CPSC 483

Data Mining and Pattern Recognition

Department of Computer Science

California State University, Fullerton
Spring 2016

Group Project

Submitted by:

Name	Email ID	CWID
Anurag Patsariya	anurag.patsariya@csu.fullerton.edu	803000082
Nitesh Joshi	niteshjoshi@csu.fullerton.edu	802385211
Vidhi Patel	vidhi.patel38@csu.fullerton.edu	893375105

1. Introduction

Kaggle competition is a data science competition where different companies put their data for predication and analysis and data miners from all over the world participate and produce best models using the given data. In Kaggle, we choose a particular competition and we download the training data for the same. Next step is, building a model using our choice of methods or tools. Finally, the predications are uploaded and Kaggle scores the solution, which are then displayed on the leaderboard.
The Kaggle competitions can be found on this url: https://www.kaggle.com/competitions

We had to choose a competition which was active on Kaggle. Therefore, we looked at different competitions and found one which was on San Francisco Crime Classification: https://www.kaggle.com/c/sf-crime

This dataset is brought by SF Open Data, the central clearinghouse for data published by the City and County of San Francisco. This dataset contains incidents derived from SFPD Crime Incident Reporting system. The data ranges from 1/1/2003 to 5/13/2015. The training set and test set rotate every week, meaning week 1,3,5,7... belong to test set, week 2,4,6,8 belong to training set.

1.1 Environment:

Python -3.5

1.2 Libraries:

1. **Pandas: -** It offers data structures and operations for manipulating numerical tables and time series
2. **Sklearn: -** They have all algorithm with their functions.
3. **Numpy: -** It helps in mathematical computation.
4. **Matplotlib: -** This library is used to plot graphs in python, it has inbuilt functions for it.
5. **Math:** - This library is used to access the mathematical function.

1.3 Tools Used:

1.Visual Studio Code.
2. RapidMiner

2. Installation of Python and it libraries

Step 1:
Download python: - **https://www.python.org/downloads/**
Step 2:
Install all the libraries needed:

1. Install pip(This will help us install other libraries)
2. Install numpy (pip install numpy)
3. Install sklearn (pip install sklearn)
4. Install pandas (pip install pandas)
5. Install matplotlib (pip install matplotlib)
6. Install math (pip install math)

3. Download data from Kaggle

Data: https://www.kaggle.com/c/sf-crime/data

- Download all the files on this link.
- Train.csv: - this data will help us to model the data using the algorithm.
- Test.csv: - the modeled algorithm will help us to predict on the test data.
- samplesubmission.csv: - the predicted values need to be stored in the format shown in this file.

Data: -

- Dates - timestamp of thse crime incident
- Category - category of the crime incident (only in train.csv). This is the target variable you are going to predict.
- Descript - detailed description of the crime incident (only in train.csv)
- DayOfWeek - the day of the week
- PdDistrict - name of the Police Department District
- Resolution - how the crime incident was resolved (only in train.csv)
- Address - the approximate street address of the crime incident
- X - Longitude
- Y - Latitude

4. Preprocessing of the Data

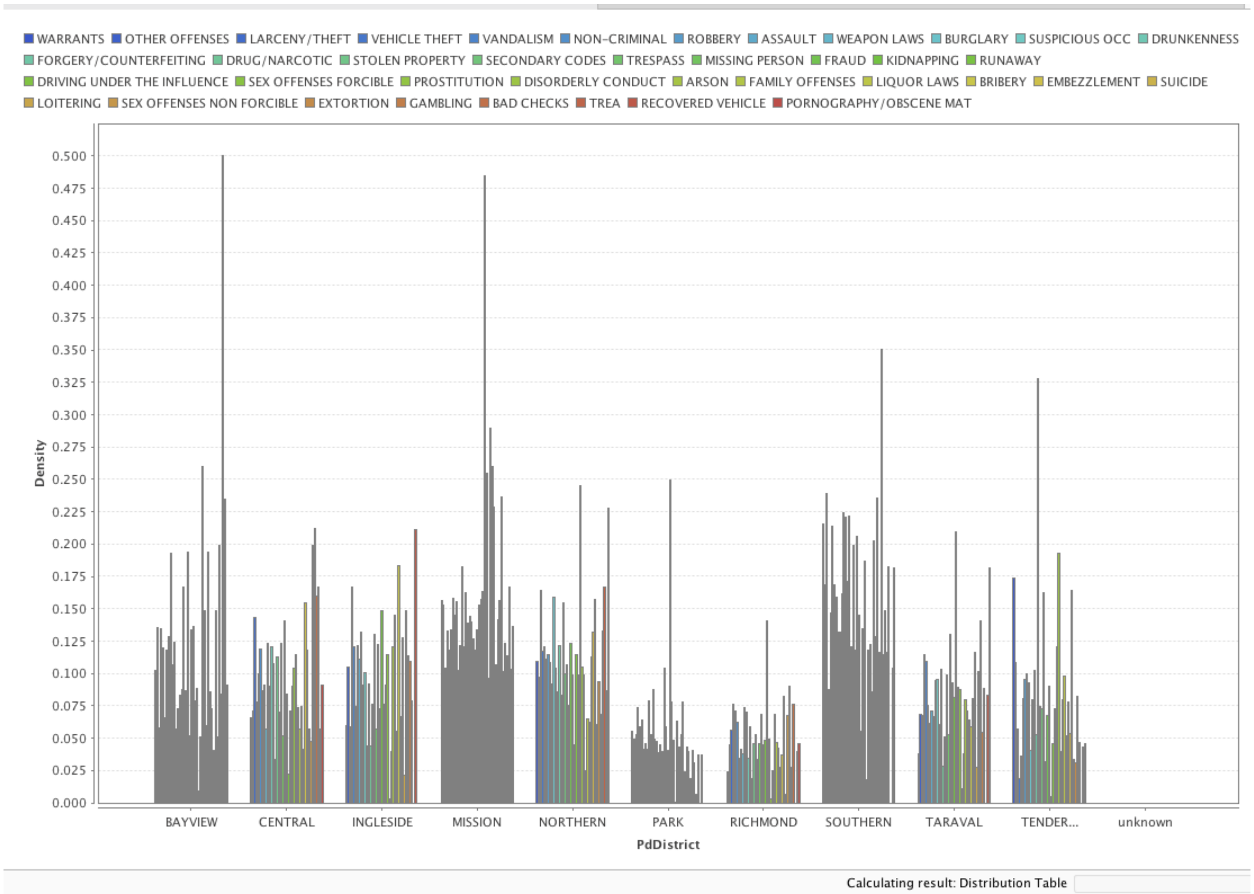
- **Date:** Data we got from kaggle has date in String format. So we used parse_dates to convert into it in date time format. Then we used get_dummies from Pandas library to extract hour.

- **Categories:** We used label_encoder to convert crime categories to unique integer values.
- **Day of week:** We used get_dummies function from Pandas to convert days into binarized array.
- **PdDistrict:** We used get_dummies function from Pandas to convert districts into binarized array.

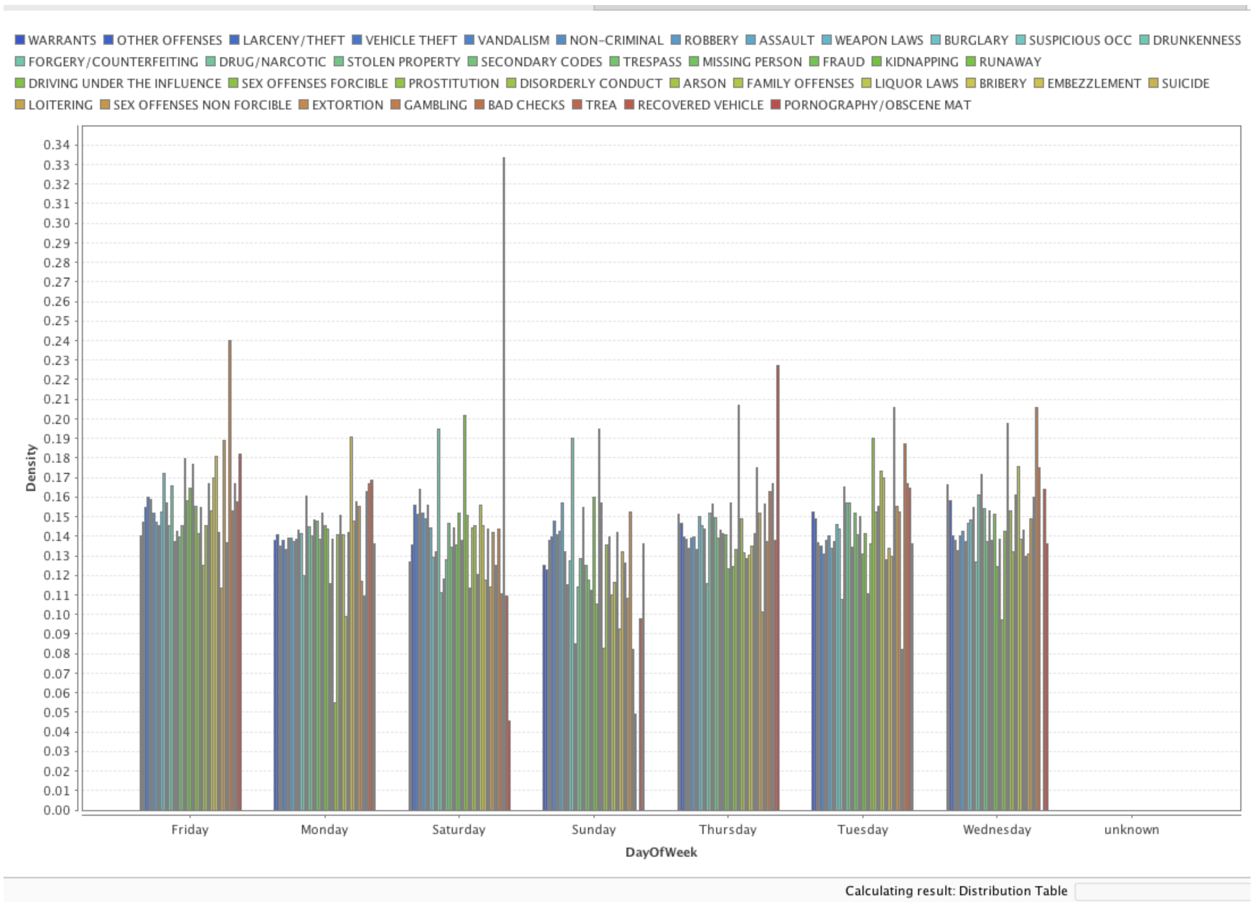
```
import pandas as pd

train =
pd.read_csv("/Users/nikunjpatel/Desktop/DataMiningGroup/train.csv",parse_dates
= ['Dates'])
test = pd.read_csv("/Users/nikunjpatel/Desktop/DataMiningGroup/test.csv",
parse_dates = ['Dates'])

le_crime = preprocessing.LabelEncoder()
crime = le_crime.fit_transform(train.Category)

days = pd.get_dummies(test.DayOfWeek)
district = pd.get_dummies(test.PdDistrict)

hour = test.Dates.dt.hour
hour = pd.get_dummies(hour)
test_data = pd.concat([hour, days, district], axis=1)
```

