San Francisco Crime Classification
by AJANS

rev.01: 2016.06.25 (start from 2016.04.30)

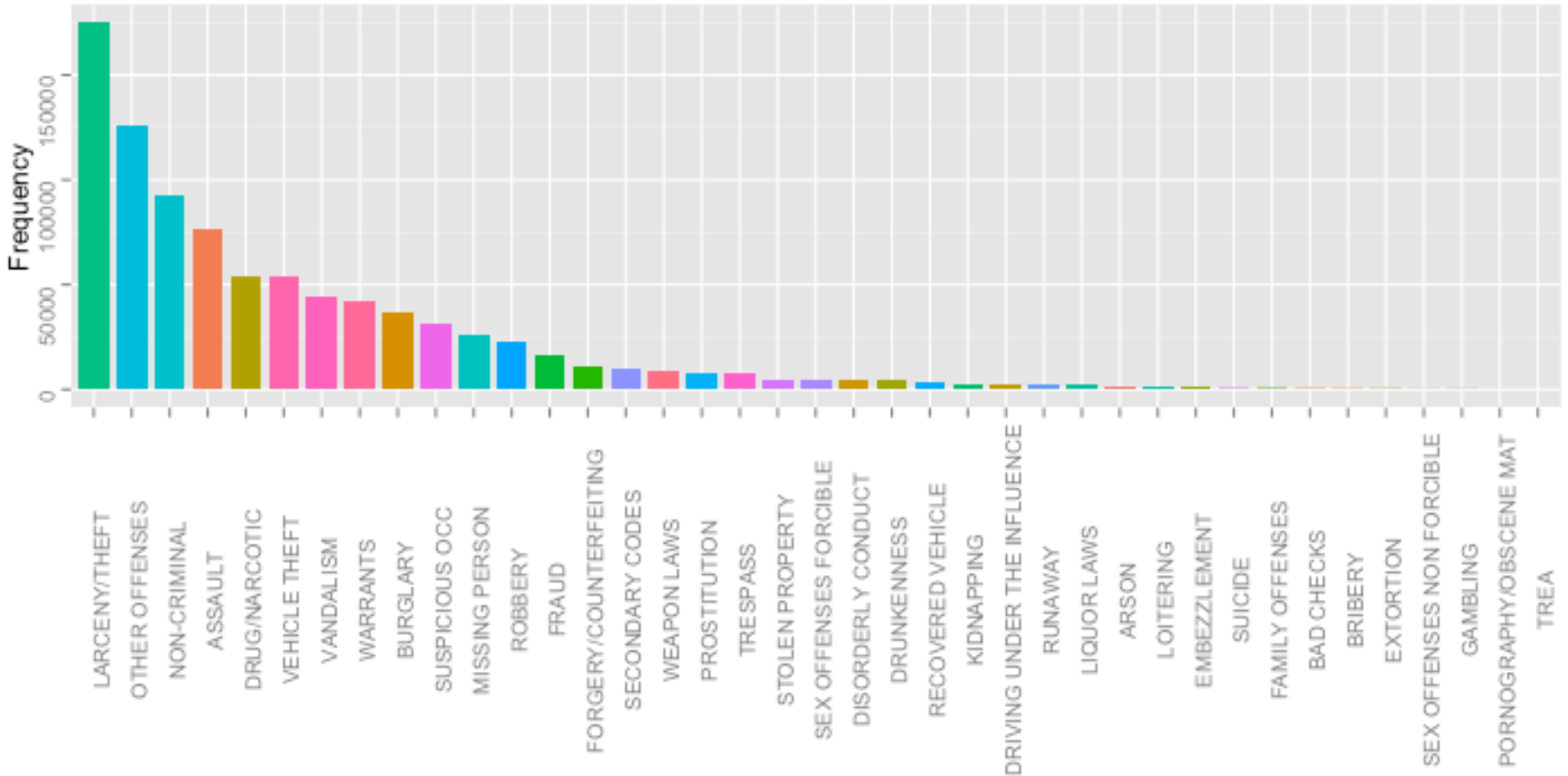
Data exploration

Types of variables for training data set

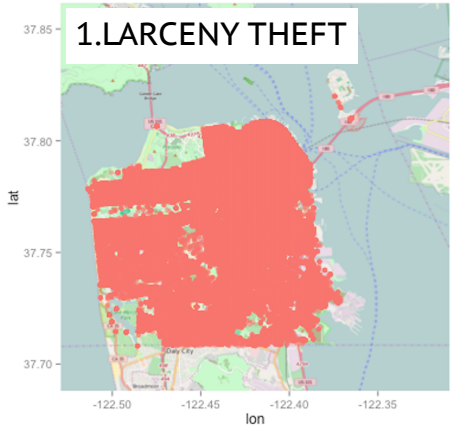
Variables	Types of Data	The number of types	Test data
Dates	Numerical (Continuous)	389257 values	Yes
Category	Categorical (Regular)	39 types	No
Descript	Categorical (Regular)	879 types	No
Day of Week	Categorical (Original)	7 levels	Yes
PdDistrict	Categorical (Regular)	10 types	Yes
Resolution	Categorical (Regular)	17 types	No
Address	Categorical (Regular)	23228 types	Yes
X,Y	Numerical (Continuous)	found some errors (-120.5, 90)	Yes

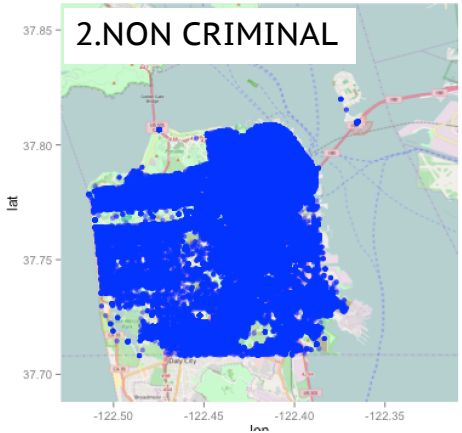
Category data (39 types) should be predicted by dates, day of week, PdDistrict, Address and X, Y.

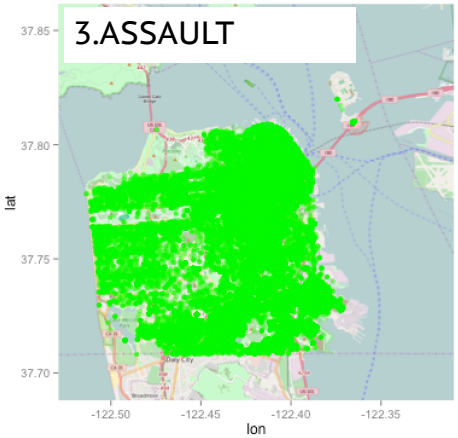
Sorting in order of frequency (bar chart)



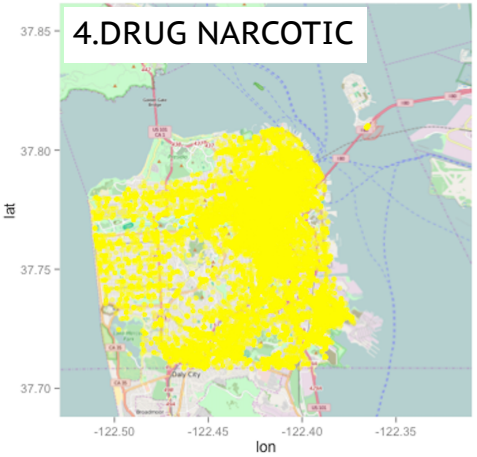
Longitude & Latitude (X, Y) information (1)

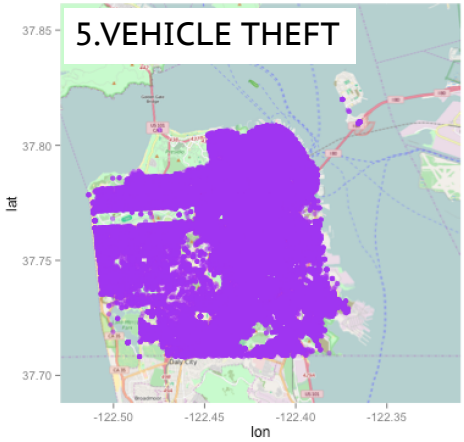






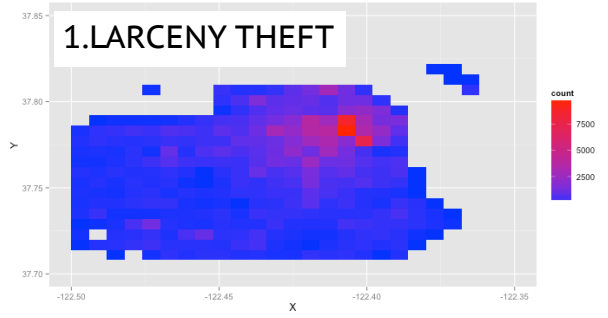
	Category	Counts
1	LARCENY/THEFT	174900
2	OTHER OFFENSES	126182
3	NON-CRIMINAL	92304
4	ASSAULT	76876
5	DRUG/NARCOTIC	53971
6	VEHICLE THEFT	53781
7	VANDALISM	44725
8	WARRANTS	42214
9	BURGLARY	36755
10	SUSPICIOUS OCC	31414

