Tobias Weis

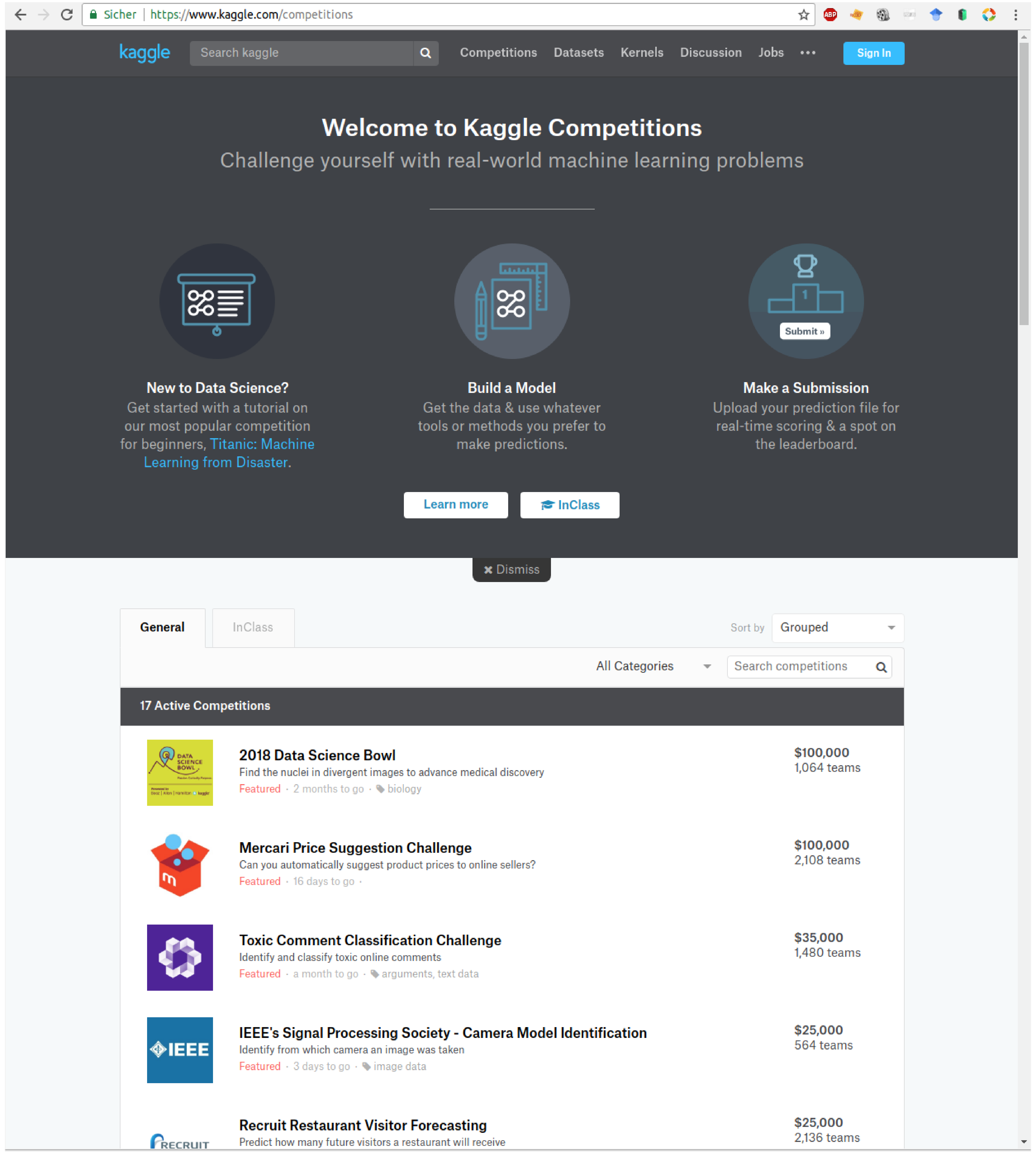
ML Praktikum 17/18

Kaggle: San Francisco Crime Challenge

Kaggle

https://www.kaggle.com

- Machine learning competitions + datasets
  1. Download a dataset
  2. Build a model
  3. Upload predictions or script
  4. Get ranked against others
  5. $$$ Profit



San Francisco Crime Challenge

The task is to predict the Category of a crime given the time and location. The dataset contains incidents from the SFPD Crime Incident Reporting system from 2003 to 2015 (878049 datapoints for training) with the following variables:

- Dates – timestamp of the crime incident
- Category – category of the crime (target variable) – 39 different categories
- Descript – detailed description of the crime incident (only in training set)
- DayOfWeek – the day of the week
- PdDistrict – name of the Police Department District
- Resolution – how the crime incident was solved (only in training set)
- Address – approximate address of the crime incident
- X – Longitude
- Y – Latitude

San Francisco Crime Challenge

2003-01-07 07:52:00	WARRANTS	WARRANT ARREST	Tuesday	SOUTHERN	ARREST, BOOKED	5TH ST / SHIPLEY ST	-122.402843	37.779829
2003-01-07 04:49:00	WARRANTS	ENROUTE TO OUTSIDE JURISDICTION	Tuesday	TENDERLOIN	ARREST, BOOKED	CYRIL MAGNIN STORTH ST / EDDY ST	-122.408495	37.784452
2003-01-07 03:52:00	WARRANTS	WARRANT ARREST	Tuesday	NORTHERN	ARREST, BOOKED	OFARRELL ST / LARKIN ST	-122.417904	37.785167
2003-01-07 03:34:00	WARRANTS	WARRANT ARREST	Tuesday	NORTHERN	ARREST, BOOKED	DIVISADERO ST / LOMBARD ST	-122.442650	37.798999
2003-01-07 01:22:00	WARRANTS	WARRANT ARREST	Tuesday	SOUTHERN	ARREST, BOOKED	900 Block of MARKET ST	-122.409537	37.782691
2003-01-06 23:30:00	WARRANTS	ENROUTE TO OUTSIDE JURISDICTION	Monday	BAYVIEW	ARREST, BOOKED	REVERE AV / INGALLS ST	-122.384557	37.728487
2003-01-06 23:14:00	WARRANTS	WARRANT ARREST	Monday	CENTRAL	ARREST, BOOKED	BUSH ST / HYDE ST	-122.417019	37.789110
2003-01-06 22:45:00	WARRANTS	WARRANT ARREST	Monday	SOUTHERN	ARREST, BOOKED	800 Block of BRYANT ST	-122.403405	37.775421
2003-01-06 22:45:00	WARRANTS	ENROUTE TO OUTSIDE JURISDICTION	Monday	SOUTHERN	ARREST, BOOKED	800 Block of BRYANT ST	-122.403405	37.775421
2003-01-06 22:19:00	WARRANTS	ENROUTE TO OUTSIDE JURISDICTION	Monday	NORTHERN	ARREST, BOOKED	GEARY ST / POLK ST	-122.419740	37.785893
2003-01-06 21:54:00	WARRANTS	ENROUTE TO OUTSIDE JURISDICTION	Monday	NORTHERN	ARREST, BOOKED	SUTTER ST / POLK ST	-122.420120	37.787757

- Evaluation by computing Logarithmic Loss (logloss)
- Classifier needs to assign probability to each class (instead of just outputting most likely one)
- Probabilities have to be calculated on test.csv (does not contain labels, descs or resolution)

Over all the N datarows, the mean of the log of the probability that the classifier assigned to the true label is calculated (see also: [1,2]):

