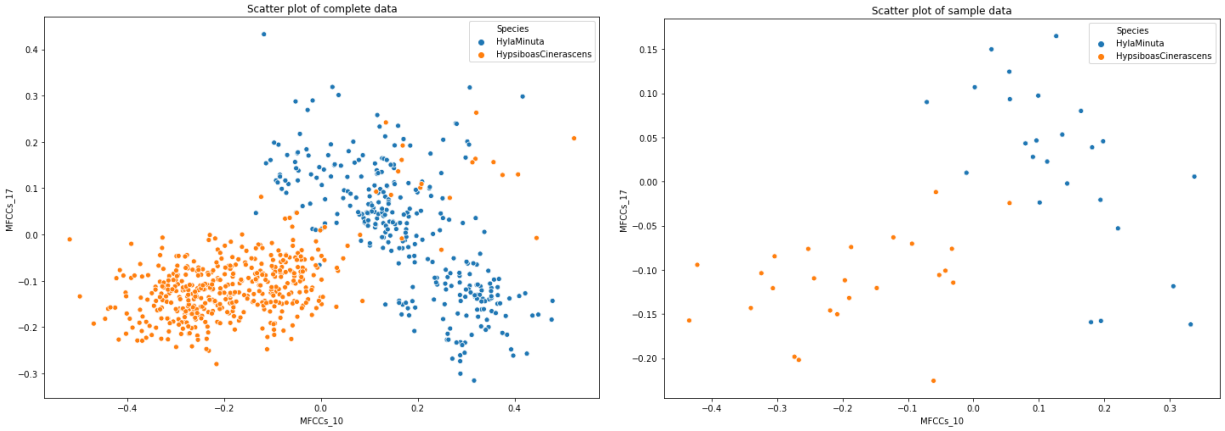
Pandas seaborn matplotlib and numpy were used for visualization and descriptive statistics

Visualization

Plotting Raw Features

Scatter Plots

(Refer to the Scatter plots)



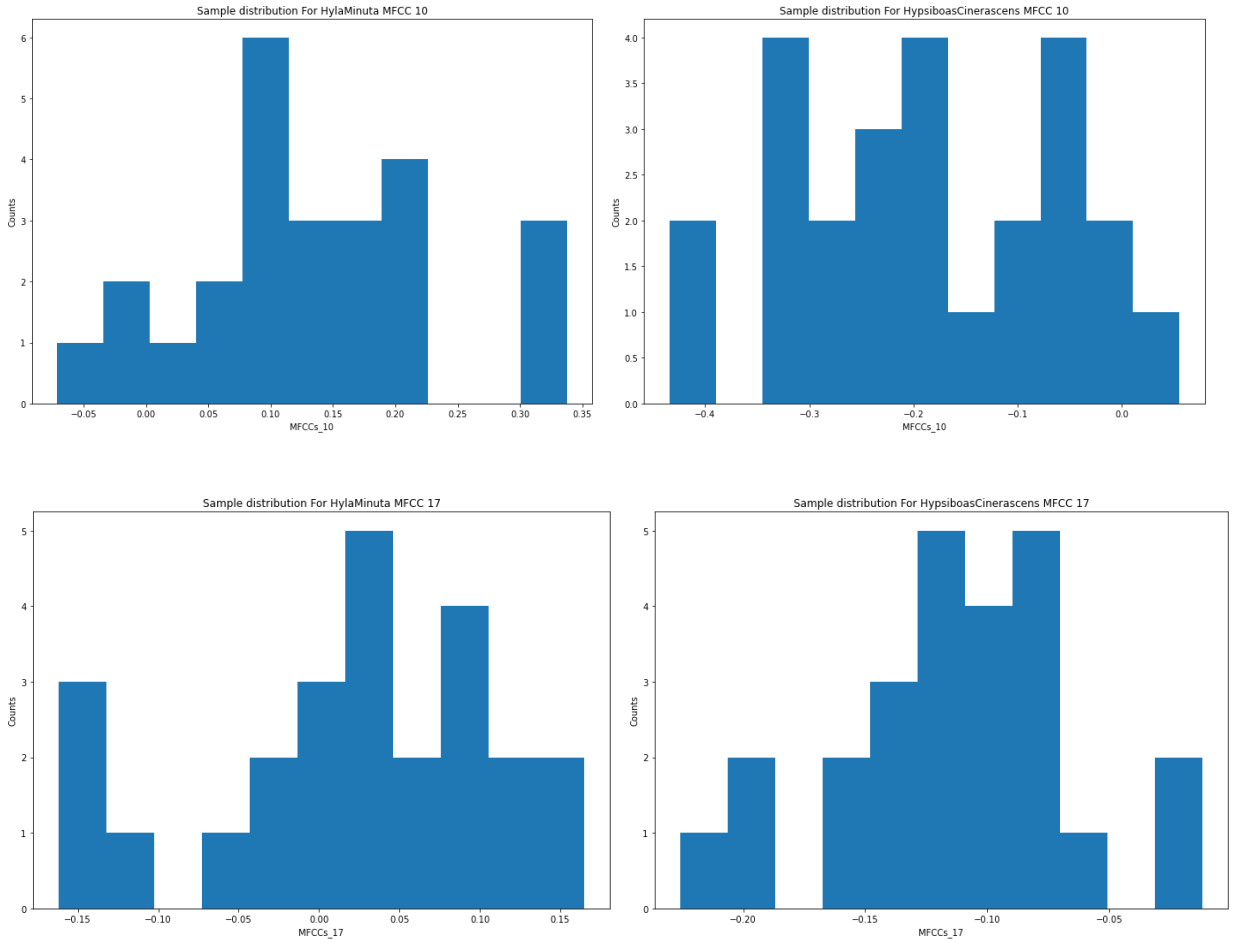
The complete data helps us better see the clusters of the two classes. Plus for the sample data scatter plot we see the data can be easily separated into two classes and both the frog species appear to be quite distant in terms of features captured.However the complete data does give us an idea of the complete distribution which might exist in the world.