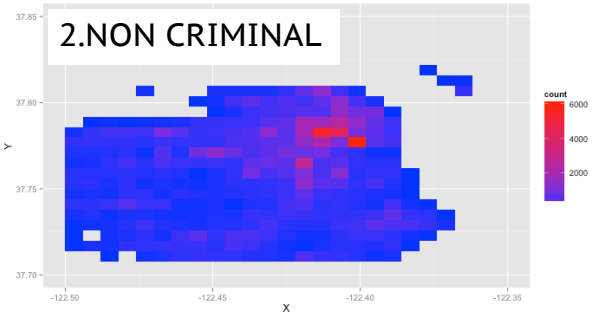


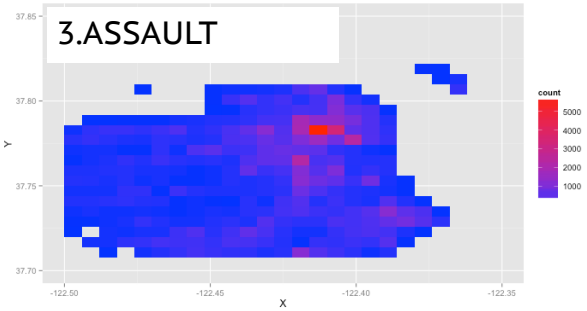


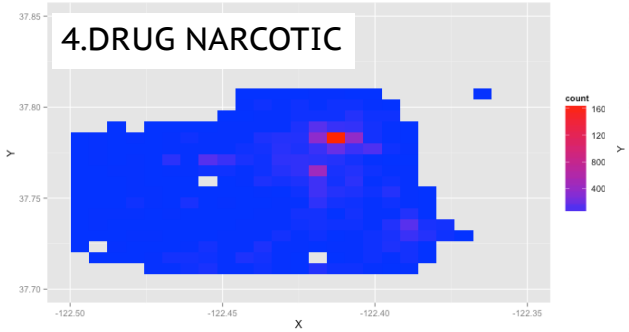
Mapping the data points with ggmap (top 1 ~ 5, remove other offenses). No clear trend was found.

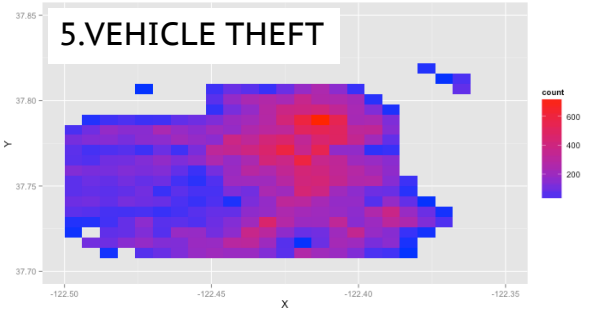
Longitude & Latitude (X, Y) information (2)







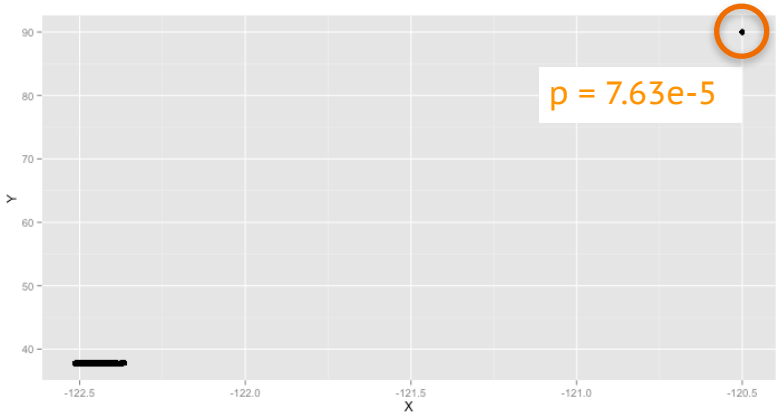




Vehicle Theft occurs all over the area unlike other crimes. Drug/Narcotic looks like more localized than others.

Mapping the data points with 2D histogram (top 1 ~ 5, remove other offenses).
Red color indicates high crime rate (not absolute value).

X,Y error information




```
> tmp <- data.med[data.med[,Address] %in% data.error[,Address]]
> tmp[Y!=90]
                  Address         X        Y
1: I-280 / CESAR CHAVEZ ST -121.4460 63.87504
2: I-280 / PENNSYLVANIA AV -121.4463 63.87588
3:    BRYANT ST / SPEAR ST -121.4440 63.89360
```

Calculated median X and Y with correct data.

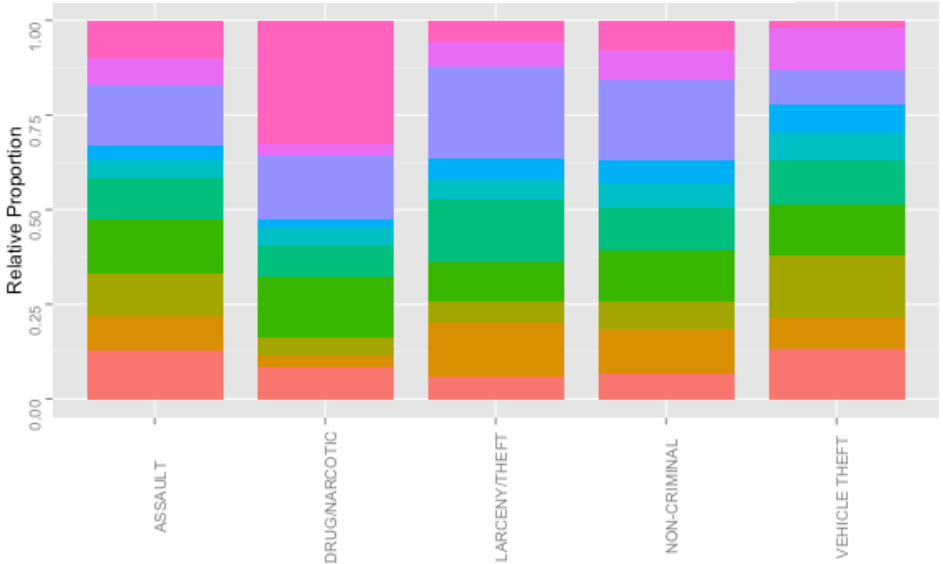
Out of 41 addresses, X/Y of only 3 addresses can be compensated with correct data. GPS error would happen at same locations almost.

Removing error values would be safe because the probability ,7.63e-5, is quite low (ignorable).

