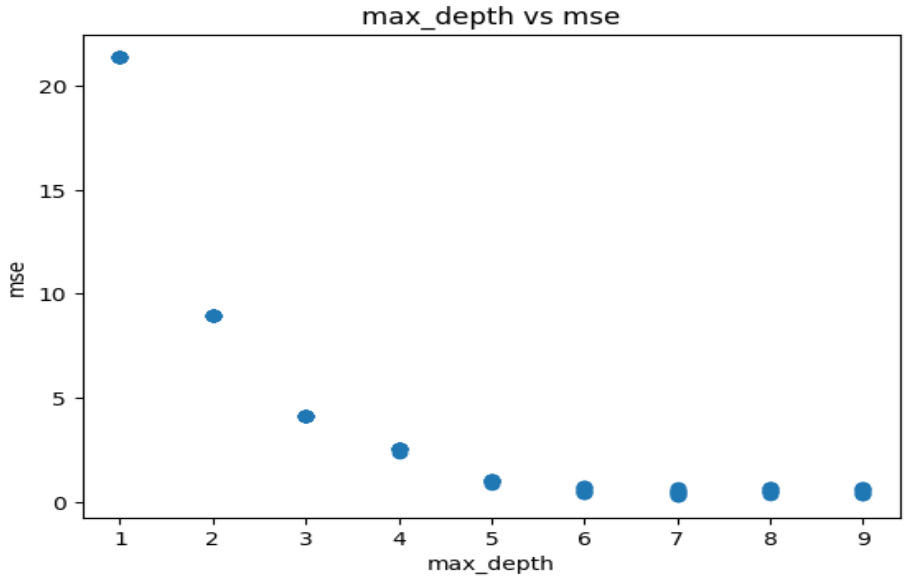
PRML LAB 2 REPORT

MANISH(B21CS044)

QUESTION 1:

1) In this part we have preprocessed the data and splitted in into Training data ,Validation data and Test data in ration 70:10:20. And here is no column to drop.
2) In this part we have used two features max_depth and min_sample split for hypertuning and finding best fit for our decision tree regressor.
   After tuning by using a for loop the hyperparameters come out to be max_depth=7 and min_sample_split = 3.
   We have stored Mean_square_error and plotted max_depth because max_depth was our prime parameter of tuning.Here is the plot between these two parameters.



3) In this part we have plotted our Decision Tree for our best parameters and have calculated mean_squared_error for our best model.
   Mse = 0.3231581810402186

   We have performed Hold out Validation , 5-fold-cross Validation , Repeated 5-fold-cross Validation.Here are the values and scores of each Validation.
   Mse_holdout = 0.31578041811334867
   Score_holdout = 0.9969703950593245

   5-fold-cross Validation values and score:
   Accuracy = [0.9969704 0.99779971 0.99750576 0.99662998 0.99738653]
   Mean_Accuracy = 0.997258475466845

   Repeated 5-fold-cross Validation:
   Accuracy = [0.99668782 0.99693124 0.99710518 0.99738464 0.99726382 0.99739911 0.9969944 0.99752128 0.99788233 0.99751797 0.99730749 0.99737353 0.99796867 0.99684421 0.99726517 0.99561743 0.99669398 0.99745355 0.9950344 0.99750392 0.99721223 0.99216258 0.99770985 0.99752449 0.99675124]
   Mean_Accuarcy = 0.9969244218719945

