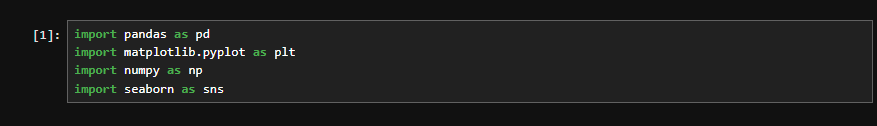
# Code Alpha Internship Task 1 (Iris classification)

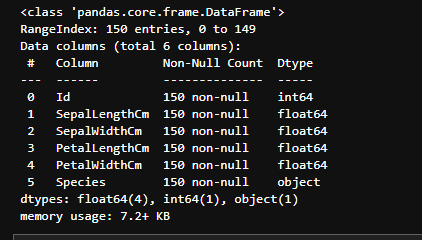
In the first part of the task, firstly I imported all the necessary libraries for Exploratory data analysis and data visualization.



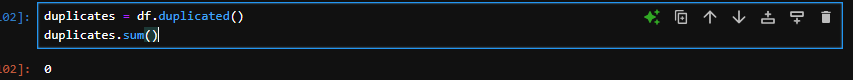
The very first step I took is manually checking my data set, firstly I manually checked data set to see if there is any kind of imbalance or unwanted things, after that I started my Exploratory data analysis.

**Exploratory data analysis**

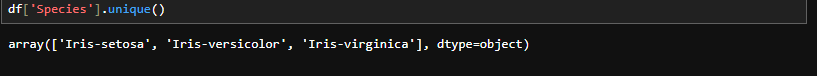
In EDA part, firstly I used function like **describe()** and **info()** to see the mean, median, standard deviation, missing value analysis, fortunately the data set didn’t contain any missing value.So, in this part I didn’t had to do much.



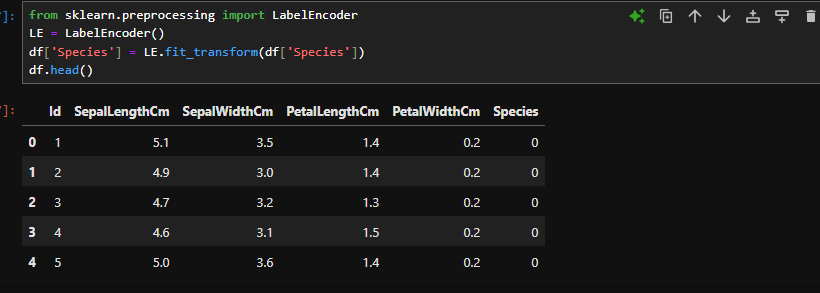
I again did duplicate analysis and again there was no duplicates in my data set



The target column which is species column had object data type, Machine learning models don’t understand Characters so I had to use Label encoder to convert it into numerical, firstly I analyzed the unique values.



I found out that there are 4 unique values , So I used label encoder which converted these 3 values to 0 1 and 2

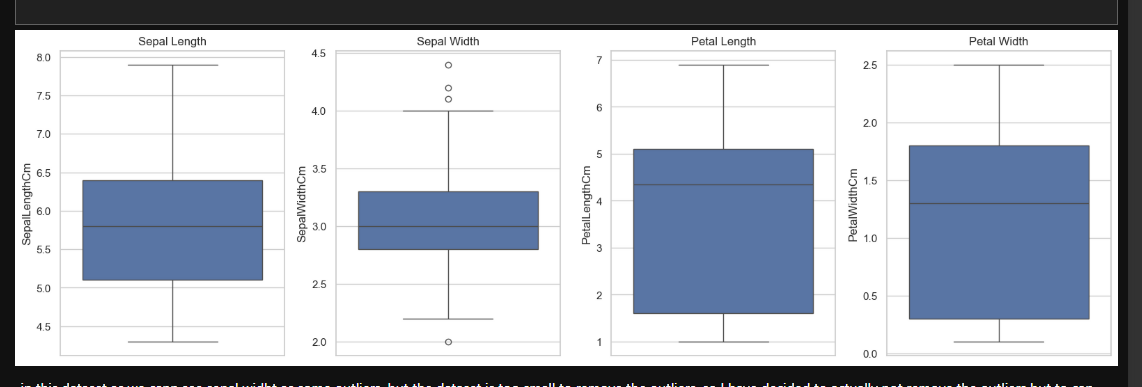


I removed ID column, as it had no pattern and created collinearity for no reason .

**Outliers analysis**

I did some outliers analysis and found out that only the sepal width column had some outliers.

I used box plot to analyse that .

