PRML LAB-2

Name: Shalin Jain

Roll No.: - B21CS070

Problem 1

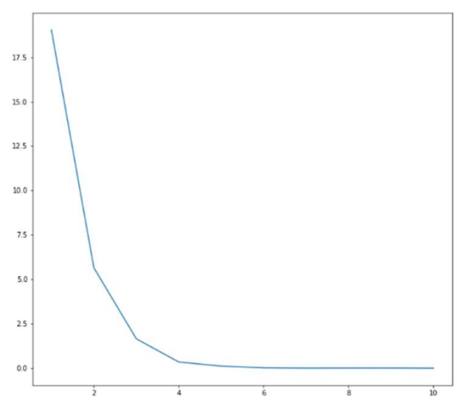
Part 1 Pre-processing the data

The data was pre-processed by dropping the null values and normalizing the dataset

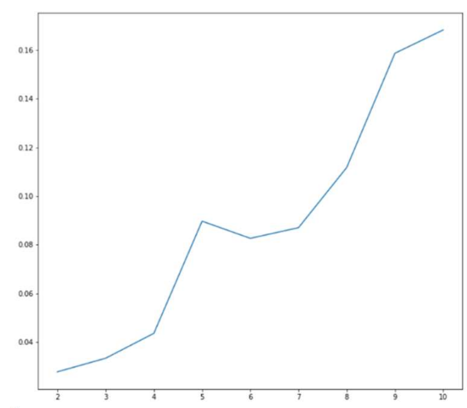
Part 2 Varying Hyper-Parameters

We varied three hyper-parameters that are max_depth, min_sample_leaf and max_features. The obtained graph for mse are as shown

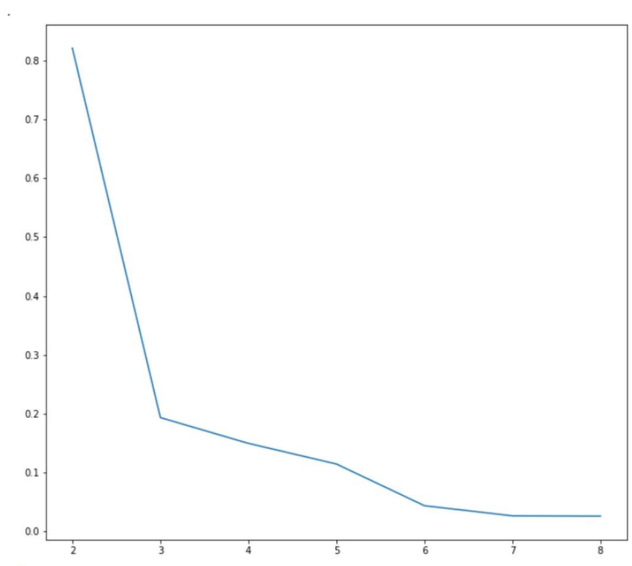
i) Max_depth



ii) Min_Samples_Leaf



iii) Max_Features



Because max_depth and min_sample_leaf are very useful while defining the height and complexity of the tree we used these hyper-parameters. We took the value of hyper-parameters for which the mse is the lowest.

Part 3 Performing cross validations

1) Hold out cross validation

```
0.9992459348170605
0.08070792316017326
0.9904549748083173
```