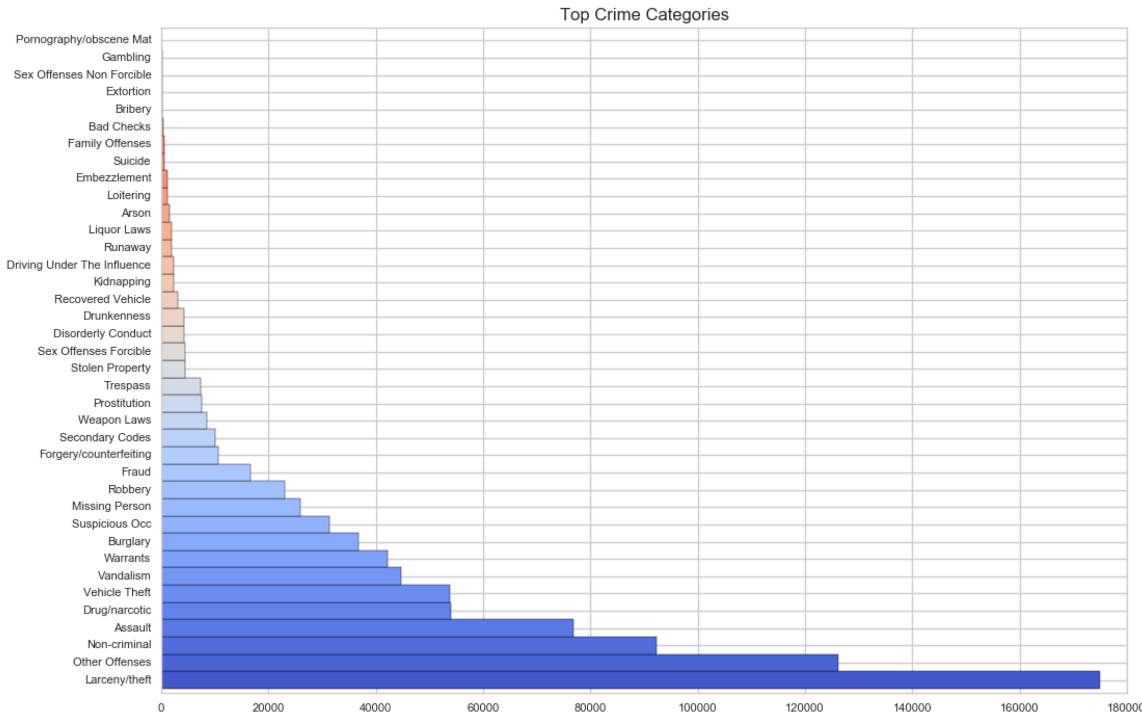
5. Analysis of the data (using RapidMiner)

Here are some graphs produced using RapidMiner for analyzing the data. After analyzing the data, we can say that crime rates are high in Bayview, Southern and Mission districts.



By analyzing data for the week days we can say that crime rate is high on Fridays.


WARRANTS
OTHER OFFENSES
LARCENY/THEFT
VEHICLE THEFT
VANDALISM
NON-CRIMINAL
ROBBERY
ASSAULT
WEAPON LAWS
BURGLARY
SUSPICIOUS OCC
DRUNKENNESS
FORGERY/COUNTERFEITING
DRUG/NARCOTIC
STOLEN PROPERTY
SECONDARY CODES
TRESPASS
MISSING PERSON
FRAUD
KIDNAPPING
RUNAWAY
DRIVING UNDER THE INFLUENCE
SEX OFFENSES FORCIBLE
PROSTITUTION
DISORDERLY CONDUCT
ARSON
FAMILY OFFENSES
LIQUOR LAWS
BRIBERY
EMBEZZLEMENT
SUICIDE
LOITERING
SEX OFFENSES NON FORCIBLE
EXTORTION
GAMBLING
BAD CHECKS
TREA
RECOVERED VEHICLE
PORNOGRAPHY/OBSCENE MAT
Density
0.34
0.33
0.32
0.31
0.30
0.29
0.28
0.27
0.26
0.25
0.24
0.23
0.22
0.21
0.20
0.19
0.18
0.17
0.16
0.15
0.14
0.13
0.12
0.11
0.10
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday
unknown
DayOfWeek
Calculating result: Distribution Table



6. Implemented Algorithms:

1. K nearest neighbor algorithm:

For this algorithm we have considered

- Category
- Dates
- x and y axis.

We took these features so that we might get the exact location, but when we tried it on real algorithm it gave us some astonishing results as follows.

Output:

```
zipfile.ZipFile('/Users/nikunjpatel/Desktop/DataMiningGroup/newData.zip')
test = pd.read_csv(open('test.csv'), parse_dates=['Dates'])
x_test = test[['DayOfWeek', 'PdDistrict']]
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(x, y)
outcomes = knn.predict(x_test)
submit = pd.DataFrame({'Id': test.Id.tolist()})
for category in y.cat.categories:
submit[category] = np.where(outcomes == category, 1, 0)
```

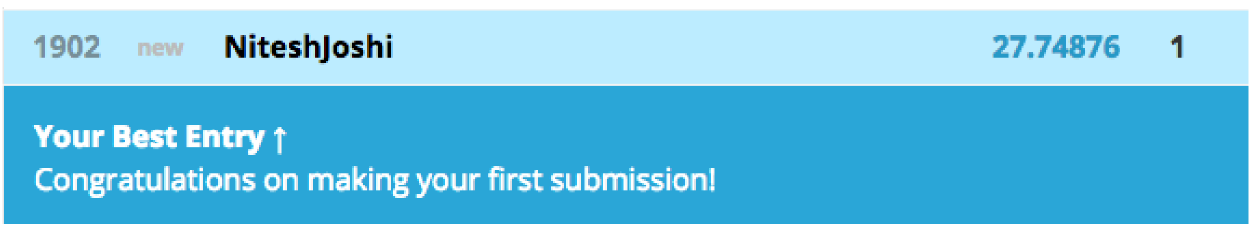
print("Printing")
submit.to_csv('k_nearest_neigbour.csv', index = False)

Output:

A1 fx 135567

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	135567	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	135568	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	135569	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
4	135570	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	135571	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	
6	135572	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
7	135573	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
8	135574	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
9	135575	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
10	135576	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
11	135577	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12	135578	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
13	135579	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
14	135580	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
15	135581	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
16	135582	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
17	135583	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
18	135584	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
19	135585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
20	135586	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
21	135587	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
22	135588	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
23	135589	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
24	135590	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
25	135591	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
26	135592	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
27	135593	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
28	135594	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
29	135595	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
30	135596	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
31	135597	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
32	135598	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
33	135599	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
34	135600	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
35	135601	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
36	135602	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
37	135603	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
38	135604	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
39	135605	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
40	135606	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
41	135607	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	
42	135608	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

k_nearest_neigbour



For KNN we tried changing the K value ranging from 1 to 20. But we got the best results for these features when k=3.

2. Random Forest:

For this algorithm we have considered

- Category
- Dates
- x and y axis.

We took these features so that we might get the exact location, but when we tried it on real algorithm it gave us some astonishing results as follows.

Output:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Id	ARSON	ASSAULT	BAD CHEC	BRIBERY	BURGLARY	DISORDER	DRIVING U	DRUG/NA	DRUNKEN	EMBEZZLE	EXTORTIO	FAMILY OI	FORGERY/	FRAUD	GAMBLIN(	KIDNAPPI	LARCENY/	LIQUOR LA	LOITERING	MISSI
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6	4	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8	6	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
11	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
12	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
13	11	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
14	12	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
15	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
16	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
17	15	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
18	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
19	17	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
20	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
21	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
22	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
23	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

criminals

`

Kaggle Ranking:

1661 new **AnuragPatsariya** 25.02448 1

Your Best Entry ↑
Congratulations on making your first submission!