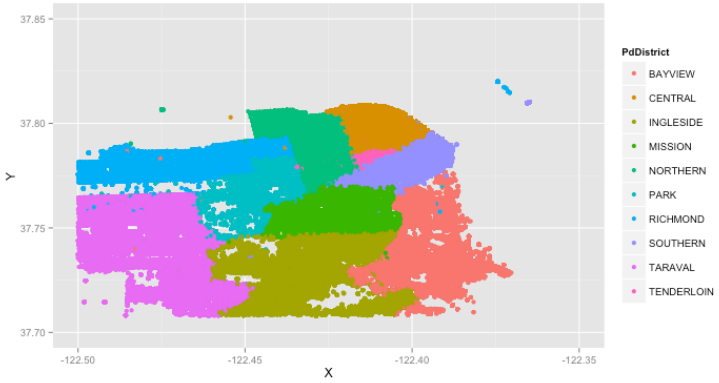
```
> data.med[data.med[,Address] %in% data.error[,Address]]
                                    Address         X        Y
 1:                 5THSTNORTH ST / OFARRELL ST -120.5000 90.00000
 2:              JAMESLICKFREEWAY HY / SILVER AV -120.5000 90.00000
 3:                   5THSTNORTH ST / EDDY ST -120.5000 90.00000
 4:                  5THSTNORTH ST / ELLIS ST -120.5000 90.00000
 5:                  ELLIS ST / 5THSTNORTH ST -120.5000 90.00000
 6:                YOSEMITE AV / WILLIAMS AV -120.5000 90.00000
 7:               BRENHAM PL / WASHINGTON ST -120.5000 90.00000
 8:            AVENUE OF THE PALMS / GEARY BL -120.5000 90.00000
 9:                  STCHARLES AV / 19TH AV -120.5000 90.00000
10:               OFARRELL ST / 5THSTNORTH ST -120.5000 90.00000
11:                   BRANNAN ST / 1ST ST -120.5000 90.00000
12:                 TURK ST / STJOSEPHS AV -120.5000 90.00000
13:      MONTGOMERY ST / THE EMBARCADERONORTH ST -120.5000 90.00000
14:                  FITCH ST / DONNER AV -120.5000 90.00000
15:              7THSTNORTH ST / MCALLISTER ST -120.5000 90.00000
16:            AVENUE OF THE PALMS / EUCLID AV -120.5000 90.00000
17:                  VANNESS AV / BEACH ST -120.5000 90.00000
18:                 PERSIA AV / LAGRANDE AV -120.5000 90.00000
19:                  3RD ST / ISLAISCREEK ST -120.5000 90.00000
20:            GEARY BL / AVENUE OF THE PALMS -120.5000 90.00000
21:                  EDDY ST / 5THSTNORTH ST -120.5000 90.00000
22:                  AUSTIN ST / LARKIN ST -120.5000 90.00000
23:                JENNINGS CT / INGALLS ST -120.5000 90.00000
24:              GENEVA AV / INTERSTATE280 HY -120.5000 90.00000
25:               ARGUELLO BL / NORTHRIDGE DR -120.5000 90.00000
26:            CHARLES J BRENHAM PL / CLAY ST -120.5000 90.00000
27:                  STELMO WY / MONTEREY BL -120.5000 90.00000
28:                  I-280 / CESAR CHAVEZ ST -121.4460 63.87504
29:      JAMES LICK FREEWAY HY / CESAR CHAVEZ ST -120.5000 90.00000
30:                  I-280 / PENNSYLVANIA AV -121.4463 63.87588
```

PdDistrict information

bar chart (relative proportion)



link to X and Y info.



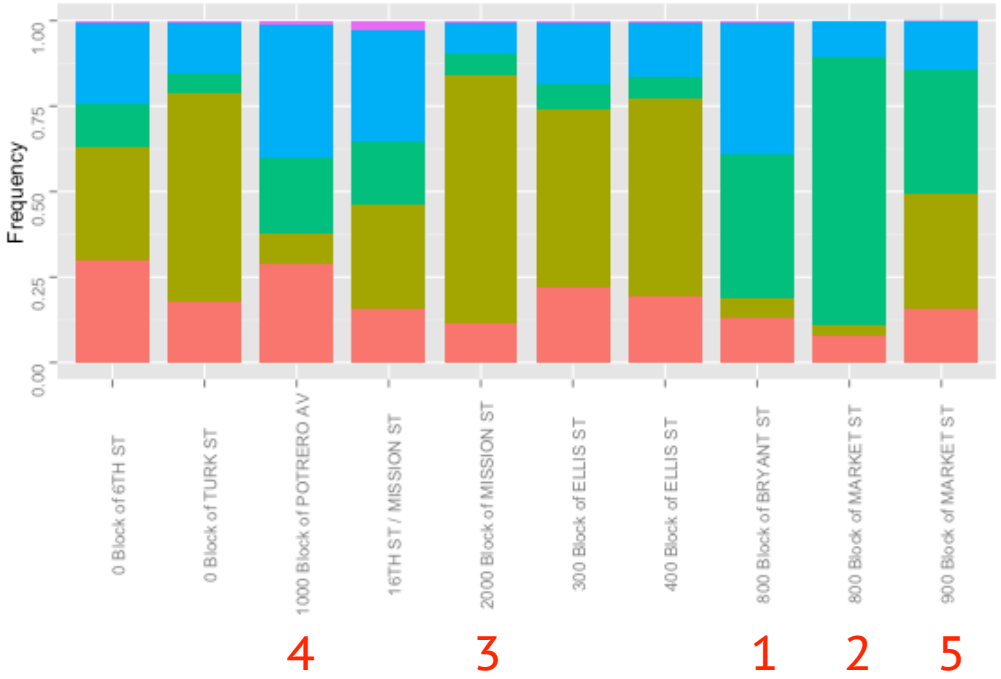
	Category	Counts
1	LARCENY/THEFT	174900
2	OTHER OFFENSES	126182
3	NON-CRIMINAL	92304
4	ASSAULT	76876
5	DRUG/NARCOTIC	53971
6	VEHICLE THEFT	53781
7	VANDALISM	44725
8	WARRANTS	42214
9	BURGLARY	36755
10	SUSPICIOUS OCC	31414

In TENDERLOIN, Drag/Narcotic happens many times, however, the frequency of other crimes is comparatively low.

SOUTHERN, MISSION and BAYVIEW area have high crime rate comparatively.

Address information

bar chart (relative proportion)



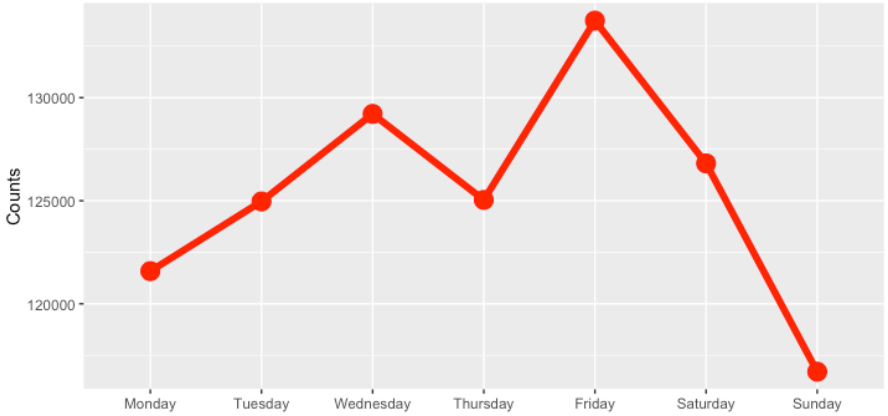
worst 10 addresses

	Address	Counts
1	800 Block of BRYANT ST	26533
2	800 Block of MARKET ST	6581
3	2000 Block of MISSION ST	5097
4	1000 Block of POTRERO AV	4063
5	900 Block of MARKET ST	3251
6	0 Block of TURK ST	3228
7	0 Block of 6TH ST	2884
8	300 Block of ELLIS ST	2703
9	400 Block of ELLIS ST	2590
10	16TH ST / MISSION ST	2504

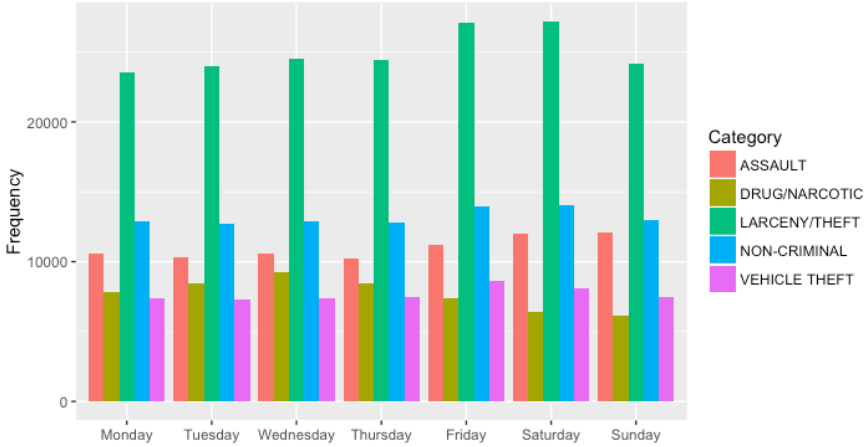
Address information has clearer dependency compared to PdDistrict.

Day of Week (1)

Total.



