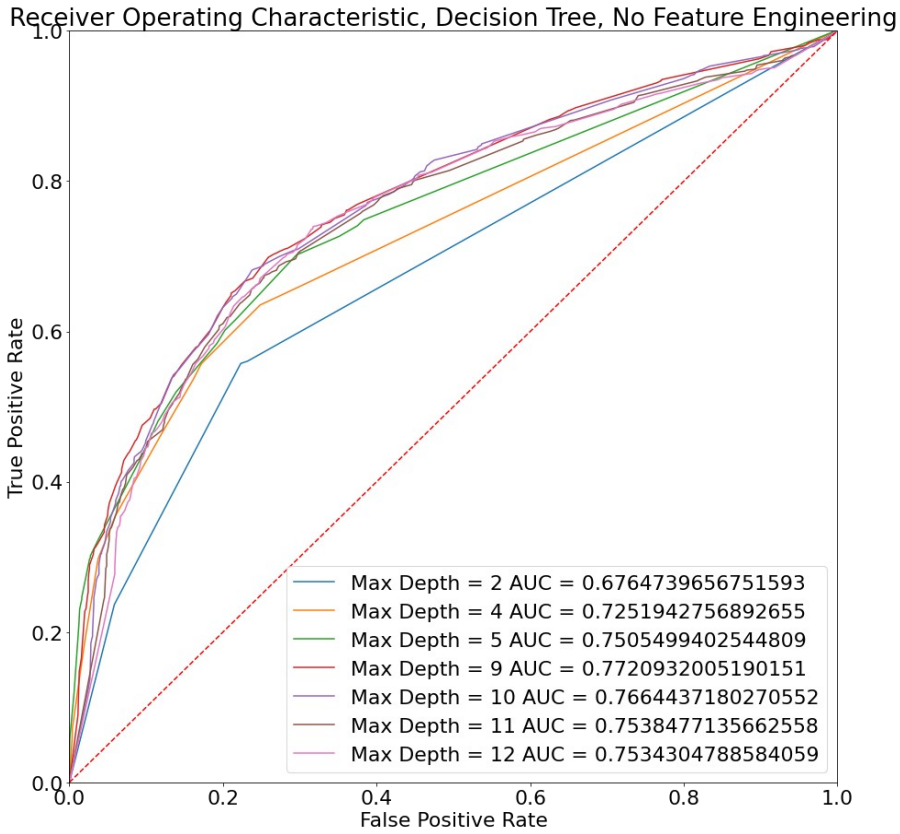
Classification Project

Earthquake Data Analysis

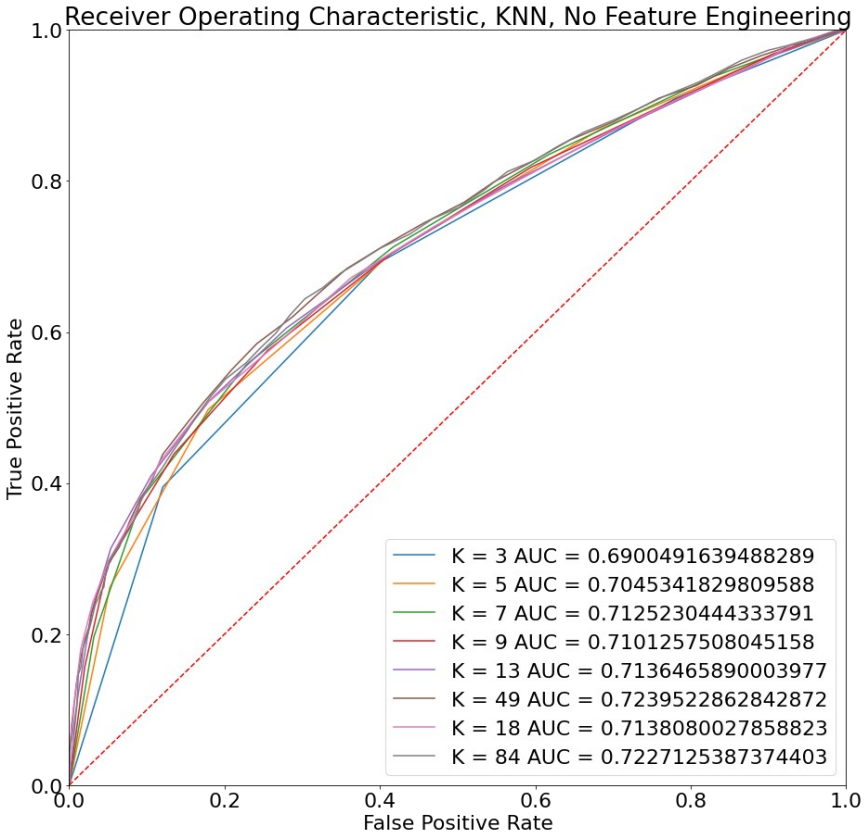
Team - 33
Varun Chhangani - 2019121011
Shivaan Sehgal - 2018111026