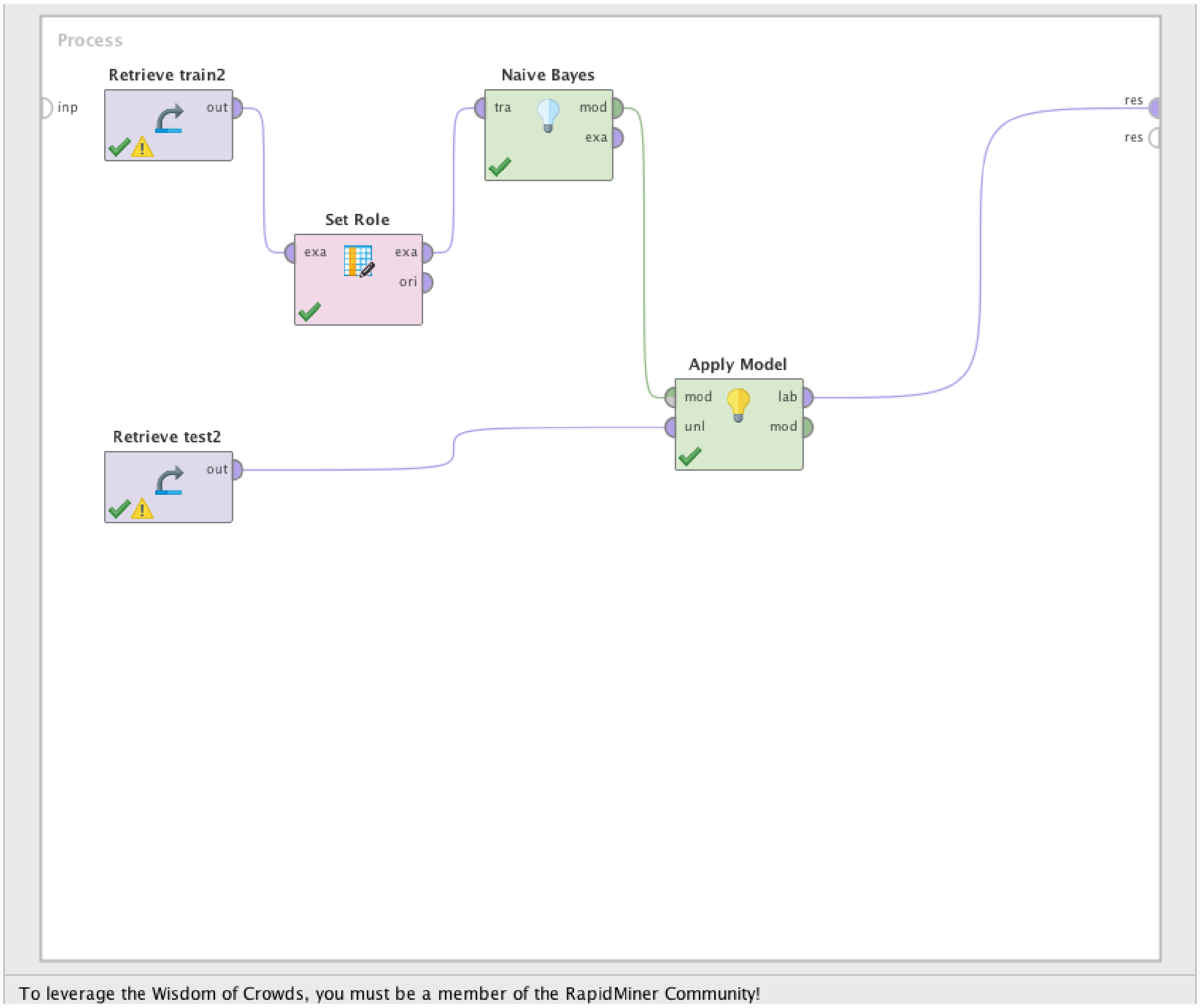
3. Naïve bayes:

For the first run of kaggle submission we used Naïve Bayes with two feature.

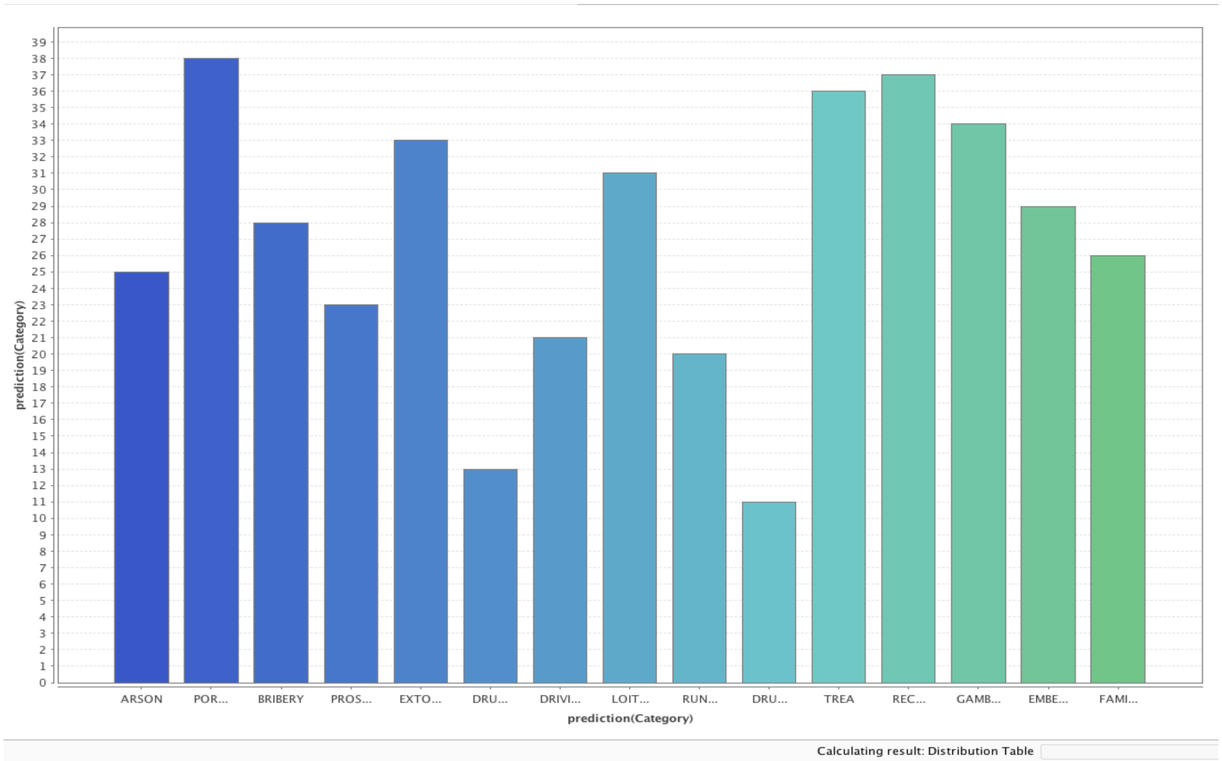
Feature used:

Day of the week: As in our analyzed data we can see that crime rates become high on a specific day so we choose this feature for our algorithm

PdDistrict: Crime rates can be different based on the location, so instead of choosing exact X & Y coordinates for the location we choose PdDistrict so that crime location can be more generalized.



After running the above model for Naïve Bayes algorithm we got the following output.



Below is the small part of code to apply Naïve Bayes algorithm in python:

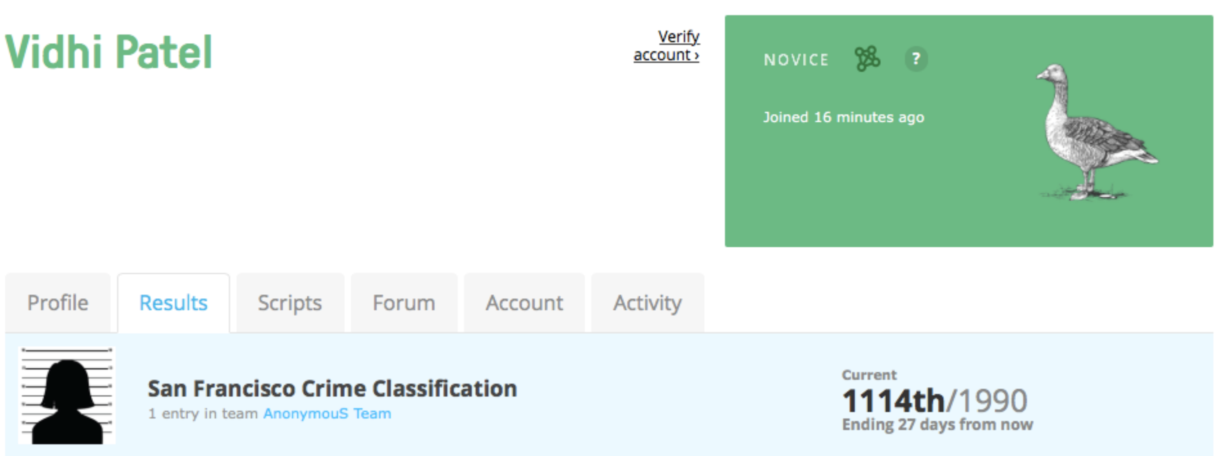
```
import pandas as pd
from sklearn.naive_bayes import BernoulliNB
.
.
features = ['Friday', 'Monday', 'Saturday', 'Sunday', 'Thursday', 'Tuesday',
 'Wednesday', 'BAYVIEW', 'CENTRAL', 'INGLESIDE', 'MISSION',
 'NORTHERN', 'PARK', 'RICHMOND', 'SOUTHERN', 'TARAVAL', 'TENDERLOIN']

model = BernoulliNB()
model.fit(train_data[features], train_data['crime'])
predicted = model.predict_proba(test_data[features])

#Write results

result=pd.DataFrame(predicted, columns=le_crime.classes_)
result.to_csv('NaivetestResult14.csv', index = True, index_label = 'Id' )
```

