Chicago Crimes - Report

Authors

Submitted By: Linoy Palas (Id – 205845407), Omer Liberman (Id – 206336406)
Date – June 6th, 2019
Done as part of the course Introduction to Machine Learning (67555), HUJI.

Introduction

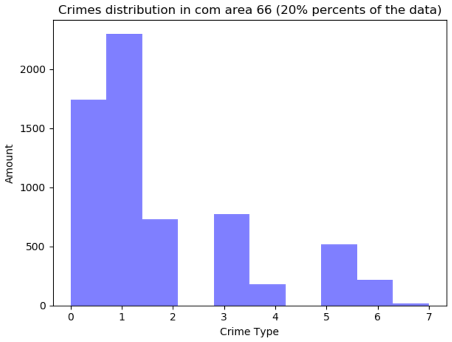
This report presents our analyzation over the data set "Chicago Crimes" which presents the crimes which happened in the city Chicago, Illinoi since 2005.
Our algorithm is trained over a data set of over million crimes cases.
The algorithm aims to classify future cases to one of the crime categories we have examined with as much accuracy as possible.

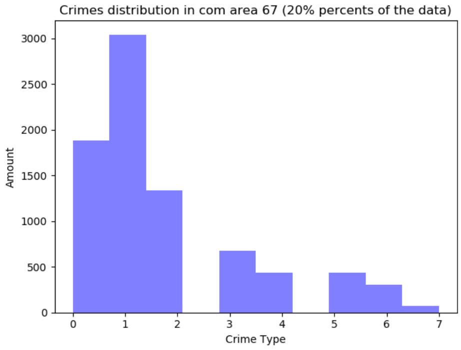
Algorithm

We decided to use the algorithms "K-nearest-neighbors" and "Decision Trees".
KNN was chosen because we found similarity in crime cases (presented below) in close areas in the city.

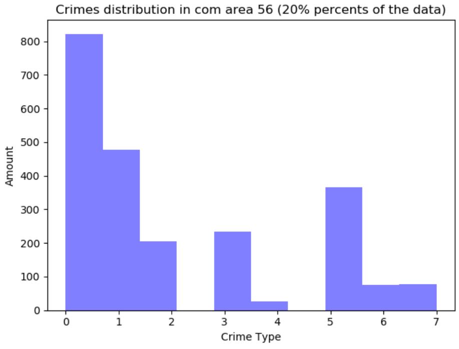
Pre-Training Research

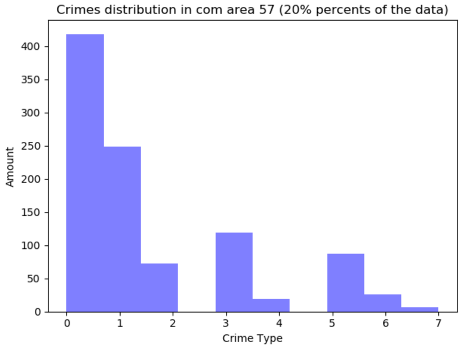
1) Similarity of crimes in close areas -
We have checked if closer communities (physically) suffer from similar types of crimes.
2 examples :
Communities 66 and 67 are very close and we can see :



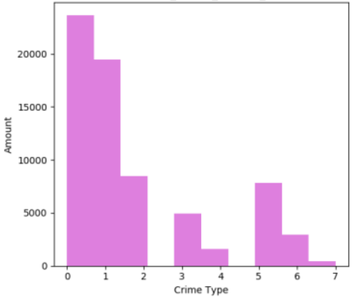


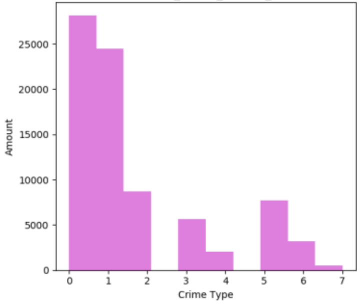
Easy to find the similarity between those two.
Another example are communities 56 and 57:

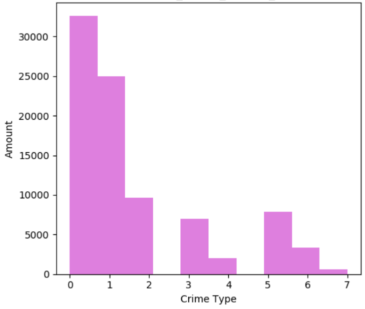


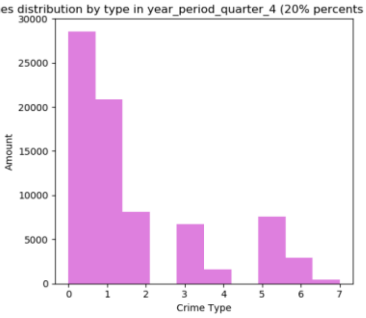


2) Distributions over different months in the year -
We has a suspicious that different months in the year has different criminal types.
We found it right.
During the first quarter of the year (Jan-Mar) there is a decrease in number of crimes ,
But in the third quarter (there are more crimes.