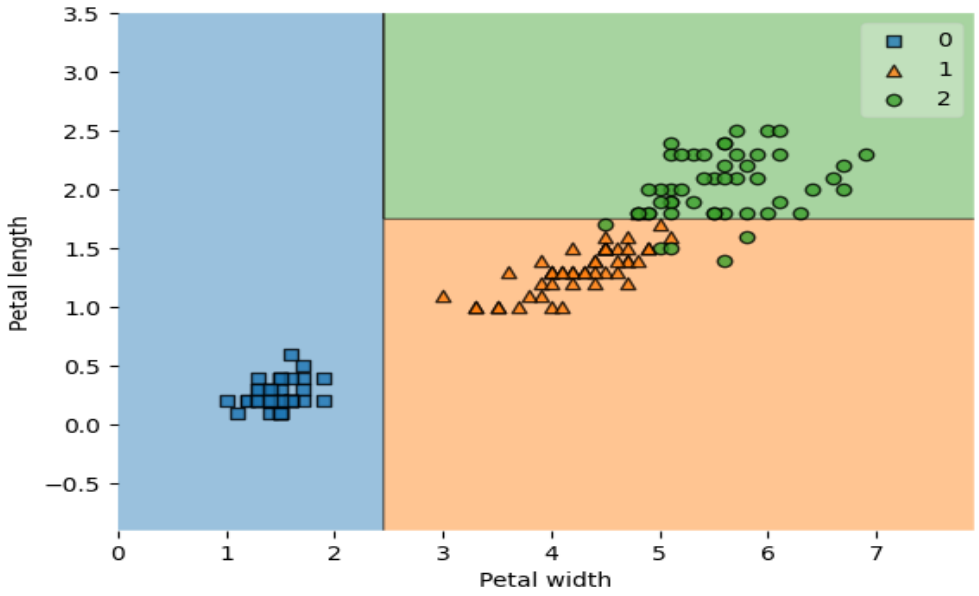
4) In this part L1 and L2 denotes the criterion squared_error and absolute_error and here the scores :
   0.9969703950593245

   0.9974361512498044
   From here we can see that absolute_error performs slightly better.

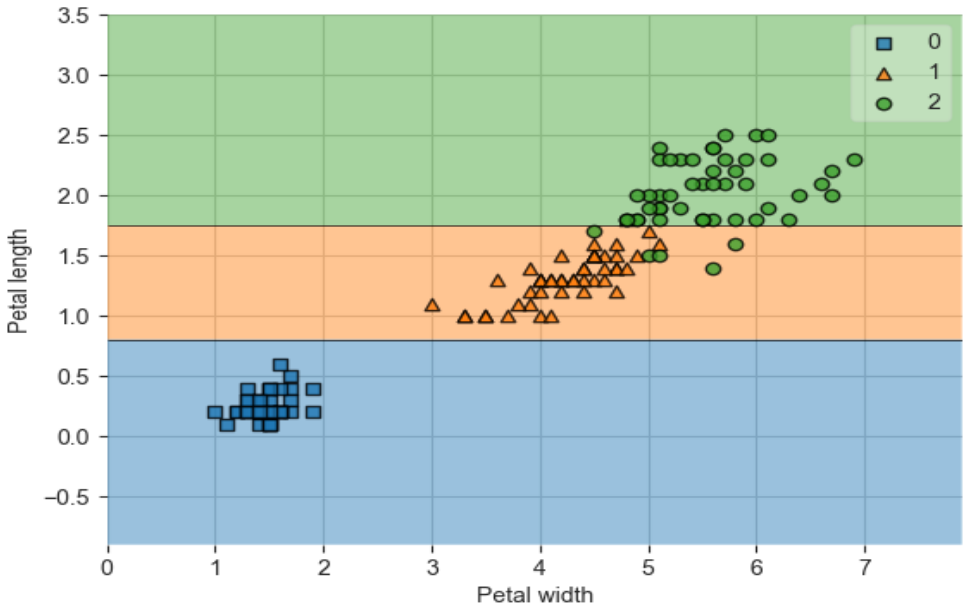
QUESTION 2:

CLASSIFICATION:

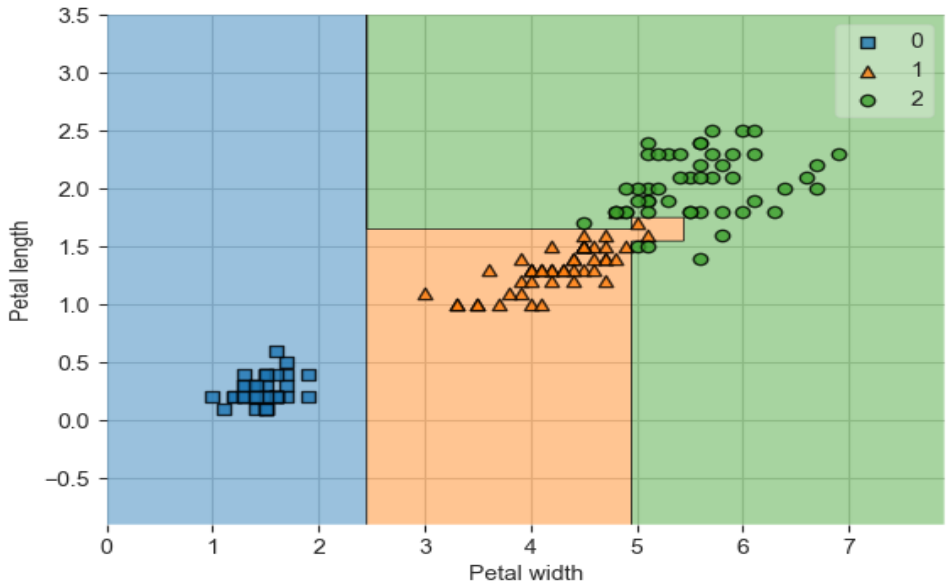
1) In this part we have preprocessed the data and splitted in into Training data and Test data in ration 80:20. And here are Sepal length and Sepal width to drop
   We have trained the Decision Tree using max_depth = 2. We have plotted the decision boundary using mlxtend library function plot_decision_regions
   Here are the decision boundary plots:



2) In this part we have dropped the data point with petal length 4.8 cm and 1.8 cm width.Here is the decision boundary:

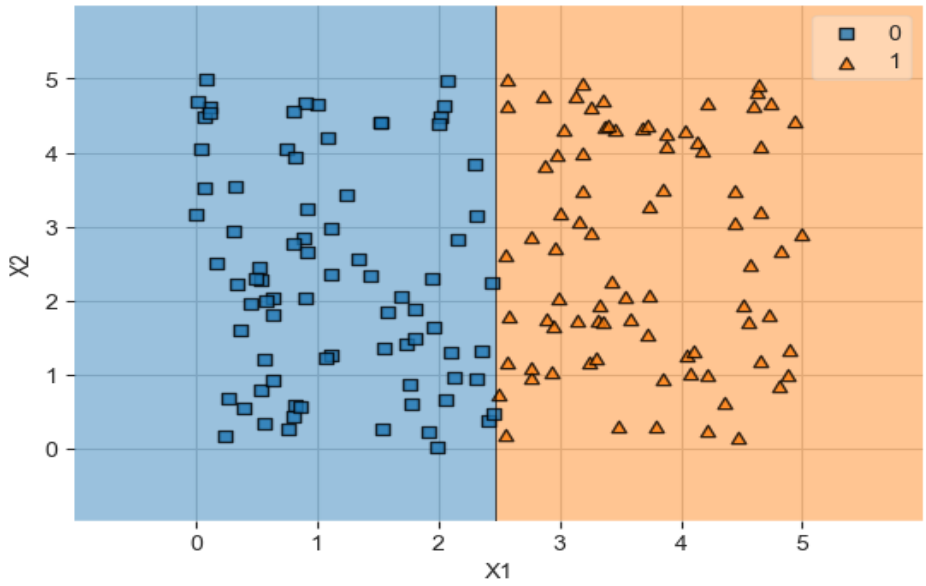


3) In this part we have taken max_depth = None.
Here is the decision boundary:



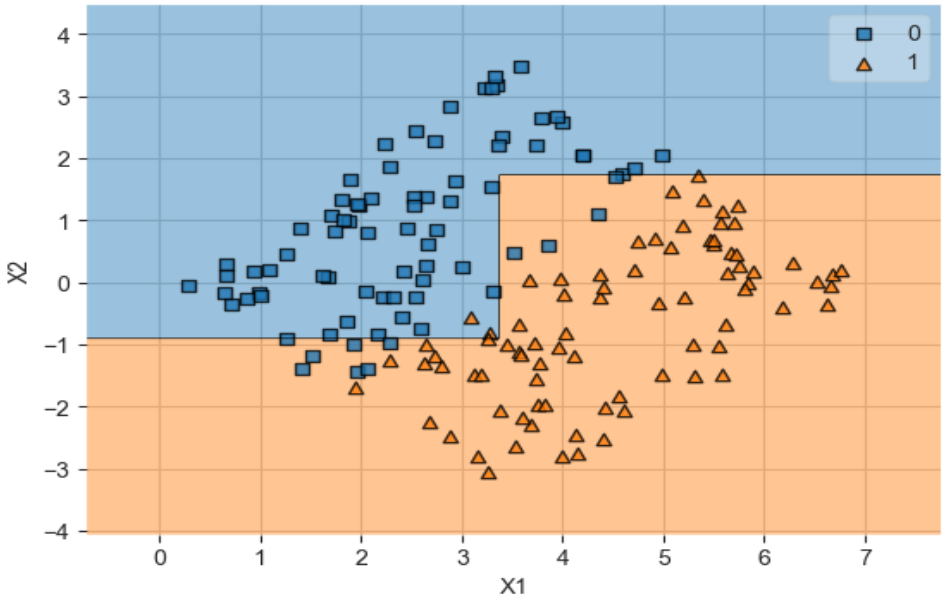
We can clearly see that in part 1 data was not perfectly fitted between Class 1 and 2 but in part 3 data is perfectly fitted.This shows that not setting the max_depth will lead to the overfitting of the dataset.