Problem Statement: Given various attributes for different earthquakes, predict whether an earthquake with some given attributes will be a major earthquake or minor earthquake (or as given in the document, “earthquake” or “no earthquake”). PS: The attributes used for classification are given later on.



About Classifiers and the Results, without feature eng:

The threshold was chosen to be 4.4 as this is the median and thus near equally divides the data. The train test split was chosen to be 0.8-0.2.



Without any feature engineering, we got 71.42% accuracy and an F1 score of 0.72 on decision tree. While we got an accuracy of 66.36% and F1 score of 0.6959 for the kNN classifier. This was also without scaling or normalization.

We can observe a difference of 5.06% between the two classifiers with the Decision Tree outperforming the kNN. This may be attributed to the intelligent branching of Decision Tree according to gini index or entropy of different variables, while kNN just takes into account the distance between the feature vectors, thus penalising the attributes where the range is too small or too large.

The best parameter in both the classifiers would be considered the one with more area under the ROC curve and F1 score. This is because the Area under ROC curve corresponds to accuracy, and F1 score can be calculated directly as the Y axis is directly proportional to the sensitivity and the X axis is directly proportional to 1-Specificity. We would prefer to maximize both of them, but for sake of ease, we can maximize one at a time in this case.

For choosing any 2 of the given/engineered features, it would be better to choose Latitude and Longitude as there are some active hotspots of earthquakes near the

fault lines of the tectonic plates. This is also apparent from the Decision Tree formed.

For the case wherein no feature processing was used, we used the features:

- Year, Month, Date
- Origin Time
- Latitude
- Longitude
- Depth
- And, Reference

This gave us an accuracy of 71.42% for the Decision Tree, and 66.36% for kNN. (as reported above, with F1 score)

Feature Processing:

The new features we have **added** are based on the fact that earthquakes don't occur at the exact same location each time rather they happen frequently in some prone areas (geographical fault lines) . Hence are using

1. frequency of quakes (small or large), in microseconds
2. Time since last quake (small or large)
3. Time since major earthquake
4. Total quakes
5. Total major quakes
6. Ratio of time since a quake and frequency
7. Ratio of time since a major earthquake and frequency

of earthquake in the vicinity of a location where an earthquake had occurred **(+- 0.5 degree lat and long)**. This will help in better generalization of models.
This feature processing increased the accuracy and F1 score of the decision tree to 71.6% and 0.738 respectively.

However, as we discussed the problems in kNN, we went with following features only:

1. YEAR
2. MONTH

3. DATE
4. LAT (N)
5. LONG (E)
6. DEPTH (km)
7. REFERENCE
8. Total major quakes
9. Ratio of time since a major earthquake and frequency

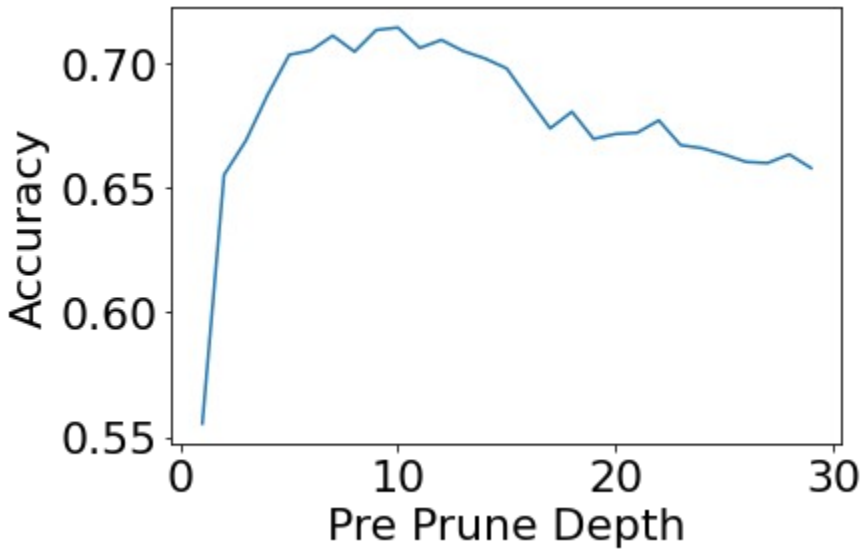
This increased the accuracy and F1 score of the kNN classifier to 67.88% and 0.716 respectively.

