PRML LAB 2

Vansh Agarwal
B21AI042

Q1. 1

The imported dataset when visualised using correlation matrix heat map (Fig1) and pair plots. The dataset doesn't contain missing entries and no Encoding of categorical variables is required.

Fig 1

The target variable increases with increase in some of the features, which are said to be strongly positively correlated. On the other hand, if the target variable decreases with increase in some of the features, these features are said to have strong negative correlation with the label.

The most relevant features thus selected are **X4 and X5.** The distribution of target with these features have been shown using scatterplot (Fig2).

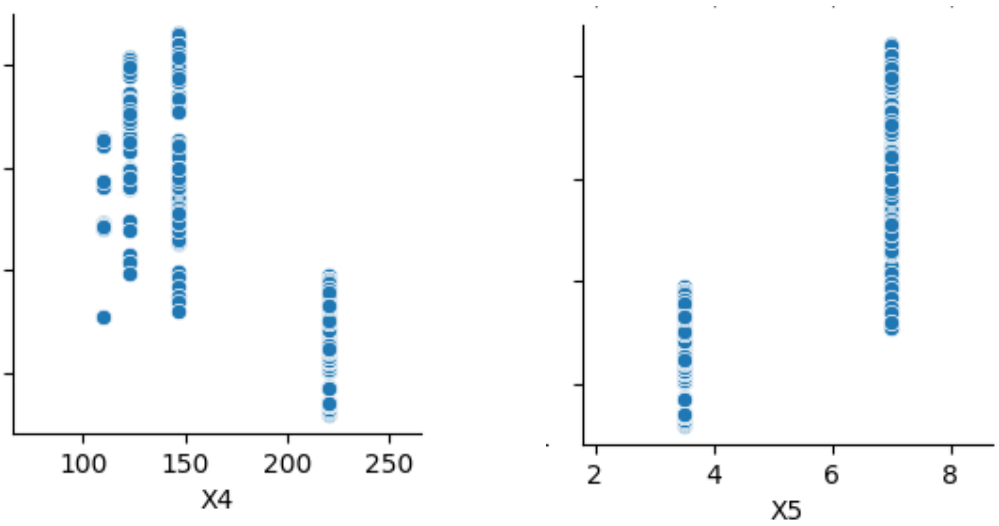


Fig 2

Q1. 2

The dataset has now been split using train, validation and test set using `sklearn's train_test_split` function.

Hyperparameter Tuning

Hyper parameters are certain parameters of the decision tree model which have been adjusted so that they control the learning process in a certain way and give the best performance.

Here we have dealt with the hyper parameter `max_leaf_nodes` (although there are others such as `max_depth` & `min_samples_leaf` etc.) Hyper parameter tuning constraints the model so that it doesn't overfit the data and hence, generalises well. The error used for hyper parameter tuning is **Root Mean Square error** as this is a regression based problem. **Also the optimal value of hyper parameter `max_leaf_nodes` comes out to be 10 with the minimum RMSE of 13.344. (Fig 3)**