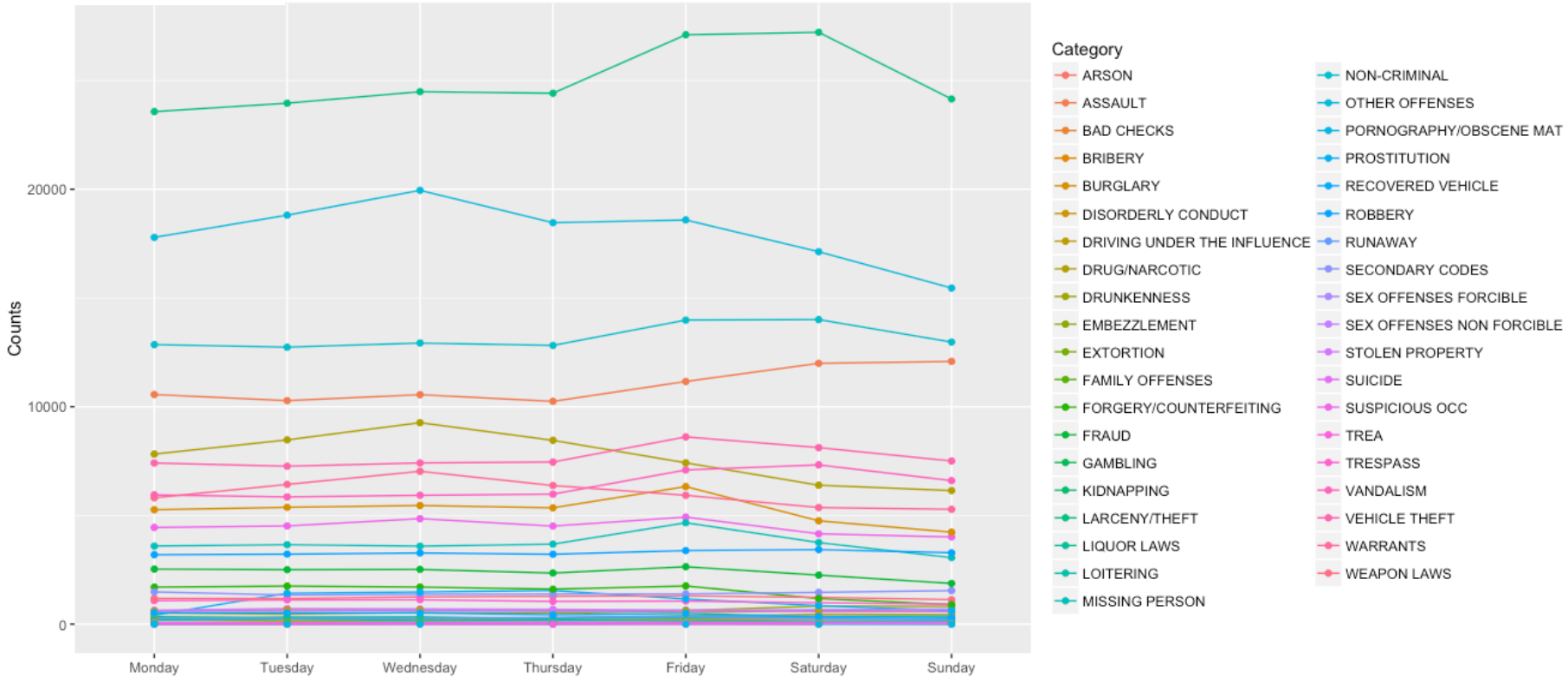
Top 5 crimes (remove other offenses).



The highest frequency of crime was obtained on Friday (Sunday was the lowest).
But, it depends on type of crime (Wednesday is the highest on Drag/Narcotic, Saturday is the highest in case of Assault).

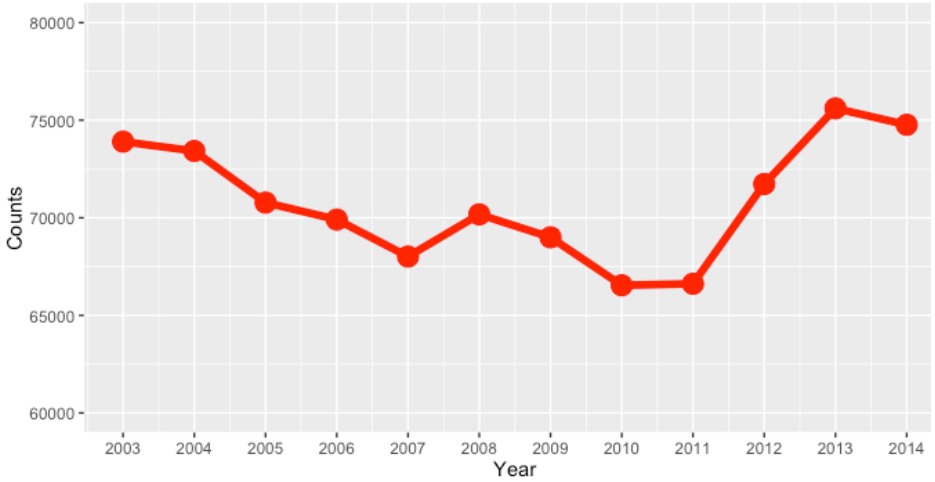
Day of Week (2)

All types.

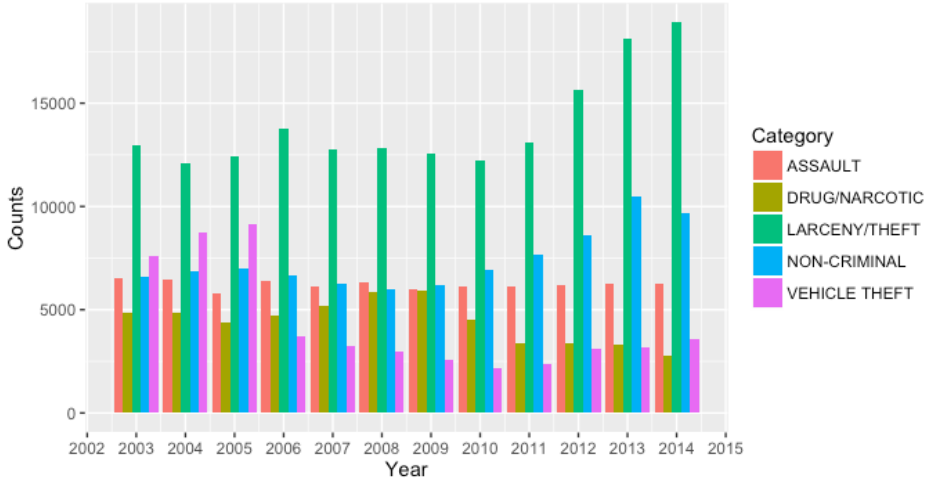


Dates (Year)

Total.



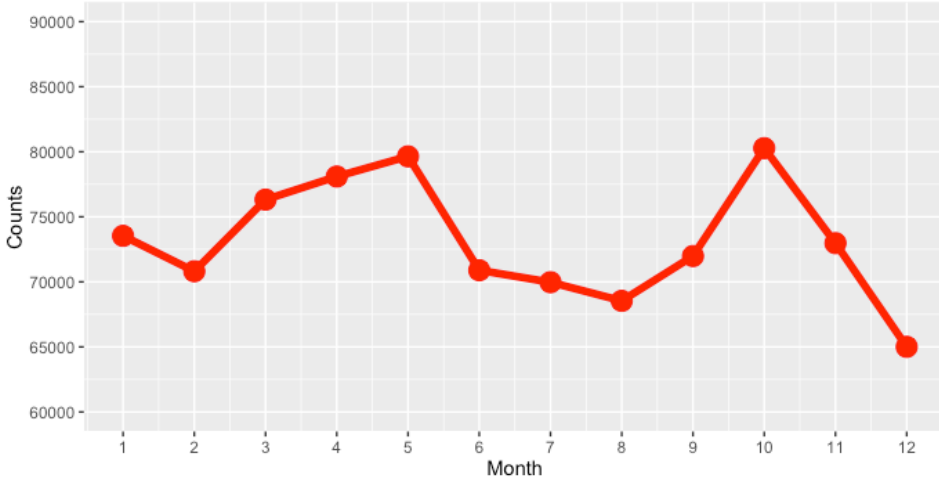
Top 5 crimes (remove other offenses).



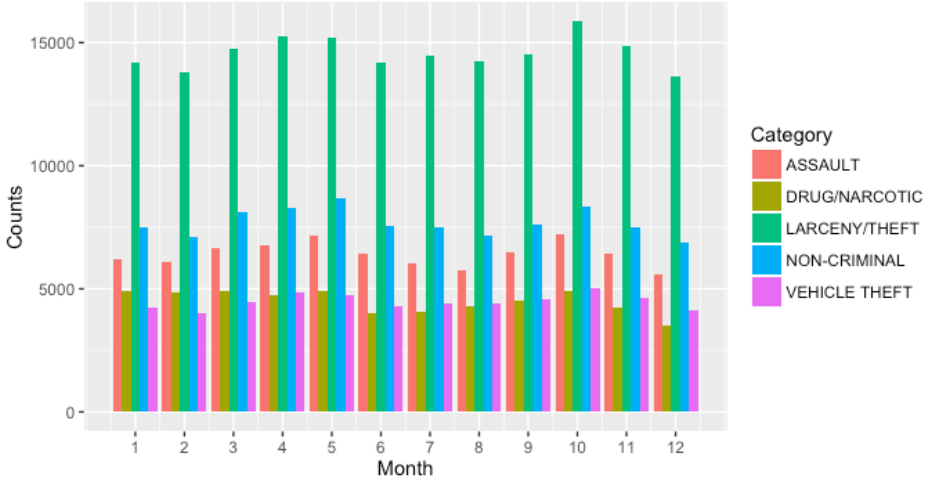
In recent years, the number of crimes has been increased especially on Larceny/Theft, Non Criminal. However, Vehicle Theft and Drug/Narcotic have been decreased on the other hand. Assault was flat almost.