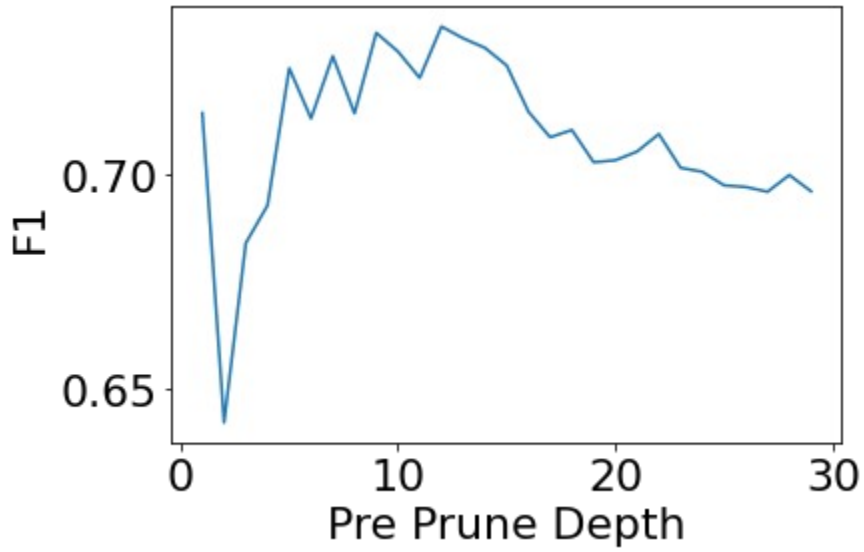
Data Cleaning:

- We have cleaned the data first by handling missing values(NaN) mainly by Imputation and in some cases dropping the row.
- For imputation, we used methods that would not affect the data in any drastic manner, eg: filling 00:00 Hrs to NaN time, ‘’ for Reference, etc.
- Removed invalid entries like, negative YYYY/MM/DD.
- String to Integers for Year,Month and Days . And String to float for Magnitude.
- Time to Time-object (YYYY-MM-DD HH:MM:SS.MS) so that we can easily take difference b/w occurrences of earthquakes and finally to ints.
- Dropping columns which provide redundant data (data already available in other columns). Like for Magnitude we had many columns which give measure of it Mw, Mb, Mb, Ms, ML.
- Formating Latitude and Longitude from XX**°N/S/E/W** to float using regular expressions

Analysis:

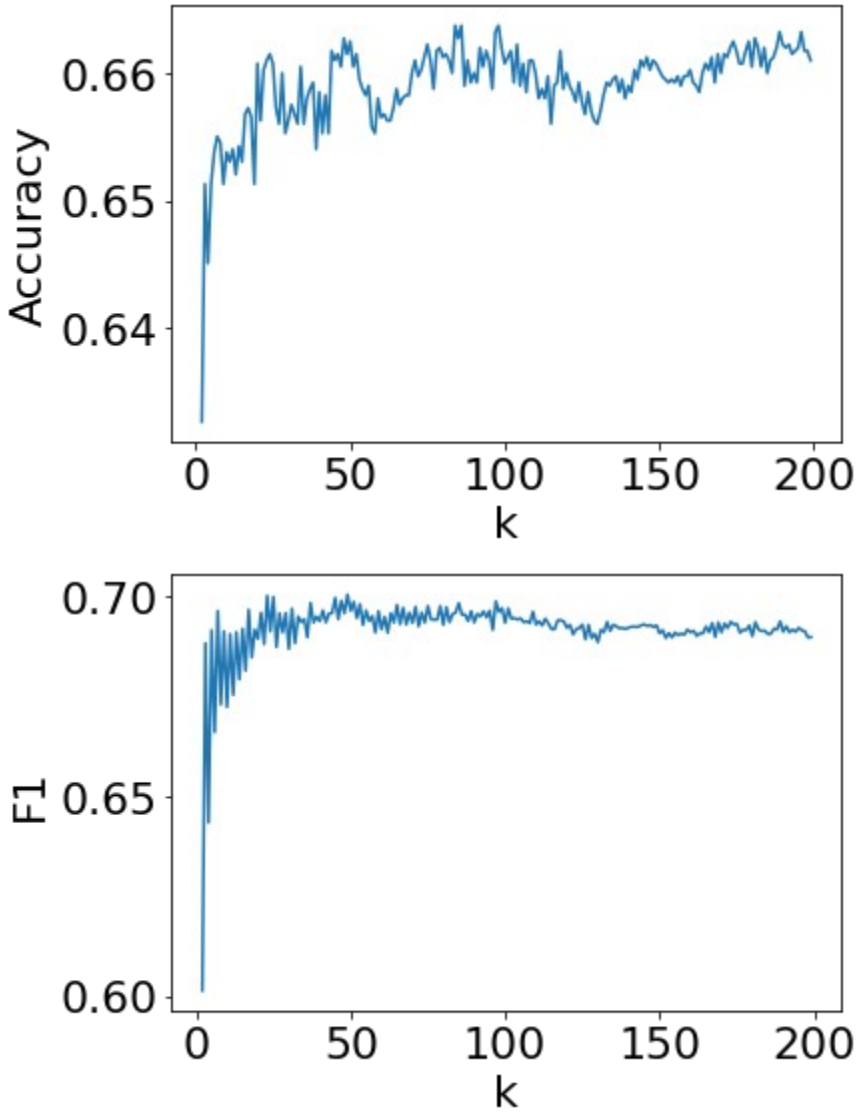
Decision Tree, without Feature engineering:



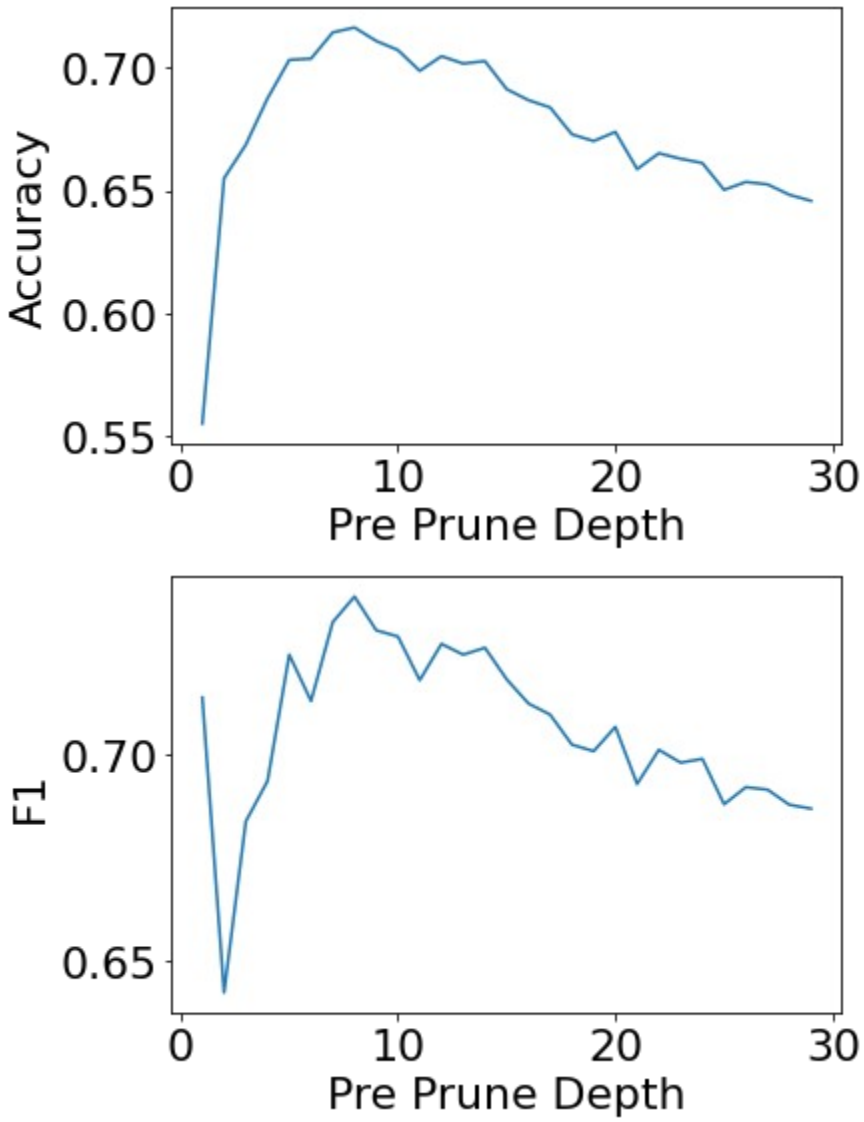


We can observe the bias variance trade off.

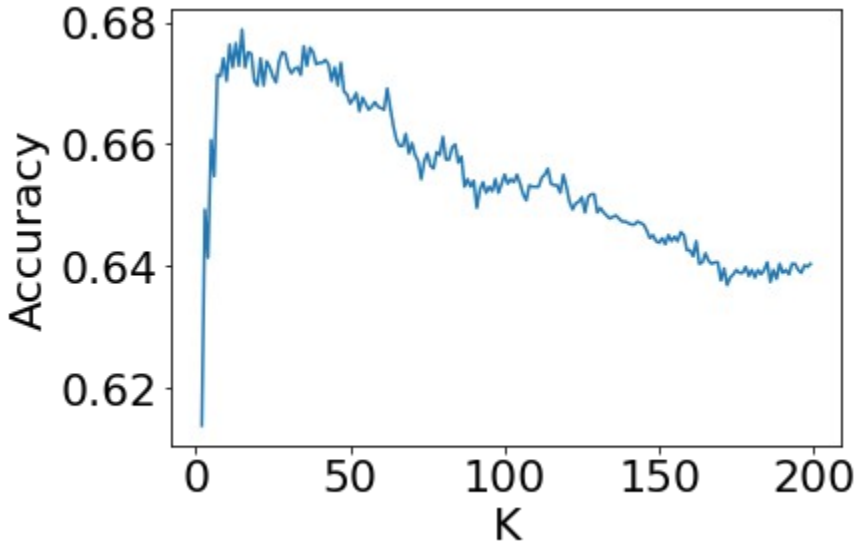
kNN, without feature engineering:

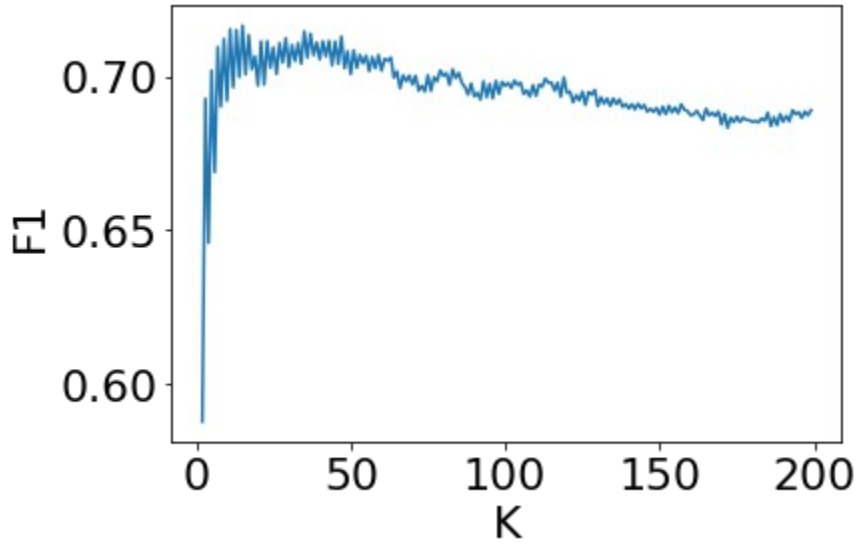


Decision Tree, After Feature Processing:



kNN, After feature processing:





More pronounced bias-variance curve after feature engineering, **may point to better direction**.