Accuracy

MSE

Cross Validation Score

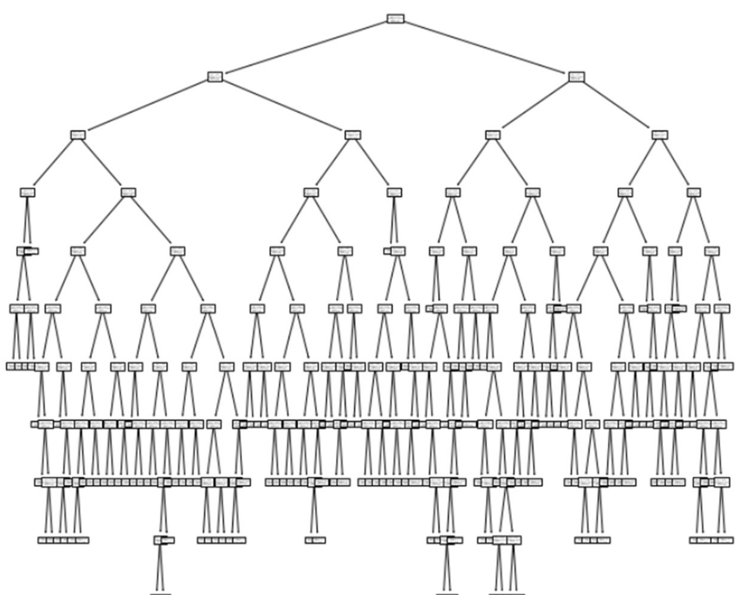
2) 5-Fold Cross Validation

```
[0.99557134 0.9895716  0.98878805 0.95162691 0.9855022 ]
0.982212020348493
```

3) Repeated 5-Fold Cross Validation

```
[0.99579498 0.99773435 0.98701509 0.94664839 0.99555791 0.98671735
 0.99204476 0.98825463 0.99124105 0.98925572 0.99624069 0.98783929
 0.97992079 0.99138788 0.99217751 0.98715791 0.99262532 0.99510298
 0.99668783 0.9810375  0.98476795 0.99582544 0.99132729 0.96677709
 0.99455043]
0.982212020348493
```

Tree Plot

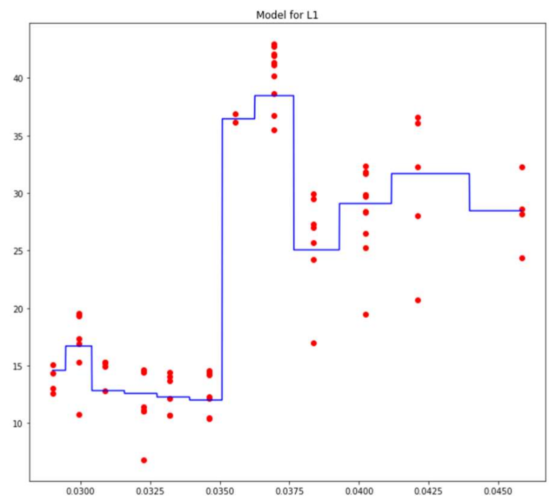


Part 4 L1 and L2 Loss

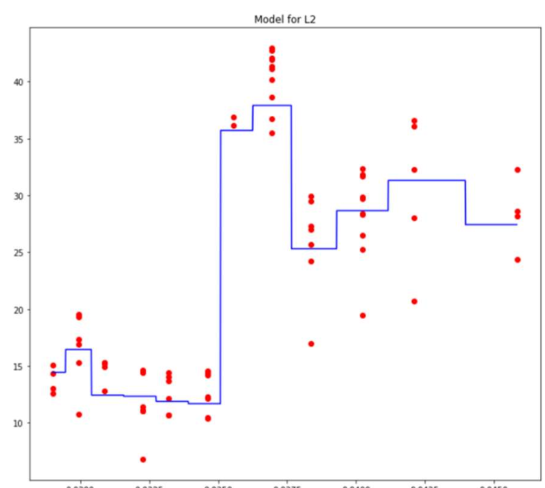
The loss function L2 that is mean squared error performs better in case of regression.

```
L1
0.9996323293581979
L2
0.9997753642918821
```