Also the complete data has outliers for both classes which sample doesn't have.

Histograms

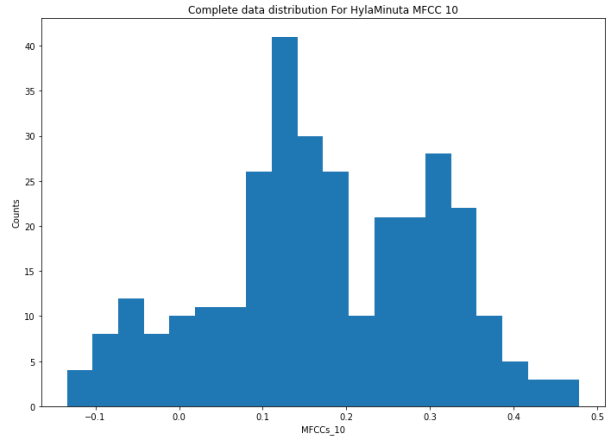
(Refer to the Histogram plots)

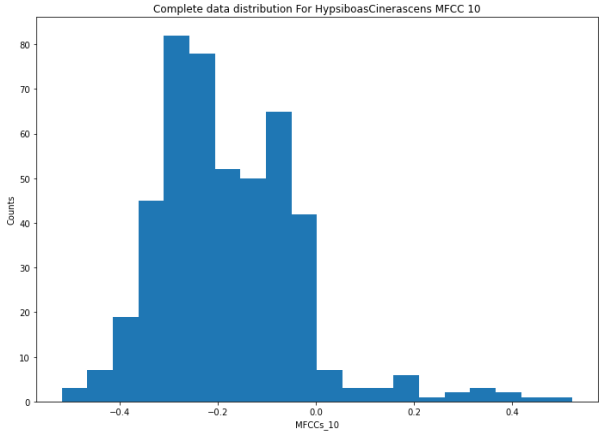
For sample Data

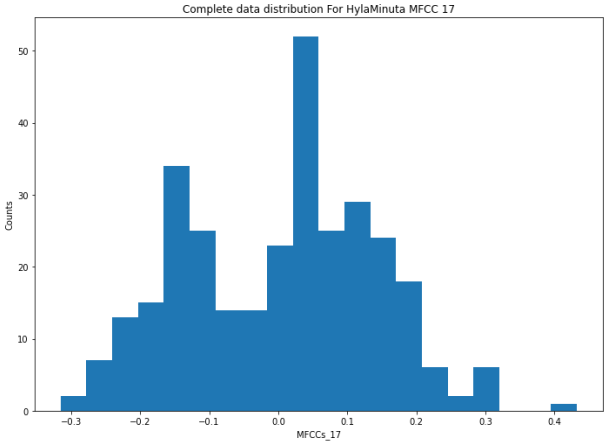


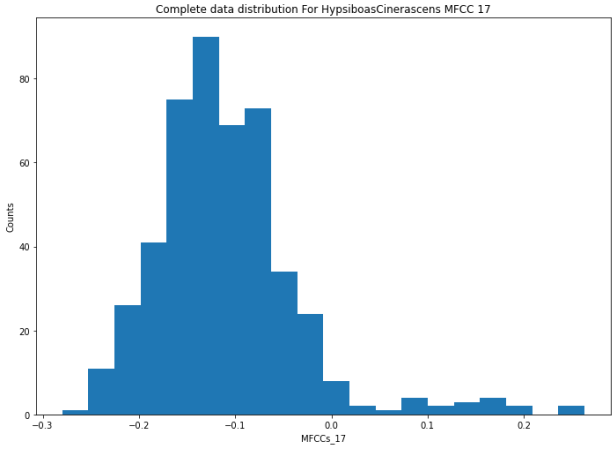
The univariate histograms show multimodal distributions for both the features for each species. With a large bin size we can see that there is some discontinuity for both the features of the two species