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij})$$

$$N = \#datarows, M = \#labels, y = binary\ indicator, p = probability$$

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij})$$

$$N = \#datarows, M = \#labels, y = binary\ indicator, p = probability$$

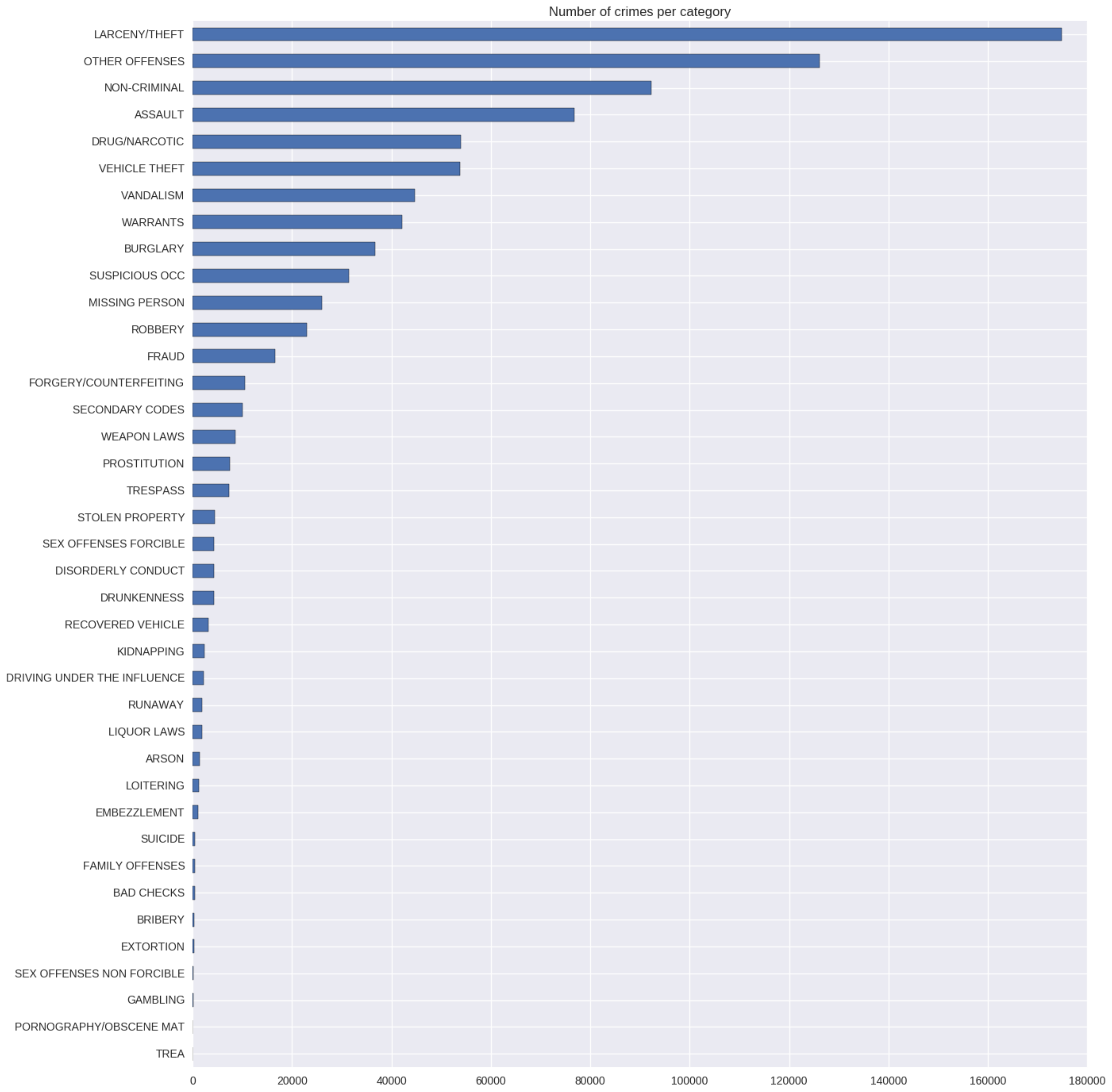
Intuition:

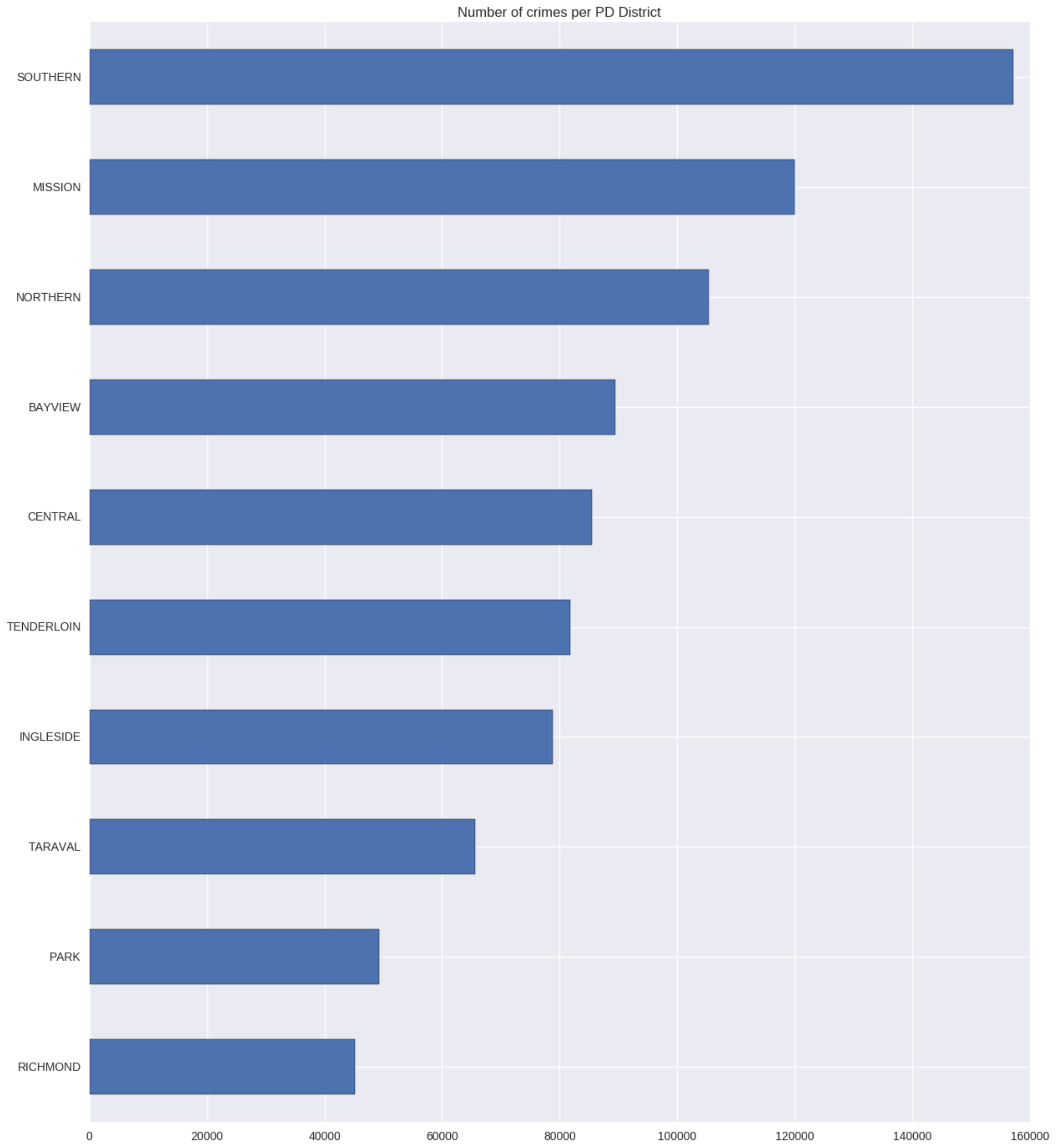
- p_{ij} is near zero for correct label: $\log(0+\epsilon)$ becomes very large
- p_{ij} is near 1 for correct label: $\log(1)$ becomes close to 0
- Uniform probability to all 39 labels: $\log\left(\frac{1}{39}\right) = 3.66$

The mean of these values over all datarows is the final logloss value for our classifer.