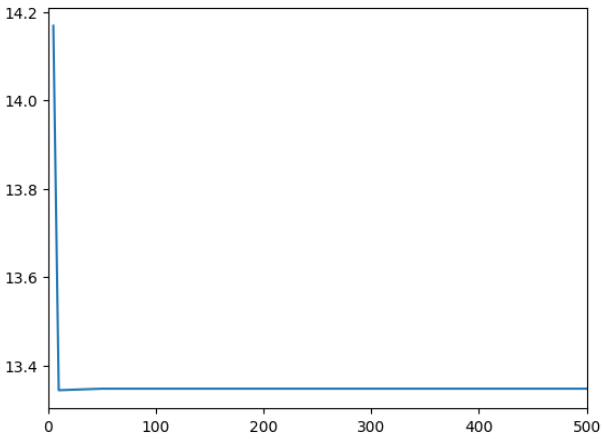


Fig 3

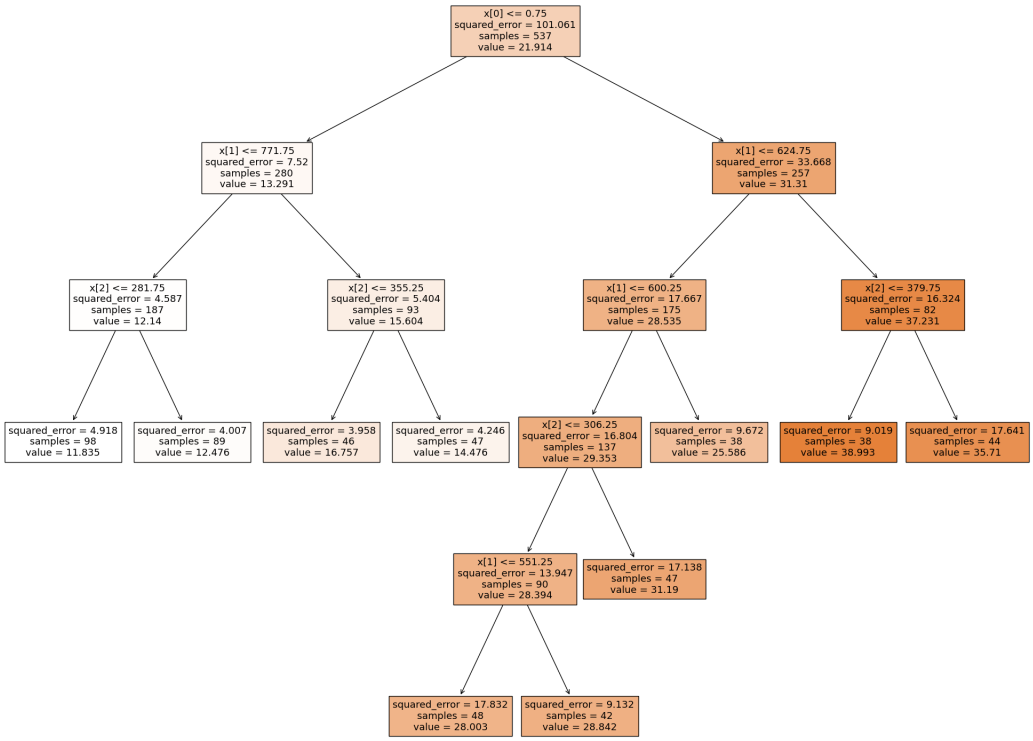


Fig 4

Q1. 3

Next Holdout cross validation, K fold cross validation and repeated K fold cross validation have been performed with k = 5 with the model and the scores are as follows:

Holdout Cross Val Score: 0.9069
5-Fold cross val score: 0.9006
Repeated 5 fold cross-val Score: 0.9017

The predictions have been made with the test set and the error has been calculated with **root mean squared error, which comes out to be 3.0421.** The decision tree has also been plotted as shown in the figure above (Fig 4)

Q1. 4 L1 & L2 Regularization

L1 regularization when applied on the model tries to minimise the L1 loss, which is the sum of absolute differences between the actual values and the predicted values. Thus, for L1 regularisation we use the criteria of split as Absolute error while training the model.
L2 regularisation when applied to the model tries to minimise the L2 loss, which is the sum of square of differences between actual and predicted values. Thus for L2 regularisation we use the criterion for split as Mean Squared Error.