Dates (Month)

Total.



Top 5 crimes (remove other offenses).

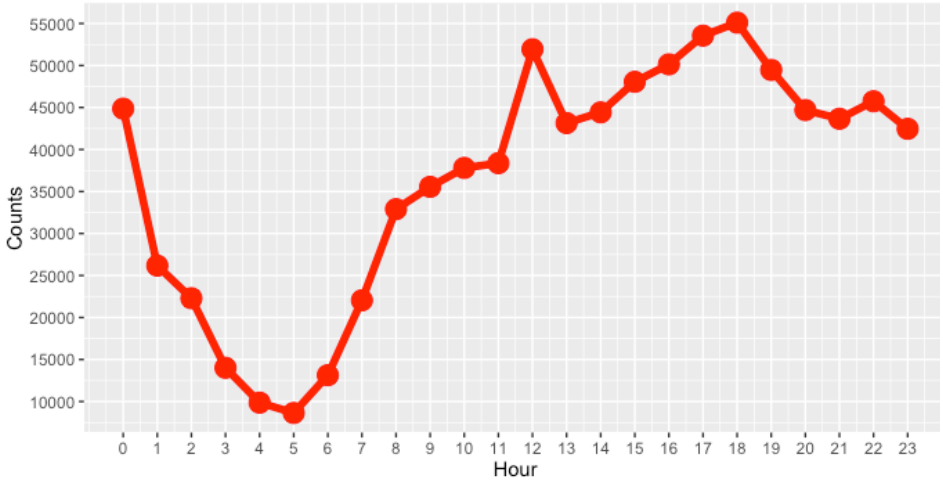


There are two peaks, on May and October.

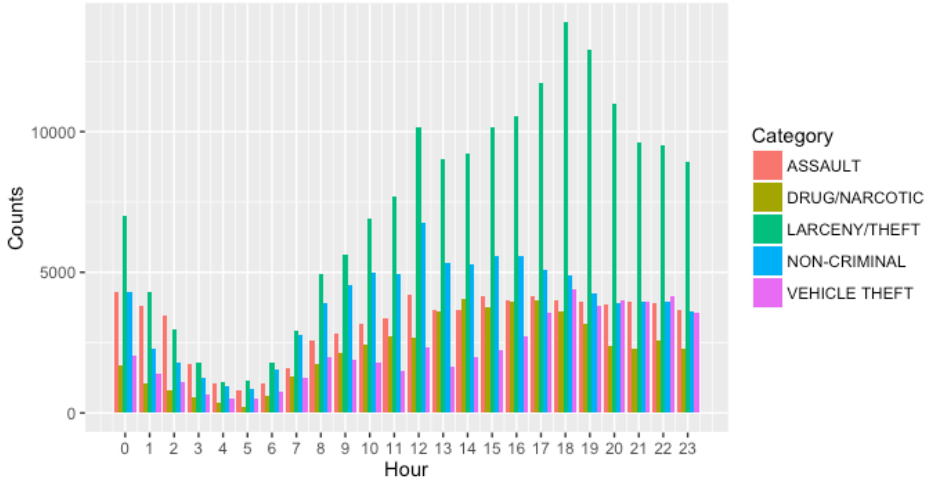
All top 5 types of crimes look like having same trend.

Dates (Hour)

Total.



Top 5 crimes (remove other offenses).



Low frequency is seen at the morning hours (2:00 ~ 7:00 o'clock). After that, crime rate is increased gradually to the highest peak at 18:00. At noon time, there is comparatively high peak. It would depend on human activity.

Machine Learning

Preparation

Selected Algorithm,

Neural Network, Support Vector Machine, Random Forest, Naive Bayes,
Linear Model, Xgboost

Pick up Parameters,

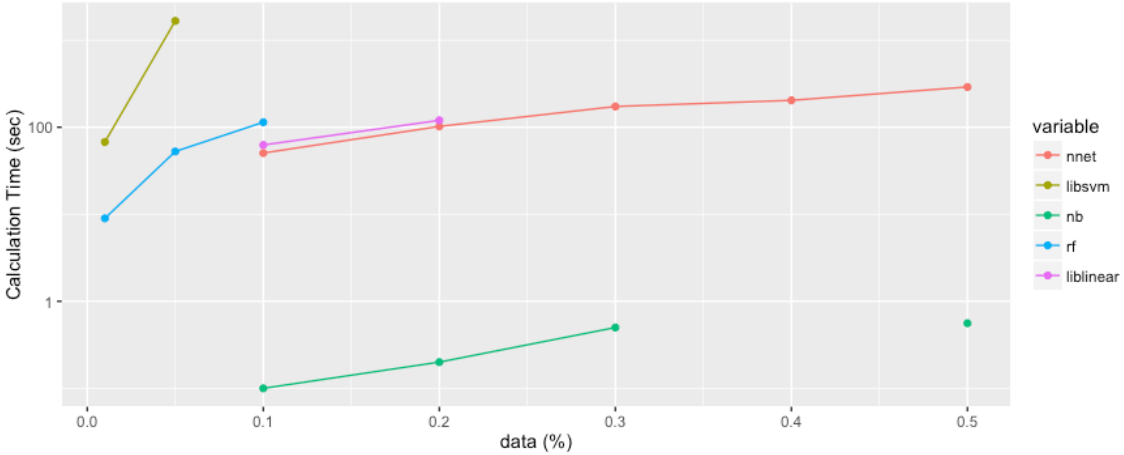
X, Y, Year, Month, Hour, DayOfWeek

*error values for X,Y were removed.
**Numerical parameters (X, Y, Year, Month, Hour) were scaled to 0 ~ 1.
***DayOfWeek, which is categorical variable, was converted into dummy variables.

Data set,

Train test set (878049 rows) was separated to two data sets (half and half).
One was used for training and another was used to check prediction accuracy.
In order to train, top 10 types of crimes were treated (42 % of all the data).
Prediction power was measured on test data set with all the 39 levels.

Problem with R



Tools,
MacBook Pro
R version 3.3.0
Processor: 2.2 GHz Intel Core i7
Memory: 16 GB 1600 MHz DDR3
IBM workbench (RS)
(https://datascientistworkbench.com/)

R has a problem with calculation speed because single core is used for data processing usually. Here is techniques for fast calculation with R.

Parallel Processing,

foreach, doMC (for mac and linux) packages can be used to speed up the calculation for training with caret package and prediction for test set.

High speed processing for large data frame,

dplyr, data.table packages were suitable for very large data frame.

SparkR,

Support R in Spark (in memory DB). But, my impression is not enough library prepared for ML still.

Neural Network (Testing Condition)

Feature of Algorithm,

Feed forward neural networks with a single hidden layer. Multi class classification is possible. Sigmoid function is used for the activation function in nnet.

Data set for training/test,

train = 42 % (treat with top 10 levels of category)
test = 50 % (treat with all 39 levels of category)

Training Method,

10 fold cross validation (1 repeated)

Parameters (search for optimum parameter with grid search),

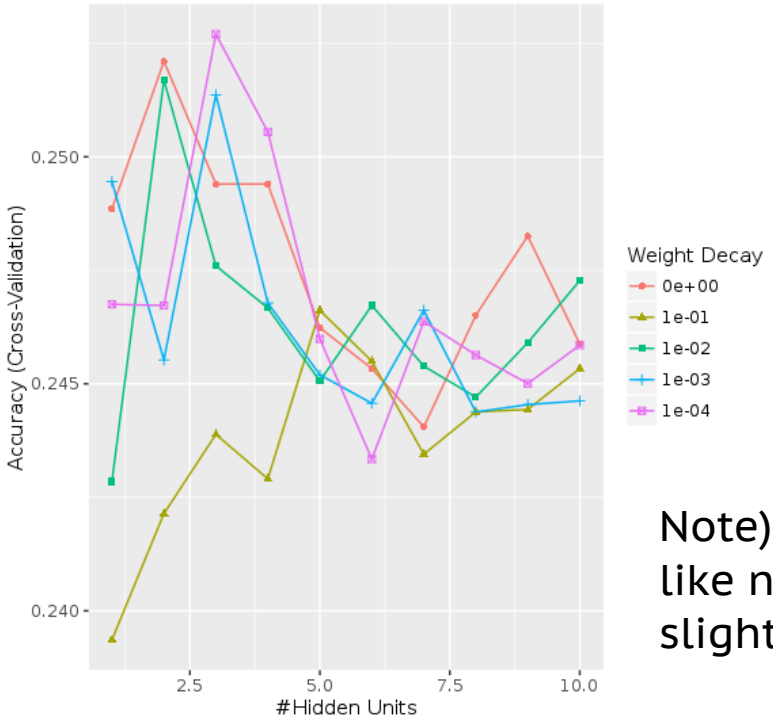
decay (prevent over fitting), size (the number of hidden layers)

