HEART OF RANDOM FORESTS

- Splitting criteria

- Suppose we have only two columns, C1 and C2 which act as the predictor variables for our model and we have a target value.
- Now there are lots of methods with which we can define the **SPLIT FOR OUR NODE IN THE TREE.**
- **Remember that this split is done on the data values of the predictor variable and it loops through all the values of the feature and splits them into a subset >(greater than) or <= (less than equal to the split).**
- **BASIS- this is done on the basis that the standard deviations of the two subtrees formed is AS LOW AS POSSIBLE. Standard deviation of what? The Y VALUES IN THE SUB-TREEEEEEES.**
- **SCORE is the weighted sum of the STANDARD DEVIATION OF THE LST AND THE RST.**
- **This is called the information gain of our tree. How much better does our score get if we find a better split?**
- **So minimizing the group standard deviations is the same as minimization of the root mean squared errors.**

SPEEDING UP RANDOM FOREST SPLIT LEVEL

- First of all speed up STANDARD DEVIATION.
- Formula to use-> root of(mean of squares -square of means)
- HOW? Sort the data.
- Now as you go from one observation to the next you will find that you **DON'T HAVE TO CALCULATE STANDARD DEVIATION EACH AND EVERY TIME YOU GO TO THE NEXT VALUE.**
- You can just divide the data set into LHS and RHS , lhs is the data <= **CURRENT VALUE** rhs is the data > **CURRENT VALUE.**
- Now in std. For lhs you calculate the sum and sum of squares by just subtracting the value and its square and for the rhs adding the value and its square gives the sum and the sum of squares.
- Thus, sorting takes O(nlog(n)) time to make it a faster algorithm.
- Note- argsort method is used to gather the sorted indices.