i) L1 Loss

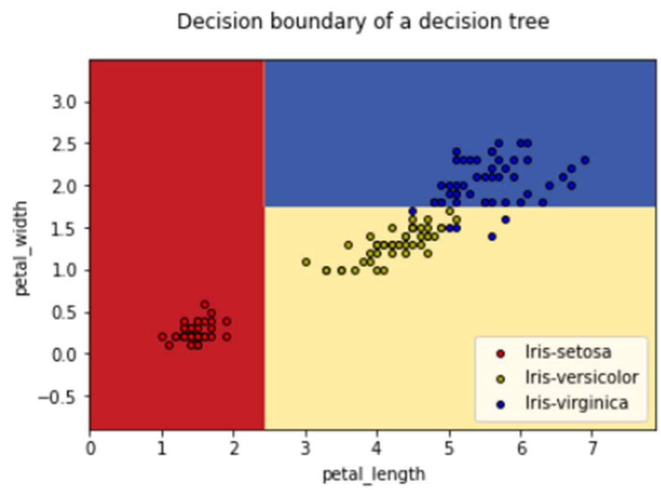


ii) L2 Loss

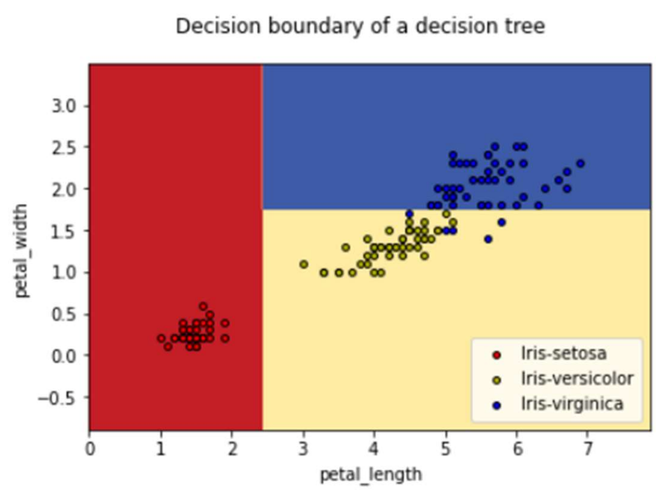


Problem 2

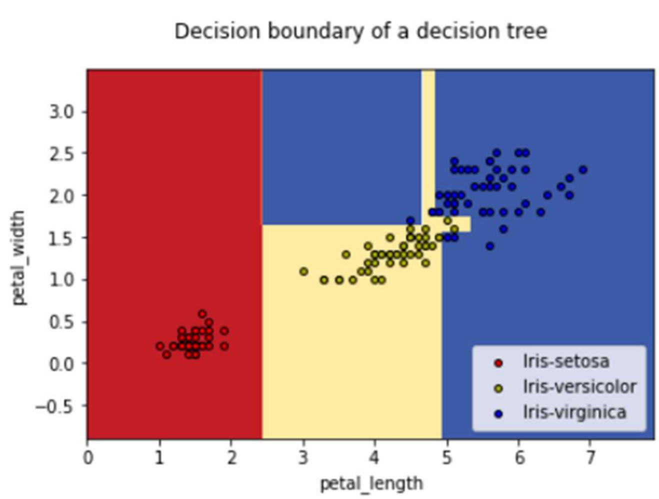
Part 1 Training a Decision Tree Classifier and Plotting Decision Tree Boundary



Part 2 Dropping one value and plotting the decision boundary



Part 3 Changing max_depth to None and plotting decision boundary



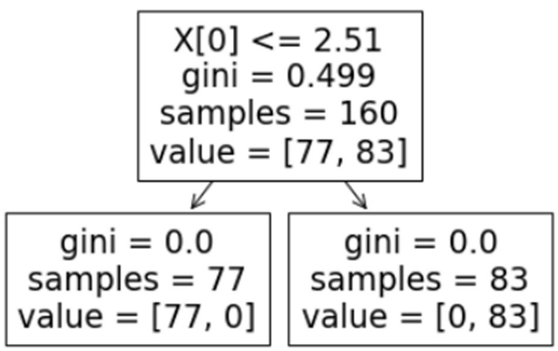
The difference between first and the third plot is that the decision boundary is more precise with increase in max_depth.

