Classification Project

September 08,2020

Team Number: 6

Team Members:

Aayush Goel - 20171188

Gaurav Batra - 20171114

Problem Statement

- We are given earthquake data for India over different years. This data contains attributes like date and time of earthquake along with its locations (Longitude, Latitude) and Magnitude (Mw).
- We need to first convert the dataset into a binary classification dataset using an appropriate threshold and then apply K-Nearest Neighbours and Decision Trees to it to solve the classification problem.
- We use metrics like accuracy, recall, precision and F1-score to test the models and decide which works better in different conditions.
- We also suggest ways to improve these metrics at the end of the report.

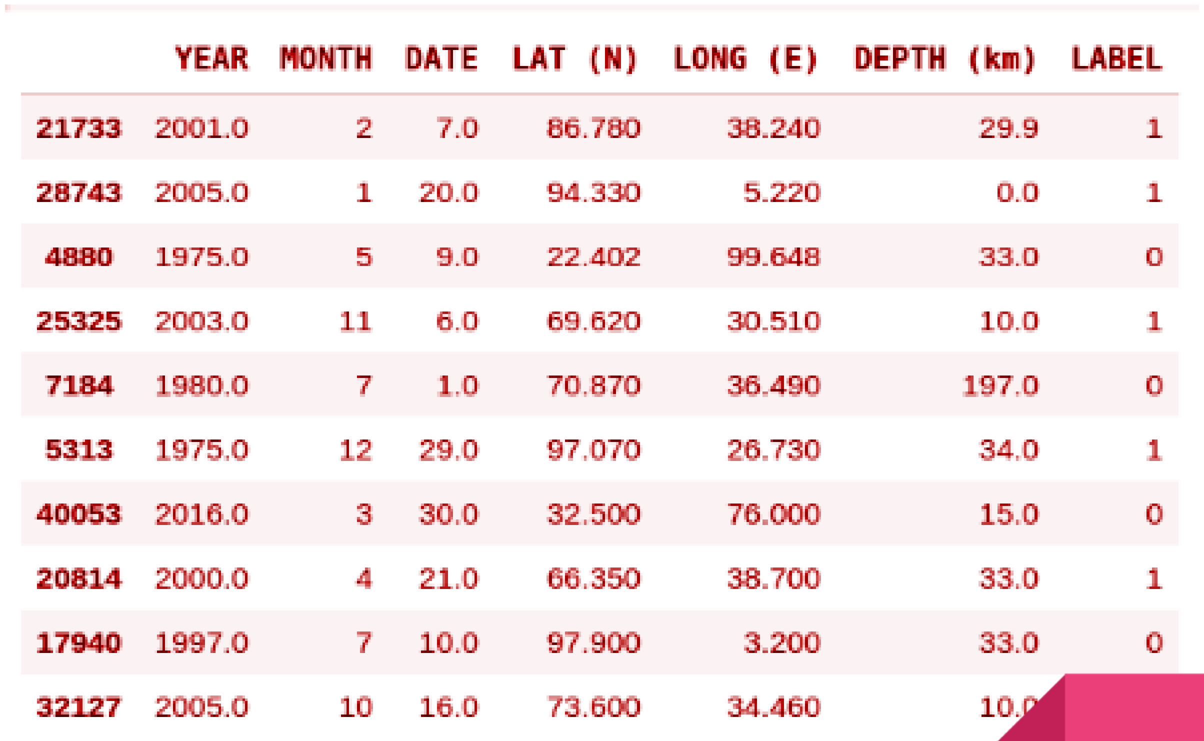
Data Cleaning and Threshold Selection

-> Data Cleaning

- We first removed redundant columns like **Reference, Serial number** etc, as these attributes do not have a significant effect on the magnitude of the earthquake.
- Rows in which the value of **Magnitude** was missing have been removed, as these points cannot be used for classification.
- Fault values for Latitude and Longitude were removed that didn't belong to their defined range that is (-90,90) and (-180,180) respectively.

- Fields of **Longitude** and **Latitude** contained special characters and extra spaces, which made processing difficult, so the special characters were removed.
- For Nan/empty strings in all other columns, we put in default values to maintain consistency.
- We also insured that data was logically consistent like Year >= 0 etc.
- The final attributes used are:
    - **Year:** Float
    - **Month:** Float
    - **Date:** Float
    - **Magnitude:** Float
    - **Latitude:** Float
    - **Longitude:** Float
    - **Depth:** Float

Fews rows of the cleaned version (after adding label column) of dataset is given below:



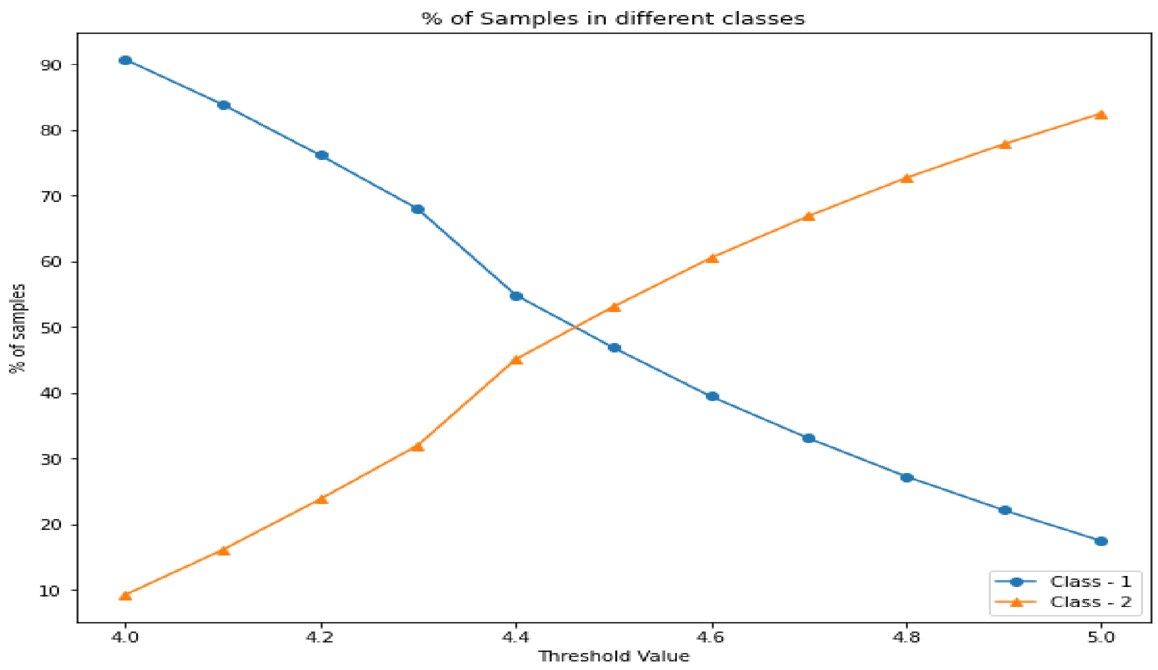
	YEAR	MONTH	DATE	LAT (N)	LONG (E)	DEPTH (km)	LABEL
21733	2001.0	2	7.0	86.780	38.240	29.9	1
28743	2005.0	1	20.0	94.330	5.220	0.0	1
4880	1975.0	5	9.0	22.402	99.648	33.0	0
25325	2003.0	11	6.0	69.620	30.510	10.0	1
7184	1980.0	7	1.0	70.870	36.490	197.0	0
5313	1975.0	12	29.0	97.070	26.730	34.0	1
40053	2016.0	3	30.0	32.500	76.000	15.0	0
20814	2000.0	4	21.0	66.350	38.700	33.0	1
17940	1997.0	7	10.0	97.900	3.200	33.0	0
32127	2005.0	10	16.0	73.600	34.460	10.0	

-> Threshold Selection :

- We need to decide a threshold so that the data can labelled for binary classification.
- We tried values between [4,5] and observed the number of samples belonging to class **1** and **0**, which is depicted in the table below.
- Total Samples are - 40935

	Threshold	% in Class 1	% in Class 0
1	4.0	90.74	9.25
2	4.1	83.89	16.11
3	4.2	76.16	23.83
4	4.3	67.98	32.01
5	4.4	54.9	45.1
6	4.5	46.9	53.09
7	4.6	39.46	60.5
8	4.7	33.09	66.9
9	4.8	27.28	72.71
10	4.9	22.16	77.84
11	5.0	17.46	82.53

- Plot of the above values



- We observe from the above plot that the best value for the threshold should be 4.4 - 4.5, as in this range the bias of the chosen split of the data will be minimum possible (as both the classes have equal number of samples)
- If we choose the threshold closer to 4.0 or 5.0, the number of samples of a particular class will increase and data will become biased towards that class.
- For all the future analysis we use threshold (**T**) as **4.5**.
- We have used a training/test split of **80%-20%.**

Classification Models:

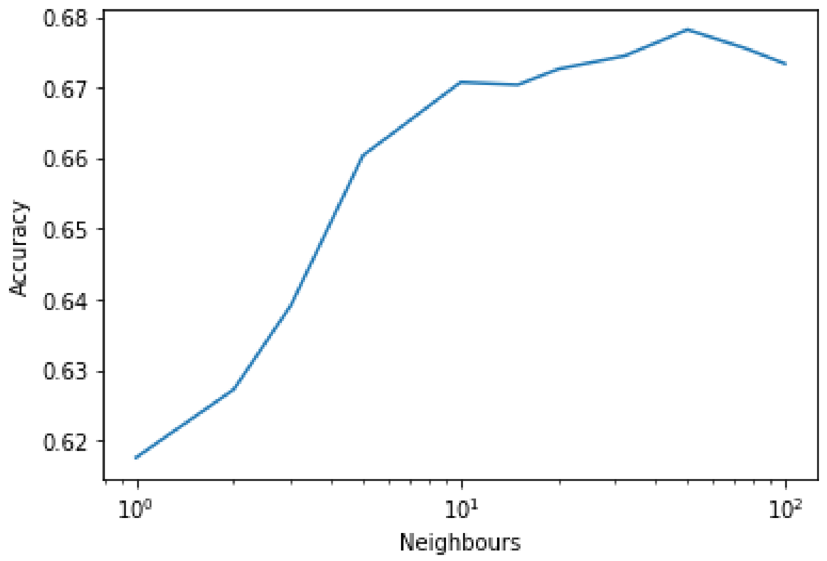
-> K-Nearest Neighbour:

- We applied the K-Nearest Neighbour model to the clean dataset with 80% of the data used for training and 20% used for validating.
- Given below is the result for different values of K. We have used **Accuracy, f1 score, Recall and Precision** to evaluate our model.
- Training Samples = 32748, Validation Samples = 8187

Value of K	Accuracy	F1-Score	Recall	Precision
1	0.650	0.686	0.688	0.683
2	0.636	0.609	0.763	0.507
3	0.662	0.697	0.698	0.696
5	0.668	0.711	0.713	0.709
10	0.673	0.700	0.744	0.660
15	0.671	**0.720**	0.728	**0.712**
20	0.674	0.715	0.746	0.687
32	0.675	0.716	0.745	0.689
50	0.675	0.716	0.744	0.689
75	**0.677**	0.719	0.746	0.696
100	0.674	0.715	0.746	0.686
240	0.661	0.707	**0.748**	0.671

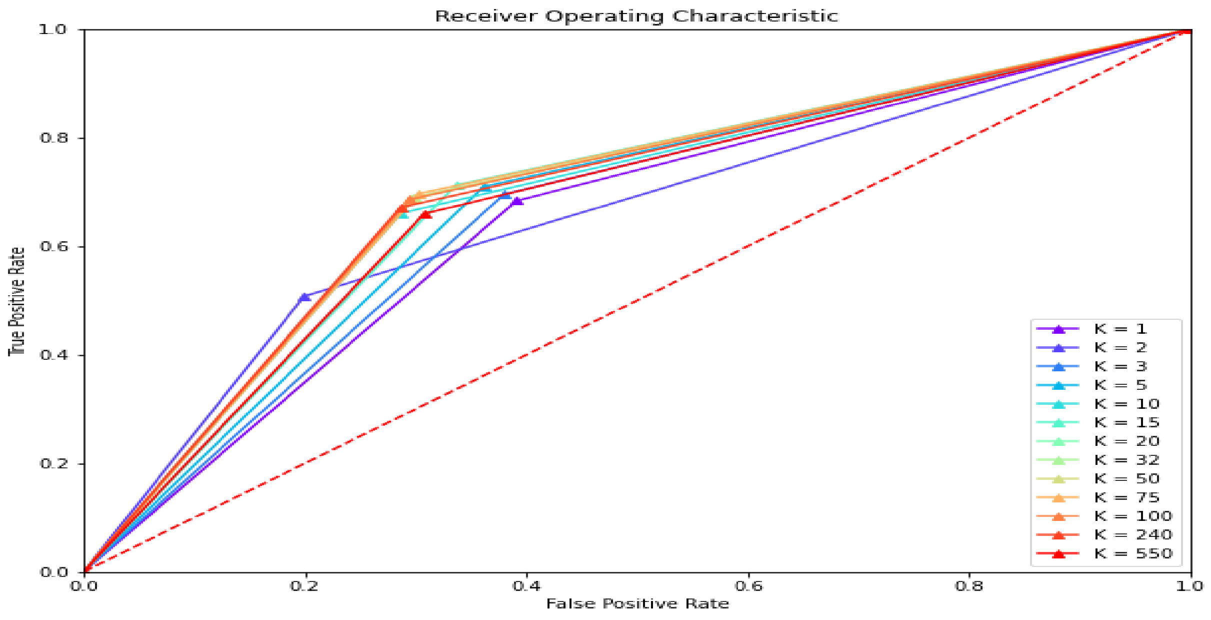
500	0.654	0.693	0.730	0.659

- Plot of K versus accuracy is given below:



- As we can observe from the above plot and table, we get a maximum accuracy of **0.67** for K = **75**.

- ROC plot for KNN Classifier for different values of K is given below:

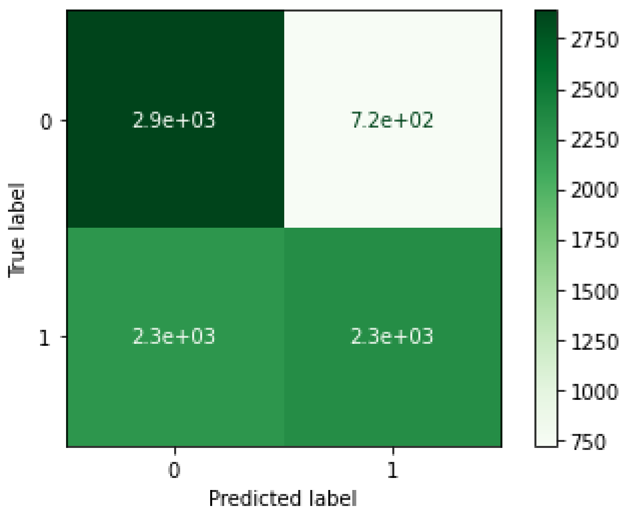


Observations from the above ROC:

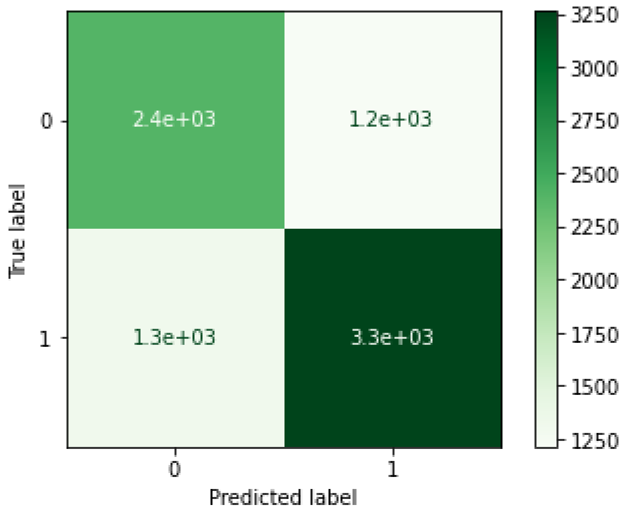
- We observe almost similar classification results for in range (10-100) and that seems the peak for the data, and we get almost the same ROC plots for this range. Below this range we get low accuracies because maybe we don't consider all the samples in the neighbourhood and above this range accuracy decreases because we might consider samples too far that may not influence our test sample.

Confusion Matrix for different values of K:

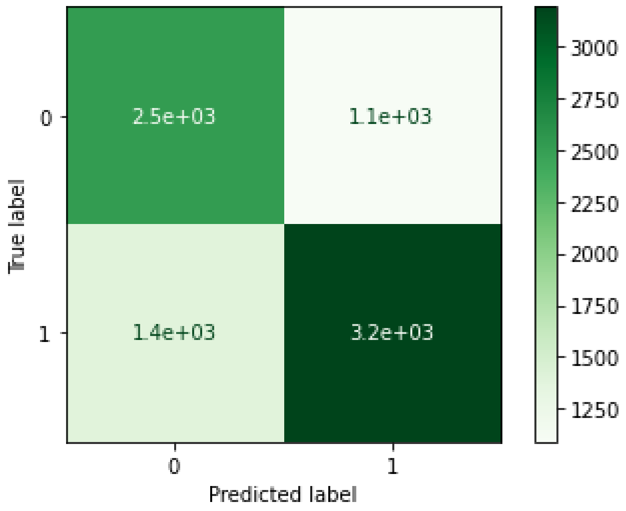
- K = 2


2.9e+03
7.2e+02
2.3e+03
2.3e+03
True label
Predicted label
0
1
2750
2500
2250
2000
1750
1500
1250
1000
750

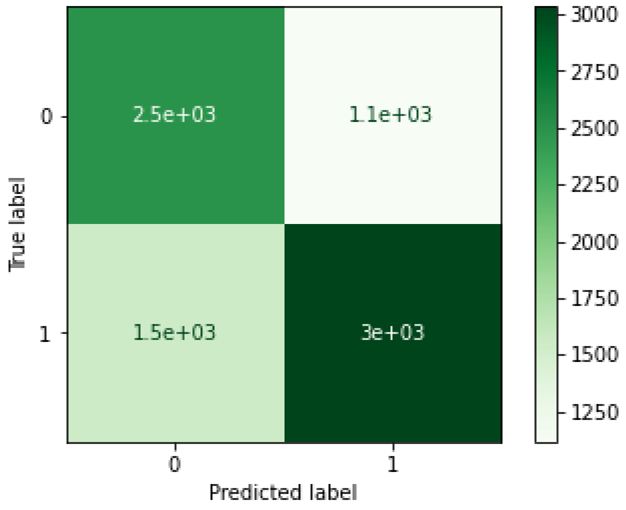
- K = 15


2.4e+03
1.2e+03
1.3e+03
3.3e+03
True label
Predicted label
0
1
3250
3000
2750
2500
2250
2000
1750
1500
1250

- K = 75



- K = 500



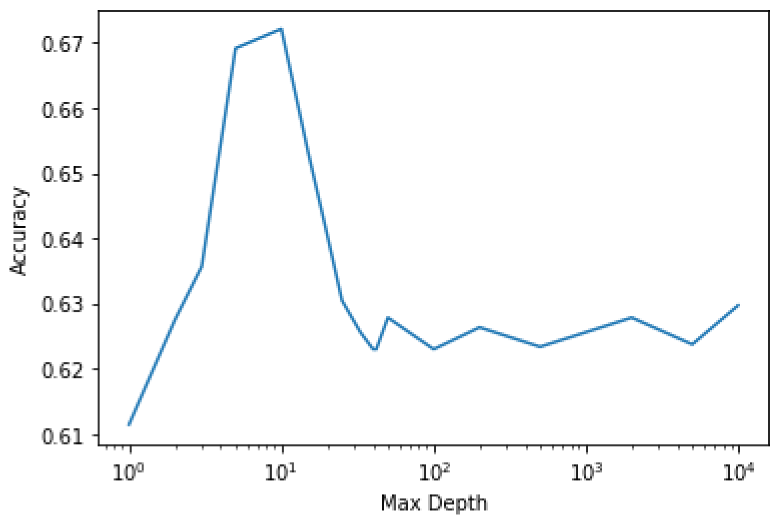
DecisionTree Classifier:

- We apply Decision Tree Classifier with different values of maximum depth to influence the level of pruning in the tree.
- Given below is the result for different values of K. We have used **Accuracy, f1 score, Recall and Precision** to evaluate our model.