Output

1	Id	ARSON	ASSAULT	BAD CHECKS	BRIBERY	BURGLARY	DISORDERLY	DRIVING UNI	DRUG/NARC	DRUNKENNE	EMBEZZLEM	EXTORTION	FAMILY OFFE	FORGERY/CC	FRAUD	GAMBLING	KIDNAPPING	LARCENY/TH	LIQUOR LAW	LOITERING	MISSING PEF	NON-CRIN
2	0	0.00433437	0.12730671	0.00090785	0.00113643	0.03749721	0.0033797	0.00351432	0.04304396	0.0047559	0.00150668	0.00090352	0.00126128	0.00620067	0.00893253	0.00086979	0.0043008	0.11594598	0.0019101	0.00137046	0.0490945	0.07152
3	1	0.00433437	0.12730671	0.00090785	0.00113643	0.03749721	0.0033797	0.00351432	0.04304396	0.0047559	0.00150668	0.00090352	0.00126128	0.00620067	0.00893253	0.00086979	0.0043008	0.11594598	0.0019101	0.00137046	0.0490945	0.07152
4	2	0.00204786	0.09210604	0.00092309	0.00086391	0.04732236	0.00477218	0.00385265	0.03688122	0.00542546	0.0015249	0.00089055	0.00086339	0.00838445	0.01658442	0.00072829	0.00312085	0.27455109	0.00177566	0.00209813	0.01682327	0.101016
5	3	0.00273818	0.12383945	0.00094351	0.00117779	0.03572148	0.00322949	0.00411469	0.02600915	0.00453081	0.00146986	0.00103336	0.00132205	0.00838218	0.01323478	0.00084306	0.00507445	0.13137923	0.00199156	0.0012971	0.03493676	0.089677
6	4	0.00273818	0.12383945	0.00094351	0.00117779	0.03572148	0.00322949	0.00411469	0.02600915	0.00453081	0.00146986	0.00103336	0.00132205	0.00838218	0.01323478	0.00084306	0.00507445	0.13137923	0.00199156	0.0012971	0.03493676	0.089677
7	5	0.00249224	0.09651136	0.00105083	0.00104772	0.0444494	0.00359541	0.00464787	0.02079037	0.00621582	0.00175195	0.00116733	0.00120662	0.01102311	0.0196849	0.00086424	0.00389889	0.18204965	0.00237634	0.0014852	0.04452659	0.108639
8	6	0.00273818	0.12383945	0.00094351	0.00117779	0.03572148	0.00322949	0.00411469	0.02600915	0.00453081	0.00146986	0.00103336	0.00132205	0.00838218	0.01323478	0.00084306	0.00507445	0.13137923	0.00199156	0.0012971	0.03493676	0.089677
9	7	0.00273818	0.12383945	0.00094351	0.00117779	0.03572148	0.00322949	0.00411469	0.02600915	0.00453081	0.00146986	0.00103336	0.00132205	0.00838218	0.01323478	0.00084306	0.00507445	0.13137923	0.00199156	0.0012971	0.03493676	0.089677
10	8	0.00187327	0.10932621	0.00083944	0.00103589	0.02721181	0.00918776	0.00437283	0.06317575	0.00891584	0.00138337	0.00085516	0.00142377	0.00744059	0.01423426	0.00072602	0.00346065	0.15670468	0.0038647	0.00192562	0.02452111	0.108425
11	9	0.00204043	0.09384631	0.00107634	0.00090976	0.04473578	0.00603126	0.00335944	0.01893544	0.00816045	0.00204943	0.00112173	0.00102782	0.00909851	0.02281514	0.00088909	0.00327706	0.29225719	0.0021383	0.00152478	0.01449703	0.130824
12	10	0.00273818	0.12383945	0.00094351	0.00117779	0.03572148	0.00322949	0.00411469	0.02600915	0.00453081	0.00146986	0.00103336	0.00132205	0.00838218	0.01323478	0.00084306	0.00507445	0.13137923	0.00199156	0.0012971	0.03493676	0.089677
13	11	0.00187327	0.10932621	0.00083944	0.00103589	0.02721181	0.00918776	0.00437283	0.06317575	0.00891584	0.00138337	0.00085516	0.00142377	0.00744059	0.01423426	0.00072602	0.00346065	0.15670468	0.0038647	0.00192562	0.02452111	0.108425
14	12	0.00187327	0.10932621	0.00083944	0.00103589	0.02721181	0.00918776	0.00437283	0.06317575	0.00891584	0.00138337	0.00085516	0.00142377	0.00744059	0.01423426	0.00072602	0.00346065	0.15670468	0.0038647	0.00192562	0.02452111	0.108425
15	13	0.00204043	0.09384631	0.00107634	0.00090976	0.04473578	0.00603126	0.00335944	0.01893544	0.00816045	0.00204943	0.00112173	0.00102782	0.00909851	0.02281514	0.00088909	0.00327706	0.29225719	0.0021383	0.00152478	0.01449703	0.130824
16	14	0.00204043	0.09384631	0.00107634	0.00090976	0.04473578	0.00603126	0.00335944	0.01893544	0.00816045	0.00204943	0.00112173	0.00102782	0.00909851	0.02281514	0.00088909	0.00327706	0.29225719	0.0021383	0.00152478	0.01449703	0.130824
17	15	0.00175839	0.11097109	0.0008865	0.00096704	0.01672419	0.01003883	0.00298841	0.18946401	0.00777007	0.00155743	0.00092621	0.00113692	0.00525531	0.01231074	0.00083247	0.00362646	0.12604088	0.00260257	0.00258355	0.01036278	0.09660
18	16	0.00249224	0.09651136	0.00105083	0.00104772	0.0444494	0.00359541	0.00464787	0.02079037	0.00621582	0.00175195	0.00116733	0.00120662	0.01102311	0.0196849	0.00086424	0.00389889	0.18204965	0.00237634	0.0014852	0.04452659	0.108639
19	17	0.00273818	0.12383945	0.00094351	0.00117779	0.03572148	0.00322949	0.00411469	0.02600915	0.00453081	0.00146986	0.00103336	0.00132205	0.00838218	0.01323478	0.00084306	0.00507445	0.13137923	0.00199156	0.0012971	0.03493676	0.089677
20	18	0.00204786	0.09210604	0.00092309	0.00086391	0.04732236	0.00477218	0.00385265	0.03688122	0.00542546	0.0015249	0.00089055	0.00086339	0.00838445	0.01658442	0.00072829	0.00312085	0.27455109	0.00177566	0.00209813	0.01682327	0.101016
21	19	0.00204786	0.09210604	0.00092309	0.00086391	0.04732236	0.00477218	0.00385265	0.03688122	0.00542546	0.0015249	0.00089055	0.00086339	0.00838445	0.01658442	0.00072829	0.00312085	0.27455109	0.00177566	0.00209813	0.01682327	0.101016
22	20	0.00204786	0.09210604	0.00092309	0.00086391	0.04732236	0.00477218	0.00385265	0.03688122	0.00542546	0.0015249	0.00089055	0.00086339	0.00838445	0.01658442	0.00072829	0.00312085	0.27455109	0.00177566	0.00209813	0.01682327	0.101016
23	21	0.00187327	0.10932621	0.00083944	0.00103589	0.02721181	0.00918776	0.00437283	0.06317575	0.00891584	0.00138337	0.00085516	0.00142377	0.00744059	0.01423426	0.00072602	0.00346065	0.15670468	0.0038647	0.00192562	0.02452111	0.108425
24	22	0.00204043	0.09384631	0.00107634	0.00090976	0.04473578	0.00603126	0.00335944	0.01893544	0.00816045	0.00204943	0.00112173	0.00102782	0.00909851	0.02281514	0.00088909	0.00327706	0.29225719	0.0021383	0.00152478	0.01449703	0.130824
25	23	0.00273818	0.12383945	0.00094351	0.00117779	0.03572148	0.00322949	0.00411469	0.02600915	0.00453081	0.00146986	0.00103336	0.00132205	0.00838218	0.01323478	0.00084306	0.00507445	0.13137923	0.00199156	0.0012971	0.03493676	0.089677
26	24	0.00204786	0.09210604	0.00092309	0.00086391	0.04732236	0.00477218	0.00385265	0.03688122	0.00542546	0.0015249	0.00089055	0.00086339	0.00838445	0.01658442	0.00072829	0.00312085	0.27455109	0.00177566	0.00209813	0.01682327	0.101016
27	25	0.00286264	0.08177423	0.00117928	0.00112316	0.04954996	0.00381874	0.00825132	0.02012886	0.00585411	0.00165614	0.00123341	0.00130139	0.00921918	0.02070057	0.00094419	0.00361706	0.21708031	0.00248283	0.00151534	0.02311128	0.128639
28	26	0.00204786	0.09210604	0.00092309	0.00086391	0.04732236	0.00477218	0.00385265	0.03688122	0.00542546	0.0015249	0.00089055	0.00086339	0.00838445	0.01658442	0.00072829	0.00312085	0.27455109	0.00177566	0.00209813	0.01682327	0.101016
29	27	0.00172173	0.09047621	0.00080451	0.00077467	0.02646283	0.0037375	0.00313658	0.05033009	0.00825543	0.00173634	0.00077856	0.00080538	0.01001748	0.01846519	0.00062614	0.00292313	0.26932912	0.00251588	0.00271443	0.01743187	0.130109
30	28	0.00273818	0.12383945	0.00094351	0.00117779	0.03572148	0.00322949	0.00411469	0.02600915	0.00453081	0.00146986	0.00103336	0.00132205	0.00838218	0.01323478	0.00084306	0.00507445	0.13137923	0.00199156	0.0012971	0.03493676	0.089677
31	29	0.00249224	0.09651136	0.00105083	0.00104772	0.0444494	0.00359541	0.00464787	0.02079037	0.00621582	0.00175195	0.00116733	0.00120662	0.01102311	0.0196849	0.00086424	0.00389889	0.18204965	0.00237634	0.0014852	0.04452659	0.108639
32	30	0.00249224	0.09651136	0.00105083	0.00104772	0.0444494	0.00359541	0.00464787	0.02079037	0.00621582	0.00175195	0.00116733	0.00120662	0.01102311	0.0196849	0.00086424	0.00389889	0.18204965	0.00237634	0.0014852	0.04452659	0.108639
33	31	0.00273818	0.12383945	0.00094351	0.00117779	0.03572148	0.00322949	0.00411469	0.02600915	0.00453081	0.00146986	0.00103336	0.00132205	0.00838218	0.01323478	0.00084306	0.00507445	0.13137923	0.00199156	0.0012971	0.03493676	0.089677
34	32	0.00249224	0.09651136	0.00105083	0.00104772	0.0444494	0.00359541	0.00464787	0.02079037	0.00621582	0.00175195	0.00116733	0.00120662	0.01102311	0.0196849	0.00086424	0.00389889	0.18204965	0.00237634	0.0014852	0.04452659	0.108639
35	33	0.00204786	0.09210604	0.00092309	0.00086391	0.04732236	0.00477218	0.00385265	0.03688122	0.00542546	0.0015249	0.00089055	0.00086339	0.00838445	0.01658442	0.00072829	0.00312085	0.27455109	0.00177566	0.00209813	0.01682327	0.101016
36	34	0.00433437	0.12730671	0.00090785	0.00113643	0.03749721	0.0033797	0.00351432	0.04304396	0.0047559	0.00150668	0.00090352	0.00126128	0.00620067	0.00893253	0.00086979	0.0043008	0.11594598	0.0019101	0.00137046	0.0490945	0.07152
37	35	0.00286264	0.08177423	0.00117928	0.00112316	0.04954996	0.00381874	0.00825132	0.02012886	0.00585411	0.00165614	0.00123341	0.00130139	0.00921918	0.02070057	0.00094419	0.00361706	0.21708031	0.00248283	0.00151534	0.02311128	0.128639
38	36	0.00172173	0.09047621	0.00080451	0.00077467	0.02646283	0.0037375	0.00313658	0.05033009	0.00825543	0.00173634	0.00077856	0.00080538	0.01001748	0.01846519	0.00062614	0.00292313	0.26932912	0.00251588	0.00271443	0.01743187	0.130109
39	37	0.00233057	0.08329555	0.00104727	0.00109747	0.04932257	0.00596953	0.00519632	0.04444741	0.01001602	0.00166805	0.00109993	0.00129765	0.00749929	0.01684042	0.00091194	0.00334295	0.18710402	0.00314973	0.00161022	0.04730424	0.123473
40	38	0.00249224	0.09651136	0.00105083	0.00104772	0.0444494	0.00359541	0.00464787	0.02079037	0.00621582	0.00175195	0.00116733	0.00120662	0.01102311	0.0196849	0.00086424	0.00389889	0.18204965	0.00237634	0.0014852	0.04452659	0.108639
41	39	0.00249224	0.09651136	0.00105083	0.00104772	0.0444494	0.00359541	0.00464787	0.02079037	0.00621582	0.00175195	0.00116733	0.00120662	0.01102311	0.0196849	0.00086424	0.00389889	0.18204965	0.00237634	0.0014852	0.04452659	0.108639
42	40	0.00286264	0.08177423	0.00117928	0.00112316	0.04954996	0.00381874	0.00825132	0.02012886	0.00585411	0.00165614	0.00123341	0.00130139	0.00921918	0.02070057	0.00094419	0.00361706	0.21708031	0.00248283	0.00151534	0.02311128	0.128639