Part 4 Creating a random dataset

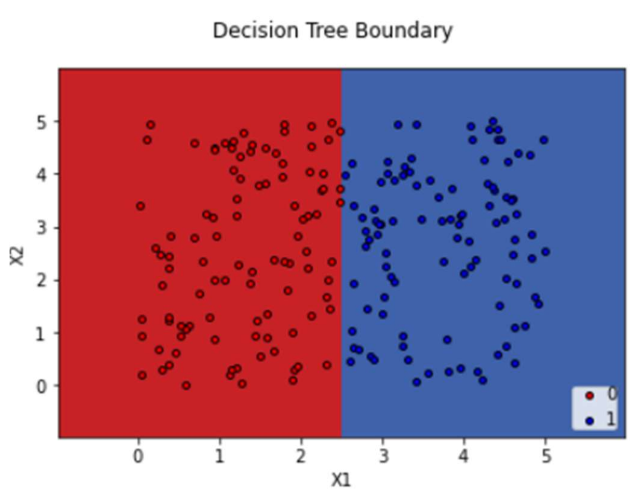
	X1	X2	Y
0	3.73	3.13	1
1	4.39	3.07	1
2	2.33	4.65	0
3	3.59	3.90	1
4	1.25	3.92	0
...	...	...	...
195	1.67	2.39	0
196	4.60	2.49	1
197	1.78	3.95	0
198	2.81	1.46	1
199	1.90	0.09	0

200 rows × 3 columns

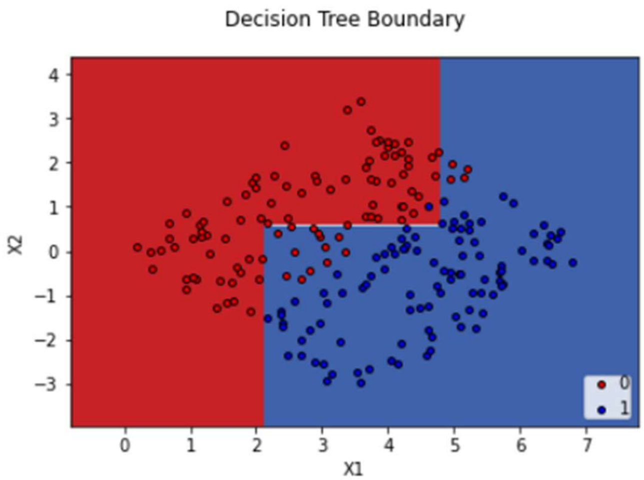
Decision Tree (Max_depth = 2)