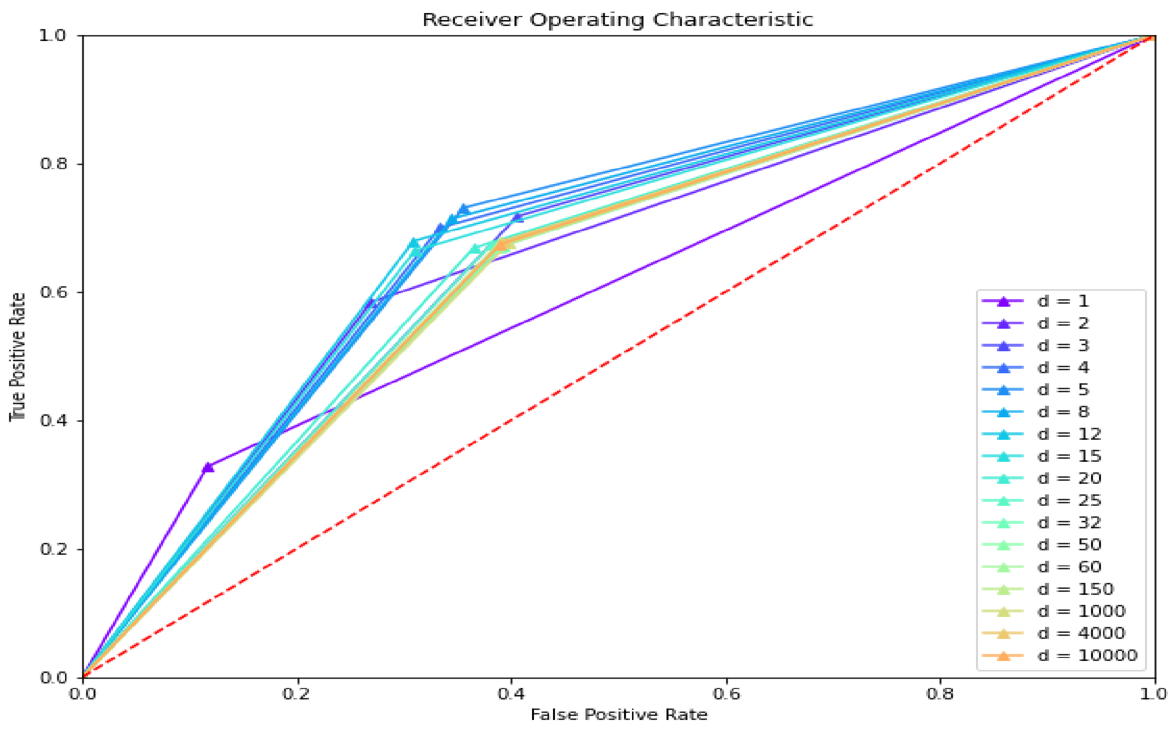
- Training Samples = 32748, Validation Samples = 8187

Value of Max Depth	Depth	Accuracy	F1-Score	Recall	Precision
1	1	0.572	0.461	0.780	0.328
2	2	0.648	0.650	0.733	0.583
3	3	0.663	0.704	0.691	0.717
4	4	0.665	0.713	**0.726**	0.699
5	5	0.669	**0.726**	0.722	**0.730**
8	8	**0.673**	0.718	0.724	0.713
12	12	0.664	0.706	0.735	0.678
15	15	0.665	0.695	0.730	0.664
20	20	0.653	0.683	0.698	0.668
25	25	0.649	0.681	0.691	0.672
32	32	0.643	0.678	0.684	0.671
50	43	0.643	0.677	0.686	0.668
60	43	0.647	0.682	0.686	0.677
150	41	0.647	0.682	0.687	0.676
1000	41	0.643	0.678	0.683	0.674
4000	43	0.647	0.682	0.686	0.678
10000	41	0.645	0.680	0.686	0.673

- Plot of Maximum Depth versus Accuracy is given below:



- ROC plot for different values of Max Depth [d] is given below:

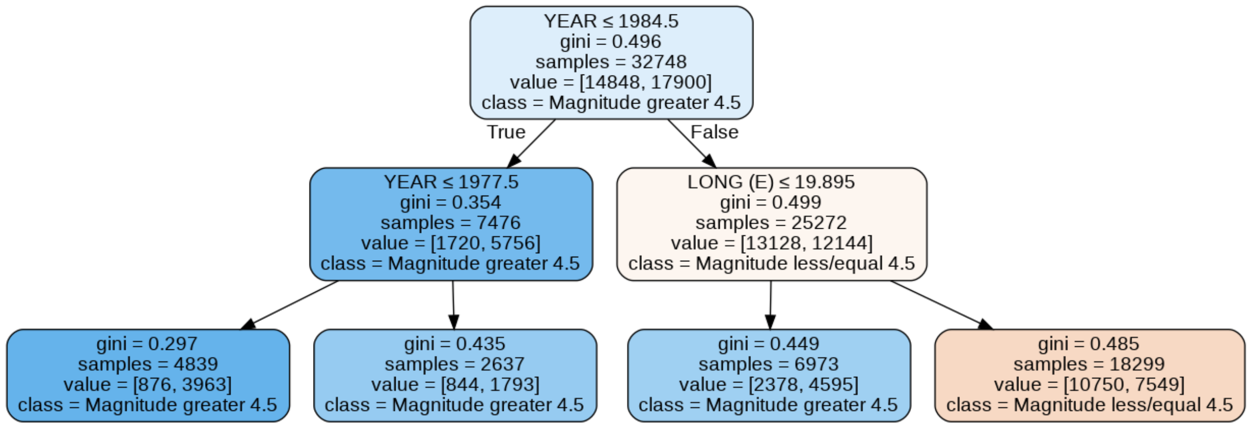


- From the above plots and table we see that the maximum accuracy is **0.673** for a max depth of **8.**

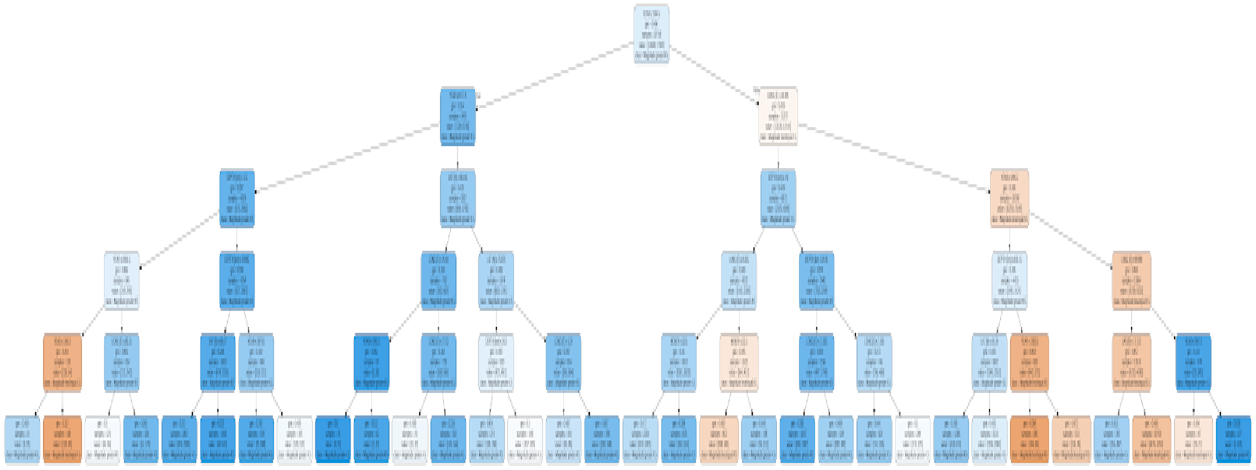
- This is the same as the maximum accuracy obtained in KNN.
- Also, we see that the graph the depth of the graph saturates at ~41/43 for larger values of max-depth. This is the reason that accuracy becomes almost constant at larger depth values.
- Also, we see that in greater depth values have relatively smaller values of accuracies. This is as the depth/complexity of the decision tree increases it overfits the training data, but gives relatively poor accuracy on validation data.
- A depth of range (4-12) might be the ideal as it does not make the decision tree too complex which makes it too much oriented on the training set and also not with too low depth that might fail to capture the essential features.

Given below are images of the decision trees generated for different values of maximum depth:

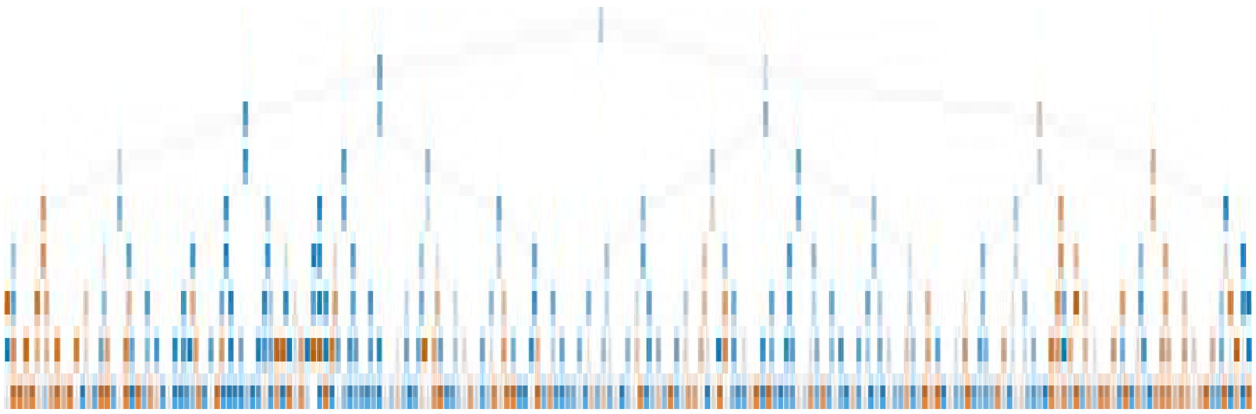
Depth - 2



Depth - 5

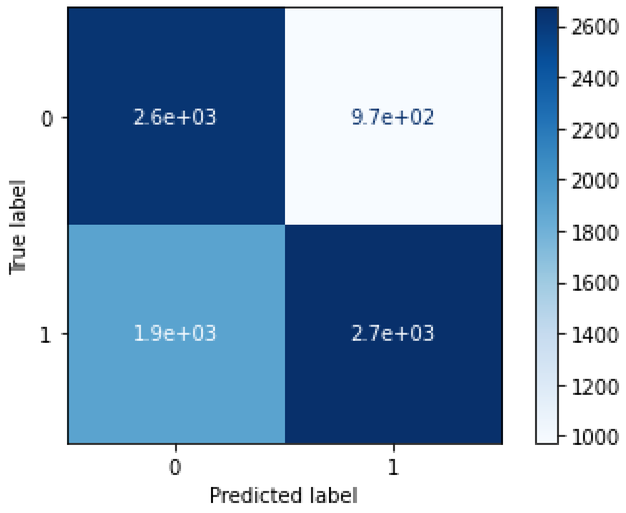


Depth - 8

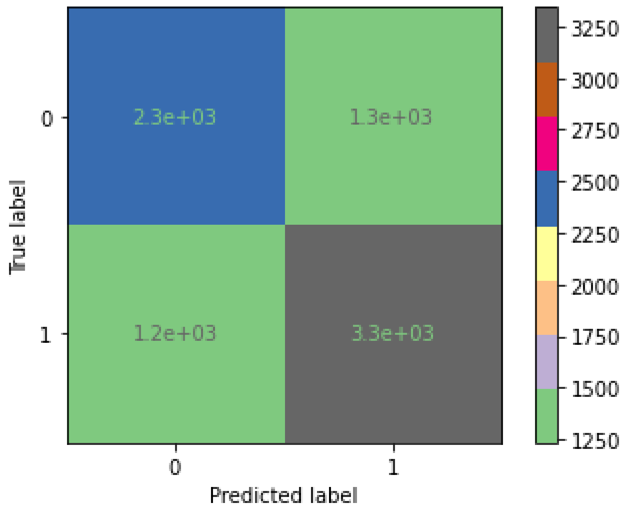


Confusion Matrices for different values of Maximum Depth:

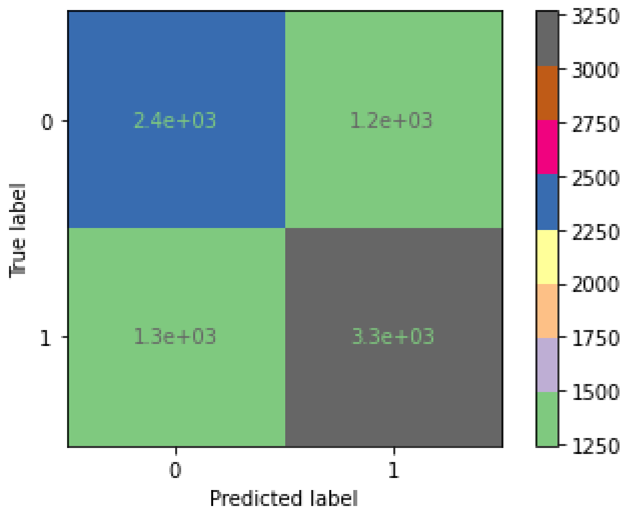
- D = 2


2.6e+03
9.7e+02
1.9e+03
2.7e+03
True label
Predicted label
0
1
2600
2400
2200
2000
1800
1600
1400
1200
1000

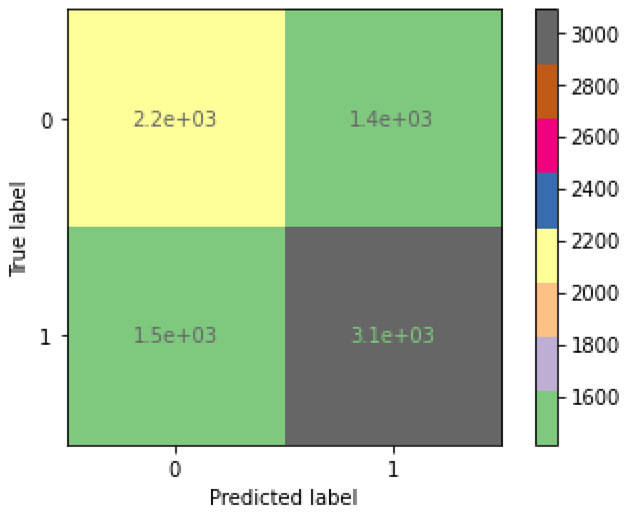
- D = 5


2.3e+03
1.3e+03
1.2e+03
3.3e+03
True label
Predicted label
0
1
3250
3000
2750
2500
2250
2000
1750
1500
1250

- D = 8



- D = 32



Comparison between the two classifiers

- We find that on the cleaned dataset for the earthquake data the maximum accuracy seems to be almost the same.
- If we consider accuracy as the judging criteria we might feel both to be almost the same as they give around 70% accuracies.

- But, in such a scenario the decision of which classifier is based upon our requirement. So, **if we wish to have low training time even over the cost of high testing time we prefer K-nearest neighbour classifier** and **if we prefer low testing time, we might prefer Decision Tree classifier.**
- This is on the basis of the basic models but certain improvisions over the features and the model parameters might change accuracies for these models and also may guide classifier decision.

