REPORT - IML Lab5

Vrushali Pandit - B20BB047

Task 1:
https://colab.research.google.com/drive/1J3WdiAG2OTFOdJwkwkK2bj_0NTaxhS52?usp=sharing

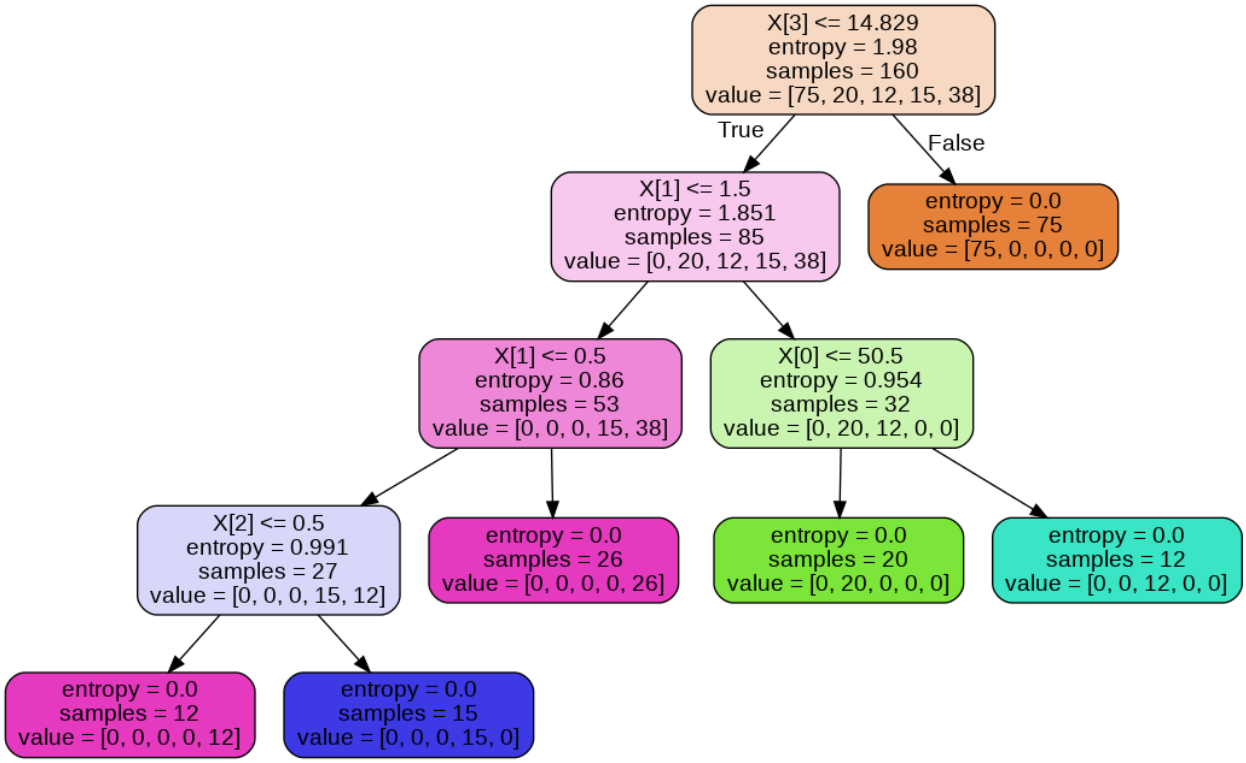
Data: Data collected for which drug should be administered to the patient (target variable) depending on features (Age, BP, cholesterol, Na_to_K and sex).

Categorical Features:

- BP, Cholesterol - ordinal, encoded using Ordinal Encoder or Label Encoder
- Sex - Categorical (no order) - encoded using OneHotEncoder

Steps:

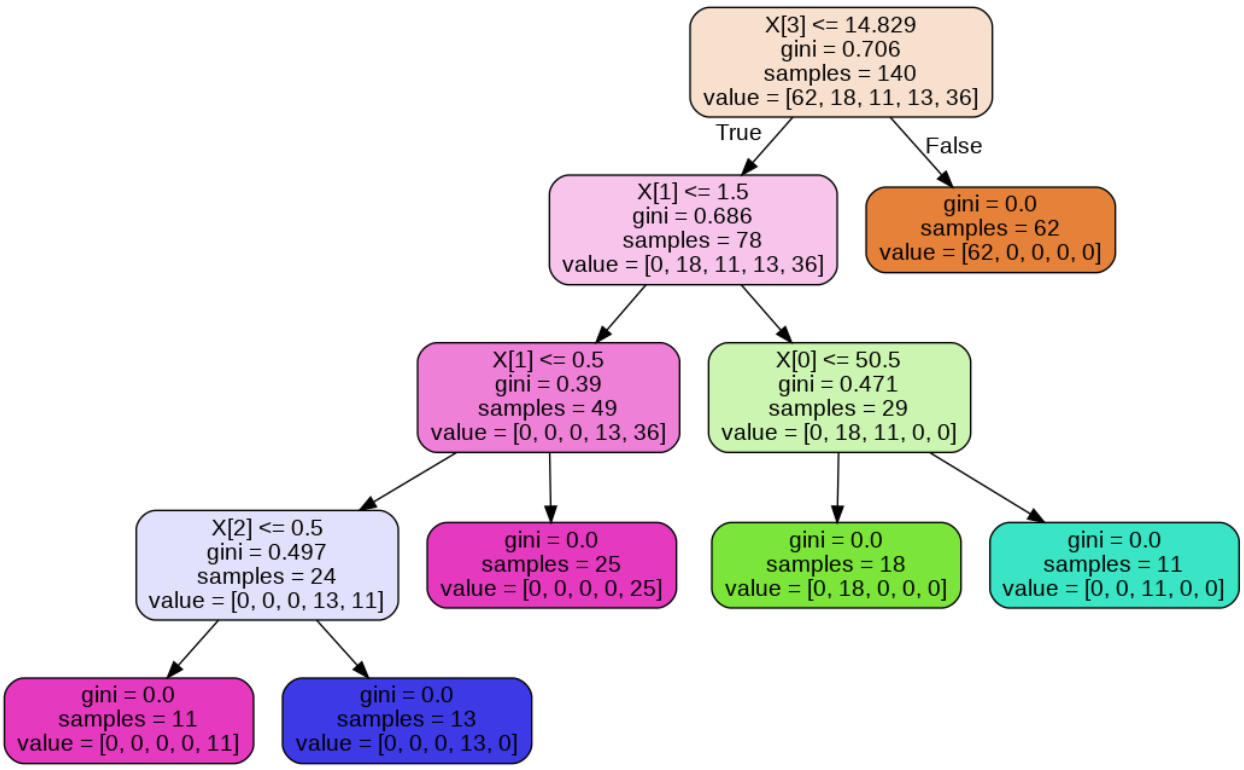
1. Split the dataset using 3 splits.
2. Using sklearn library implemented a Decision Classifier to classify the target on the **80:20 split using Entropy**



```
Model Accuracy |  1.0
Test Accuracy |  1.0
```

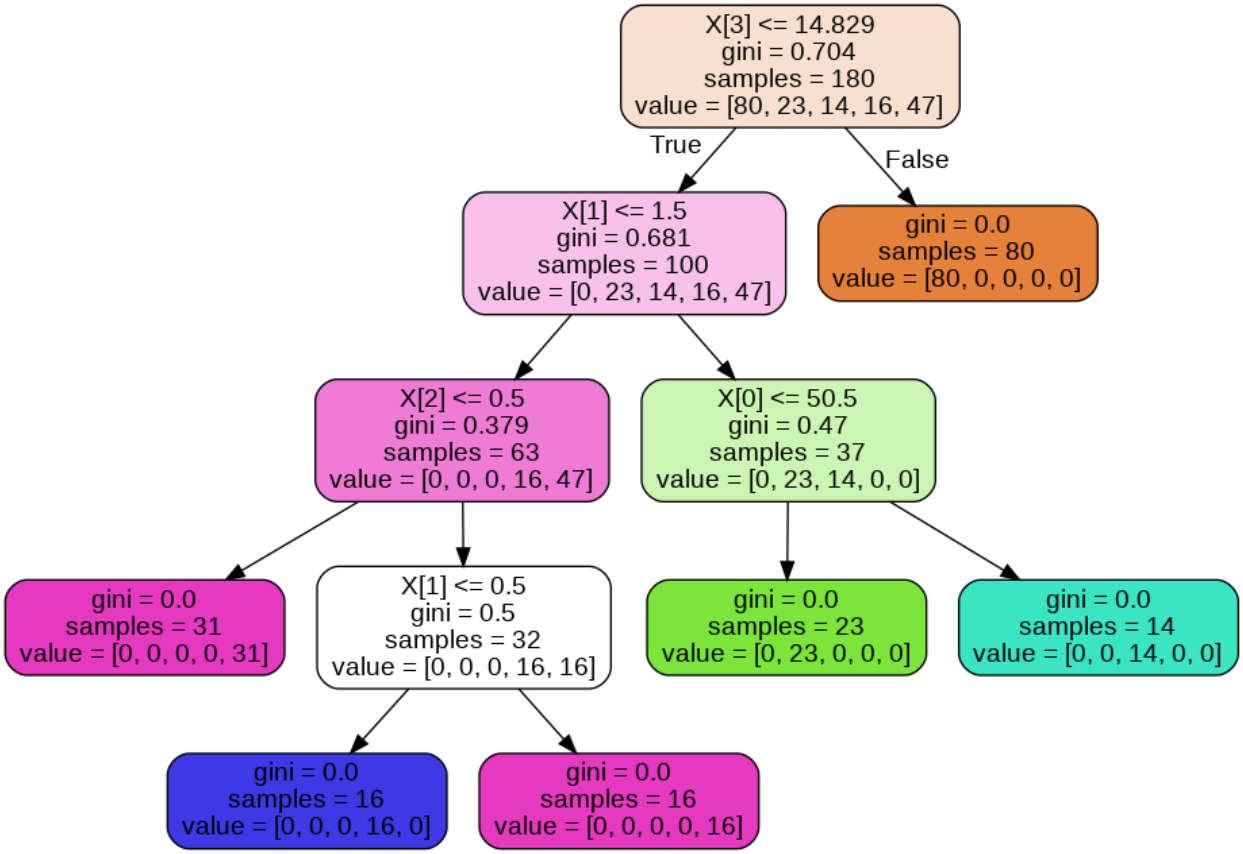
As we can see, the model is performing extremely well. While this is unusual it can be explained by the fact that our testing data is very small.