4) In this part we have generated the data points regarding the given conditions dataset having 2 attributes(X1 and X2), and 2 classes (y=0 and y=1) .X1,X2 are randomly sampled from the range (0,5). y=0 when X1<2.5, and y=1 when X1>2.5. The dataset should have 100 data points for both the classes.
We have trained the Decision Tree on max_depth = 2
Here is the decision boundary:



Now we have rotated the data points by 45 degree by setting x = x1+x2/2**0.5 and

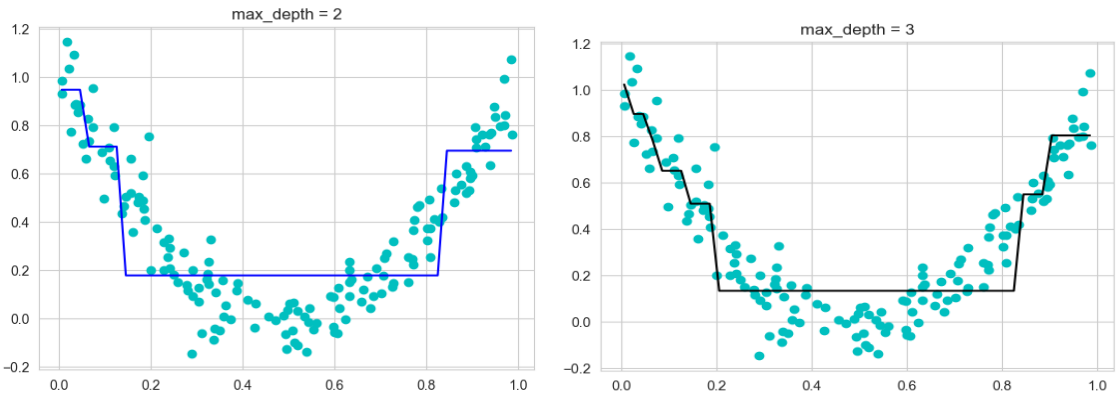
y = y1+y2/2**0.5 and Here is the decision boundary for the same:



5) In part 2,3,4 we have plotted the different boundary and we have observed that if we drop the biggest datapoint then the boundary separates the features better Mse reduces. In part 3 we have observed that not setting max_depth will lead to overfitting and in part 4 we have seen that rotating the data points by 45 degrees will lead to rotation of the decision boundary by the same.

REGRESSION:

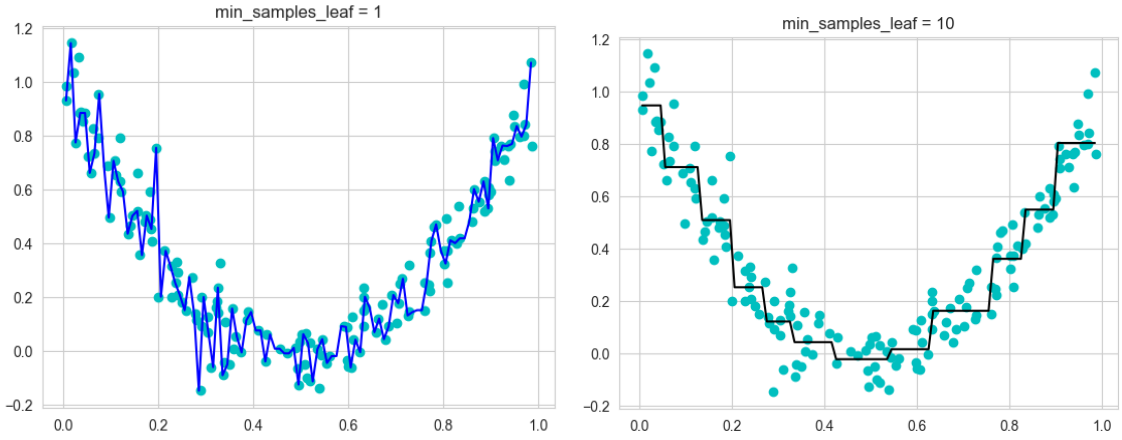
1) In this part we have trained two decision trees on max_depth parameter given values = 2,3 to max_depth.
   We have plotted the decision regions regarding each depth and the data points along in the same plots and here are plots:



From here we can clearly see that Tree having max_depth = 3 is fitting the dataset more than that of Tree with max_depth = 2. Blue boundary plot is of max_depth= 2

and Black Boundary plot is of max_depth = 3 .

