Case Study Report

San Francisco Crime Classification

Gabriela HERNÁNDEZ LARIOS

Contents

AKNOWLEDGMENTS

First of all, I would like to thank the Prof. Peter Neuss for his support during the elaboration of this work and for giving me valuable advices and guidelines.

I would also like to extend my thanks to each member of this team for their support, their commitment and their brilliant ideas shared in the elaboration of this project.

1 Introduction

This project was conducted jointly by the DMKM students in Italy, we enrolled the San Francisco Crime Classification competition in Kaggle [1]. This competition aims to classify and predict the crimes in the city of San Francisco, California given geographical information and time data. This report briefly introduces the work performed by each team member, focusing mostly on the machine learning and data mining techniques I carried out.

This report is structured as follows: in this section, I present the main motivations of this project and a general overview of its structure. Section 2 summarizes the principal methods and techniques used. In Section 3, we can find the description of the train and test datasets used. Section 4 shows an exhaustive exploration in the data, identifying some patterns, outliers and noisy data. Section 5 introduces the main techniques used to preprocess and transform the data, while section6 describes the implementation of the models and its results. Section 7 summarizes the best results obtained by all the members of the team and finally in Section 8 I draw my final conclusions and comments for this work, introducing future work that can help to improve our results.

1.1 Motivation

In the last years, the task of crime prediction has gained significant popularity in the research literature. Although, there exist many procedures and approaches used to carry out investigations to respond to different crimes, predicting where and when a crime is likely to occur leads to develop more efficient strategies either to prevent crimes or to improve the investigation efforts.

Among several investigations carried out in this field, many of them provide a strong evidence that the crime is in fact predictable in a statistical way, some of them are presented in the survey carried out by the RAND corporation in [2]. This theory is mainly supported by the fact that people tend to follow certain life habits and patterns, and even though it is not always true, the amount of crimes occurred under the same circumstances make these kind of methods work reasonably good. Therefore, one of the main justification of crime prediction is to identify these patterns in order to prevent crimes with strategic interventions.

In this work, we particularly worked with the crime occurrences in the city of San Francisco. This dataset contains two of the most difficult data types to model: time series and spatial data. Therefore, working and dealing with this information in a real noisy dataset was especially important for our professional formation.

1.2 Project Structure and Overview

As previously mentioned, we worked in team and each member performed different tasks with respect to the feature engineering and models. Aleksandra worked mostly with the "Address" feature by identifying corners, type of streets (street, Avenue, Boulevard, Plaza, Road, etc.) and whether a crime has happened in the same location or not (labeled as repeated or unique). Maira worked mostly with the time data, identifying seasons (Spring, Summer, etc), Periods of months ('middle', 'beginning', etc.), she also performed PCA in the data, in order to keep the data with more variance, and she separated the less likely crimes from the training dataset by assigning them pre-defined probabilities. Kienka worked identifying hot-spots of the crimes and with a novel model of deep neural networks that helped us to improve our results. I was mainly focused on the spatial data, finding the best approaches to use Longitude and Latitude as features for our classifiers. With this purpose, I divided the San Francisco map into 400 cells by creating a 20x20 grid map and each data point was labeled with a cell number according to its longitude and latitude coordinates, later I created a 39-D vector of each crime identifying the relative and absolute probabilities of the crime to happen in each cell, in section 2 I provide more specific details about this implementation. In general, we used the following four different models: "Logistic Regression", "Naive Bayes Classifier", "Random Forest" and "Deep Neural Networks".

In general, for developing this project we used two phases. In the first phase, as shown in figure 2, we worked with the training dataset by exploring it, finding patterns, cleaning it, extracting the most important features and transforming these features as necessary for our models, afterwards we validated each classification algorithm with a partition cross validation with 0.6 for training and 0.4 for testing and we obtained the log-loss result, finally we selected the model that gave us the best result. In the second phase, as shown in figure 3, we transformed the testing dataset accordingly to the transformations we made in the training set, we obtained the crimen prediction probabilities for each observation, afterwards we submitted our results to Kaggle and obtained the log-loss result.