Visualization and Pre-Processing

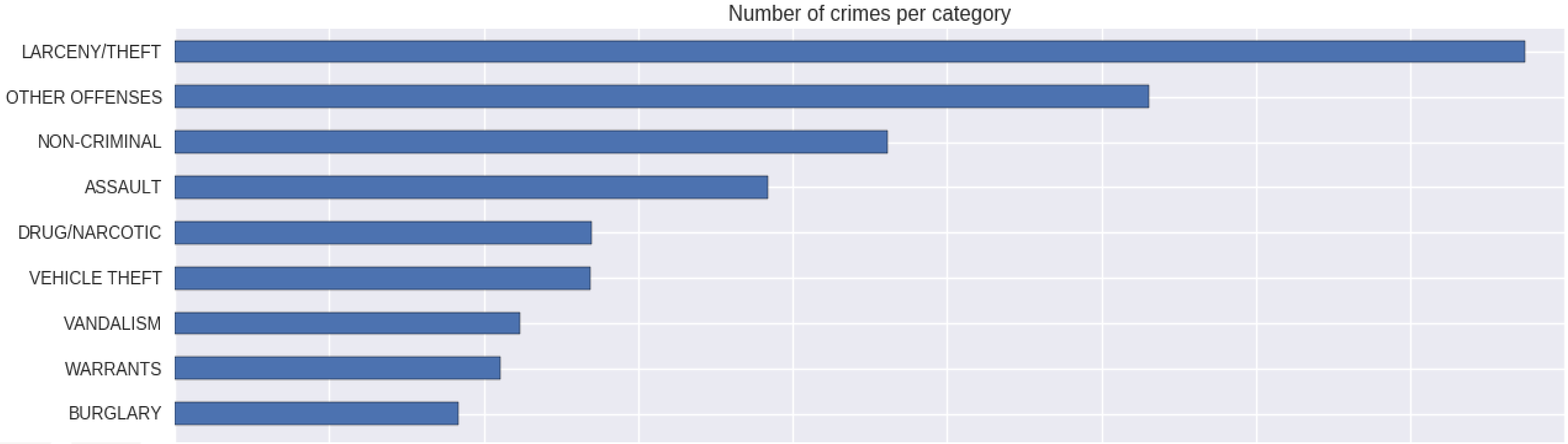
As a first step, I visualized the variables of the dataset to get an understanding of the involved variables, and identify which variables could be used to differentiate between crime-categories.





Visualization and Pre-Processing

As a first step, I visualized the variables of the dataset to get an understanding of the involved variables, and identify which variables could be used to differentiate between crime-categories.

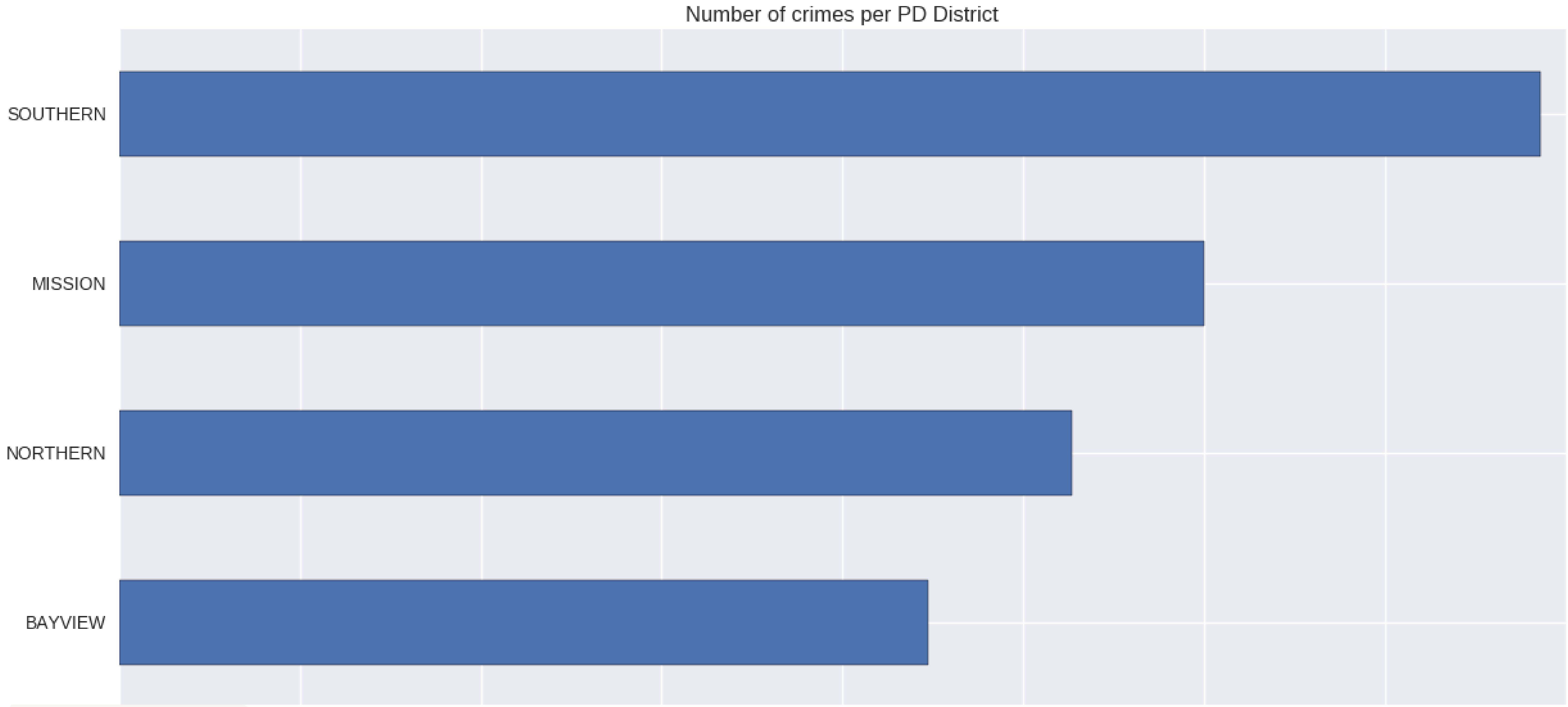


The histogram of the categories revealed that there exists a clear ordering in the amounts of different crimes. The top 6 in descending order:

1. Larceny/Theft
2. Other Offenses
3. Non-Criminal
4. Assault
5. Drug/Narcotic
6. Vehicle Theft

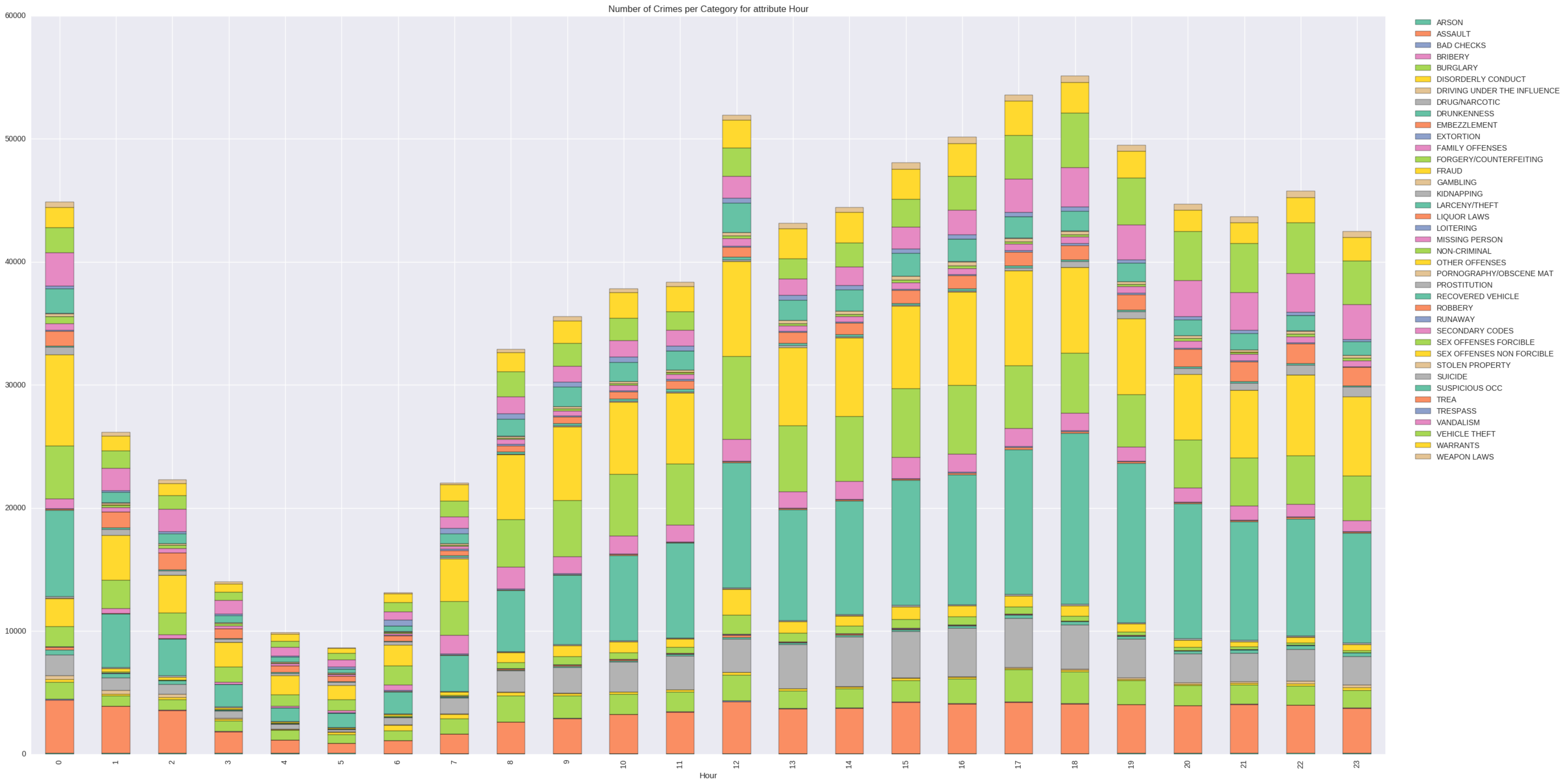
Visualization and Pre-Processing

As a first step, I visualized the variables of the dataset to get an understanding of the involved variables, and identify which variables could be used to differentiate between crime-categories.

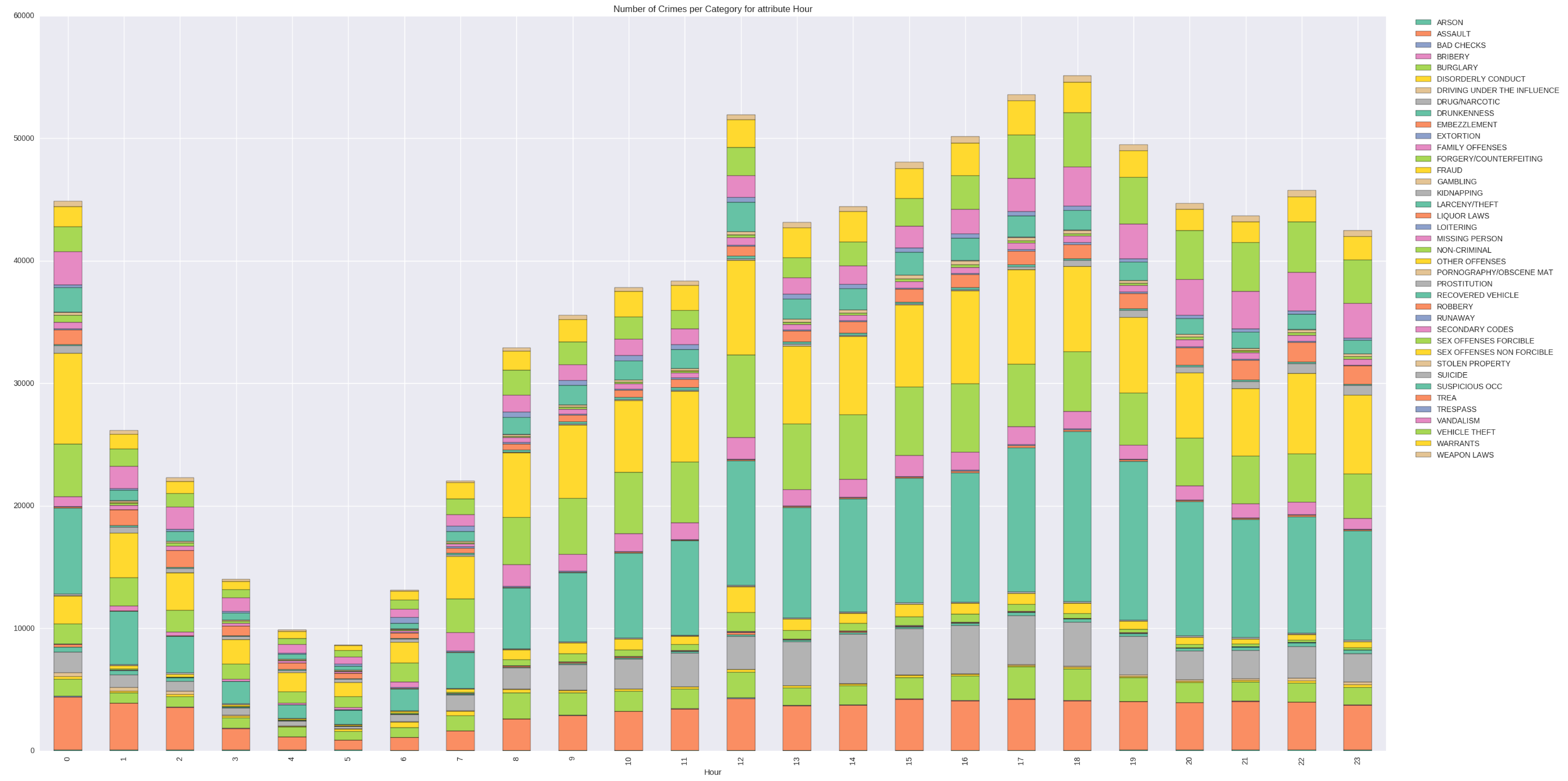


San Francisco Crime Challenge

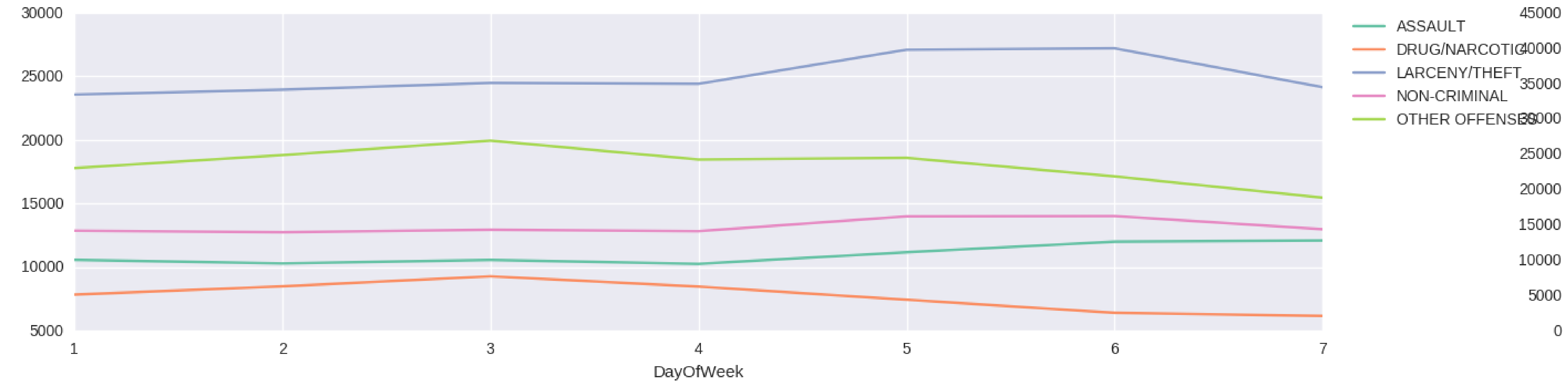
The timestamp seems to be a good indicator, different crimes seem to have different days and times at which they tend to happen most often, which might give additional hints to the classifier.