Packages (R),

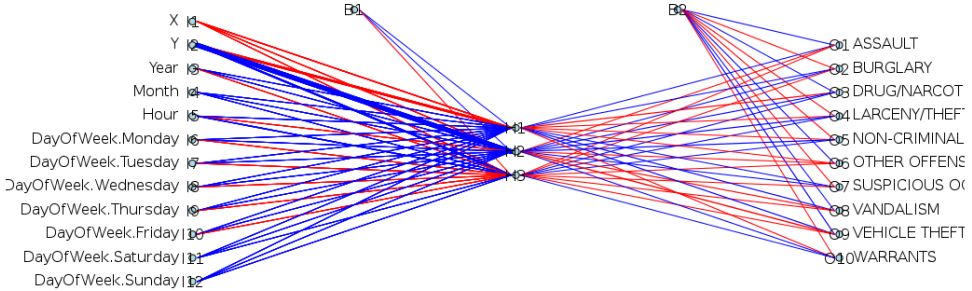
nnet (for neural network), caret (for parameter tuning)

Neural Network (Calculation Results)

Parameter tuning.



Neural network with best parameters.
(size = 3, decay = 1e-4)



Note) If taking into account all 39 levels, NN seemed like not working well. But, prediction power was just slightly lower than 10 levels case.

Prediction power was 0.2085234.

Random Forest (Testing Condition)

Feature of Algorithm,

Classification based on a forest of trees using random input. Random forest can make groups of trees with low correlation by sampling predictors randomly.

Data set for training/test,

train = 4.1 % (treat with top 10 levels of category)
test = 50 % (treat with all 39 levels of category)

Training Method,

10 fold cross validation (1 repeated)

Parameters (search for optimum parameter with grid search),

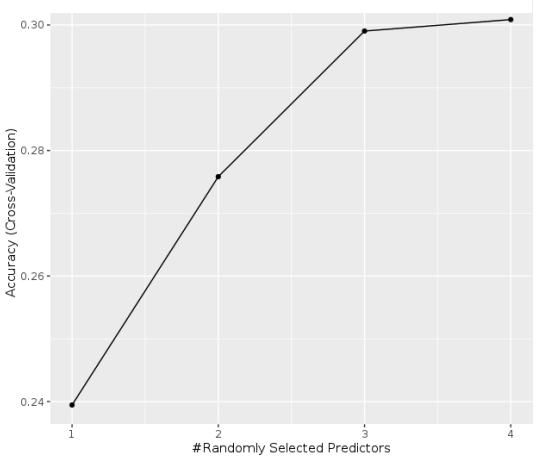
mtry (a random selection of m predictors)

Packages (R),

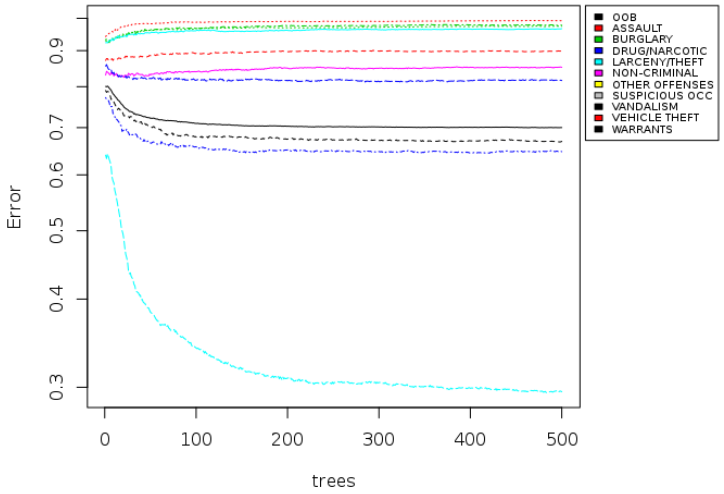
randomForest (for random forest), e1071 & caret (for parameter tuning)

Random Forest (Calculation Results)

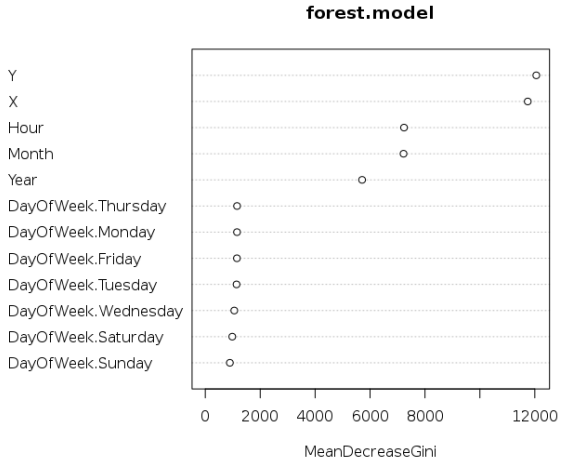
mtry (the number of selected predictors)



ntree Dependency



Paraemter Importance



Optimum mtry was 4, which is close to theoretical value, sqrt(the number of predictors).
Error values were saturated at ntree = 500 (default).
X and Y were important parameters for fitting.

Prediction power for test set was 0.2436467.

Naive Bayes (Testing Condition)

Feature of Algorithm,

Multi class classification is possible based on the assumption of each predictor being independent.

Data set for training/test,

train = 42 % (treat with top 10 levels of category)
test = 50 % (treat with all 39 levels of category)

Parameters (search for optimum parameter with grid search),

use kernel or not

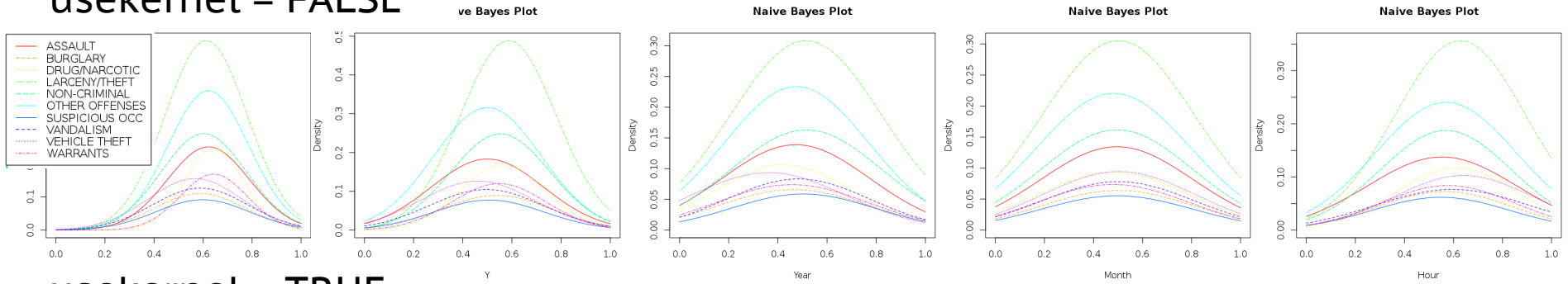
*Note that kernel was used for density estimation (see next page for detail)

Packages (R),

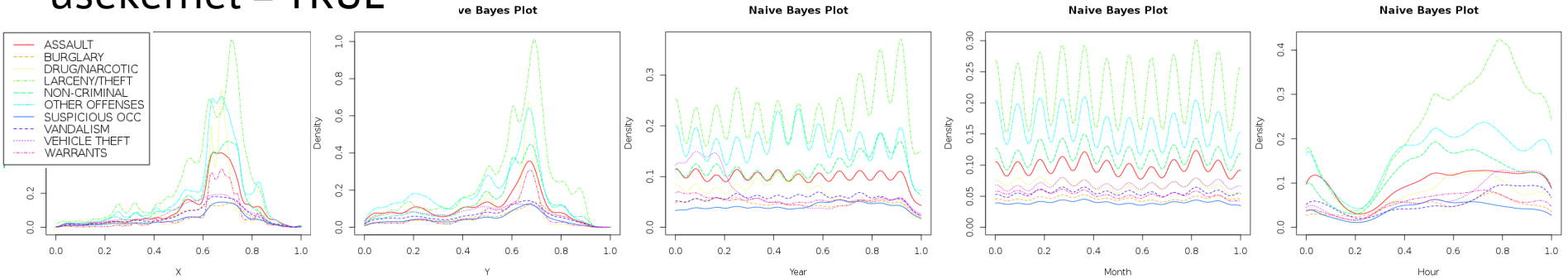
klaR (possible to use a kernel for density estimation and caret package for training), e1071

Naive Bayes (Density Plots)

usekernel = FALSE



usekernel = TRUE



With density estimation, each predictor showed clearer dependency.
Prediction power on test set without density estimation was 0.2096558.
Prediction power on test set with density estimation was 0.212649.