For our dataset L2 performs slightly better. If there are outliers in the dataset L2 performs exceptionally well, because the error squared turns out to be a very large quantity and L2 loss tends to minimise this. So L2 is better for data having outliers. The decision boundaries and scores are:
Score with L1: 3.678
Score with L2: 3.653

$$L1LossFunction = \sum_{i=1}^{n} |y_{true} - y_{predicted}|$$

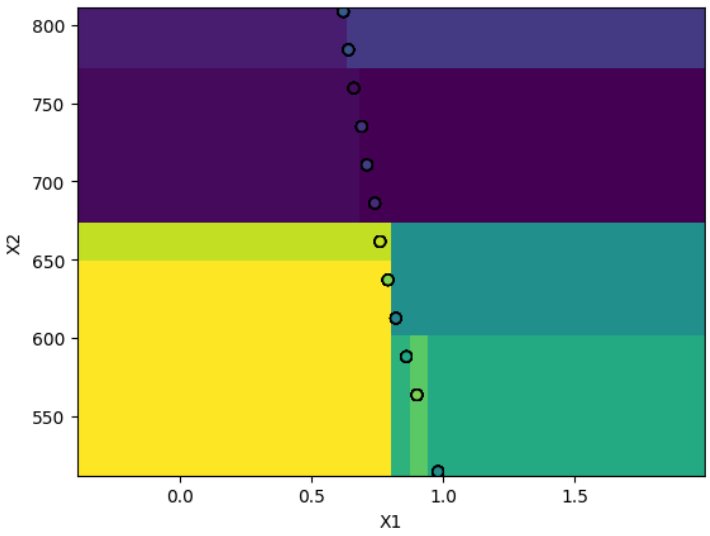


Fig 5a - L1 Loss

$$L2LossFunction = \sum_{i=1}^{n} (y_{true} - y_{predicted})^2$$

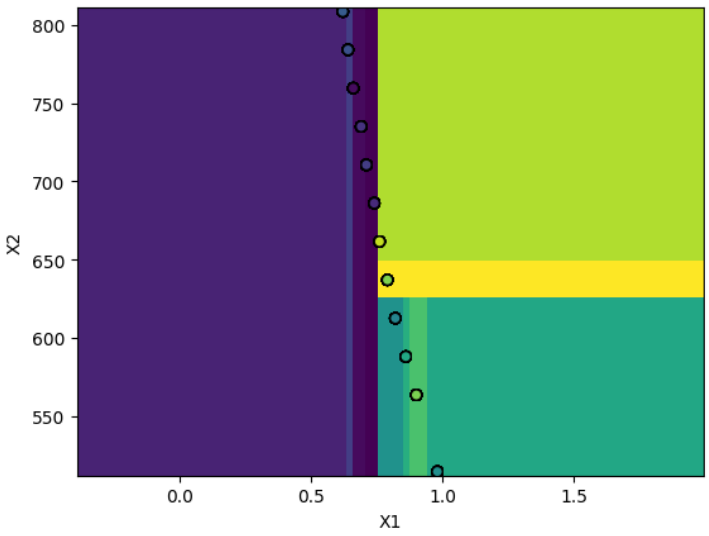


Fig 5b - L2 Loss

Q2. Classifier

1.

Decision boundaries and splits have been shown in the figure below (Fig 6). The tree has been built using the criteria as Gini Impurity. The optimal split for root node is when X[1] $<$ 0.8 because it produces the least gini impurity of all possible splits i.e. 0.667. This leads to a leaf node (Pure node containing all samples from same class). Further a split has been made with X[0] $<=$ 4.75 which leads to almost pure splits, as shown in the figure.