Decision Boundary

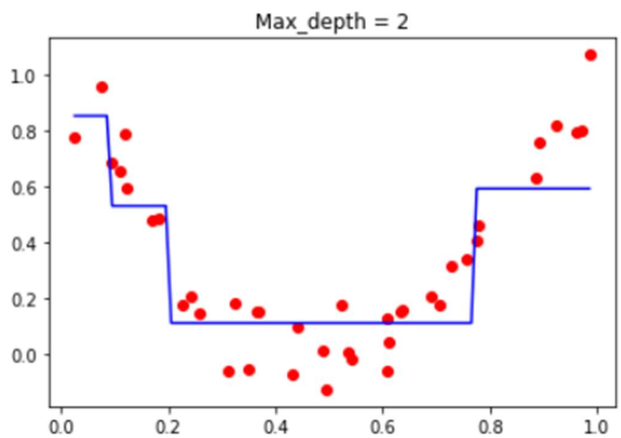


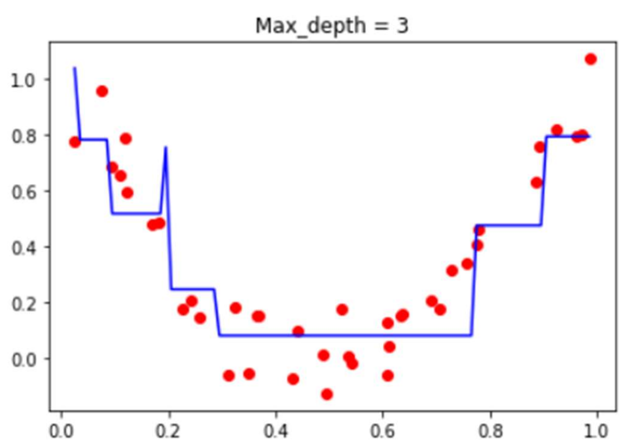
Decision boundary after rotating



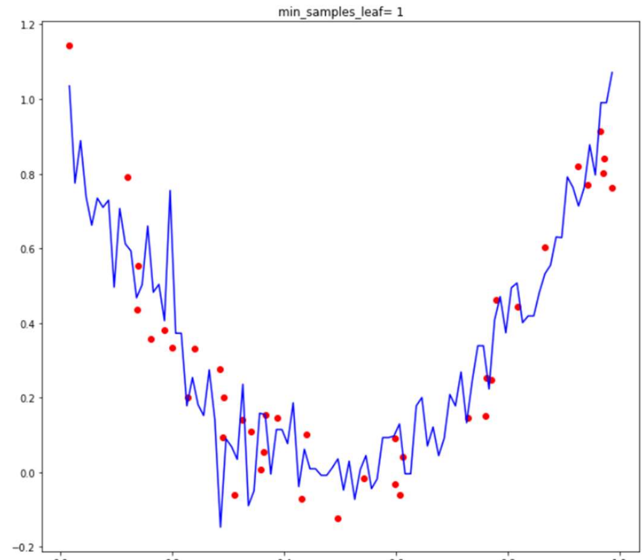
Regression

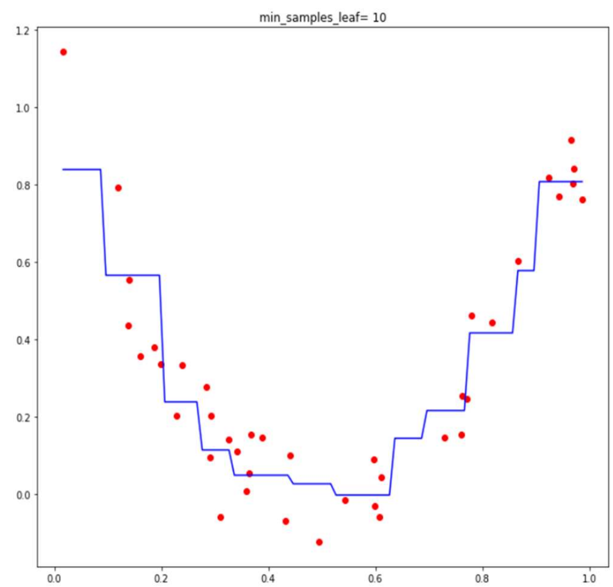
Part 1





Part 2



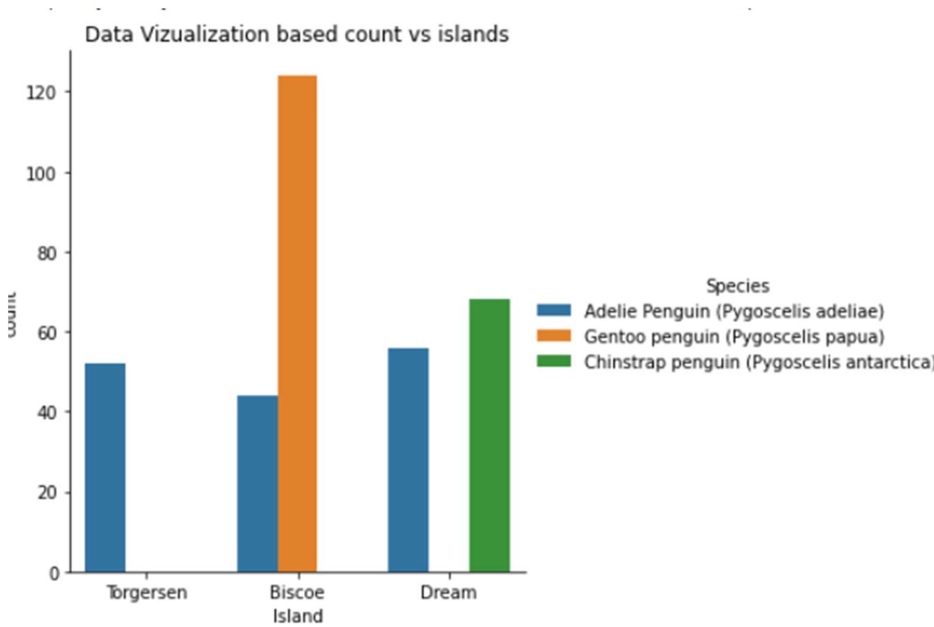


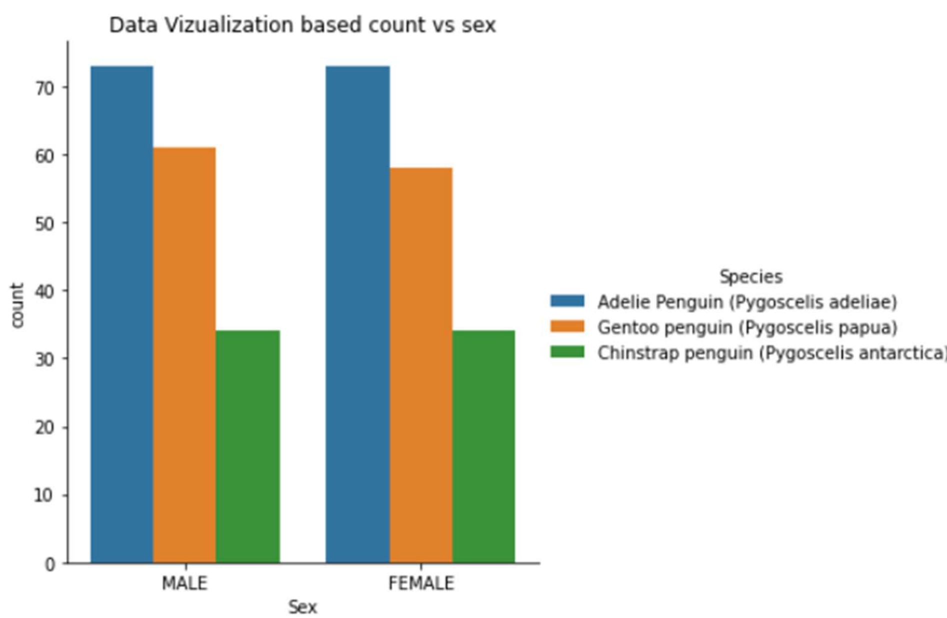