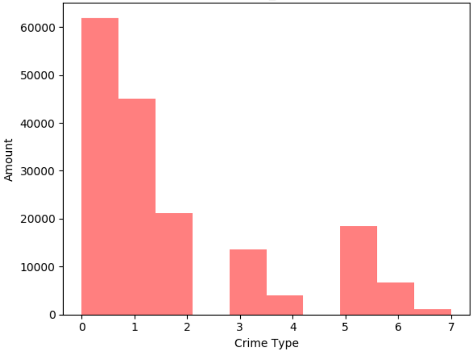

'imes distribution by type in year_period_quarter_1 (20% percents of the dat

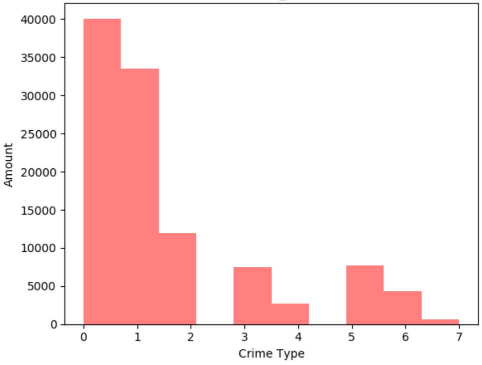

'imes distribution by type in year_period_quarter_2 (20% percents of the dat


'imes distribution by type in year_period_quarter_3 (20% percents of the dat

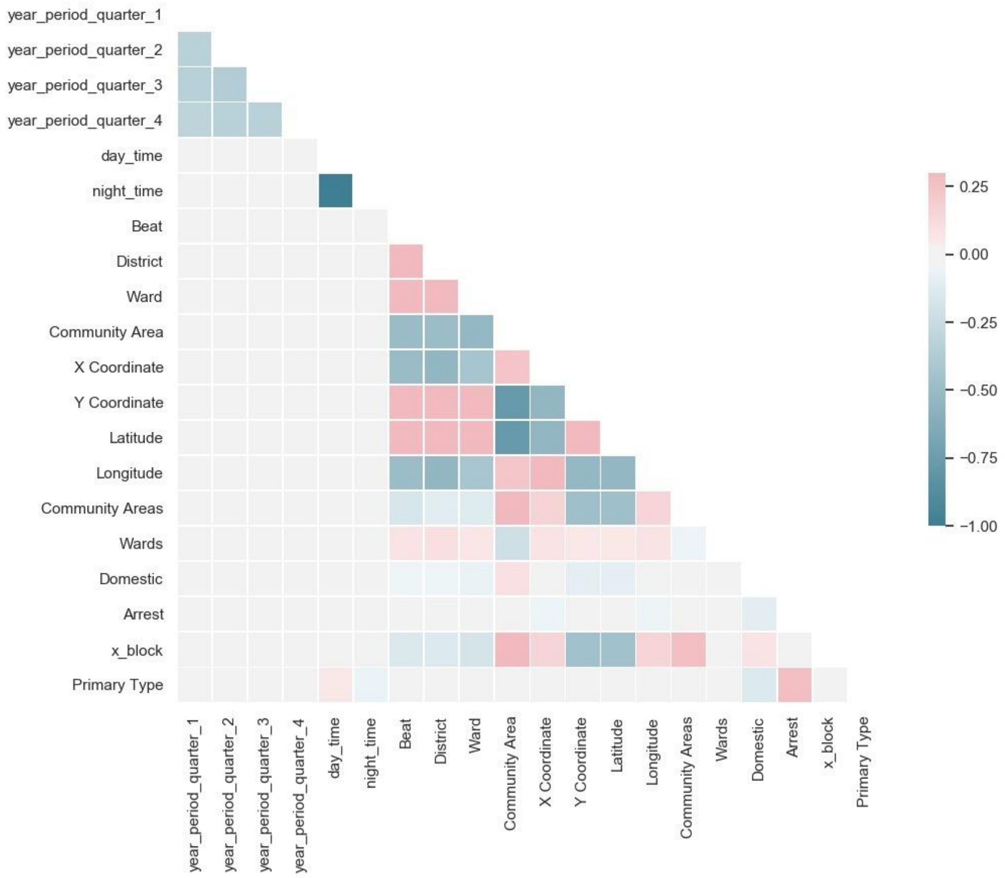

'imes distribution by type in year_period_quarter_4 (20% percents of the dat

3) Distributions over different hours of the day -
We has a suspicious that different times in the day have different criminal types.
We found it wrong! (Even surprising)
We expected to find growing crimes number at nights but find the contrary.


Crimes distribution by type in day_time (20% percents of the data)


Crimes distribution by type in night_time (20% percents of the data)

4) Correlation between features -



Training Process

We split the given data set to two parts: train (80%) and validation (20%).
We trained the algorithm over different sizes of the data.

How is the problem going to be solved?