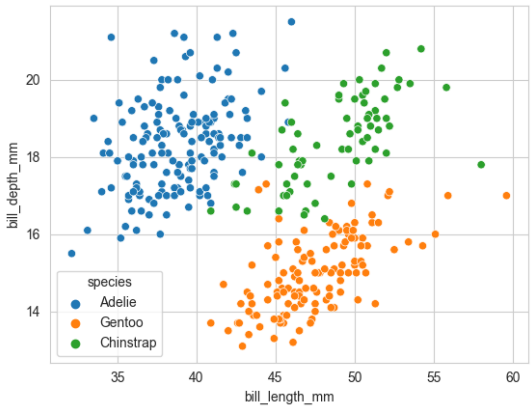
2) In this part we have trained two decision trees on max_samples_leaf parameter given values = 1,10 to max_samples_leaf.
We have plotted the decision regions regarding each leaf split and the data points along in the same plots and here are plots:

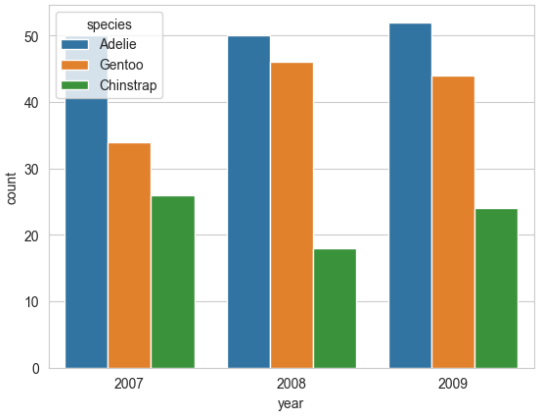


In these plots the one with blue boundary is for min_samples_leaf = 1 and with black boundary is for min_samples_leaf = 10. From the plots we can see that if we declare the leaf node having 1 sample is overfitted in comparison to one with a minimum 10 sample points in the leaf node.

QUESTION 3

1) In this part we have performed preprocessing of data filling the NA with means and dropping the column of Sex. Then , we have performed the categorical encoding of the columns year and island , and we have also performed ordinal encoding for Species. Then plotted graphs for visualization of data and after we have splitted the data into 80:20 ratio.
Here are some graphs:





2) We have implemented the cost function as Entropy and also along with that we have also implemented the function of Information gain.
Entropy of y_train = 0.5307198966433953

3) In this part we have performed the categorical encoding of the features in 2 categories 1 and 0 based on the selecting the data point which will give us best Information Gain.
For less complexity we have iterated the above function to check over the array of thresholds only not over every single data point in that Feature column.
We have implemented a function to get the array of thresholds.
In that function we are checking for the points where the y changes for two continuous points and appending the mean of those two points.

   Here are the values of the following features for optimal encoding:
Bill_length_mm = 41.150000000000006

   Bill_depth_mm = 16.35

   Flipper_length_mm = 208.0

   Body_mass_g = 4150.0

4) In this part we have implemented Two functions to select the best attribute according to the Information Gain ,the one having maximum Information Gain will be selected and to split the data according to that attribute.The left side data of that Node will be having that attribute value = 0 and right side data of that Node will be having that attribute value = 1.

5) In this part we have implemented the Decision Tree using a class DTClassifier and set the data set to grow According to the two parameters:
    a) One having max_depth = 5
    b) If Information gain is = 0 then it is leaf node then no need to split it further.
    c) For this we have implemented the fit function which will fit the data and make the Decision Tree for us.

6) In this part we have implemented the predict function which will predict the Class for X_test data and will give the whole array having the values 0,1,2 according to the class.

7) In this part we have calculated the avg_accuracy and Classwise_accuracy of the X_test data.
Here are the values for the same:

Average Accuracy: 0.9130434782608695

Class Wise Accuracy: [0.8148148148148148, 1.0, 0.9615384615384616]