We can clearly see the difference that if minimum samples in a leaf is less then more accurate boundary is present.

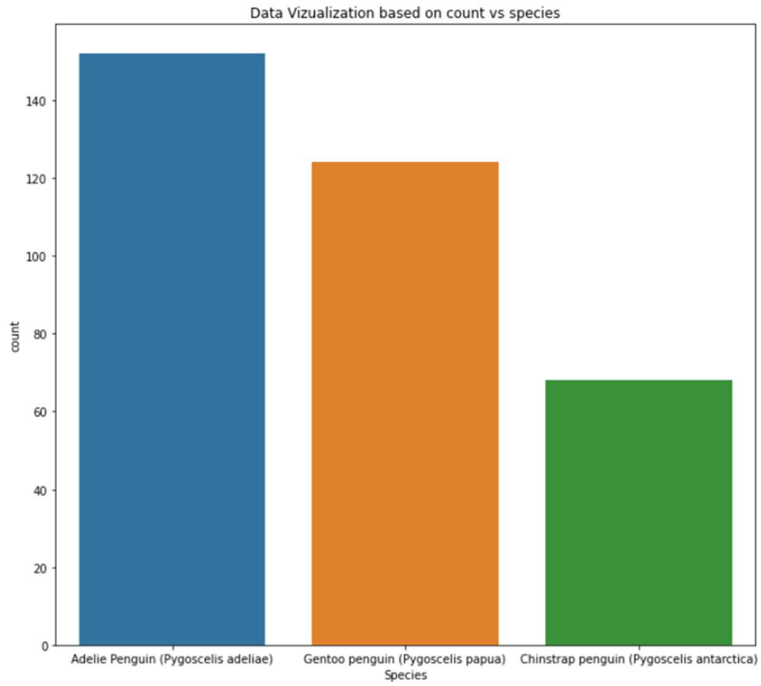
In the part 1 we can decision boundary's precision and accuracy increases with increase in max_depth. The increase in max_depth increase more accurate splits as well as predictions.