NaivetestResult

Kaggle Ranking on submission:



After getting the above results we tried the same code for three features given below:

- **hour**
- **day of week**
- **PdDistrict**

For the second run, we used time, days of Week and PdDistrict.
This time we choose “Time” as a feature because there are possibilities of happening certain crimes at a particular time. e.g. drunk and drive happens specially in the night.

We changed the python code for Naïve Bayes as shown below:

```
import pandas as pd
from sklearn.naive_bayes import BernoulliNB
.
.
features = ['Friday', 'Monday', 'Saturday', 'Sunday', 'Thursday', 'Tuesday',
 'Wednesday', 'BAYVIEW', 'CENTRAL', 'INGLESIDE', 'MISSION',
 'NORTHERN', 'PARK', 'RICHMOND', 'SOUTHERN', 'TARAVAL', 'TENDERLOIN']

features2 = [x for x in range(0,24)]
features = features + features2

model = BernoulliNB()
model.fit(train_data[features], train_data['crime'])
predicted = model.predict_proba(test_data[features])

#Write results

result=pd.DataFrame(predicted, columns=le_crime.classes_)
result.to_csv('NaivetestResult14.csv', index = True, index_label = 'Id' )
```

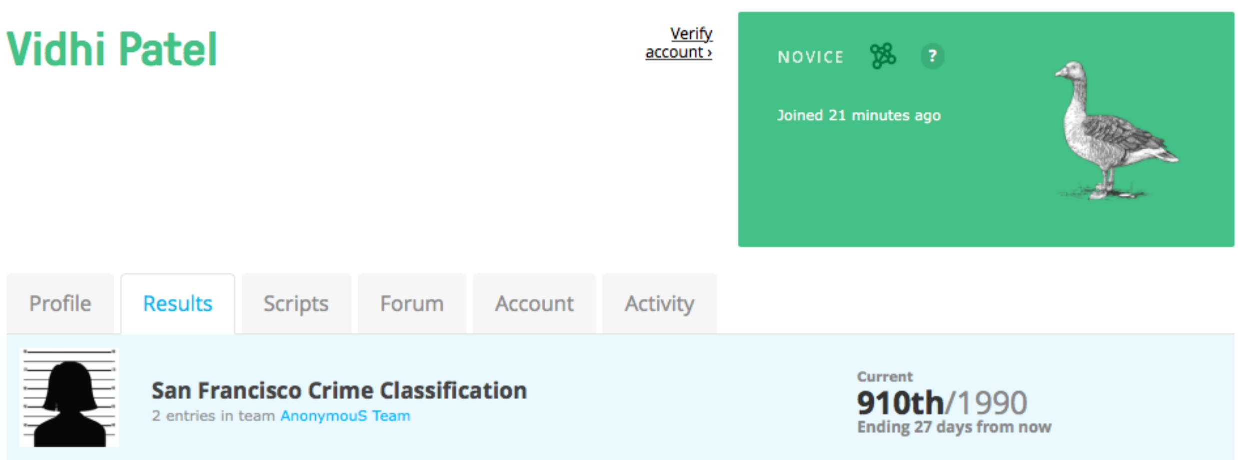
Here we created sparse matrix of days of week, time and PdDistrict.