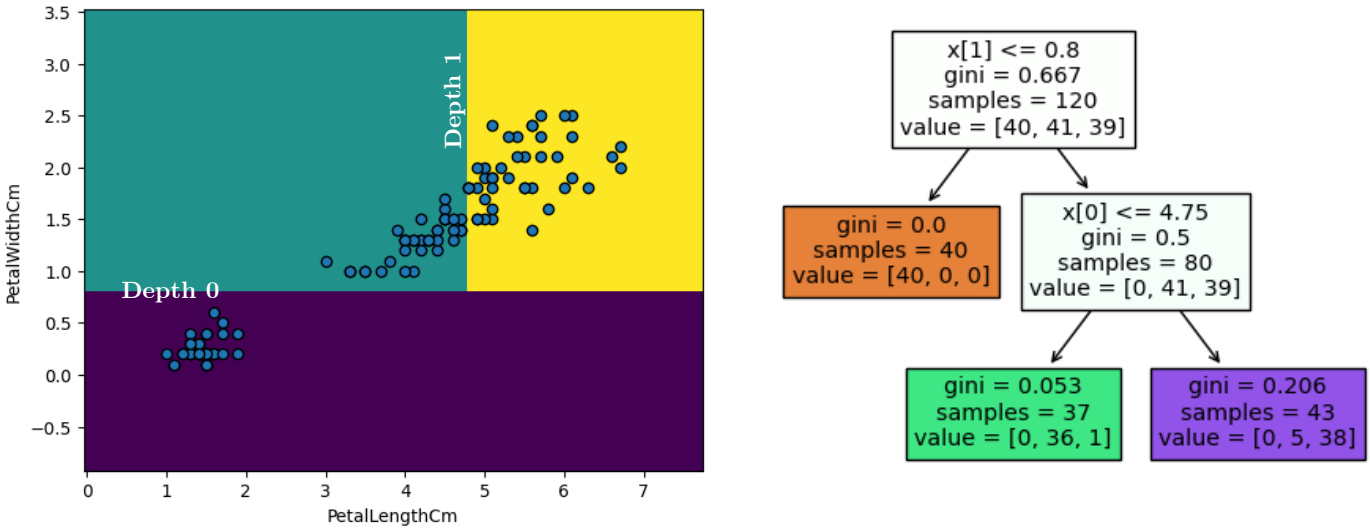


Fig 6

2. Remove Widest Iris Versicolor

After removing the Wiest Iris versicoloured, we get the following decision boundary (Fig 7). Again splits have been made based on the gini impurity. For each non leaf node we select the feature and its value which gives least gini impurity. Decision trees are very sensitive to small variations in data.