For Complete Data









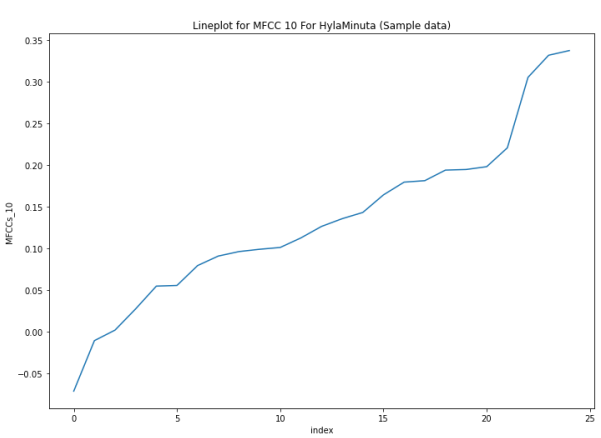
The univariate histogram shows that for HypsiboasCinerascens both MFCC 10 and MFCC 17 are left skewed compared to HylaMinuta

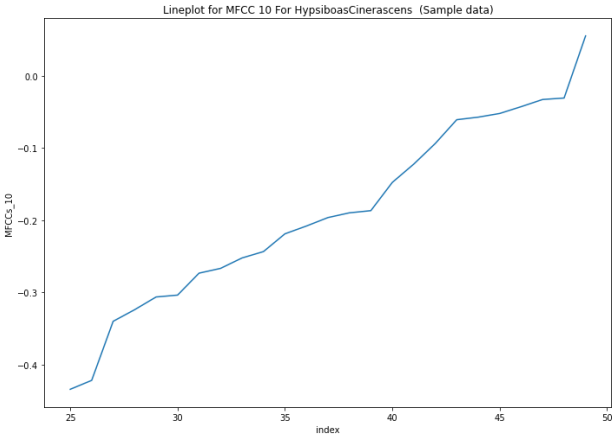
For sample data discontinuity is evident however we don't see it for the complete data

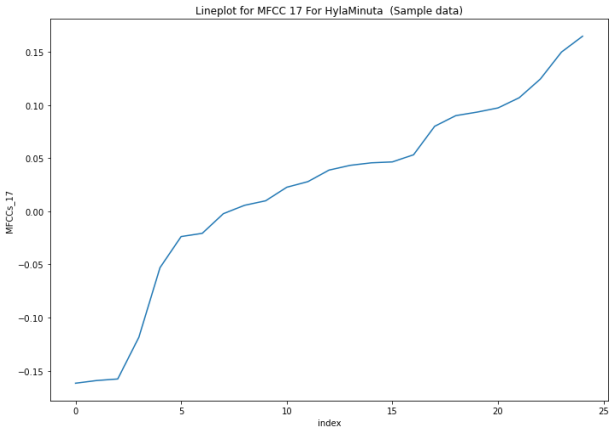
Line plots

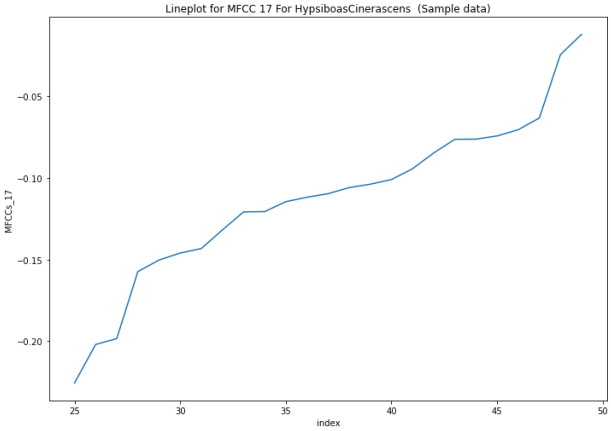
(Refer to the sample line plots)