Linear Model (Testing Condition)

Feature of Algorithm,

Multi class classification is possible with **one vs. one** or **one vs. the rest** approach. In LiblineaR package, it seems like using one vs. the rest approach for multi class classification.

Data set for training/test,

train = 42 % (treat with top 10 levels of category)
test = 50 % (treat with all 39 levels of category)

Training Method,

2 fold cross validation (1 repeated)

Parameters (search for optimum parameter with grid search),

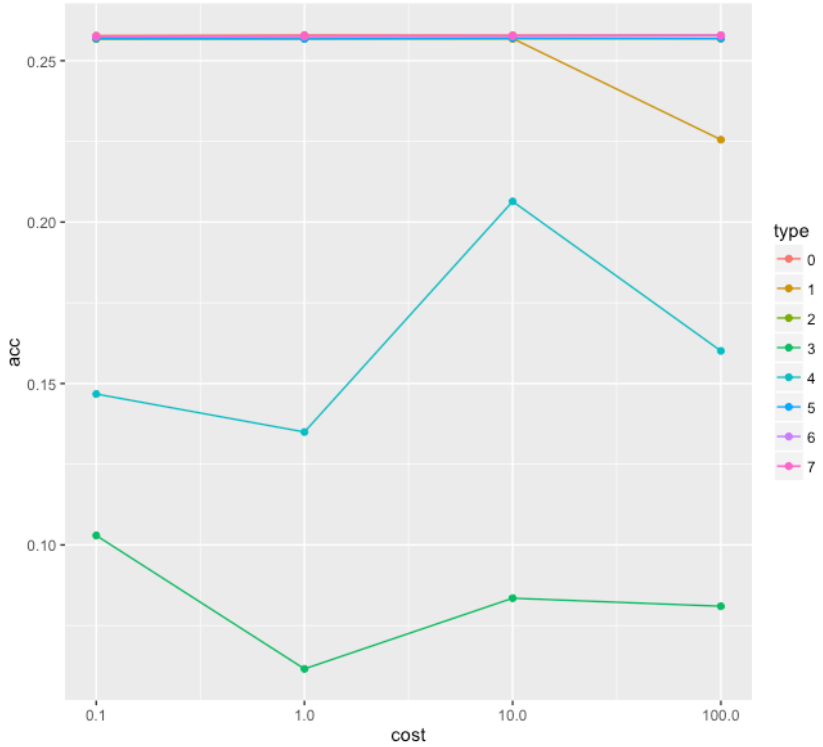
type (L1/L2 regulation, Support Vector Machine/Logistic Regression, see next page for the detail), cost (prevent over fitting), bias = TRUE

Packages (R),

LiblineaR (fast calculation is possible with linear model)

Linear Model (Calculation Results)

Parameter Tuning



 # for multi-class classification

0 – L2-regularized logistic regression (primal)
1 – L2-regularized L2-loss support vector classification (dual)
2 – L2-regularized L2-loss support vector classification (primal)
3 – L2-regularized L1-loss support vector classification (dual)
4 – support vector classification by Crammer and Singer
5 – L1-regularized L2-loss support vector classification
6 – L1-regularized logistic regression
7 – L2-regularized logistic regression (dual)

 # for regression

11 – L2-regularized L2-loss support vector regression (primal)
12 – L2-regularized L2-loss support vector regression (dual)
13 – L2-regularized L1-loss support vector regression (dual)

Prediction power on test set with best model (type = 0, cost =1) was 0.2154942.

Xgboost (Testing Condition)

Feature of Algorithm,

Xgboost (eXtreme Gradient Boosting) is an efficient implementation of gradient boosting framework.

Data set for training/test,

train = 50 % (treat with all 39 levels of category)
test = 50 % (treat with all 39 levels of category)

Training Method,

20 fold cross validation

Parameters,

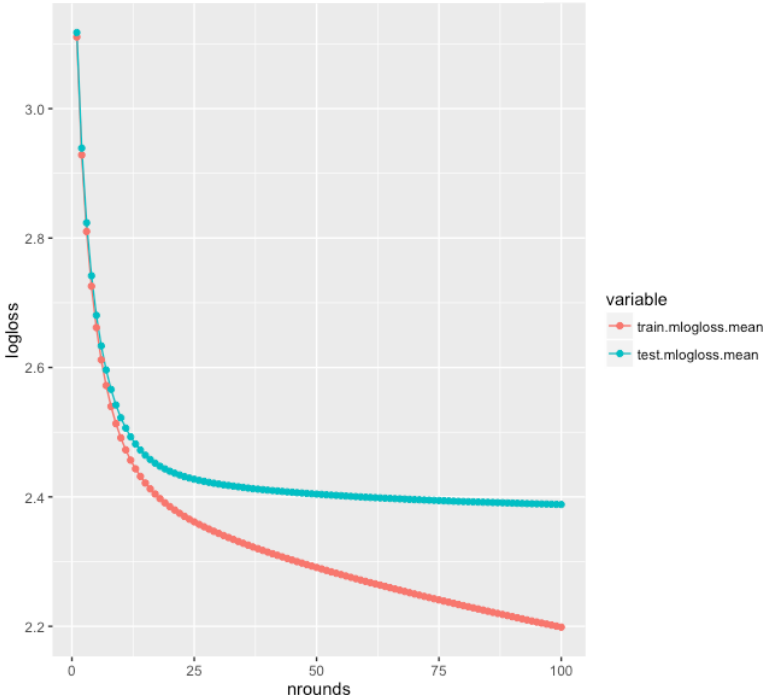
nrounds: the number of decision trees in the final model

Packages (R),

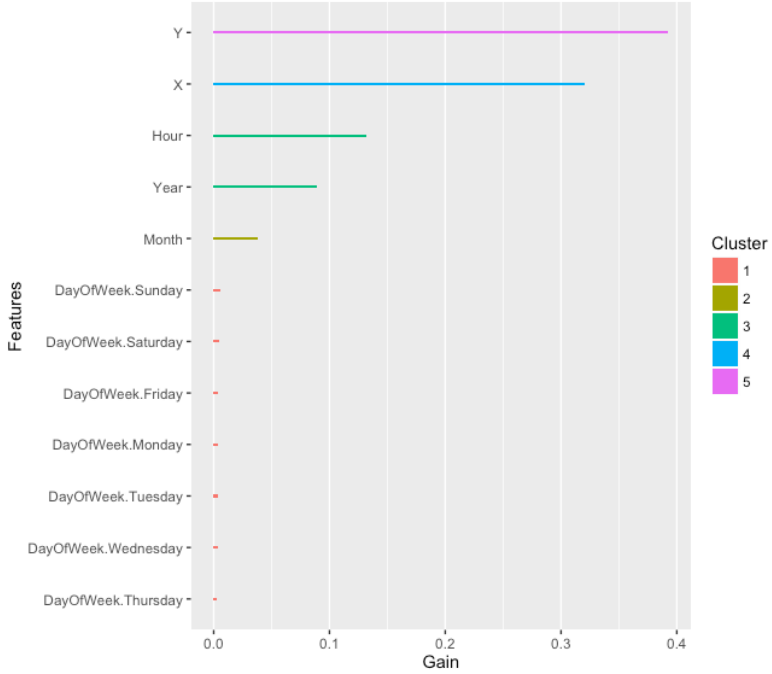
xgboost (the package can automatically do parallel computation on a single machine).

