Classification whether it's a criminal case

Identifying the classification problem

After skimming through each file `acts_section`, cases, keys, and judges, I soon saw that there was a binary classification possible in terms of whether it's a criminal case or not through `act_section`.csv and cases as they have the same `ddl_case_id`.

Problem

The major issue I had to face for the whole task was the enormous amount of data I had to go through. This was a burden to my laptop and I wasn't able to run commands as I usually do. So, the first task I did was to go through the columns and remove irrelevant ones by scanning them manually. This was to reduce the amount of time and to be able to read all the datasets by my laptop. So, by this, I removed columns like dates and the codes court and state etc. which would have no bearing on whether its a criminal case.

Preprocessing - Cleaning

Cases

Since, I don't know law, I couldn't just sift through and determine each attribute as being pertinent but I was at least able to reduce the columns by quite a few. Then, I loaded each year individually. I first removed all the rows that carried unclear names or values (NaN values, -9999, missing name, etc.) as we had enough dataset that even if we lost data like this, we can still build a strong classification algorithm.

Then, I outputed this individual year file to a new file, naming it cleaned csv.

I did this for the past 5 years (2014 - 2018). Using dask, I created parts of each of the file for faster parallel processing

Acts Section

For this, I wasn't able to load the dataset on my computer, and uploading the 3gb dataset would have been too time consuming and strainous. Furthermore, I couldn't merge it with the cases to get the merged shorter version on my laptop.

So, I took the columns which were useful, `ddl_case_id`, acts, and criminal attributes.

Merging Cases and Acts

With dask, I was able to easily read the multiple parts of files into a dataframe. Plus the acts cleaned was also read easily. Furthermore, merging was able to efficiently work on this dataframe based on the attributes

After getting the merged file, I was able to run various measures such as the chi2 test. I selected the top 7 attributes from them.

The acts and type names were the most important attributes

After removing the least important ones from chi2 and the ddl case id, I was able to get the final dataset on which I can run my classification on.

This created a file in Mbs which I could work with. I was able to upload this file on google collab.

I then ran various classifiers - Random Forest, Decision Trees, KNN, and SVM

I was unable to finish the classification of SVM cause of time constraints but I found out that it doesn't work well with big datasets. The best result I got was with Decision Trees algorithm with the classification accuracy at 1.

Conclusion

All classifiers were giving good accuracies. $> 99\%$ Decision Trees was the best classifier Acts and the type names were the best attributes which influenced the classification accuracy

I faced many problems with resource constraints but I optimised and cleaned the data thoroughly to reduce the data to be able to be run easily.