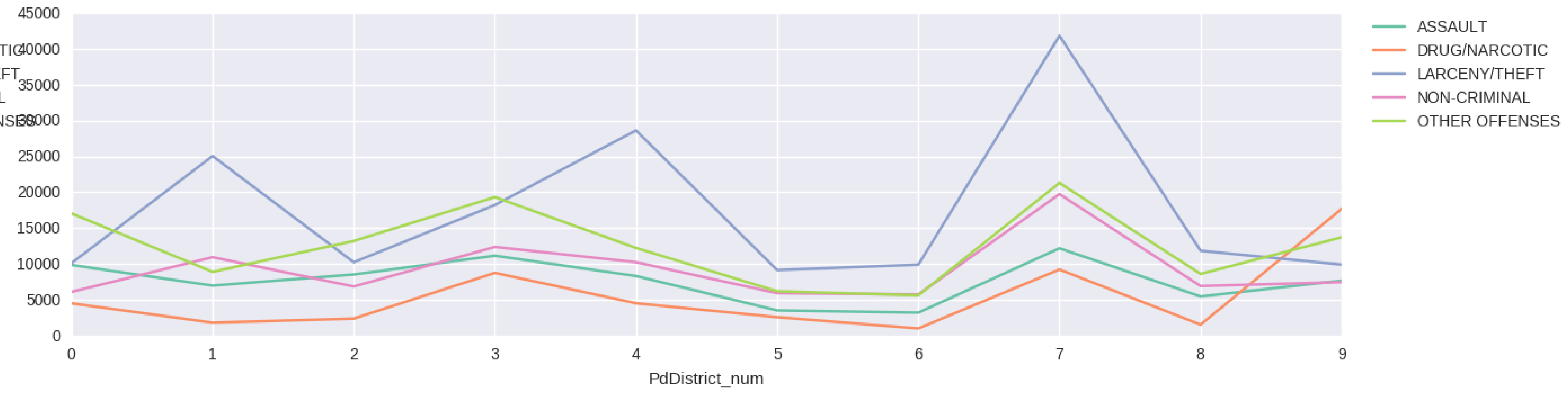


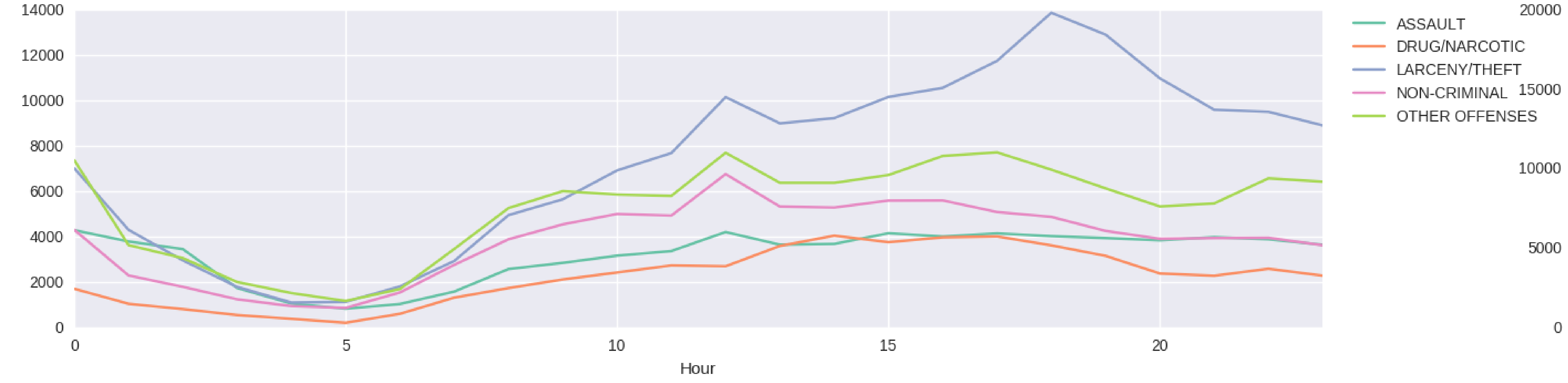
San Francisco Crime Challenge

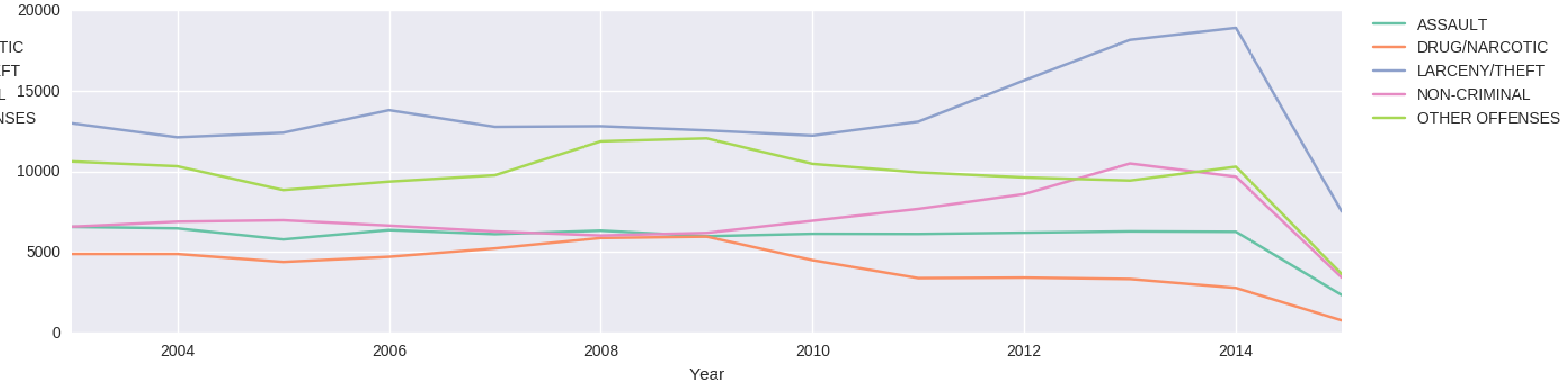


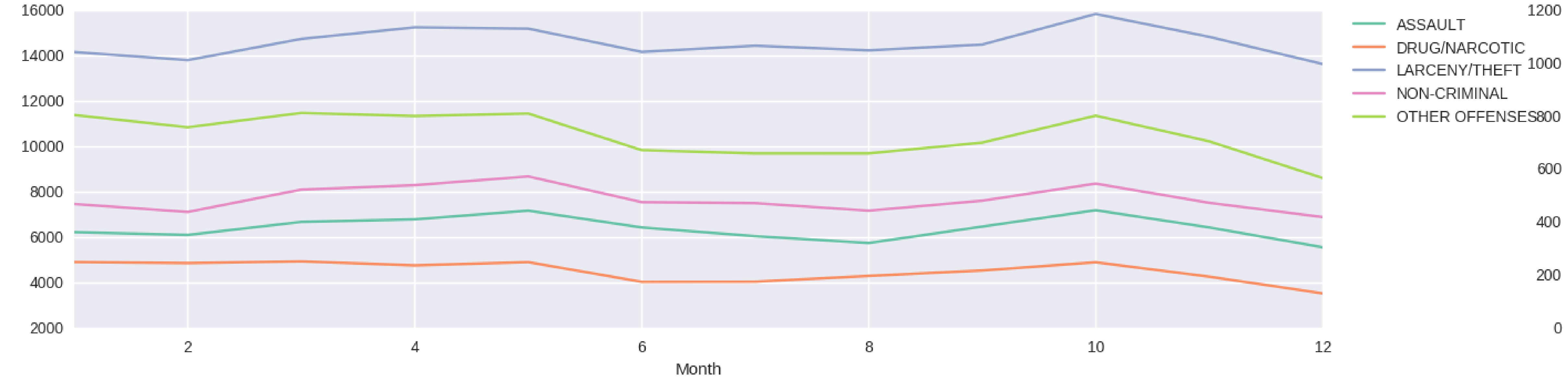
San Francisco Crime Challenge

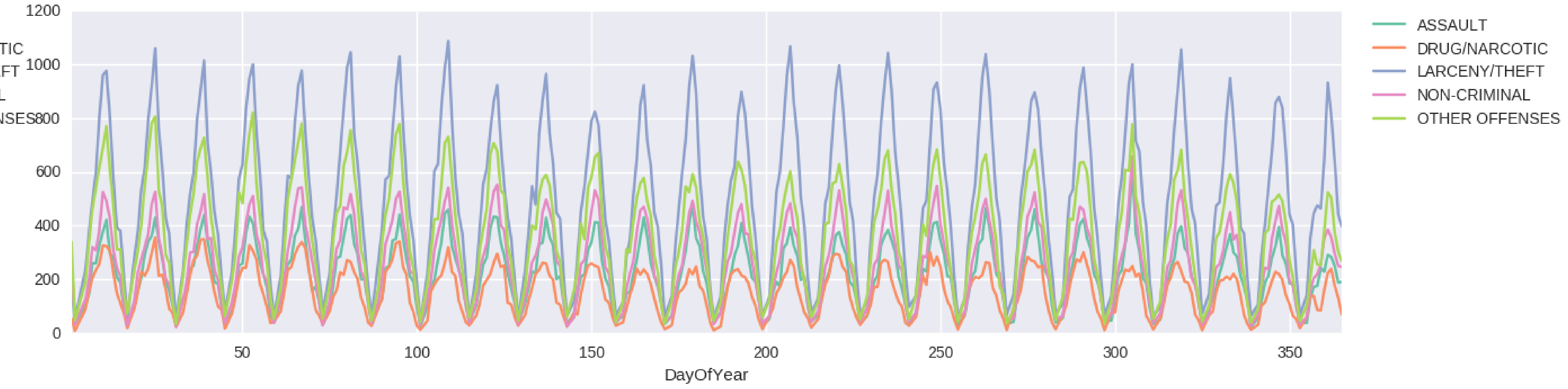


