Classification Assignment

Name: Tushar Choudhary

Roll Number: 2019111019

Attributes I have considered to predict Mw

List of all attributes present in the data:

```
0 Sl. No. nan
1 YEAR  nan
2 MONTH  nan
3 DATE nan
4 ORIGIN TIME (UTC)
5 nan (IST)
6 MAGNITUDE Mw
7 nan Mw
8 nan Mb
9 nan Mb
10 nan Ms
11 nan ML
12 LAT (N) nan
13 LONG (E) nan
14 DEPTH (km) nan
15 INTENSITY MM
16 nan MMI
17 nan MME
18 LOCATION nan
19 REFERENCE nan
```

Out of these, I picked **all** the numerical attributes initially:

Column Index	Attribute Name	Number of null values
1	Year	0
2	Month	18
3	Date	57
4	Time	31803
12	Lat	0
13	Long	0
14	Depth	2178
15	Intensity	52948
6	Magnitude	12054

Location was skipped as location is already covered when including latitude and longitude. And reference was skipped as it is not a numerical attribute.

The data set had nearly 52989 rows, and as a lot of these rows had null readings in the entry for “Time” and “Intensity”, these attributes were skipped and I used the remaining 6 attributes to predict Mw.

Final Attributes used along with data representation:

In [20]:
```
attributes = [1,2,3,12,13,14,6]
att_names = ['Year', 'Month', 'Date', 'Lat', 'Long', 'Depth', 'Magnitude']
X=df.iloc[last_row+1:,attributes]
X
```

Out[20]:

	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 12	Unnamed: 13	Unnamed: 14	Unnamed: 6
10	-2474	0	0	71	24	0	7.5
11	-325	0	0	71	24	0	7.5
12	25	0	0	72.9	33.72	0	7.5
13	26	5	10	17.3	80.1	NaN	6.1397
14	26	5	10	26	97	80	6.1397
...	...	...	...	...	...	...	...
52994	2019	7	28	32.8°N	78.4°E	10	3.2
52995	2019	7	28	25.5°N	90.4°E	70	3.6
52996	2019	7	28	23.2°N	86.5°E	22	4
52997	2019	7	29	32.8°N	76.4°E	20	4.3
52998	2019	7	31	20.0°N	72.8°E	10	3

52989 rows × 7 columns

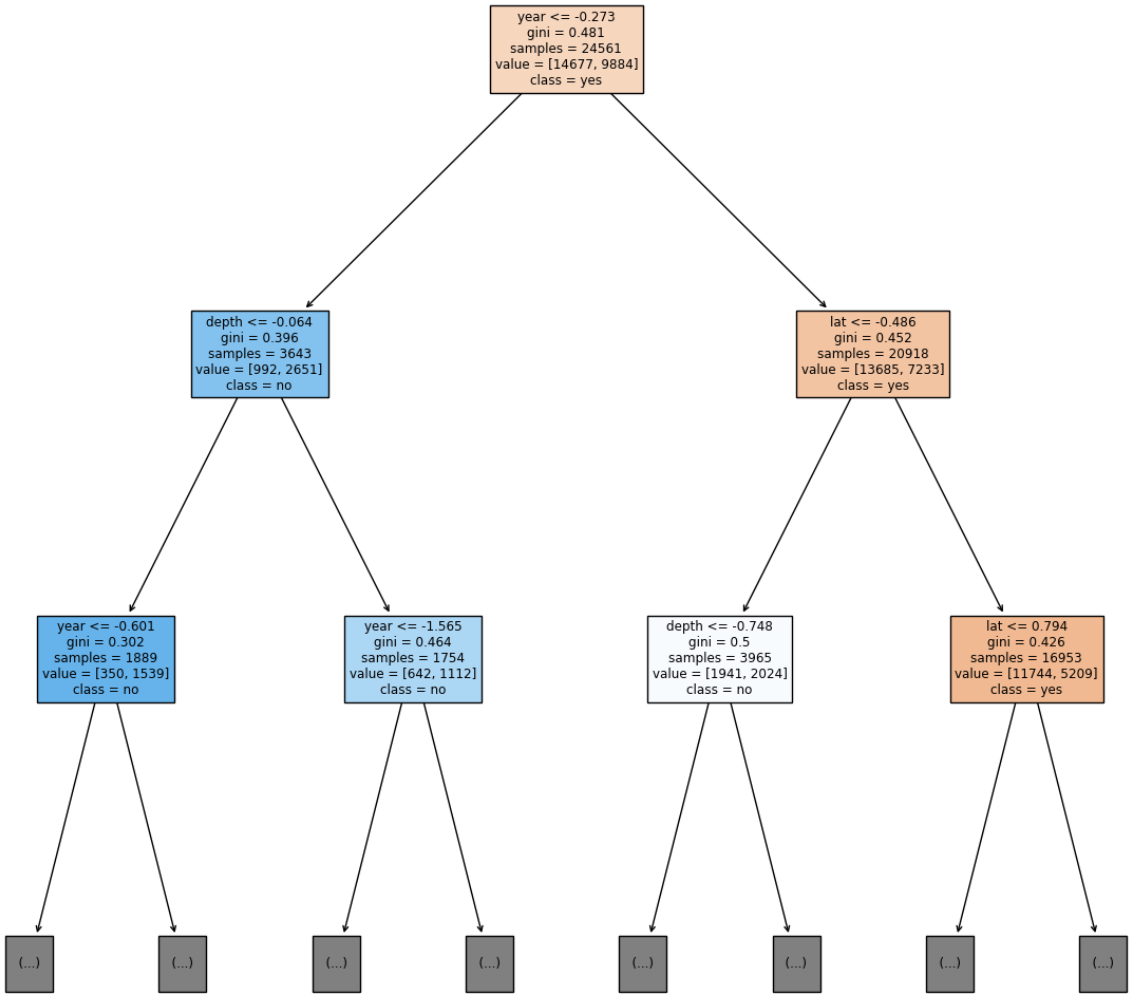
After this:

- All rows with null value in place for Mw were removed as this is the attribute we will be predicting.
- The latitude and the longitude were parsed into numerical values.
- All the incomplete (nan entries) for different columns were imputed by “most frequent” or “mean” values.

Once the data was cleaned, it was split into training, validation and testing sets in the proportion 60:20:20. For all the classifiers, the training was done on the train set, validation set was used to choose the best value of the parameter and then performance of the best parameter was reported on the test set.

Threshold value was kept equal to the mean of Mw which was equal to **4.537564295818176**.

Earlier the data was standardized as well, but this made the results non-interpretable for decision trees. Hence I commented out this part of the code. A sample tree for standardized data is below:

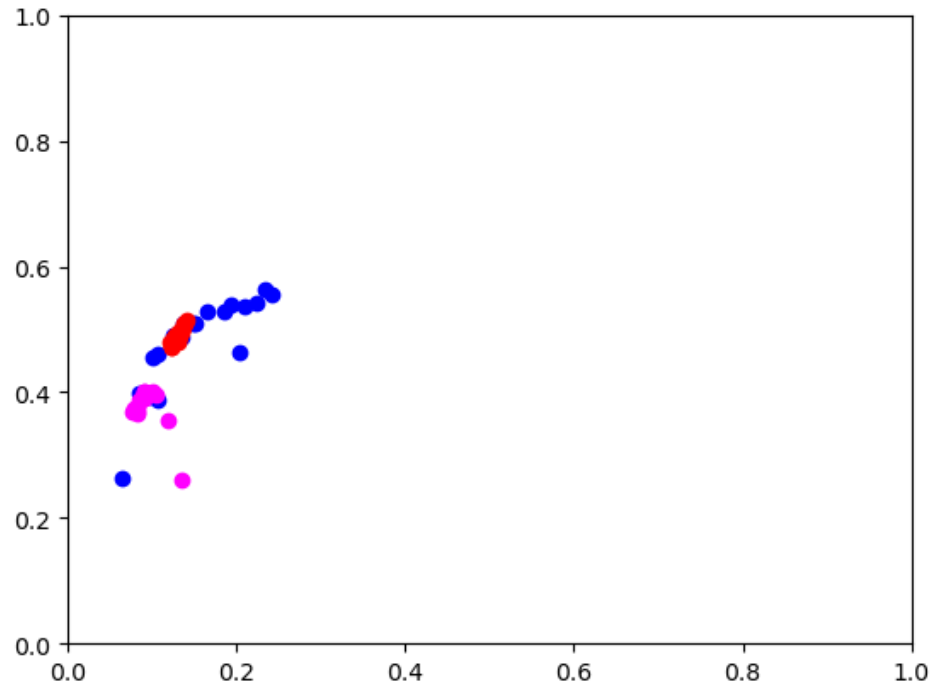


Parameters

1) KNN: The value of K varies from 35 to 65 inclusive.
2) Decision Tree: The value of D varies from 1 to 20 inclusive.
3) Random forest classifier: The value of n_estomator varies from 1 to 20 inclusive.

In every case, the confusion matrix was calculated for each value of the parameter, the best value of the parameter was selected based on this.

ROC plot:



In the above plot, red corresponds to KNN, blue corresponds to decision trees, and magenta corresponds to random forest classifiers.