For sample Data



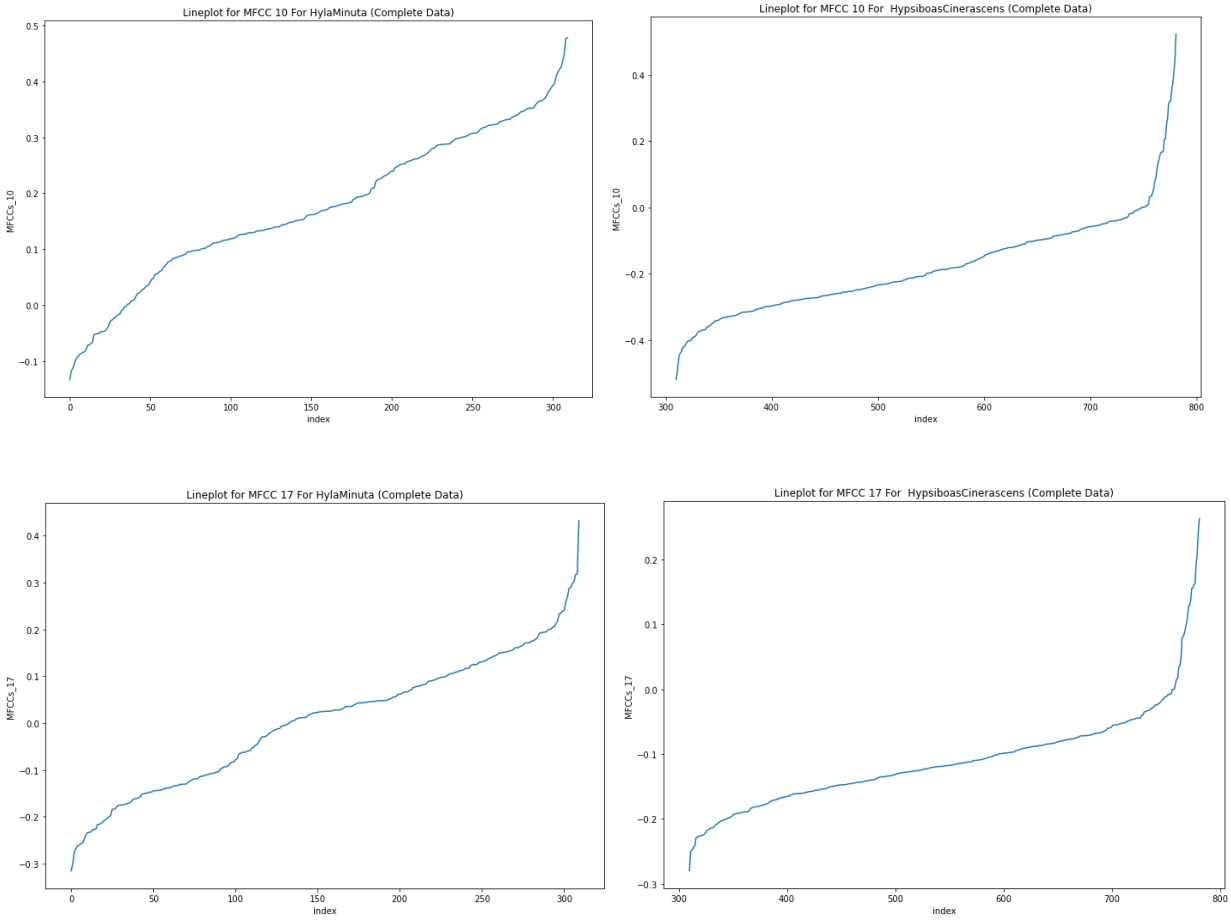






Here we can see the the discontinuity of the data also we can see that for HypsiboasCinerascens MFCC 10 shoots up gradually compared to MFCC 10 for HylaMinuta for the higher values while the trend reverses for MFCC 17

For Complete Data



Those steep rises in value are very evident now with the complete data especially for HypsiboasCinerascens for both MFCC 10 and 17.