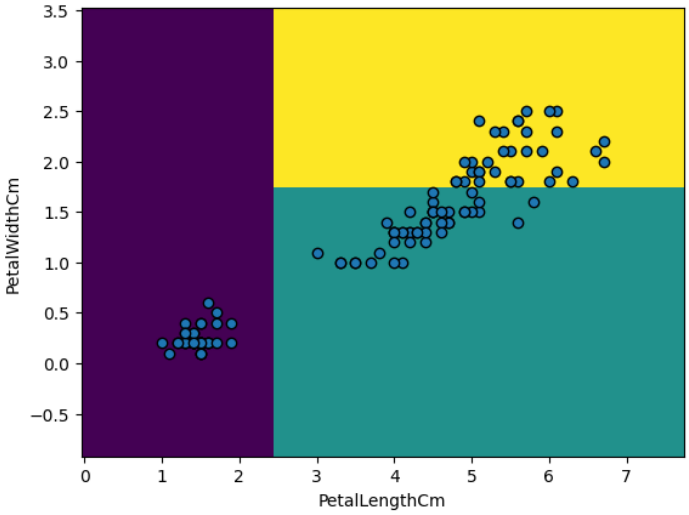


Fig 7

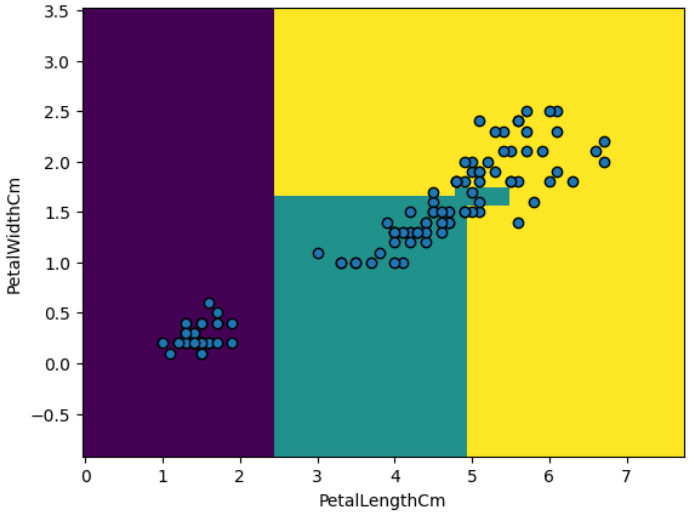
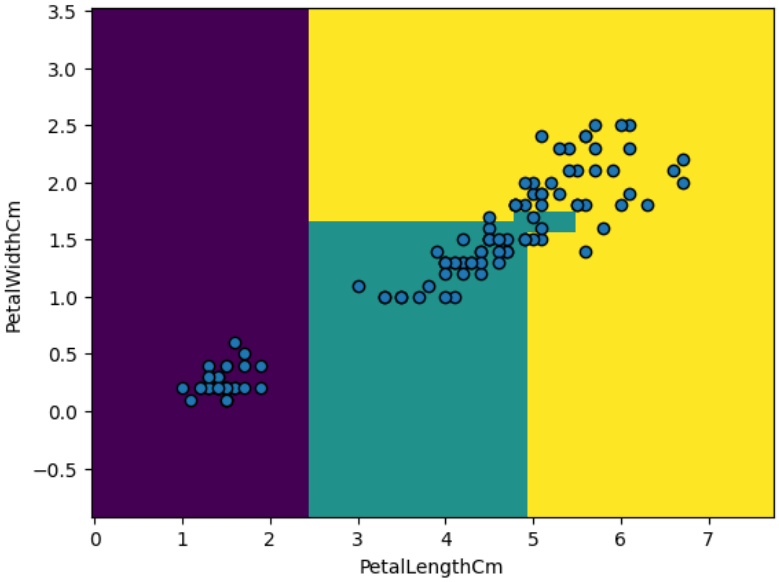


Fig 8

3. Max Depth = None

On training a tree with max_depth equal to none, we have set no condition on the hyper parameter and the tree grows in a way to fit the train data in the best possible manner, not keeping a track of its depth, number of leafs etc. Thus, the model is prone to overfitting and may not generalise well on the test data. The decision boundary for the same is shown below (Fig 9)



4. Random Dataset

With unrotated data, the split is easily made at X0 <= 2.525 which leads to two pure leaf_nodes classifying the samples into purely single categories having gini impurity = 0.

Decision trees are very sensitive to small variations in data. Also, their decision boundaries are always parallel to the axes. Thus, after rotation the linearly separable dataset (which could have been separated easily by a single straight line) has been separated with overly complicated decision boundaries of the tree. This means the model has been overfit and will not generalise well on test data. (Fig 10)