Testing results for KNN:

```
Best K = 56

Confusiong matrix after using best K on testing set
[[4329  634]
 [1647 1577]]
```

Recall = 0.48914392059

Testing results for Decision Trees:

```
Best D=5

Confusiong matrix after using best D on testing set
[[4537  426]
 [1908 1316]]
```

Recall = 0.4081885856

Testing results for Random Forest Classifier:

```
Best n_estimator = 7

Confusiong matrix after using best n_estimator on testing set
[[4473  490]
 [1904 1320]]
```

Recall = 0.40942928039

Higher recall value implies lower false positives which is important in an earthquake setting. Since the recall value for **KNN** is highest, it is recommended to use that **when classifying Mw to above or below average when data is processed in this fashion.**

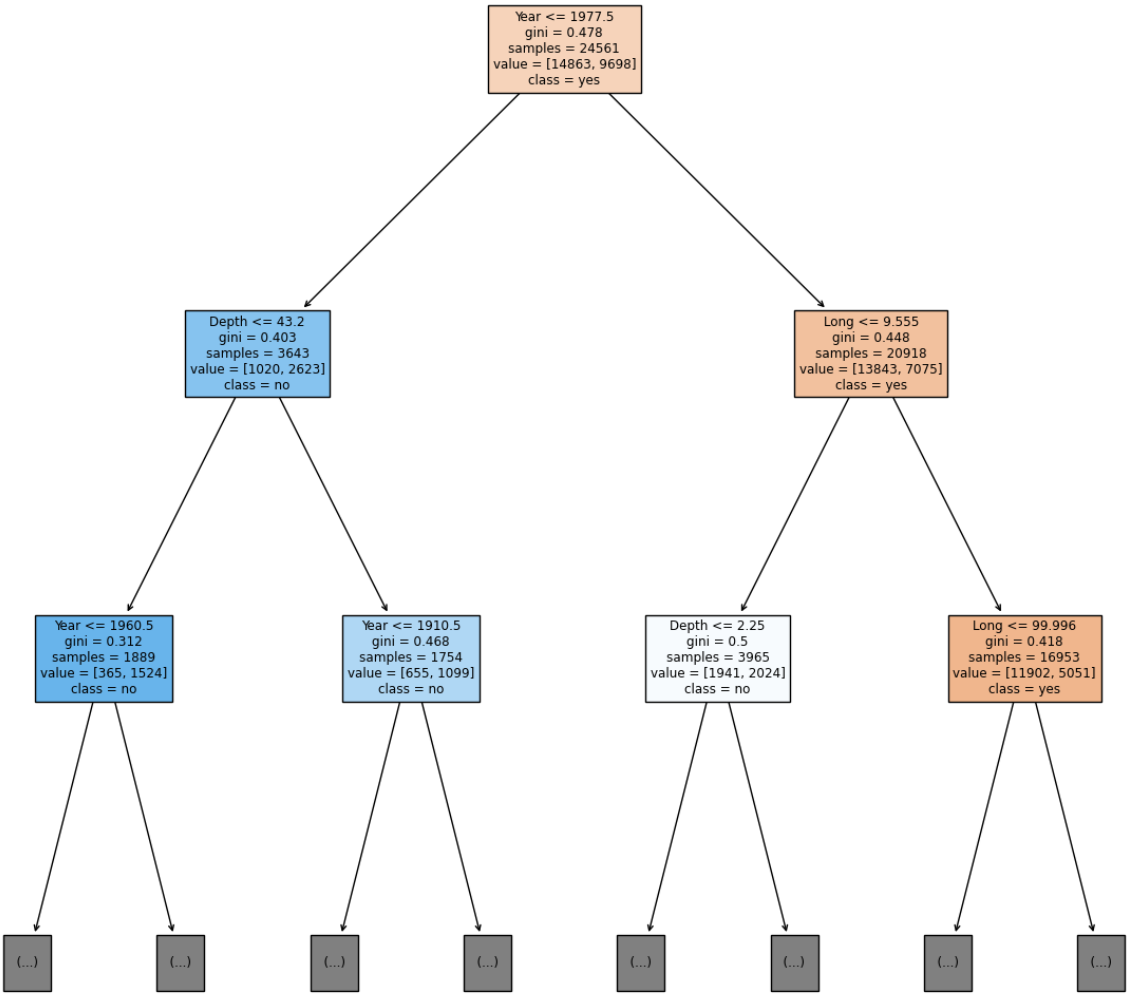
Other ways of data cleaning and preprocessing may give other classifiers in the result.

We can get the best value of the parameter by calculating the area under the curve and choosing the parameter with maximum area. The parameters and the area obtained by this method are:

KNN:
K = 41, area = 0.752

Decision Trees:
D = 8, area = 0.741

Random Forest Classifier:
N_estimator = 8, area = 0.747



If we have to consider only 2 values then we will consider the **year** for sure as it is at the top. The other feature we will consider is **depth** as it is at level 1 and has a lower gini value compared to longitude.

Feature Transform which we can do to improve the classification is generating another parameter **Day number** = 365*year + 30*month + day. This will help us get better differentiation between earthquakes that have occurred in a close time gap. Another is proper processing of degree based attributes such as latitude and longitude by multiplying by minutes/seconds instead of directly replacing degree sign by decimal. After doing these changes the output improved to:

```
Best K = 59
Confusiong matrix after using best K on testing set
[[4375  841]
 [1632 1648]]
```

Recall = 0.50243902439.

We can see improvement in recall as earlier it was 0.48914392059.