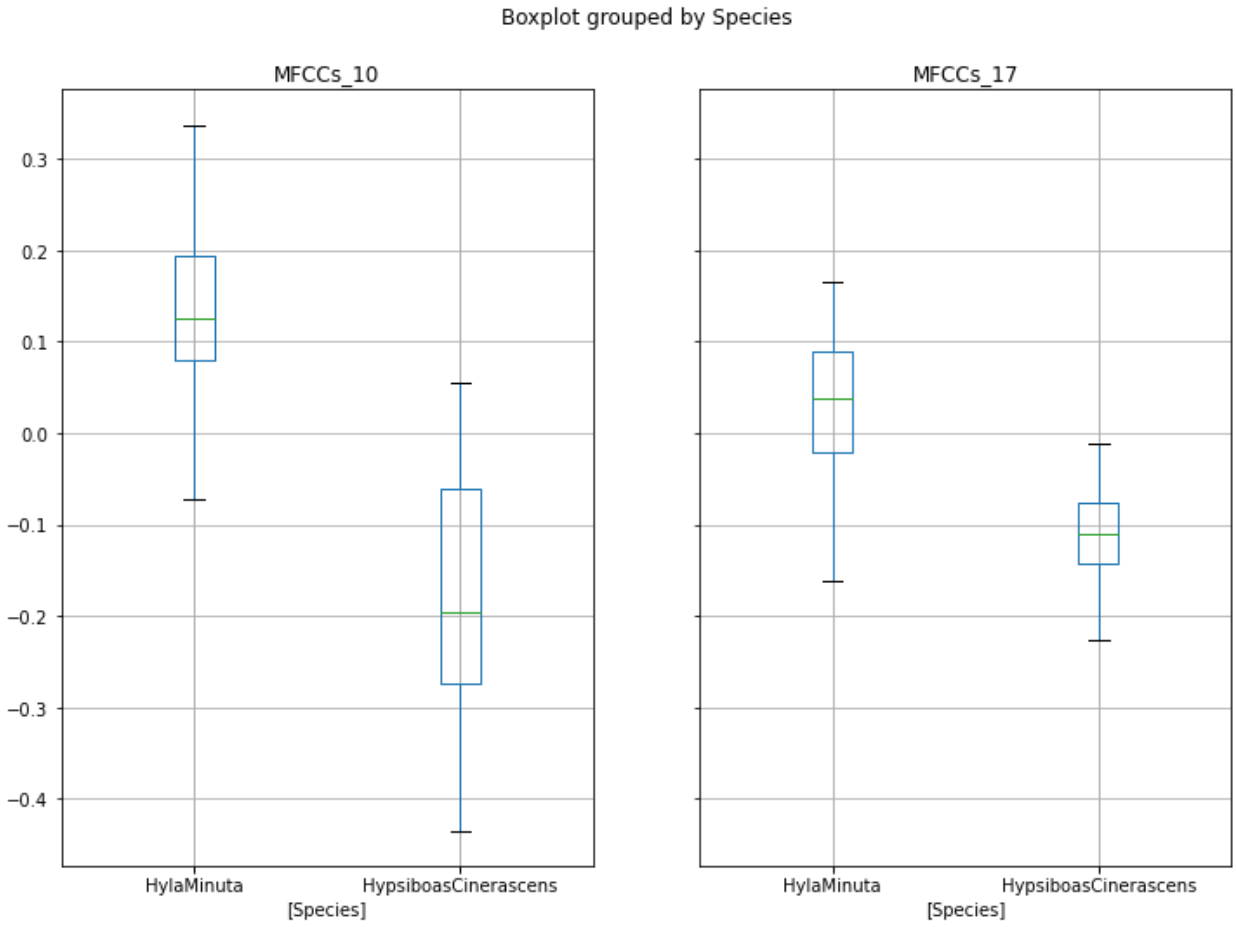
The sample data wasn't able to capture this trend which explains the presence of outliers among classes in the complete data