3. Now on the 70:30 and 90:10 splits using Gini Index




```
Model Accuracy |  1.0
Test Accuracy |  1.0
```

<u>70:30 split</u>




```
Model Accuracy |  1.0
Test Accuracy |  1.0
```

90:10 split

In all cases we get a test and train accuracy of 100%
We can't say exactly if the decision tree is overfitting or not due to its performance on the test data. However as the number of samples is rather small, we have too little information to make out whether it is overfitting or not.

Task 2 :

https://colab.research.google.com/drive/1B0a09X_2A5z_P_5H1cWNVtITLbBLfLkK?usp=sharing

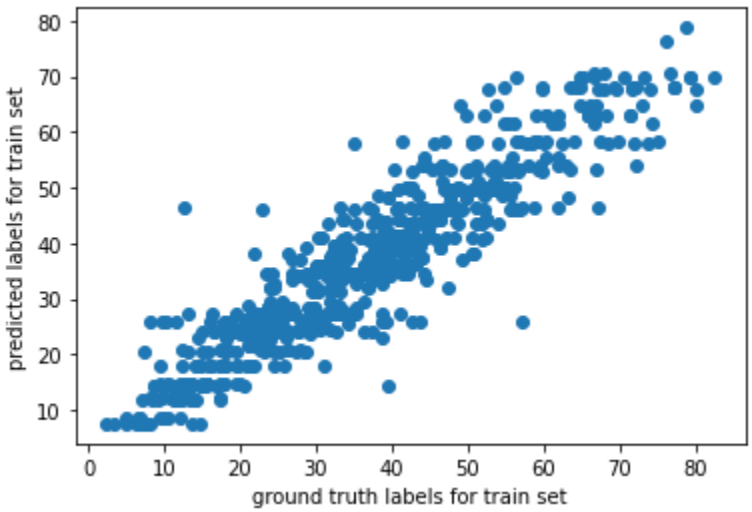
Data: Data collected for Concrete compressive strength (target variable) based on the other features.

Steps: No categorical data, hence no need to encoding.

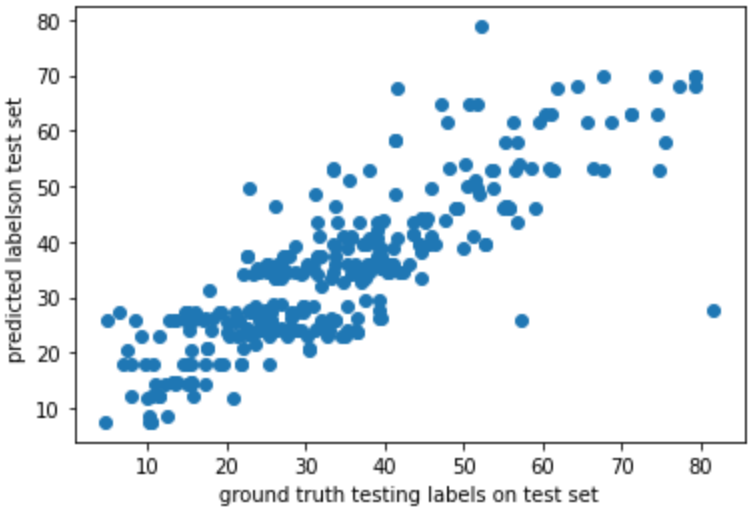
- Split the data according to my roll number and apply the DecisionTreeRegressor function from sklearn library to fit using the MAE criterion.
- However this led to overfitting because the default *tree_depth = None, min_samples_split=2, min_samples_leaf=1.*

- To prevent overfitting, I looped through a couple of values for *min_samples_split.*

- If there are *samples <= min_samples_split* at any internal node, then the tree will not split further and it will become a leaf node.
- To choose the right value of *min_samples_split* , I iterated through some fixed values and chose the one which returned the smallest MSE.
- MSE and MAE or any regression based errors only provide us with the "error" in predictions but no metric to compare whether that error is small or large.
- Hence to visualize the tree, I plotted ground truth training labels against predicted training labels.



- This shows an approximate relation of slope = 1.
- I also plotted ground truth testing labels against the predicted ones,



- While there isn't as clear of a linear relationship here, it is still there.
- Finally the errors were found as follows

```
Mean squared error in testing =  79.36268848826317
Mean absolute error in testing=  6.616807073365326
```

- Next I repeated the same process but with a DecisionTreeRegressor with "MSE" as the criterion.
- This underperformed the previous "MAE" based tree.

```
Mean squared error in testing =  102.30420137540452
Mean absolute error in testing=  7.421893203883495
```

- We cannot print accuracy reports or confusion matrices because those are only defined for classification tasks and not regression analysis.