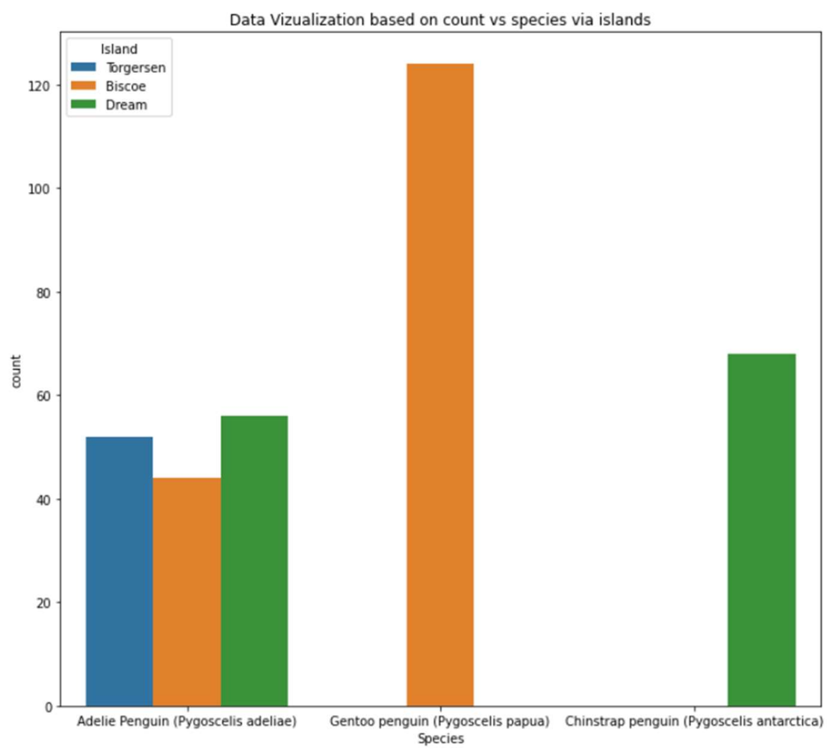
Problem 3

Data visualization






Data Vizualization based on count vs species
count
0
20
40
60
80
100
120
140
Adelie Penguin (Pygoscelis adeliae)
Gentoo penguin (Pygoscelis papua)
Chinstrap penguin (Pygoscelis antarctica)
Species


Data Vizualization based on count vs species via islands
Island
Torgersen
Biscoe
Dream
count
0
20
40
60
80
100
120
Adelie Penguin (Pygoscelis adeliae)
Gentoo penguin (Pygoscelis papua)
Chinstrap penguin (Pygoscelis antarctica)
Species

Decision Tree Class

For the third problem I have used entropy for the loss function and used information gain to get gain for each parent, child attribute and the best gain is used find the best split. The split is calculated with the help of information gain as well as the threshold.

We used the formula given below to find the entropy gain: -

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

The overall as well as class wise accuracy for my model is as follows

```
0.95
{0: 0.975609756097561, 1: 0.8, 2: 1.0}
```

Where 0,1,2 are the respective classes of species.