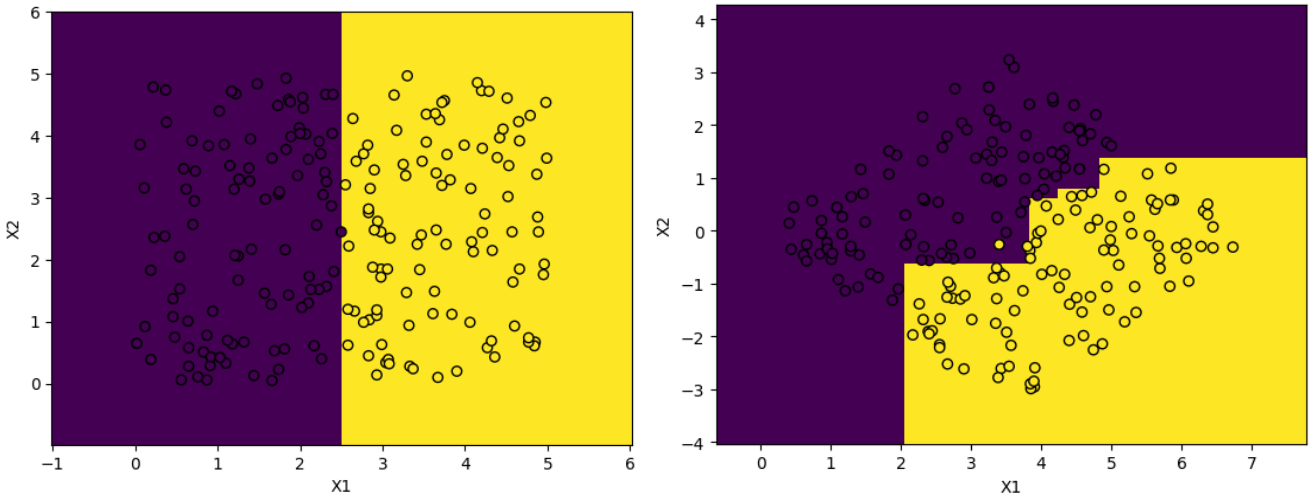


Fig 10

Q2. Regression

With max depth =3 the tree tends to fit the train data better than the tree with max depth = 2. However increasing max depths may lead to overfitting model. The depths of split have been indicated in the Fig 11.

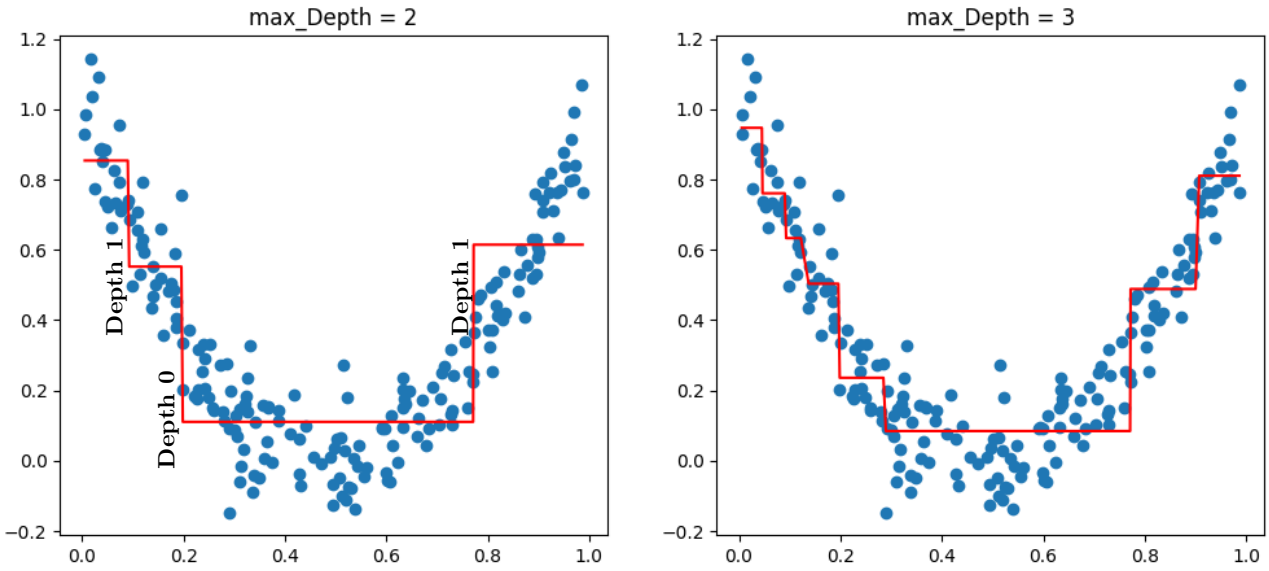


Fig 11

With min samples leaf = 1 the model tends to overfit the training data very strongly. However with min samples leaf = 10 the model is better than the previous one and more generalised. Fig 12.