Plotting Feature Distributions

Box plots

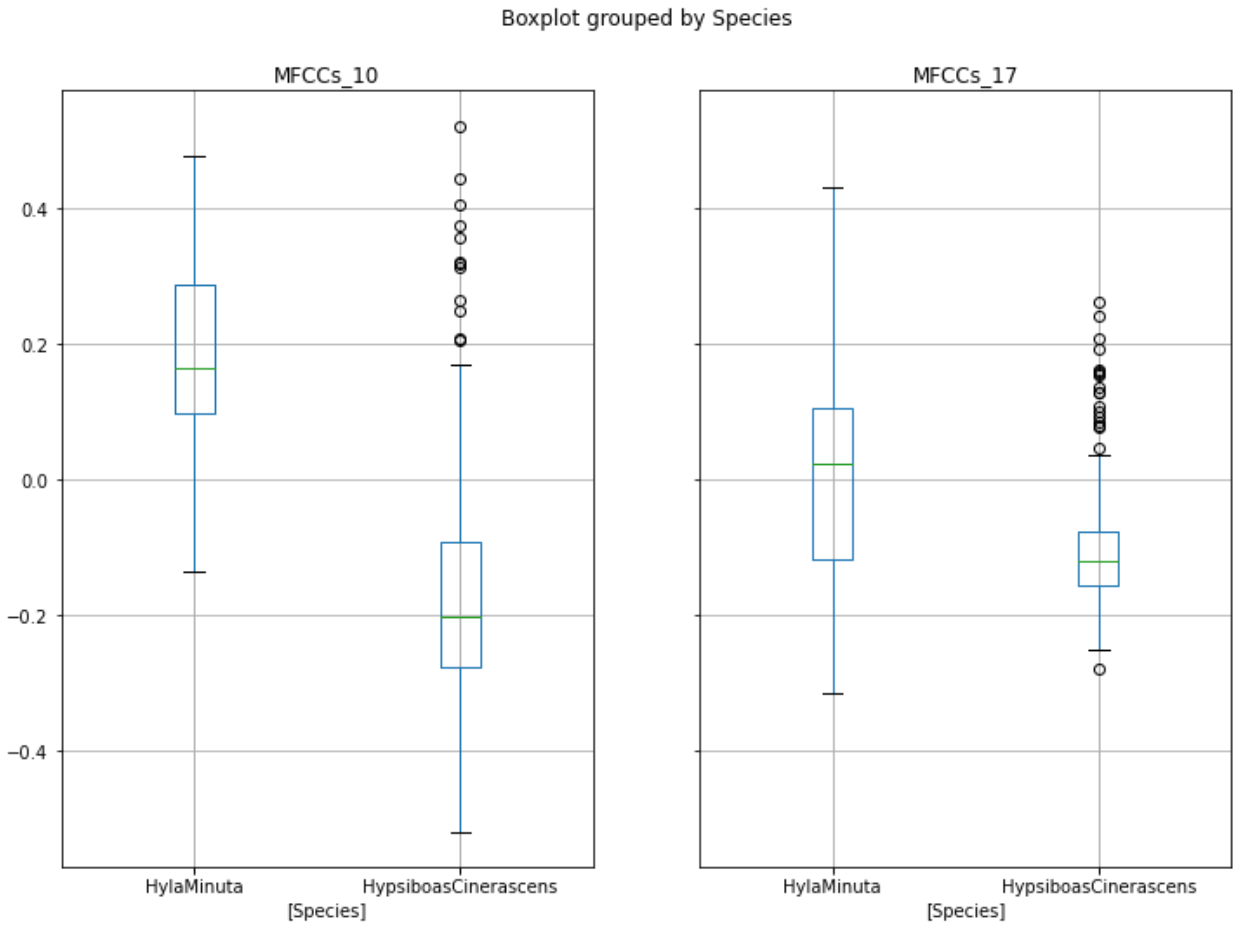
(Refer to the boxplots in the code)

For Sample Data



I have plotted the boxplots feature wise against each species for comparison we can see the 1.5 times IQR spread of the data we can also hypothesis that standard deviation for sample of HypsiboasCinerascens MFCC 10 will be more than for HylaMinuta also we can see the difference in there min max values are quite evident from boxplots

For Complete Data



We don't see the spread difference in the complete data but we do see that min max values are different for both classes. Also the outliers are quite evident in the box plot as any point outside 1.5 time IQR is marked as black circle. Also outliers are quite common for HypsiboasCinerascens.

Descriptive Statistics

(refer to the code that print the descriptive stats)

Sample Data		
Metrics	HylaMinuta	HypsiboasCinerascens
Mean	MFCCs_10 0.134064 MFCCs_17 0.020093	MFCCs_10 -0.190087 MFCCs_17 -0.112754
Standard Deviation	MFCCs_10 0.099519 MFCCs_17 0.090617	MFCCs_10 0.126790 MFCCs_17 0.049717
Covariance	'MFCCs_10', 'MFCCs_17' MFCCs_10 [0.01031672 -0.00597868] MFCCs_17 [-0.00597868 0.0085536]	'MFCCs_10', 'MFCCs_17' MFCCs_10 [0.0167455 0.00236565] MFCCs_17 [0.00236565 0.00257479]
Complete Data		
Metrics	HylaMinuta	HypsiboasCinerascens
Mean	MFCCs_10 0.175102 MFCCs_17 0.004276	MFCCs_10 -0.178493 MFCCs_17 -0.109695
Standard Deviation	MFCCs_10 0.131494 MFCCs_17 0.138070	MFCCs_10 0.145468 MFCCs_17 0.074566
Covariance	'MFCCs_10', 'MFCCs_17' MFCCs_10 [0.01734669 -0.01235759] MFCCs_17 [-0.01235759 0.01912491]	'MFCCs_10', 'MFCCs_17' MFCCs_10 [0.021206 0.00635566] MFCCs_17 [0.00635566 0.00557183]

Descriptive statistics appear to vary a lot among both species and also among the two datasets. HylaMinuta mean for MFCCs vary a lot compared to HypsiboasCinerascens . Also

we see that the mean for HypsiboasCinerascens for both MFCCs is negative and for hyla minuta it is positive. For HylaMinuta we see that the standard deviation between MFCCs is almost similar for both sample and complete while for HypsiboasCinerascens it varies a lot

Q2

(refer to q2.py or q2.ipynb)

Pytorch was used to create the logistic regression model on both sample and complete data of frogs

For both the file data was split into train and test 80:20 split. The idea is to train the model on one set of data and use the other set to predict to check for under and over fitting.