Output from algorithm:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	166479	0.00075236	0.06780536	0.00075693	0.00024122	0.02343186	0.00208021	0.00033851	0.08463321	0.00354086	0.00248304	0.00019104	0.00035839	0.02000119	0.02335886	0.00012866	0.00186903	0.27392563	0.00127973	0.0036726	0.01912998	0.14345
2	166480	0.00134825	0.07959297	9.68E-04	0.00030596	0.04643132	0.00175296	0.00061422	0.03463523	0.00228864	0.00205292	5.19E-04	0.00085369	0.02349983	0.02659312	1.74E-04	0.00283007	0.18283419	0.00100437	0.00057752	0.0620356	0.1234
3	166481	0.00174951	0.10649279	7.01E-04	0.00086682	0.03647037	0.00151975	0.00051447	0.04471881	0.0013551	0.00137901	0.00033437	0.00151945	0.01694641	0.01670469	2.86E-04	0.00454942	0.12826685	0.00070993	0.00036281	0.04728639	0.10016
4	166482	0.00377742	0.1058737	6.30E-04	0.00078213	0.03697664	0.0016598	0.00037269	0.07459772	0.00148022	0.00151866	1.23E-04	1.32E-03	0.01109991	0.00984779	4.72E-04	0.003395	0.1069423	0.00064684	0.00059019	0.06880133	0.07457
5	166483	0.00377742	0.1058737	6.30E-04	0.00078213	0.03697664	0.0016598	0.00037269	0.07459772	0.00148022	0.00151866	0.00012306	0.00132431	0.01109991	0.00984779	4.72E-04	0.003395	0.1069423	0.00064684	0.00059019	0.06880133	0.07457
6	166484	0.00075236	0.06780536	7.57E-04	0.00024122	0.02343186	0.00208021	0.00033851	0.08463321	0.00354086	0.00248304	0.00019104	0.00035839	0.02000119	0.02335886	1.29E-04	0.00186903	0.27392563	0.00127973	0.0036726	0.01912998	0.14345
7	166485	0.00075236	0.06780536	7.57E-04	0.00024122	0.02343186	0.00208021	0.00033851	0.08463321	0.00354086	0.00248304	0.00019104	0.00035839	0.02000119	0.02335886	1.29E-04	0.00186903	0.27392563	0.00127973	0.0036726	0.01912998	0.14345
8	166486	0.00170865	0.06749671	1.23E-03	0.00022103	0.05273765	0.00172705	0.00153172	0.03340212	0.00191812	0.00135467	0.00047447	0.00082766	0.0188565	0.02836685	1.40E-04	0.00222005	0.22521005	0.00097684	0.0002169	0.02945083	0.15087
9	166487	0.00170865	0.06749671	1.23E-03	0.00022103	0.05273765	0.00172705	0.00153172	0.03340212	0.00191812	0.00135467	0.00047447	0.00082766	0.0188565	0.02836685	1.40E-04	0.00222005	0.22521005	0.00097684	0.0002169	0.02945083	0.15087
10	166488	0.00075236	0.06780536	7.57E-04	0.00024122	0.02343186	0.00208021	0.00033851	0.08463321	0.00354086	0.00248304	0.00019104	0.00035839	0.02000119	0.02335886	1.29E-04	0.00186903	0.27392563	0.00127973	0.0036726	0.01912998	0.14345
11	166489	0.00134825	0.07959297	9.68E-04	0.00030596	0.04643132	0.00175296	0.00061422	0.03463523	0.00228864	0.00205292	5.19E-04	0.00085369	0.02349983	0.02659312	0.00017384	0.00283007	0.18283419	0.00100437	0.00057752	0.0620356	0.1234
12	166490	0.0008041	0.07743517	8.38E-04	0.00013468	0.04355588	0.00455303	0.00028435	0.02719824	0.003149	0.00215092	5.89E-04	0.0005239	0.02083897	0.03654073	5.99E-04	0.00194606	0.30859944	0.00102324	0.00073407	0.01558508	0.15561
13	166491	0.0008041	0.07743517	8.38E-04	0.00013468	0.04355588	0.00455303	0.00028435	0.02719824	0.003149	0.00215092	5.89E-04	0.0005239	0.02083897	0.03654073	5.99E-04	0.00194606	0.30859944	0.00102324	0.00073407	0.01558508	0.15561
14	166492	0.00074059	0.09071629	3.90E-04	0.00059389	0.02430236	0.00828509	0.00048081	0.10491021	0.00353009	0.00096909	0.00018807	0.00236108	0.01632174	0.02063539	2.21E-04	0.002241	0.14795833	0.00308311	0.00152854	0.02838526	0.12349
15	166493	0.00066383	0.06879063	4.61E-04	0.0002037	0.02219278	0.0022275	0.00027431	0.07630583	0.00311374	0.00170894	0.00019571	0.00037717	0.02302721	0.02734871	1.42E-04	0.00169093	0.27325353	0.00152017	0.00303647	0.01792232	0.14829
16	166494	0.00066383	0.06879063	4.61E-04	0.0002037	0.02219278	0.0022275	0.00027431	0.07630583	0.00311374	0.00170894	0.00019571	0.00037717	0.02302721	0.02734871	1.42E-04	0.00169093	0.27325353	0.00152017	0.00303647	0.01792232	0.14829
17	166495	0.0008041	0.07743517	8.38E-04	0.00013468	0.04355588	0.00455303	0.00028435	0.02719824	0.003149	0.00215092	0.00058914	0.0005239	0.02083897	0.03654073	5.99E-04	0.00194606	0.30859944	0.00102324	0.00073407	0.01558508	0.15561
18	166496	0.0008041	0.07743517	8.38E-04	0.00013468	0.04355588	0.00455303	0.00028435	0.02719824	0.003149	0.00215092	0.00058914	0.0005239	0.02083897	0.03654073	5.99E-04	0.00194606	0.30859944	0.00102324	0.00073407	0.01558508	0.15561
19	166497	0.00066383	0.06879063	4.61E-04	0.0002037	0.02219278	0.0022275	0.00027431	0.07630583	0.00311374	0.00170894	0.00019571	0.00037717	0.02302721	0.02734871	1.42E-04	0.00169093	0.27325353	0.00152017	0.00303647	0.01792232	0.14829
20	166498	0.00336454	0.10843058	3.87E-04	0.00066672	0.03535338	0.00179417	0.00030487	0.06789548	0.00131401	0.00105512	1.27E-04	1.41E-03	0.0129004	0.01163919	5.25E-04	0.00310062	0.1076914	0.00077566	4.93E-04	0.06506911	0.07782
21	166499	0.00155809	0.109051	4.30E-04	0.00073882	0.03486498	0.00164258	0.00042079	0.04069596	0.00120279	0.00095798	0.00034575	1.61E-03	0.0196928	0.01974094	3.17E-04	0.00415442	0.1291492	0.0008512	3.03E-04	0.04471569	0.10450
22	166500	0.00066383	0.06879063	4.61E-04	0.0002037	0.02219278	0.0022275	0.00027431	0.07630583	0.00311374	0.00170894	1.96E-04	3.77E-04	0.02302721	0.02734871	1.42E-04	0.00169093	0.27325353	0.00152017	0.00303647	0.01792232	0.14829
23	166501	0.00074059	0.09071629	3.90E-04	5.94E-04	0.02430236	0.00828509	0.00048081	0.10491021	0.00353009	0.00096909	0.00018807	0.00236108	0.01632174	0.02063539	2.21E-04	0.002241	0.14795833	0.00308311	0.00152854	0.02838526	0.12349
24	166502	0.0008041	0.07743517	8.38E-04	0.00013468	0.04355588	0.00455303	0.00028435	0.02719824	0.003149	0.00215092	0.00058914	0.0005239	0.02083897	0.03654073	5.99E-04	0.00194606	0.30859944	0.00102324	0.00073407	0.01558508	0.15561
25	166503	0.00336454	0.10843058	3.87E-04	0.00066672	0.03535338	0.00179417	0.00030487	0.06789548	0.00131401	0.00105512	0.00012726	0.00140691	0.0129004	0.01163919	5.25E-04	0.00310062	0.1076914	0.00077566	4.93E-04	0.06506911	0.07782
26	166504	0.00074059	0.09071629	3.90E-04	0.00059389	0.02430236	0.00828509	0.00048081	0.10491021	0.00353009	0.00096909	1.88E-04	2.36E-03	0.01632174	0.02063539	2.21E-04	0.002241	0.14795833	0.00308311	0.00152854	0.02838526	0.12349
27	166505	0.00031741	0.08269673	8.39E-04	0.00011918	0.01655414	0.00808479	0.00012838	0.23358442	0.00209851	0.00493716	0.00013204	0.00040616	0.02038139	0.03196659	1.49E-04	0.0021563	0.10409013	0.00116947	0.00187037	0.01032196	0.10861
28	166506	0.00031741	0.08269673	8.39E-04	0.00011918	0.01655414	0.00808479	0.00012838	0.23358442	0.00209851	0.00493716	1.32E-04	4.06E-04	0.02038139	0.03196659	1.49E-04	0.0021563	0.10409013	0.00116947	0.00187037	0.01032196	0.10861
29	166507	0.00031741	0.08269673	8.39E-04	0.00011918	0.01655414	0.00808479	0.00012838	0.23358442	0.00209851	0.00493716	1.32E-04	0.00040616	0.02038139	0.03196659	1.49E-04	0.0021563	0.10409013	0.00116947	0.00187037	0.01032196	0.10861
30	166508	0.00249705	0.10291844	1.43E-03	0.00046857	0.04409264	0.00170292	0.00021624	0.04088752	0.00103975	0.00513272	1.97E-04	0.00076285	0.02691388	0.02262714	3.70E-04	0.00314987	0.10000841	0.00064819	0.0003865	0.07031728	0.08199
31	166509	0.00249705	0.10291844	1.43E-03	0.00046857	0.04409264	0.00170292	0.00021624	0.04088752	0.00103975	0.00513272	0.00019683	7.63E-04	0.02691388	0.02262714	3.70E-04	0.00314987	0.10000841	0.00064819	0.0003865	0.07031728	0.08199
32	166510	0.00249705	0.10291844	1.43E-03	0.00046857	0.04409264	0.00170292	0.00021624	0.04088752	0.00103975	0.00513272	0.00019683	0.00076285	0.02691388	0.02262714	3.70E-04	0.00314987	0.10000841	0.00064819	0.0003865	0.07031728	0.08199
33	166511	0.00249705	0.10291844	1.43E-03	0.00046857	0.04409264	0.00170292	0.00021624	0.04088752	0.00103975	0.00513272	0.00019683	0.00076285	0.02691388	0.02262714	3.70E-04	0.00314987	0.10000841	0.00064819	0.0003865	0.07031728	0.08199
34	166512	0.00056987	0.07018404	2.97E-03	9.04E-05	0.0518729	0.00412658	0.00019259	0.01564046	0.00237936	0.00999142	0.00087008	0.00027125	0.04151528	0.06783329	4.03E-04	0.00188781	0.27365869	0.00081652	0.00055	0.01608255	0.15655
35	166513	0.00056987	0.07018404	2.97E-03	9.04E-05	0.0518729	0.00412658	0.00019259	0.01564046	0.00237936	0.00999142	8.70E-04	2.71E-04	0.04151528	0.06783329	4.03E-04	0.00188781	0.27365869	0.00081652	0.00055	0.01608255	0.15655
36	166514	0.00055861	0.08751004	1.47E-03	0.00042419	0.03080454	0.00799207	0.00034661	0.06420943	0.00283888	0.00479116	2.96E-04	0.0013011	0.03460754	0.04077095	1.58E-04	0.00231376	0.13964522	0.0026185	0.00121892	0.03117534	0.13223
37	166515	0.00060744	0.06294678	1.17E-03	0.00010207	0.05895573	0.00400284	0.00039857	0.04226615	0.00310013	0.00464723	0.00023274	0.00046977	0.0329739	0.04809261	3.74E-05	0.00161458	0.16868049	0.00161867	0.00033375	0.06399014	0.14921
38	166516	0.00056987	0.07018404	2.97E-03	9.04E-05	0.0518729	0.00412658	0.00019259	0.01564046	0.00237936	0.00999142	0.00087008	0.00027125	0.04151528	0.06783329	4.03E-04	0.00188781	0.27365869	0.00081652	0.00055	0.01608255	0.15655
39	166517	0.00056987	0.07018404	2.97E-03	9.04E-05	0.0518729	0.00412658	0.00019259	0.01564046	0.00237936	0.00999142	8.70E-04	0.00027125	0.04151528	0.06783329	4.03E-04	0.00188781	0.27365869	0.00081652	0.00055	0.01608255	0.15655
40	166518	0.00111865	0.10013201	1.54E-03	0.00050231	0.04206554	0.0015082	0.00028874	0.02370845	0.0009207	0.00450818	0.0005173	0.00084661	0.03974499	0.03712588	2.16E-04	0.0040828	0.11602433	0.00068812	0.00022982	0.04674649	0.10651
41	166519	0.0006512	0.07081694	1.99E-03	0.00011208	0.05818928	0.00310091	0.00027208	0.03421209	0.00135345	0.0057627	0.00030701	9.76E-05	0.03953581	0.04839508	9.74E-05	0.00186581	0.26586779	0.00059098	0.00139609	0.01985773	0.12061
42	166520	0.00105932	0.0615357	2.63E-03	0.00012419	0.05897919	0.00166182	0.00083352	0.01717033	0.00126362	0.00429396	0.00071174	0.00044714	0.04288026	0.06112815	1.03E-04	0.00193177	0.19752141	0.00091805	0.00013322	0.02822943	0.15557