Parameters for the Classifiers

We based our model only on a subset of parameters based upon our knowledge for the system. We used the date features as we feel that the geographical features of a location change over a period of time and therefore, play an important role in the prediction of the earthquake. Along with the date features, we used the latitude and longitude of the place which we feel is a very important parameter as it decides if a place lies in an earthquake-prone area where the chances of the earthquake will be high.

As discussed above the best parameters for the K-Nearest Neighbour is K = **75** and Maximum Depth for Decision Tree Classifier is **8** (with default min_sample_split parameter)**.**

Best Features (Subset of features)

If we had to use a subset of the features I feel we could drop Date, month fields as parameters as I feel they won't be that important as geographical features change over many years so the difference of years is a much more important parameter than the difference in dates and months of the samples. Going by domain knowledge and results of the feature importance generated by decision trees, **I feel Year, Lat, Long are the most important fields that should be used for classification of samples.**

Features	Importance
Year	0.45
Month	0.03
Date	0.02
Latitude	0.17
Longitude	0.18
Depth	0.15

From the above table for max-depth = 8(value giving max accuracy in decision tree) we can see that year, long, lat are the most important features that influence the decision tree classifier as we expected from our intuition.

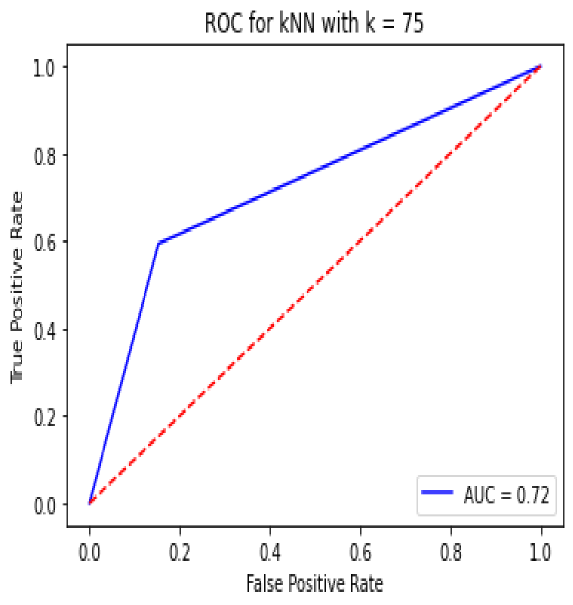
Improvements | Improvisions

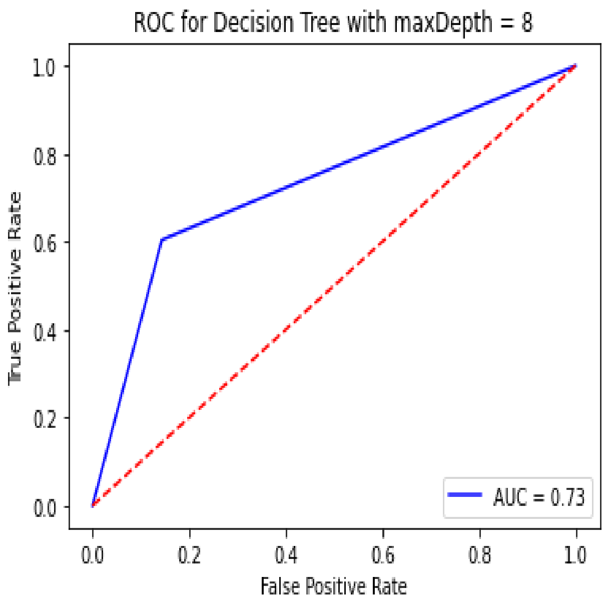
- We were able to improvise over some parameters to get an accuracy of around 73 percent for the best parameters of both the classifiers.

- One major issue we witnessed was that because the range of year varied over a large set of values compared to the range of other parameters, i.e., 180 for latitude and 360 for longitude, the importance the classifier gave to the year field was very high and almost decided the label. Therefore, we normalised the results to a comparable range (500) which increased the accuracy by around 2 percent. This was inferred from the Feature importance table of the decision tree where around 50 percent importance was initially given to ‘YEAR ‘ which was later much uniformly balanced to around 25 percent. Given below is how the updated feature importance table looked for max-depth = 8 :-

Features	**Importance**
Year	0.25
Month	0.06
Date	0.04
Latitude	0.24
Longitude	0.25
Depth	0.16

- **Weighted voting instead of Majority voting in kNN** gave better results which also seemed logical as sample nearer to the target sample should have higher importance.
- **Using min_samples_split parameter to around 100** from the default 2 helped improve the performance which avoided splitting similar nodes in some specific branch of the Decision Tree and also tackling over-fitting in few parts of the dataset.
- Improvised ROCs with better accuracies (72% for k-NN and 73% for Decision Tree) and the best parameters in the previous models are shown :-


ROC for kNN with k = 75
True Positive Rate
False Positive Rate
AUC = 0.72


ROC for Decision Tree with maxDepth = 8
True Positive Rate
False Positive Rate
AUC = 0.73