For sample data

The model was trained using following parameters-

Input dim-2 (MFCC 10 and MFCC 11)

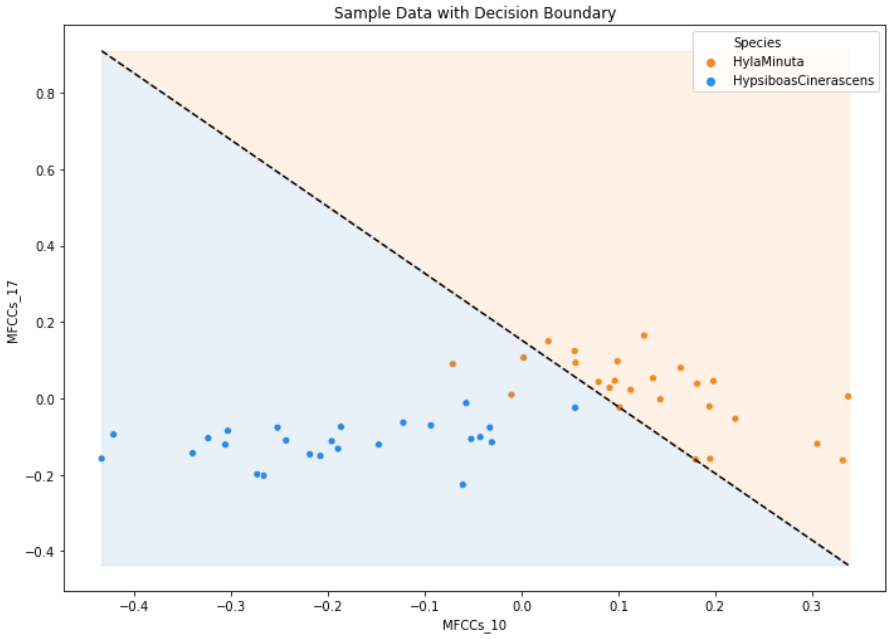
Output Layer dim - 1

Activation - sigmod

Epoch - 2000

Learning rate -0.01

Optimizer- stochastic gradient descent



Following is the decision boundary obtained. This is for the complete sample data the decision boundary for test and train data can be observed in the code. The decision boundary is linear and divides the data into two classes. Some “Hylaminuta” frog are incorrectly classified. The train accuracy is 0.9500 while the test is 0.90.

For Complete Data

The model was trained using following parameters-

Input dim-2 (MFCC 10 and MFCC 11)

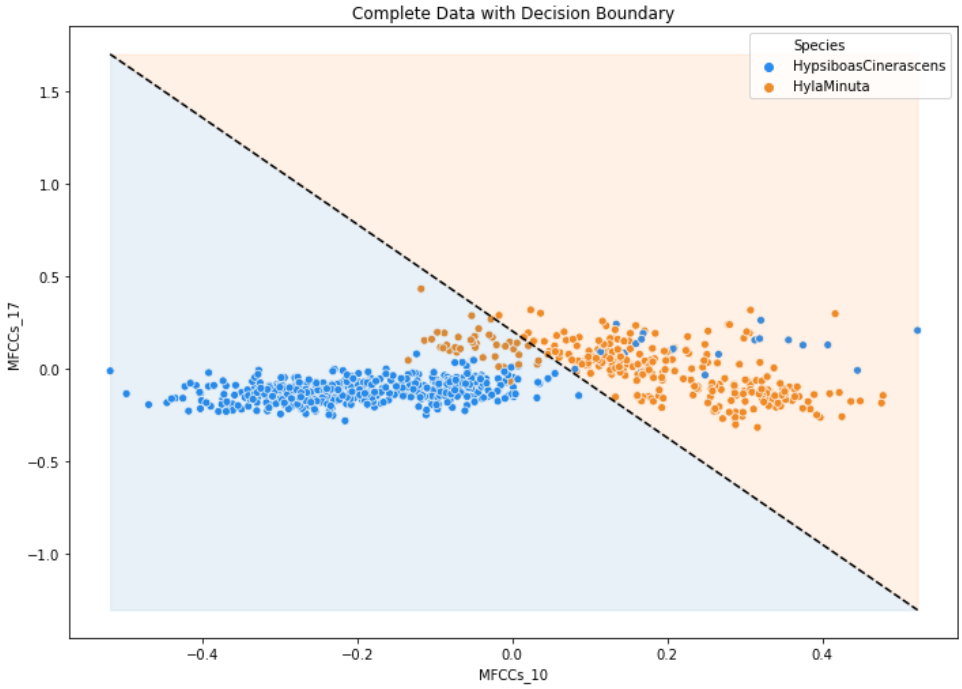
Output Layer dim - 1

Activation - sigmod

Epoch - 8000

Learning rate -0.01

Optimizer- stochastic gradient descent



Following is the decision boundary obtained. This is for the complete data the decision boundary for test and train data can be observed in the code. The decision boundary is linear and divides the data into two classes. Compared to the sample data we have data points on both classes lying on either side of the decision boundary .The train accuracy is 0.9153 while the test is 0.9103.