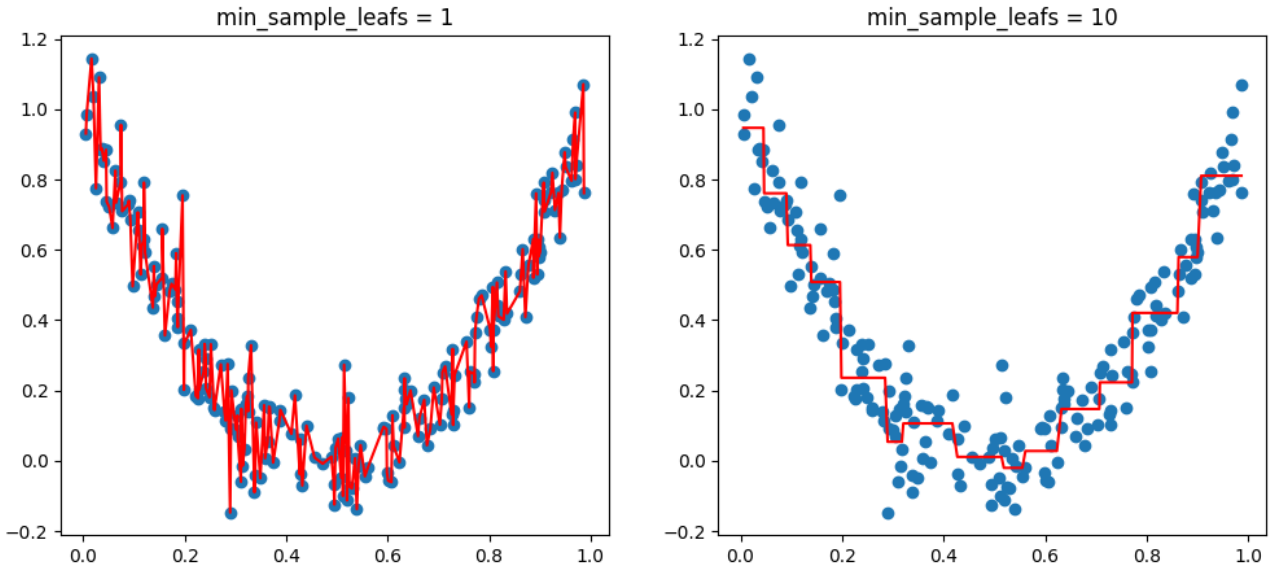
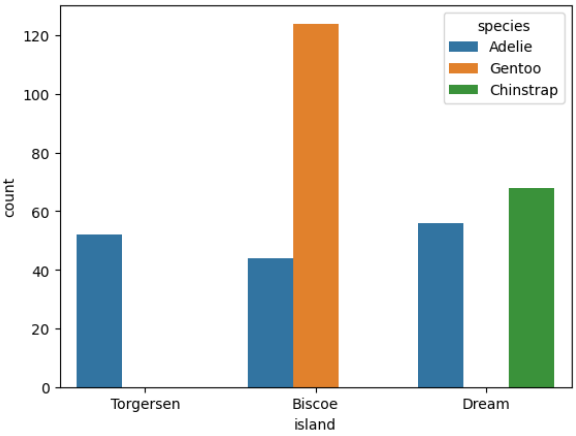


