Hackathon IML

Did you describe the dataset, and any challenging characteristics it has?

Dataset and challenges:

As we looked at the data, we noticed that the number of features is very small in compare to the number of samples which suppose to be good but we noticed as well that most of the features are very similar which is not good for learning algorithms.

So, we understood that the main challenge is to extract as much data as we can.

Did you describe (briefly) the data cleaning and preprocessing?

Pre-Processing:

First, we deleted irrelevant data such as 'ID' (which has no influence) and 'Year' (which is identical to all data). Then we began analyzing correlation between the features and we found out that 'District' and 'Beat' are correlated almost perfectly so we removed the 'District' column as well ('Beat' is a little bit more specific). 'X' and 'Y' indicate location as 'Longitude' and 'Latitude', so they are redundant as 'Location' which is a tuple of longitude and latitude- so we deleted them as well.

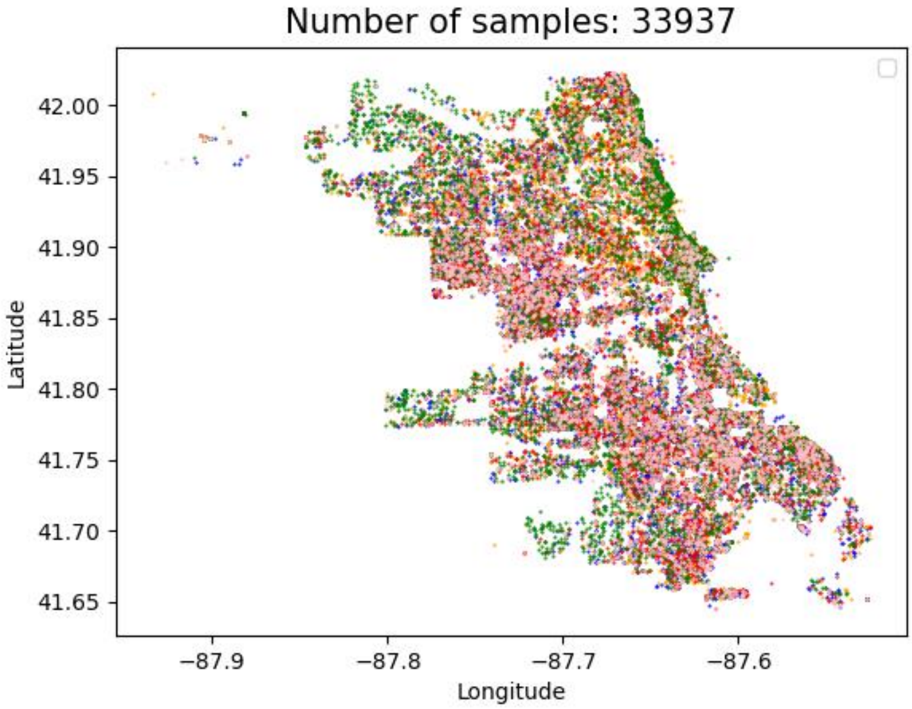
Secondly, we began creating dummies from features that represent categorical values ('Beat', 'Location Description', 'Ward', etc.)

Afterwards, we began extracting information from complicated features as 'Date'. We computed for every date which day of the week it is because we believe there is a difference in the frequency of crimes in weekends and rest of the week.

Finally, we extracted from date the time of day the crime occurred, and we split it by slots of 15 minutes so we can get a good time estimator for crimes.

Considerations that guided your design of learning systems

We created functions that plots the data by location to see some patterns of the data,



Each color represents a different type of crime.

To understand the behaviour, we looked online to read some articles and research about factors that influence urban crime.

We looked at all the classification algorithms we learned in class and then we considered which algorithms are more fitted to location and time analyze. Then we began running some simple algorithms to sense the data and to understand where we are standing.

For every algorithm we tried to figure what are the best parameters for this specific algorithm.

After that, we began using more complicated algorithms such as boosting, adaboosting and bagging.

- Did you describe (briefly) various methods you tried and the results you obtained?

We tried using KNN of various k's and we got bad results and bad time performance.

When we used LDA alone and we got best results, but we didn't succeed passing 0.5 accuracy, so we tried manipulating the data some more.

Moreover, we used k-folds algorithms to determine which algorithm is best fit and we got as we thought that LDA is the best algorithm.

- Did you describe the learning system you ended up using?

So, in the end we decided to stick to the LDA algorithm and try to maximize the prediction.

We noticed in the data that the only crime which has not "Location Description" is deceptive practice, so we added this to our predict algorithm.

- Did you provide a prediction (and explanation) of the generalization error you expect your system to have?

A guessing algorithm would get a success rate of 0.2 or a bit more if you choose always the most common crime. Our algorithm achieves a success rate of 0.48 which is much better and gives us a nice estimator.

In addition, we developed for the second problem an algorithm which finds the highest concentration points in the 3-D surface of time and location and knows to maximize the surface they cover by verifying that some points do not collide.

After that we tried to increase accuracy by dividing it to days of the week.

In the end we got for every day of the week 30 points of location and time for every police car which gets maximum number of crimes covered.