NaiveBayesResult12

Kaggle Ranking Run#2:



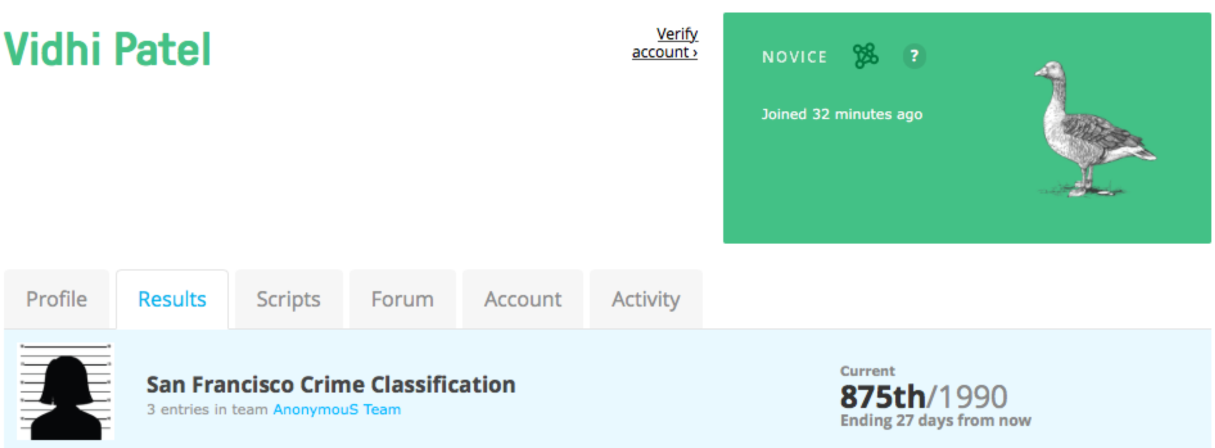
4.Logistic Regression

For this algorithm we used same feature which we selected for the second run of Naïve Bayes.