Different algorithms will be used to come up with a good result in this contest. Each of them will be explained, tried and tested, and finally we will get to see which of them works best for this case.
Cross-validation will be used to validate the models, so the database has to be split into test, train and validation subsets. This split has to be stratified to ensure that the initial proportion of elements (same amount of crimes per category) is maintained in each subdivision. The resulting train dataset is still too large and running the testing programs would take too long.

Once the data has been treated, the following algorithms will be tried :

-K-Nearest Neighbors
-Decision Tree

Each of them will be explained in next page.

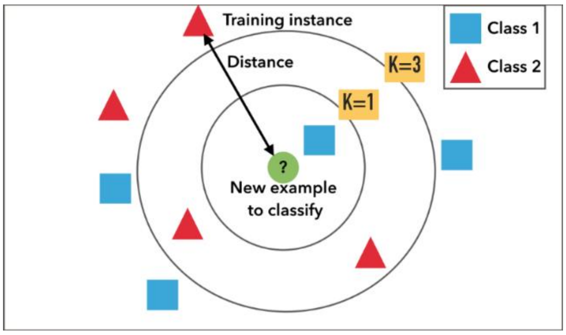
Machine Learning Algorithms Used:

KNN

The algorithm:

The k-nearest neighbor classification rule is arguably the simplest and most intuitively appealing nonparametric classifier. It assigns a sample z to class X if the majority of the k values in the training dataset that are near z are from X, otherwise it assigns it to class Y. This algorithm compares the given unclassified sample z with "all" the values in the training set and gets its k nearest values. Among them, it applies the previously described rule.

Example for better understanding:



Implementation

Although it's a pretty easy algorithm to implement , making it efficient and fast it's not trivial. For this reason, Python scikit-learn library, which has a broad set of machine learning utilities, has been used. This library has a Nearest Neighbors package with all necessary functions. From this package, Neares-tNeighbors class has been used (concretely, fit and k-neighbors methods).

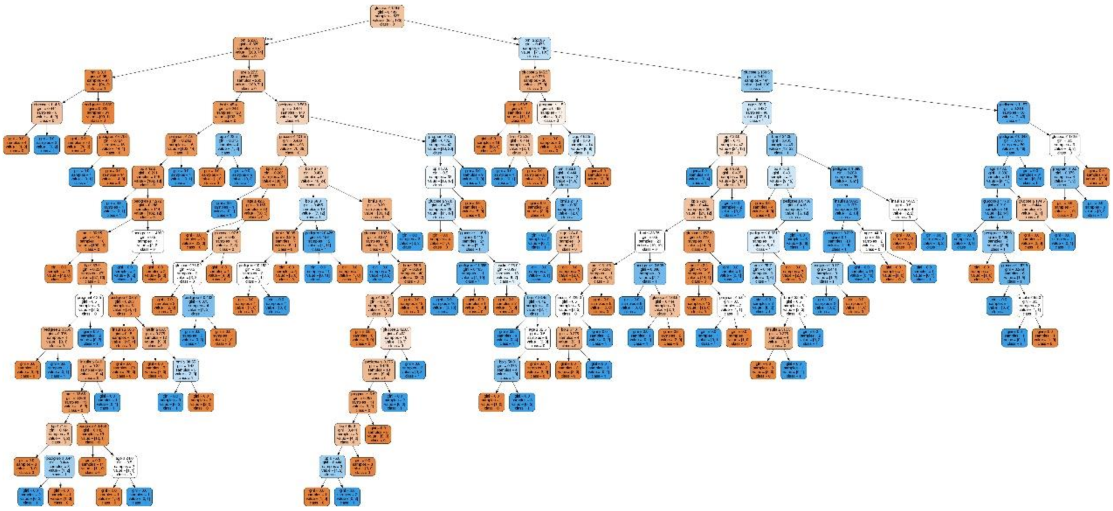
Resulte:
We explored KNN with different k's (of course, one would expect that, as the value of k is increased, results would get better. That is true, however, the computational cost of increasing this value would not be worth the output gain.)
several tests using different features were performed with this algorithm. This was made in order to find the best combination and use it afterwards with all the tests. This section is a walkthrough over the results in each feature combination try, giving the loss output of all of them.
Unfortunately we found only cons for this algorithm. In order to find the right k we explored many different k's and it costs LONG time and error rates were awful (50%-70% error).

Decision Tree

The algorithm:

Decision tree learning uses a decision tree as a predictive model which maps observations about an item (represented in the branches) to conclusions about the item's target value(represented in the leaves). It is one of the predictive modeling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a finite set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels.

Example for better understanding:



Implementation

Impanation of a decision tree can be a little tricky and inefficient, and so, like KNN, we used Python's built in scikit-learn library which has a broad set of machine learning utilities, has been used. This library has a Decision Tree package with all necessary functions

Result:

We also explored this algorithm. We found this algorithm better the previous and the lowest error we got was 40%.

Final Conclusions

Machine Learning is a really powerful field when it comes to AI and, if a model is done right, the level of accuracy that some algorithms can achieve can be astonishing. Certainly, the present and future of intelligent systems goes through ML and big data analysis.

With the crime classification problem, it has been seen that the most accurate algorithm was the Decision Tree.