Comparison

We can see the decision boundary for the whole data generalized well compared to the sample data. This can be due to the advantage that having more data to do machine learning has on the model. Also we can see outliers are more common for HypsiboasCinerascens class than Hylaminuta one problem with logistic regression is that it is prone to outliers. If we remove some outliers we might have better results. One more idea is to use a nonlinear model where we can learn a non linear decision boundary. Seeing the data scatter plot we might be able to get better performance metrics using them.

Ans3 a) Let X be event of product being defective

Let S be event of product being shipped

~~$P(X) = 1 - 0.80$~~

$P(X) = 1 - 0.85$ $\quad P(\sim X) = 0.85$

$P(\sim S|\sim X) = 0.10$ $\quad P(S|\sim X) = 1 - P(\sim S|\sim X) = 0.90$

$P(S|X) = 0.05$ $\quad P(\sim S|X) = 1 - P(S|X) = 0.05$

To find $P(X|S)$

Using Bayes

$$P(X|S) = \frac{P(S|X) \times P(X)}{P(S)}$$

$$P(S) = P(S|X) * P(X) + P(S|\sim X) * P(\sim X)$$

$$= 0.05 * 0.15 + 0.90 * 0.85$$

$$= 0.0075 + 0.765 = 0.7725$$

$$P(X|S) = \frac{0.05 * 0.15}{0.7725} = \boxed{0.0097}$$

b) Let X be random process that generates bits of length 4; $|X| = 16$

" A be a event for even number of ones

" B be event for bits that end in 1

Assuming A and B not independent

$$P(A \cap B) = \{1111, 1001, 0011, 0101\} = \frac{4}{16} = \frac{1}{4}$$

$$P(A) = 8/16 = \frac{1}{2} \qquad P(B) = \frac{8}{16} = \frac{1}{2}$$

$$P(A).P(B) = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$$

For A and B not be be independent

$$P(A \cap B) \neq P(A).P(B)$$

But they are equal Hence we prove by contradiction that ~~P(A) and P(B) are~~ A and B are independent events.

c) Let X be the random process such that $X = \{0, 1\}$.

S be the experiment that generated the sequence assuming (i.i.d assumption)

Since X has two possible outcomes $\{0, 1\}$ the obs in S follows i.i.d and bernoulli

Likelihood of bernoulli

$$L(p) = \prod_{i=1}^{n} p^{x_i} (1-p)^{(1-x_i)}$$

log both sides

$$ln(p) = \log p \sum_{i=1}^{n} x_i + \log(1-p) \sum_{i=1}^{n} (1-x_i)$$

derivate to find maxima.

$$\frac{d\, ln(p)}{dp} = \frac{\sum_{i=1}^{n} x_i}{p} - \frac{\sum_{i=1}^{n} (1-x_i)}{1-p} = 0$$

$$\sum_{i=1}^{n} x_i - p \sum_{i=1}^{n} x_i = p \sum_{i=1}^{n} (1-x_i)$$

$$p = \frac{1}{n} \sum_{i=1}^{n} x_i$$

To prove it a maxima we derivate again.

$$\frac{d^2 \ln(p)}{dp^2} = -\frac{\sum x_i}{p^2} - \frac{\sum (1-x_i)}{1-p^2}$$

Since $\frac{d^2 \ln(p)}{dp^2} < 0$ its a maxima.

For S $n = 30$

MLE for $P(X=1) = \frac{1}{30} \times$ no of ones $= \frac{13}{30}$

MLE for $P(X=0) = \frac{1}{30} \times$ no of zeros $= \frac{17}{30}$

Bonus

For MAP we maximize the posterior probabity while for MLE we maximize the likelihood.

We know that the coin is fair $P(X=1) = \frac{1}{2}$

$$MAP = \text{argmax}\left(\frac{P(X|X=1) * P(X=1)}{P(X)}\right)$$

since its argmax we dont have to worry about $P(X)$

$$MAP = MLE \times P(X=1) = \frac{13}{30} \times \frac{1}{2} = \frac{13}{60}$$

References:

https://pytorch.org/docs/stable/index.html

https://towardsdatascience.com/mle-map-and-bayesian-inference-3407b2d6d4d9

https://www.youtube.com/watch?v=nTizrDsR1x8