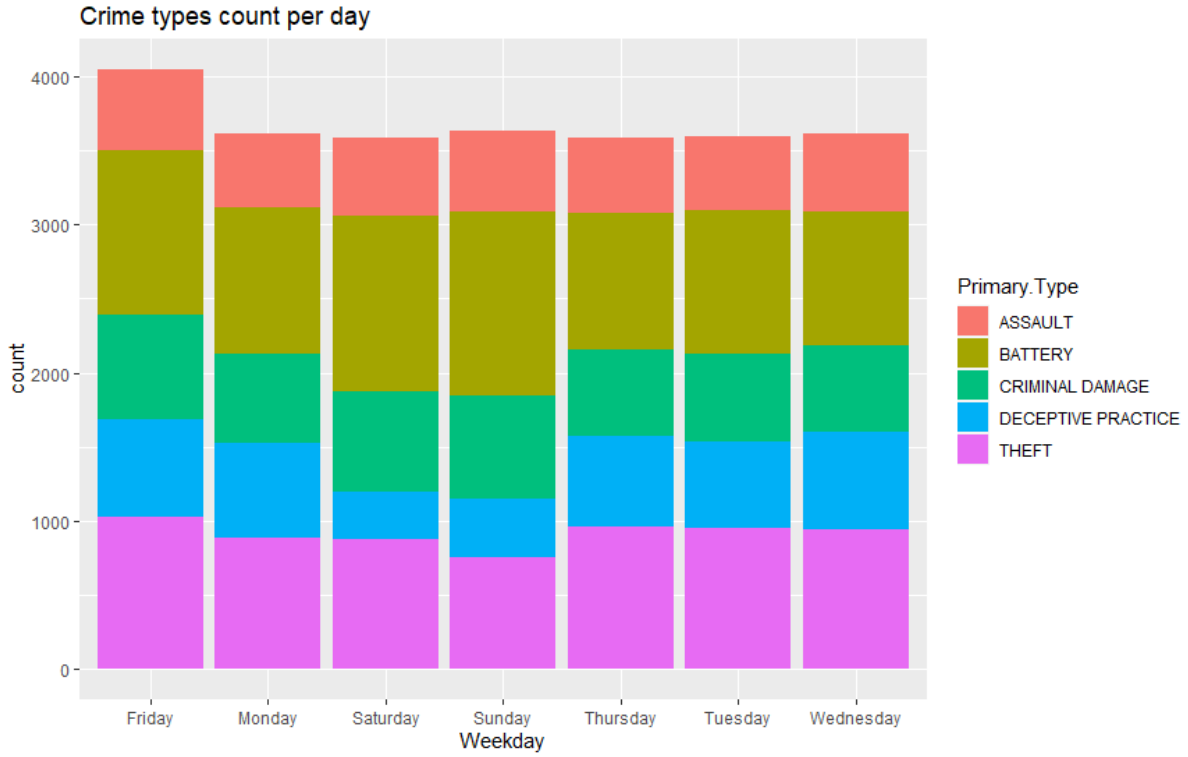
IML Hackathon 2021 - Chicago Police

Ron Moran, Shahar Guini, Dor, Tamir Golan

The data we have received Includes ~42000 samples, each sample represents a report of a crime that took place in Chicago in 2021. The reports include ~20 features, the main of them are the date and time the crime occurred, the exact datum-point, and the primary type of the crime (out of 5 options).

Dataset Manual analysis:

First we tried plotting the data, looking for some patterns between features. Although location did have an impact, we decided to leave it unchanged since the model can partition the space. It seems that other data is quite evenly spread - Crime count doesn't vary over different days (it does over different times) and crime types only seem to change over different days and hours:



For that reason we used the time as round hour and weekday as categorical variables.

Preprocess:

We have loaded the csv file into pandas data frame, and used only the features: Date, Arrest, Domestic, X Coordinate, Y Coordinate, Updated On, Beat, Location Description. The omitted features include repetitive data, data we were not allowed to use, or irrelevant data to our tasks.

We dropped the Date and 'Updated On' after calculating the difference (in seconds) between the occurrence and the update time, excluding the weekday and hour (we found that the month-day has no additional effect).
To change the categorical geographical data (ward, beat, etc…) into meaningful data the frequency of each categorical field was assessed over crime types and saved in a lookaside table.
We used CatBoost for two reasons: It's particularly optimized for categorical data, and the output model is slim relative to random forest. Both models have similar performance.
When checking the score of the model trained on the joined train and validation sets against the test score the loss is 52.4%. This is known as the complement of the generalization error, therefore the generalization error is 47.6%. In order to calculate it the model trained over said data gives the score (literally with "model.score()") for the labeled test data.

Question 2

In this part we were asked to implement a program that given a certain date it returns the best location and time for a police car to be placed during that day.
We found out after analysing the data that there isn't much difference in crimes amount and frequency during different times of the month or week. Therefore, our system essentially emits the data given to it and return the best combinations of time and locations around the city to place a police car, these combinations were directly evaluated from the data the model trained on. The way it evaluates those combinations is by doing "reversed KNN" in a way. It analysis the each crime location and time that is in our training data, and checks how many crimes are "nearby" by calculating how many crimes occurred in range of 500 meters away and 30 mins before and after the crime we are examining. we then take the crime locations with the most crimes around them, and by that essentially picking the locations and times around the city that have the highest density of crimes. We then send the police cars to those locations.

After testing the performance on our test set we got a generalization error of around 98%, meaning that we got 2% of crimes with our police cars spread.
This is around 1.22 crimes prevented a day.