- Days of week
- PdDistrict
- Time

Output:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
566655	0.00240019	0.07628601	0.00105801	0.00095714	0.0459254	0.00242661	0.00326742	0.0230667	0.00423561	0.00146854	0.00102723	0.00111286	0.00932607	0.01592356	0.00080496	0.00303885	0.21412268	0.00215246	0.00140473	0.04163648	0.083217
566656	0.00274129	0.06395558	0.00118143	0.00102055	0.05123252	0.00256674	0.00576235	0.02228219	0.00395507	0.0013804	0.00107916	0.00119412	0.00771104	0.01660481	0.00087478	0.00280153	0.25342881	0.00224009	0.00142635	0.021628	0.098966
566657	0.00425492	0.10309199	0.00092664	0.00105337	0.03884162	0.00228735	0.00252089	0.04795856	0.00329415	0.00128749	0.0008043	0.00117935	0.00539085	0.00739394	0.00082153	0.00340492	0.13931618	0.00174661	0.00130772	0.04607046	0.054660
566658	0.00274129	0.06395558	0.00118143	0.00102055	0.05123252	0.00256674	0.00576235	0.02228219	0.00395507	0.0013804	0.00107916	0.00119412	0.00771104	0.01660481	0.00087478	0.00280153	0.25342881	0.00224009	0.00142635	0.021628	0.098966
566659	0.00171765	0.06723915	0.00108612	0.00083182	0.04943369	0.00403249	0.00203962	0.02480294	0.00525614	0.00180066	0.00097829	0.00096888	0.00763113	0.01843731	0.00082372	0.00243252	0.35496388	0.00244651	0.00142125	0.01310758	0.099993
566660	0.0020035	0.061625	0.00106275	0.00101133	0.05431397	0.00401839	0.00336312	0.05753715	0.00676565	0.00148086	0.00096474	0.00123245	0.00636772	0.01378678	0.00084893	0.00252749	0.23425865	0.00363398	0.00150559	0.04196284	0.093808
566661	0.0020035	0.061625	0.00106275	0.00101133	0.05431397	0.00401839	0.00336312	0.05753715	0.00676565	0.00148086	0.00096474	0.00123245	0.00636772	0.01378678	0.00084893	0.00252749	0.23425865	0.00363398	0.00150559	0.04196284	0.093808
566662	0.0020035	0.061625	0.00106275	0.00101133	0.05431397	0.00401839	0.00336312	0.05753715	0.00676565	0.00148086	0.00096474	0.00123245	0.00636772	0.01378678	0.00084893	0.00252749	0.23425865	0.00363398	0.00150559	0.04196284	0.093808
566663	0.00245999	0.06008445	0.0011962	0.00103506	0.05507636	0.00255841	0.00527151	0.02642345	0.00389592	0.00145985	0.00108215	0.00123606	0.00768151	0.01666655	0.00087917	0.00272934	0.27062622	0.00286717	0.00141846	0.02065294	0.098265
566664	0.00383125	0.09743122	0.00094126	0.0010718	0.04193244	0.00228729	0.00231288	0.0567994	0.00325532	0.00136601	0.00080914	0.00122472	0.00538741	0.00744509	0.00082832	0.00332797	0.15073968	0.00224313	0.0013047	0.04419312	0.054403
566665	0.00168181	0.06485724	0.00091447	0.00077404	0.05080138	0.00308563	0.00229125	0.04739077	0.0034202	0.00132086	0.00076038	0.0007975	0.00708121	0.01336505	0.00066052	0.00226656	0.32779486	0.00198675	0.00191665	0.01488827	0.075356
566666	0.00171765	0.06723915	0.00108612	0.00083182	0.04943369	0.00403249	0.00203962	0.02480294	0.00525614	0.00180066	0.00097829	0.00096888	0.00763113	0.01843731	0.00082372	0.00243252	0.35496388	0.00244651	0.00142125	0.01310758	0.099993
566667	0.00168181	0.06485724	0.00091447	0.00077404	0.05080138	0.00308563	0.00229125	0.04739077	0.0034202	0.00132086	0.00076038	0.0007975	0.00708121	0.01336505	0.00066052	0.00226656	0.32779486	0.00198675	0.00191665	0.01488827	0.075356
566668	0.00168181	0.06485724	0.00091447	0.00077404	0.05080138	0.00308563	0.00229125	0.04739077	0.0034202	0.00132086	0.00076038	0.0007975	0.00708121	0.01336505	0.00066052	0.00226656	0.32779486	0.00198675	0.00191665	0.01488827	0.075356
566669	0.00171765	0.06723915	0.00108612	0.00083182	0.04943369	0.00403249	0.00203962	0.02480294	0.00525614	0.00180066	0.00097829	0.00096888	0.00763113	0.01843731	0.00082372	0.00243252	0.35496388	0.00244651	0.00142125	0.01310758	0.099993
566670	0.00148259	0.06635384	0.00081998	0.00071679	0.02934291	0.00251801	0.00199237	0.06476474	0.00549706	0.00154388	0.00068591	0.00076629	0.00844922	0.01501734	0.00058565	0.00220883	0.328808	0.00290218	0.00256508	0.01552875	0.099053
566671	0.00148259	0.06635384	0.00081998	0.00071679	0.02934291	0.00251801	0.00199237	0.06476474	0.00549706	0.00154388	0.00068591	0.00076629	0.00844922	0.01501734	0.00058565	0.00220883	0.328808	0.00290218	0.00256508	0.01552875	0.099053
566672	0.00148259	0.06635384	0.00081998	0.00071679	0.02934291	0.00251801	0.00199237	0.06476474	0.00549706	0.00154388	0.00068591	0.00076629	0.00844922	0.01501734	0.00058565	0.00220883	0.328808	0.00290218	0.00256508	0.01552875	0.099053
566673	0.00171765	0.06723915	0.00108612	0.00083182	0.04943369	0.00403249	0.00203962	0.02480294	0.00525614	0.00180066	0.00097829	0.00096888	0.00763113	0.01843731	0.00082372	0.00243252	0.35496388	0.00244651	0.00142125	0.01310758	0.099993
566674	0.00245999	0.06008445	0.0011962	0.00103506	0.05507636	0.00255841	0.00527151	0.02642345	0.00389592	0.00145985	0.00108215	0.00123606	0.00768151	0.01666655	0.00087917	0.00272934	0.27062622	0.00286717	0.00141846	0.02065294	0.098265
566675	0.00148259	0.06635384	0.00081998	0.00071679	0.02934291	0.00251801	0.00199237	0.06476474	0.00549706	0.00154388	0.00068591	0.00076629	0.00844922	0.01501734	0.00058565	0.00220883	0.328808	0.00290218	0.00256508	0.01552875	0.099053
566676	0.00171765	0.06723915	0.00108612	0.00083182	0.04943369	0.00403249	0.00203962	0.02480294	0.00525614	0.00180066	0.00097829	0.00096888	0.00763113	0.01843731	0.00082372	0.00243252	0.35496388	0.00244651	0.00142125	0.01310758	0.099993
566677	0.00171765	0.06723915	0.00108612	0.00083182	0.04943369	0.00403249	0.00203962	0.02480294	0.00525614	0.00180066	0.00097829	0.00096888	0.00763113	0.01843731	0.00082372	0.00243252	0.35496388	0.00244651	0.00142125	0.01310758	0.099993
566678	0.00239647	0.09410002	0.00097796	0.00111076	0.04028616	0.00220043	0.00265458	0.03487087	0.00305875	0.00132603	0.0009263	0.00128539	0.00723754	0.0109684	0.00080261	0.00391274	0.17040947	0.00234182	0.00123859	0.03193286	0.069266
566679	0.00216312	0.05496454	0.00114943	0.00096877	0.06455522	0.00267768	0.00435731	0.02734554	0.00342822	0.00149011	0.00103475	0.00115374	0.0089095	0.01847745	0.00086101	0.00274804	0.25985257	0.00284422	0.00148017	0.02221482	0.100173
566680	0.0013113	0.06109011	0.00079261	0.00067487	0.03477468	0.00265114	0.0016557	0.06732119	0.0048669	0.00158529	0.00065977	0.00071951	0.00985739	0.01675221	0.00057698	0.00223726	0.31846484	0.00289613	0.00269249	0.01681045	0.101581
566681	0.00211322	0.08657096	0.00094237	0.00104258	0.04749934	0.00230961	0.0021994	0.03617897	0.00269889	0.00135736	0.00088823	0.0012032	0.00841932	0.01220337	0.00078827	0.0039507	0.16320875	0.00232966	0.00129615	0.0344151	0.070870
566682	0.00211322	0.08657096	0.00094237	0.00104258	0.04749934	0.00230961	0.0021994	0.03617897	0.00269889	0.00135736	0.00088823	0.0012032	0.00841932	0.01220337	0.00078827	0.0039507	0.16320875	0.00232966	0.00129615	0.0344151	0.070870
566683	0.00152097	0.06198687	0.00105107	0.00078407	0.05841768	0.00425019	0.00169692	0.02585286	0.00465882	0.00185105	0.00094209	0.00091078	0.00891413	0.02058174	0.00081245	0.00246663	0.34475421	0.00244419	0.00149363	0.0142082	0.102655
566684	0.00337503	0.0895497	0.00090596	0.00100483	0.04937106	0.00239796	0.00191393	0.05881065	0.00286904	0.00139666	0.00077498	0.00114506	0.00626175	0.00827712	0.00081257	0.00335635	0.14404959	0.00222888	0.00136374	0.04752667	0.055621
566685	0.00211322	0.08657096	0.00094237	0.00104258	0.04749934	0.00230961	0.0021994	0.03617897	0.00269889	0.00135736	0.00088823	0.0012032	0.00841932	0.01220337	0.00078827	0.0039507	0.16320875	0.00232966	0.00129615	0.0344151	0.070870
566686	0.00337503	0.0895497	0.00090596	0.00100483	0.04937106	0.00239796	0.00191393	0.05881065	0.00286904	0.00139666	0.00077498	0.00114506	0.00626175	0.00827712	0.00081257	0.00335635	0.14404959	0.00222888	0.00136374	0.04752667	0.055621
566687	0.00211322	0.08657096	0.00094237	0.00104258	0.04749934	0.00230961	0.0021994	0.03617897	0.00269889	0.00135736	0.00088823	0.0012032	0.00841932	0.01220337	0.00078827	0.0039507	0.16320875	0.00232966	0.00129615	0.0344151	0.070870
566688	0.00337503	0.0895497	0.00090596	0.00100483	0.04937106	0.00239796	0.00191393	0.05881065	0.00286904	0.00139666	0.00077498	0.00114506	0.00626175	0.00827712	0.00081257	0.00335635	0.14404959	0.00222888	0.00136374	0.04752667	0.055621
566689	0.00149081	0.05984517	0.00088589	0.00073037	0.06007445	0.0032558	0.00190838	0.04940158	0.00303405	0.00135927	0.00073301	0.00075046	0.00828112	0.01494395	0.00065217	0.00230076	0.31831615	0.00198695	0.00201632	0.01615275	0.07749
566690	0.00149081	0.05984517	0.00088589	0.00073037	0.06007445	0.0032558	0.00190838	0.04940158	0.00303405	0.00135927	0.00073301	0.00075046	0.00828112	0.01494395	0.00065217	0.00230076	0.31831615	0.00198695	0.00201632	0.01615275	0.07749
566691	0.0013113	0.06109011	0.00079261	0.00067487	0.03477468	0.00265114	0.0016557	0.06732119	0.0048669	0.00158529	0.00065977	0.00071951	0.00985739	0.01675221	0.00057698	0.00223726	0.31846484	0.00289613	0.00269249	0.01681045	0.101581
566692	0.0013113	0.06109011	0.00079261	0.00067487	0.03477468	0.00265114	0.0016557	0.06732119	0.0048669	0.00158529	0.00065977	0.00071951	0.00985739	0.01675221	0.00057698	0.00223726	0.31846484	0.00289613	0.00269249	0.01681045	0.101581
566693	0.00175558	0.0561846	0.00101768	0.00094331	0.06346114	0.00419101	0.00276939	0.05926648	0.00593505	0.00150637	0.00091932	0.00114643	0.00736214	0.01523794	0.00082855	0.0025361	0.22370924	0.00359255	0.00156569	0.04490304	0.095316
566694	0.00152097	0.06198687	0.00105107	0.00078407	0.05841768	0.00425019	0.00169692	0.02585286	0.00465882	0.00185105	0.00094209	0.00091078	0.00891413	0.02058174	0.00081245	0.00246663	0.34475421	0.00244419	0.00149363	0.0142082	0.102655
566695	0.00138981	0.07271132	0.00081885	0.00089289	0.03543459	0.00637327	0.00219625	0.08514825	0.00506784	0.00125194	0.00071593	0.00126503	0.00742287	0.01297071	0.00066183	0.00259828	0.18978838	0.00441107	0.00188162	0.02385428	0.08448



6. Conclusion:

For this competition we tried these four algorithms. Out of which logistic regression with three features

- PdDistrict
- Hour
- Days of week

Gives us the best results and out ranking bumped up to 875 from 1902.

Algorithm	**Kaggle Ranking**
K Nearest Neighbor (k = 3)	1902
Random Forest	1667
Naïve Bayes	1114
Naïve Bayes (three features)	910
Logistic Regression	875

We can conclude that good feature selection and good algorithm can improve the ranking at Kaggle competition.

7. References:

1. http://efavdb.com/predicting-san-francisco-crimes/
2. http://scikit-learn.org/stable/
3. https://www.kaggle.com/c/sf-crime
4. http://docs.rapidminer.com/studio/operators/modeling/predictive/bayesian/naive_bayes.html