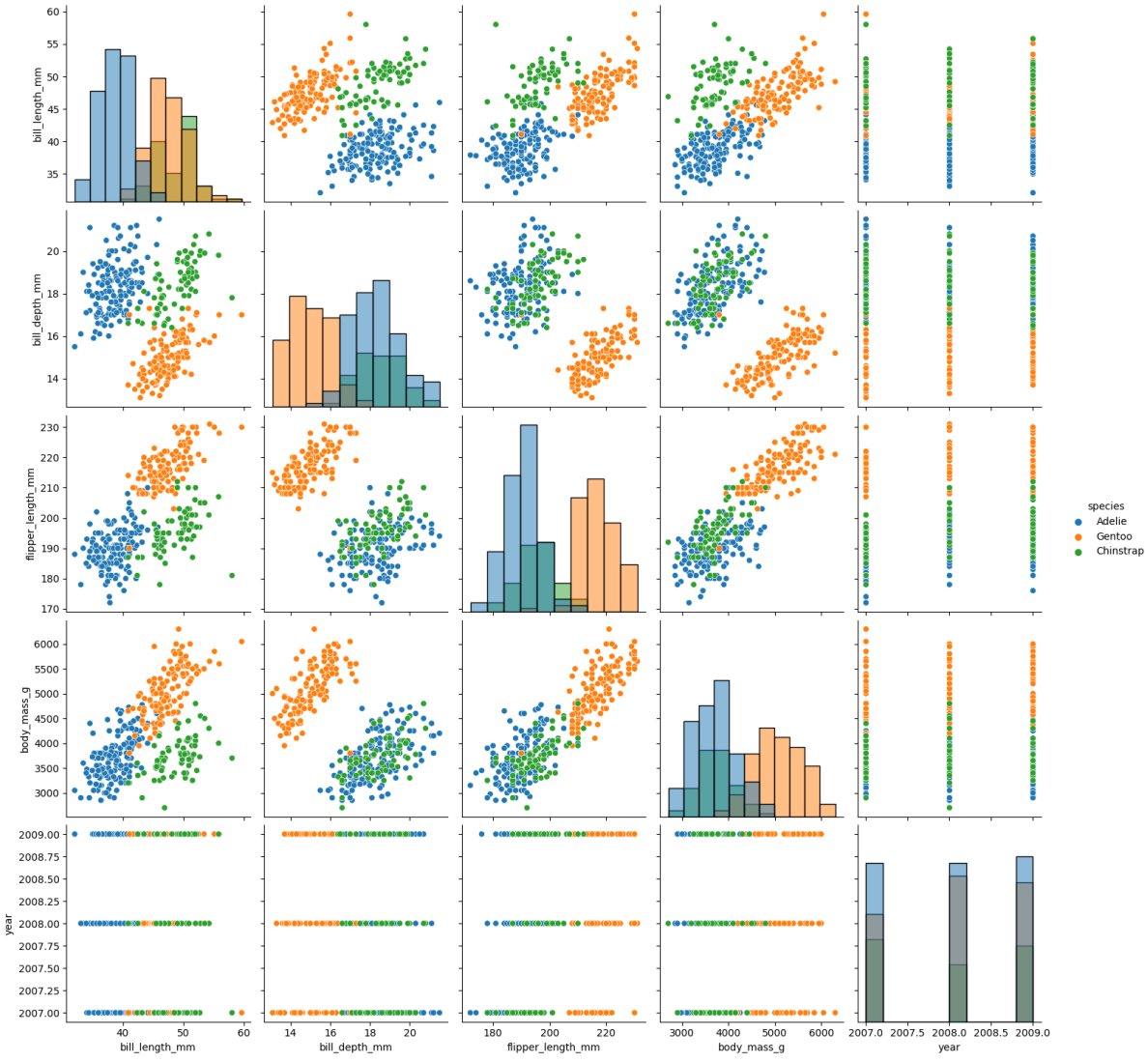
Fig 12

Q3. Scratch Implementation
1. Preprocessing & Visualisation

The columns having missing values are imputed using the most frequent entry. The categorical columns are Island, Species and Sex which have been processed with Label Encoder.
Also the pair plots have been plotted using seaborn. From pair plot we figure out that bill_length_mm and bill_depth_mm will be the two most suitable features for classification.

Also the gentoo species can be easily classified. (Orange) because it is least intermixed with other features. Also, Torgersen island only contains Adeline species. Gentoo is only found in Biscoe and Chinstrap is only found in Dream Island.





Q3. Scratch Implementation

The split function used is entropy. The formula for entropy and information gain are:
The feature which leads to the minimum entropy and hence max information gain is the most suitable criterion for split.

$$H = \sum_{i=1}^{N} p_i log_2(\frac{1}{p_i}) \qquad Gain(T, a) = Entropy(T) - \sum_{i=1}^{|a|} \frac{|a_i|}{|T|} Entropy(a_i)$$

The model has been trained on train data and tested.
The overall Accuracy produced is: 0.913

The class-wise accuracy is:
Adelie: 0.815
Gentoo: 0.9
Chinstrap: 1