Xgboost (Calculation Results)

Parameter Tuning



Paraemter Importance



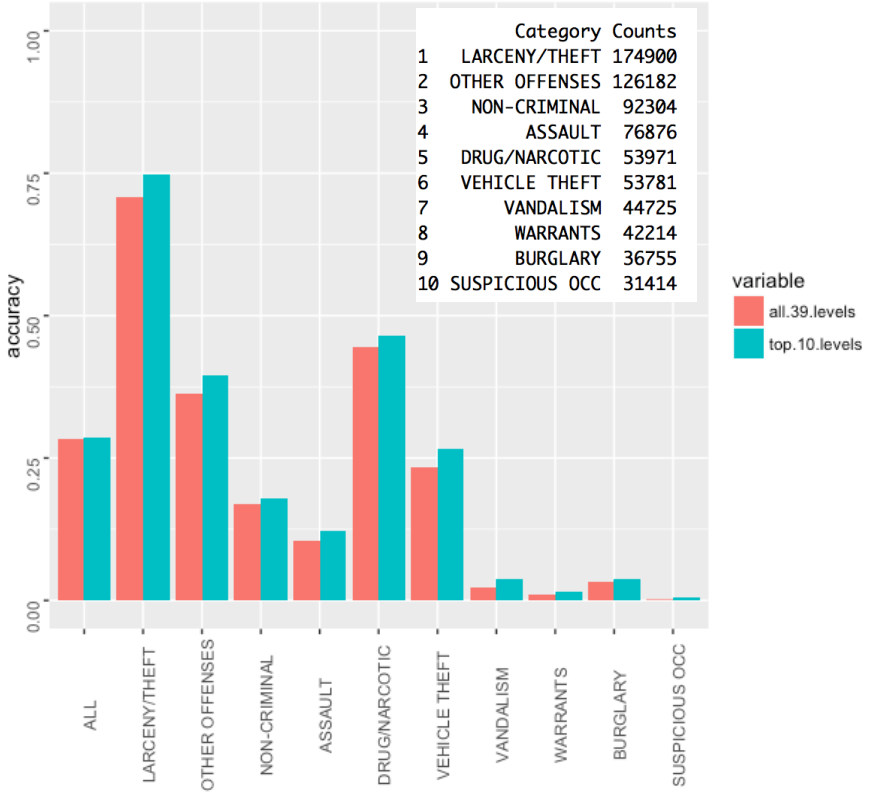
Logloss for test was almost saturated at nrounds = 50.

Parameter importance with xgboost seems to resemble with random forest.

Prediction power on test set with best model (nrounds = 100) was 0.2825001.

Xgboost (all 39 levels to top 10 levels)

Prediction Accuracy for Test Set



	Category	Counts
1	LARCENY/THEFT	174900
2	OTHER OFFENSES	126182
3	NON-CRIMINAL	92304
4	ASSAULT	76876
5	DRUG/NARCOTIC	53971
6	VEHICLE THEFT	53781
7	VANDALISM	44725
8	WARRANTS	42214
9	BURGLARY	36755
10	SUSPICIOUS OCC	31414

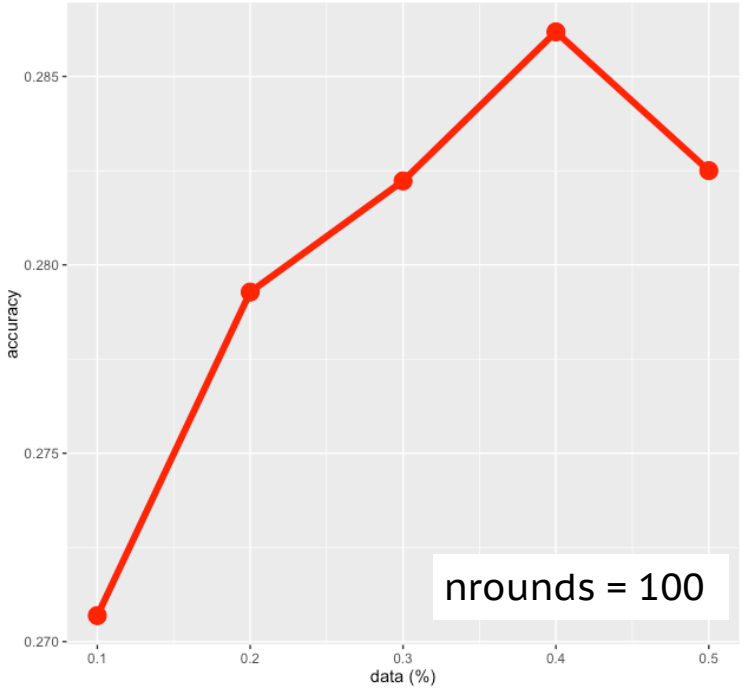
If the number of levels for target of train data set is reduced from 39 to 10 levels, prediction power for test set with all 39 levels was slightly increased from 0.2825001 to 0.2851243.

The prediction accuracy depends on category. The high frequency a type of crime has, the more prediction accuracy we get.

DRUG/NARCOTIC and VEHICLE THEFT look like special (although frequency was lower than ASSULT, NON-CRIMINAL, prediction accuracy was higher). This is because those two types of crime had distinct features on geometric/date information.

Xgboost (data amount for training)

Data Amount Dependency

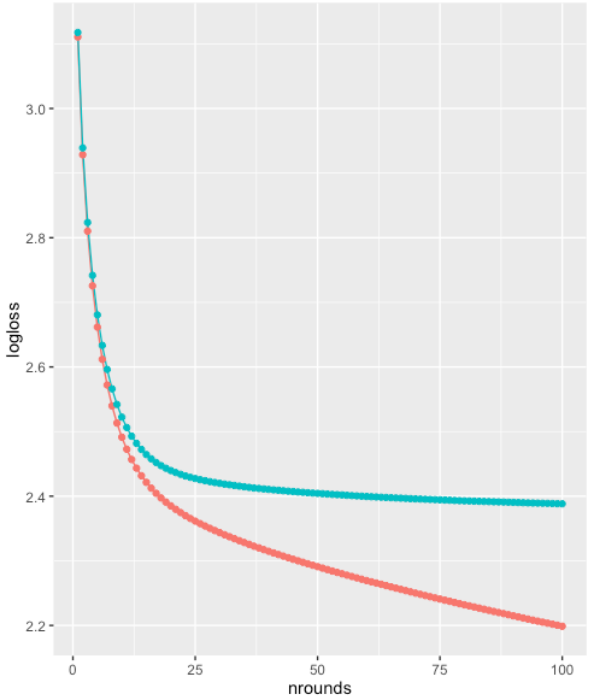


Data amount for training was changed from minimum 10 % to maximum 50 %. Then, prediction accuracy was measured on the rest 50 % test data.

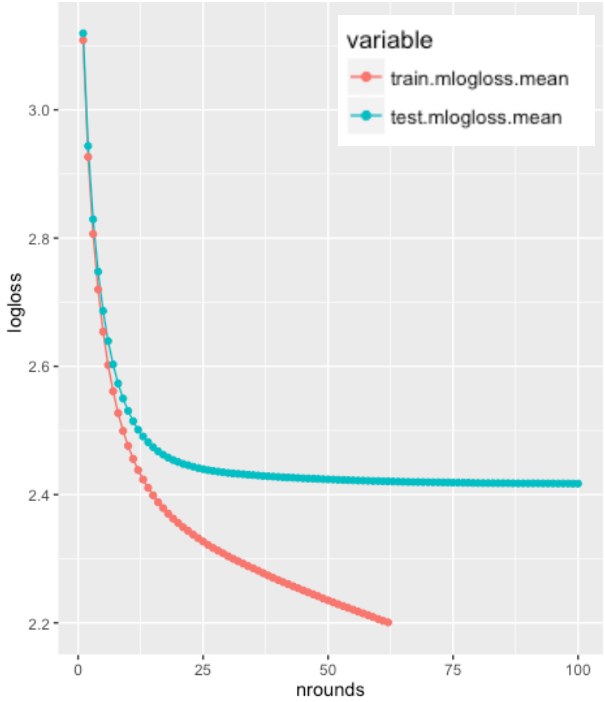
Prediction accuracy was varied within 1.5 % around.

Xgboost (k fold cross validation)

20 fold cross validation



2 fold cross validation



Logloss for test and train were confirmed with 20 fold and 2 fold cross validation.

Although logloss for train looks like different between 2 cases, logloss for test were similar. The final value, at nrouds = 100, was slightly better with 20 fold cross validation.

Summary

Out of selected Algorithms (NN, SVM, RF, NB, LN, Xgboost), Xgboost was the best and Random Forest was the second. The following is my findings„

1. Reducing the number of levels for target was effective to improve prediction power (but slightly) and speed up a calculation especially on one vs. one, one vs. the rest types of multi classification.
2. This time, Naive Bayes classification did't help much. And, the prediction accuracy was higher with random forest even with small volume of train data. It indicates that some predictors might be correlated each other.
3. Parameter Importance was measured with Random Forest and Xgboost. Then, the results were similar (x > y > hour > year > month). DayOfWeek dummy variables were much less important than others. It might be better to treat as ordinal categorical variable.
4. Prediction power was strongly dependent on algorithm rather than how to train the data/the amount of data for training.