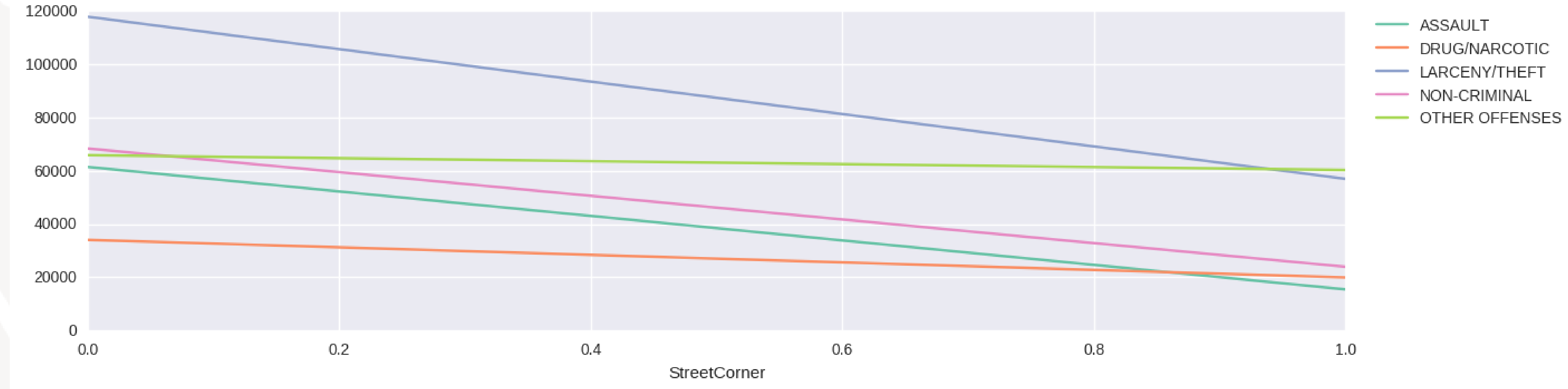




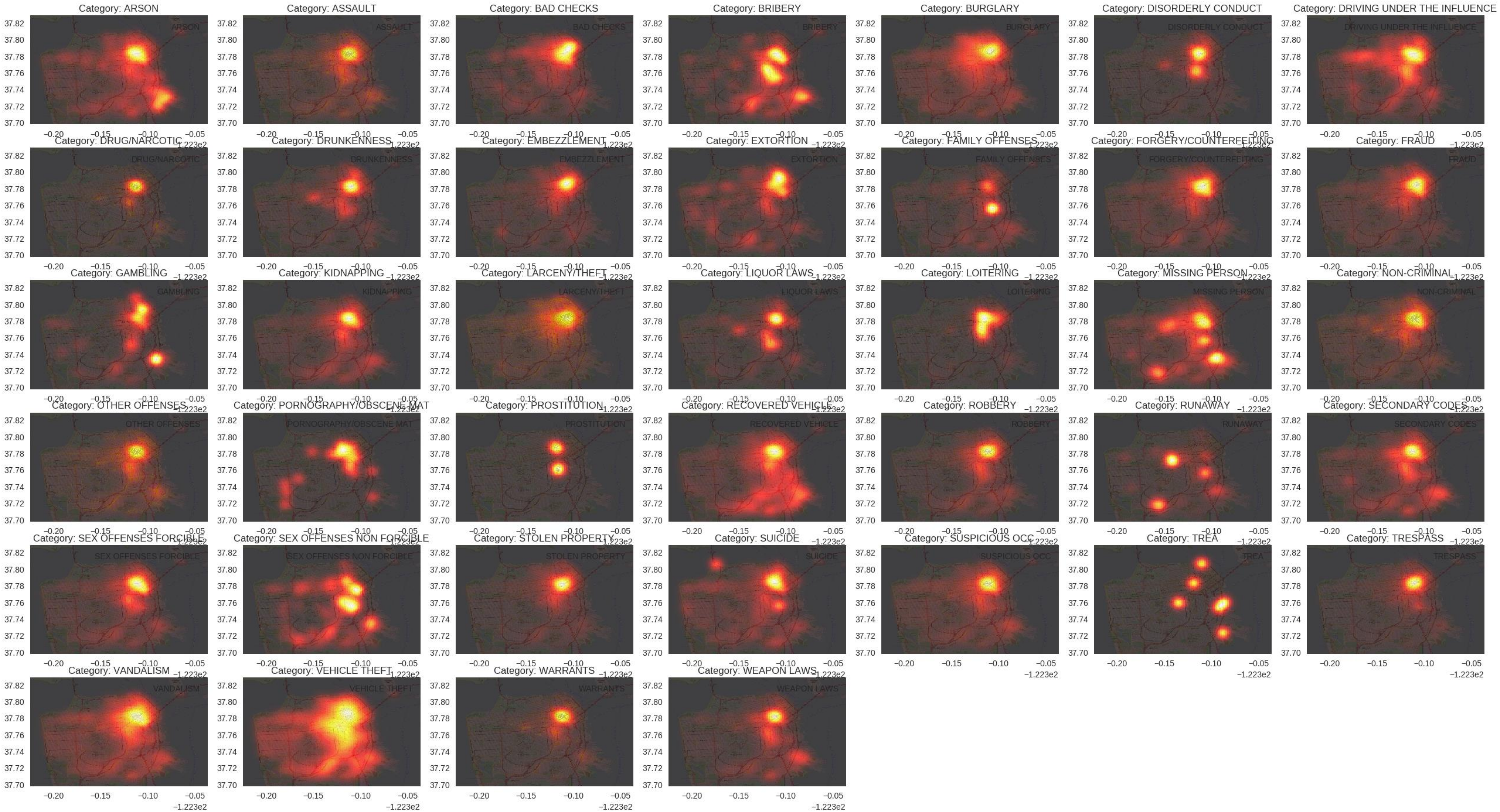




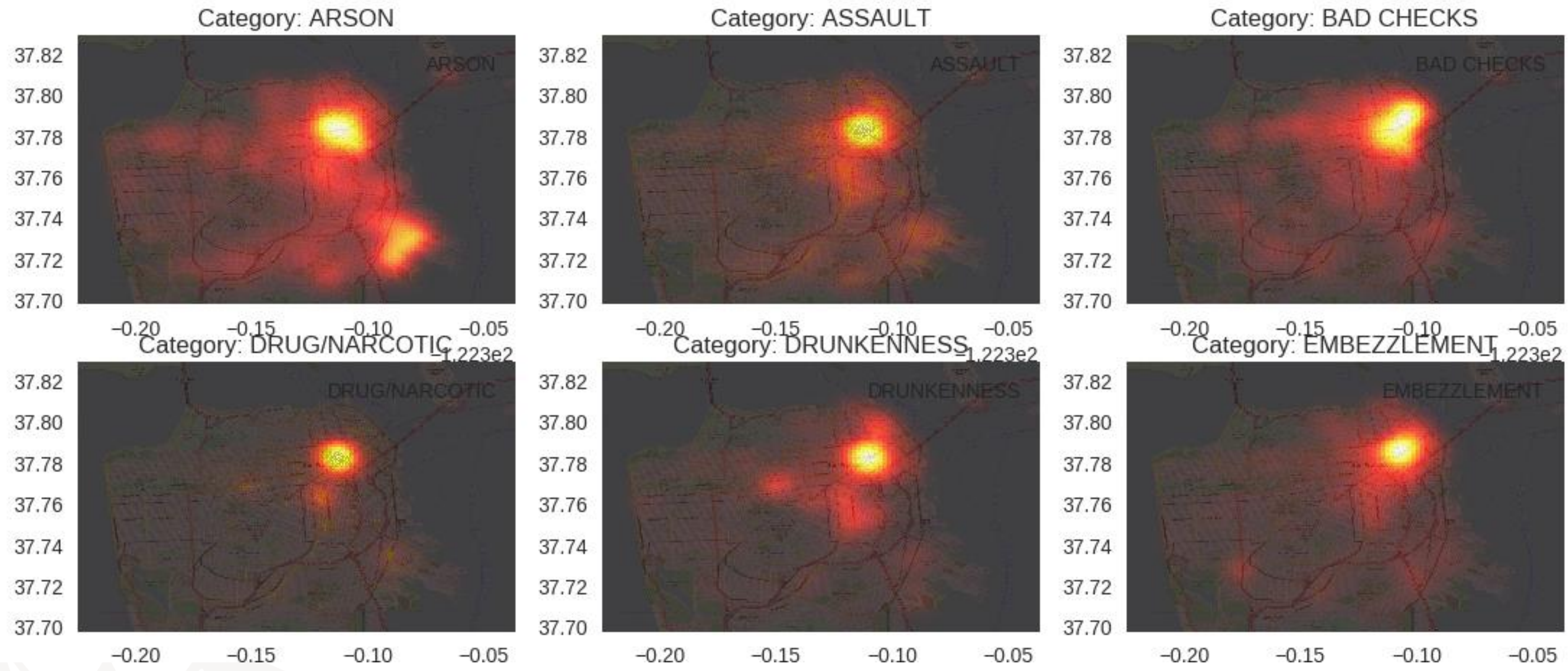




San Francisco Crime Challenge



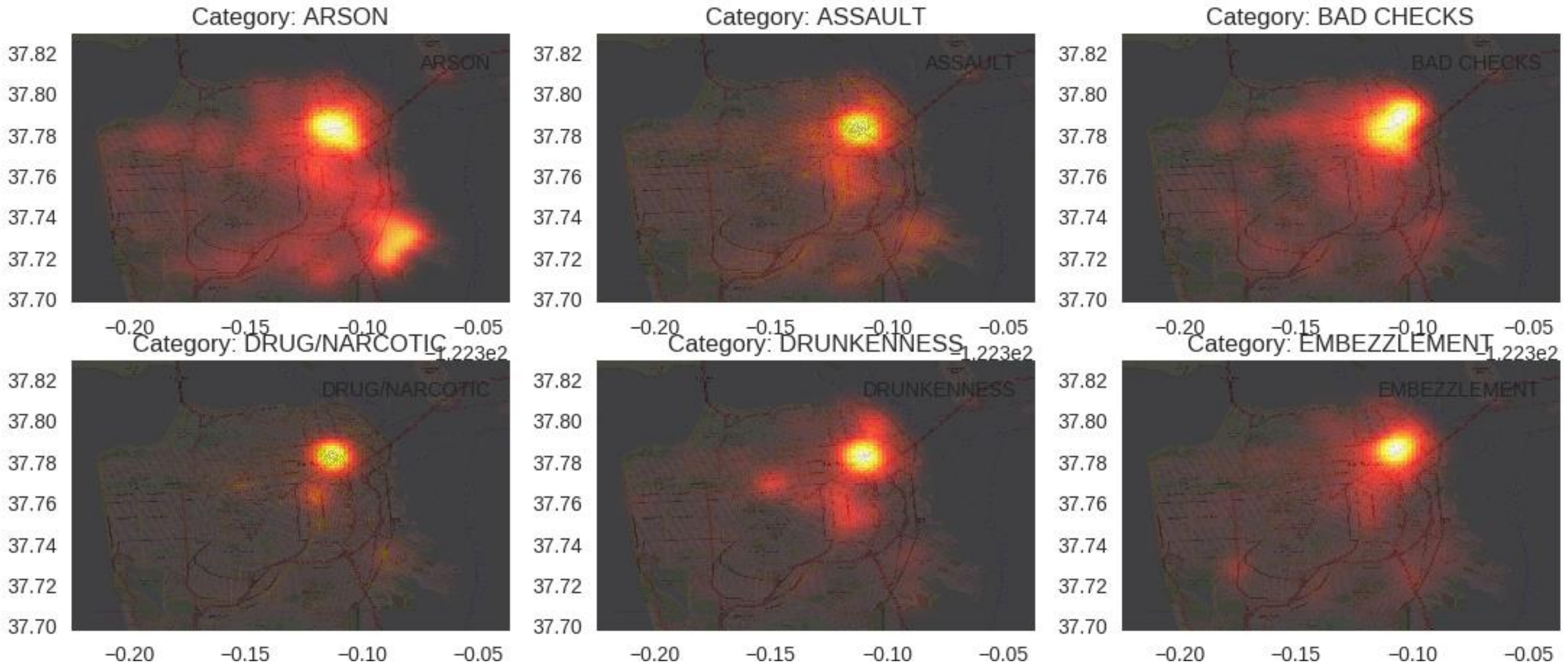
San Francisco Crime Challenge



San Francisco Crime Challenge

Here I noticed that 67 of the coordinates were far out (-120.5, 90.), so I removed those outliers.

By plotting the X/Y variables on the map of SF, I could see that the majority of crimes are concentrated on the north-east area, and that different crimes have slightly different spatial distributions.



GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Preprocessing

I decided to encode all features into numerical variables.
The order of the weekdays was random, so I transformed them to range from 0 (Monday) to 6 (Sunday).
For the PdDistricts I used the pandas functions to first transform them to categoricals and then get their index.
For the dates i also had pandas preprocess those for me, so I could directly access Day,Month,Year,Hour, etc. instead of coding this by hand.

Baseline

In order to get an understanding of the loss-function used, I first calculated the loss of two baseline ideas:

Uniform probabilities for all classes: 3.66

Always choose the most common category (LARCENY/THEFT, 174900 vs. Rest 703149), setting 0 proba as 1e-15 as log(0) is not defined: 27.66

This also tells that wrong choices with high probabilities get penalized pretty hard.

Classifier training

First I split the training data into train- and test-set (.9/.1), making sure to set the random state to a fixed value to ensure reproducability.

But the standard parameter-values of all classifiers performed poor on average, so I ran a random parameter search instead (sklearn.RandomizedSearchCV) over a wide range of parameters on each classifier/feature set before reporting scores. I did not split the data anymore, as the RandomizedSearchCV is using cross-validation internally. Furthermore, I replaced the scoring-function of the RandomizedSearch by the log-loss function to directly search for best options for this specific problem).

As we have been talking about them in the lecture, I first used a single Decision Tree on the features DayOfWeek, PdDistrict, Hour, which resulted in a score of 2.62. I chose this feature set because it already captured some aspects of time and space and was computationally cheap (basically only some categorical integer variables).

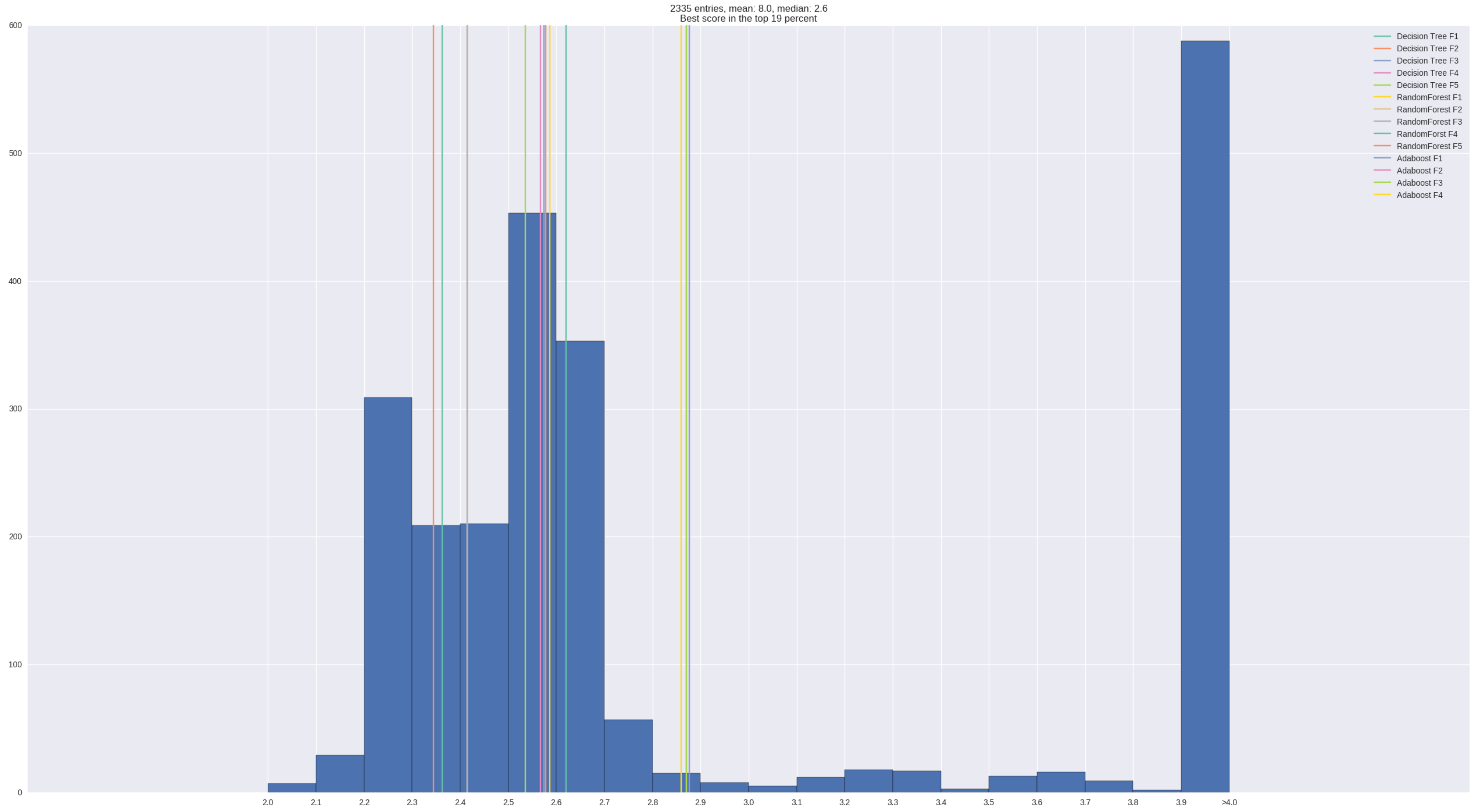
Please note that the scores reported are those that were reported by the RandomizedSearchCV

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Results

Evaluation against kaggle leaderboard

I wrote a small script that would parse the kaggle leaderboard for me, so I could build some statistics with it and see how well i did in comparison (https://github.com/TobiasWeis/kaggleLeaderboardStats).



Results

For the sake of completeness, here are the feature-sets I tried and the scores they achieved with different classifiers (I wrote a script that iterates through the feature-sets, performs a random search for each classifier and saves the result to a logfile):

Feature-Sets

F1 : DayOfWeek, PdDistrict_num, Hour

F2 : X, Y, DayOfWeek, Hour

F3 : X, Y, DayOfWeek, PdDistrict_num, Hour

F4 : X, Y, DayOfWeek, PdDistrict_num, Hour, Month, Year, Day, DayOfYear

F5 : X, Y, DayOfWeek, PdDistrict_num, Hour, Month, Year, Day, DayOfYear, Streetcorner

Explanation: I chose to use DayOfYear, Month and Year to capture seasonal dependencies, and thought that, even if I do not exploit the adresses to full extent, at least checking if the crime happened at a street corner instead of a regular adress could improve my results.

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Results

Feature-Set	Decision tree	Random forest	Adaboost
DayOfWeek, PdDistrict_num, Hour	2.62	2.59	2.877
X,Y,DayOfWeek, Hour	2.578	2.415	3.12
X,Y,DayOfWeek,PdDistrict_num, Hour	2.574	2.415	3.196
X,Y,DayOfWeek,PdDistrict_num, Hour, Month, Year, Day, DayOf Year	2.567	2.363	2.869
X,Y,DayOfWeek,PdDistrict_num, Hour, Month, Year, Day, DayOf Year, Streetcorner	2.535	2.344	2.990

Coordinates perform better than PdDistrict, and both combined give a slight improvement to the DecisionTree, have no effect on the RandomForest, but make the Adaboost-Classification worse.

Including more variables regarding the time of the crimes (set F4) improves all classifiers.

Creating the own variable StreetCorner further improved the result by a small margin.

To conclude, I visualized and cleaned the input data, was able to successively identify features that each improved the classification results, used hyperparameter-search to find a good set of hyperparameters for the chosen classifiers, and finally built a classifier that would score in the top 19% of the kaggle leaderboard of my chosen problem.

San Francisco Crime Challenge – Practice session

Now it‘s your part!

- Create an account on kaggle.com
- Go to https://www.kaggle.com/c/sf-crime, download the datasets
- Implement your own dataloader, classifier(s)
- Submit your result as „Late submission“ and see how you score
- Prepare a short presentation (2-5 min.) of the work you have done