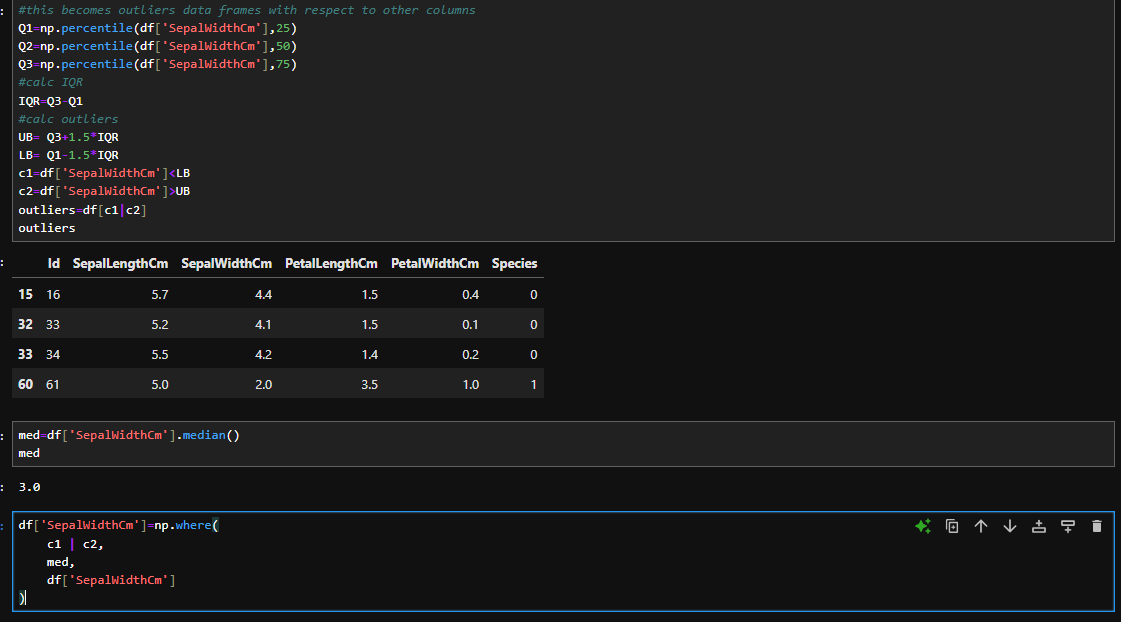


Having outliers in the data set is bad , as outliers lead to inaccurate model prediction, we have 3 ways to treat outliers :

1. Remove the outliers
2. Cap the outliers to q1 or q3
3. Cap the outliers to median or q2

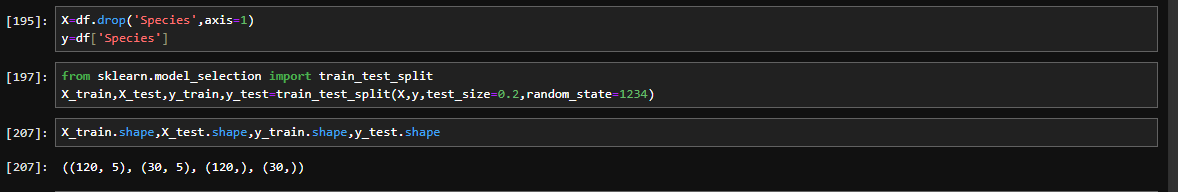
The option 1 isn’t good option because we have a very small data set (150 rows), removing it meant we lose important samples, I only use this when the data set is large, I opted for the 3rd method and removed all the outliers using numpy.

I used outliers formula to analyse outliers.

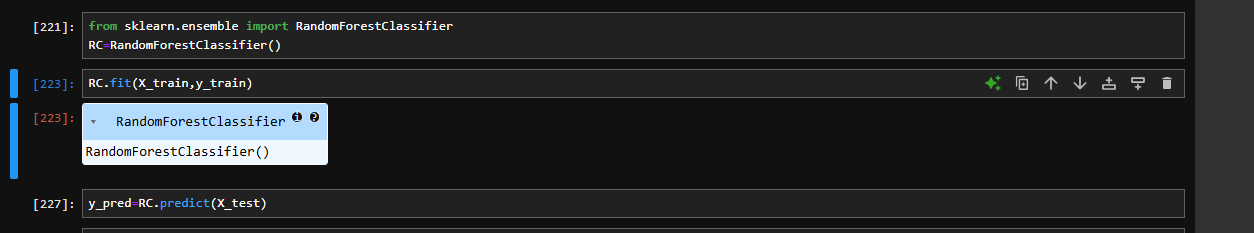


**Model Building**

In this process, I straightly opted for random forest classifier which is a great example of bagging method, the first step I did was splitting my data into train and test, firstly I separated input columns and target columns and I used train\_test\_split from sklearn.modelselection to split it into train test. I separated 20% of my data for testing purpose and used random state as 1234.



On the next step I used my random forest model to fit and predict.



**Model Evaluation**

For model evaluation I used metrics like accuracy,f1 score, precision and recall. Best thing is my model was able to predict each and every test outputs correctly giving me a 100% accuracy