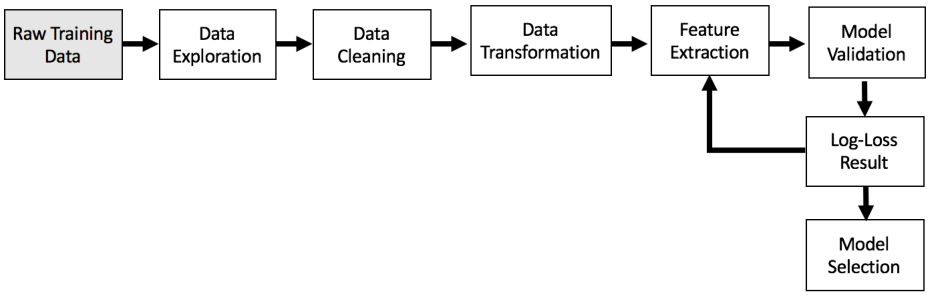

Figure 2: Overview of the first phase of the project with the training set

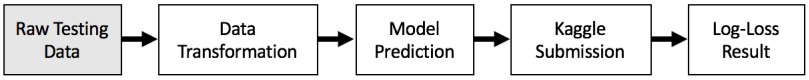

Figure 3: Overview of the second phase of the project with the testing set

2 Methodology

In this section, the software and technologies used to develop this project are detailed as well as the preprocessing and modeling methods I used.

2.1 Software and technologies

The software used to carried out this project was Python version 3.4 [3] with the libraries Pandas (Version 0.17.0), NumPy (Version 1.9.1), sklearn (Version 0.0), matplotlib (Version 1.4.2) and keras (Version 0.3.1). Below, I briefly explain each library and its function in this work.

Pandas[4], is an open source library in Python that provides a highly optimized manipulation and operations in data structures. Form this library, we mainly used the DataFrame structure for creating and manipulating both training and testing datasets as well as for creating the submission file.

NumPy[5], is a fundamental package in Python for N-dimensional array object, along with a large library of high-level mathematical functions to operate on these arrays. We used this library in order to efficiently create and manage arrays such as the grid cells, the probabilities of the crimes in the cells and the submission file.

sklearn (Scikit-Learn) [6], is an open source machine learning library for Python. It does not only contains various classification, regression and clustering algorithms but also preprocessing techniques such as standardization, scaling, normalization, binarization, encoding and others,

and model evaluation metrics like cross-validation methods, confusion matrix, log-loss error, for mentioning some of them. It is also designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. We used this library for implementing "scaling", "PCA", "stratified split for cross-validation", "Logistic regression", "Naive Bayes Classifier", "Random forest" and "log-loss".

matplotlib[7], is a python 2D plotting library and its numerical mathematics extension NumPy. It provides a variety of high quality figures in different formats and interactive methods. This library was basically used for plotting the Longitude and Latitude heatmaps for crimes.

keras[8], is a python deep Learning library able to run either TensorFlow or Theano with a fast implementation. Keras contains two models: Sequential, a linear stack of layers, and Graph, a directed acylcic graph of layers. We used the sequential model of this library.

It is also important to mention that for exploring the data, we also have used the amazing software **Tableau**[9], which provides an efficient and easy way to examine the overall dataset.

2.2 Preprocessing Methods

Although in the overall project different preprocessing methods and data transformations were used by the team members, such as "Scalability and Standardization of the Longitude and Latitude features", "Normalization and PCA", creation of new features based on the time and address features, the implementation of these are out of the scope of this report. In this section I focus on the special treatment and transformation I gave to the Longitude and Latitude variables.

Grid Map and Cells for substituting Longitude and Latitude features

In this dataset, we have two spatial features: Longitude and Latitude. The main concern was how to use these coordinates as features in our classifiers in order to improve the prediction results, with this aim I created a Grid Map with 400 cells.

Roughly speaking, the idea behind creating a Grid Map and cells is to divide the map of San Francisco city into cells and assign to each observation a cell-number accordingly to its Longitude and Latitude coordinates.

For creating the Grid Map, I assigned two axes: 'X' representing the Longitude and 'Y' for the Latitude, then I divided each axis into 20, in order to get a 20x20 Grid Map and 400 different cells. The boundaries of the X-axis were established by taking the maximum and minimum longitude in the training dataset and for the Y-axis its limits were the maximum and minimum latitude in the training dataset. It is worth mentioning that for the axes limits I excluded a potential outlier of 90 in the Latitude coordinate, and if the observation has this coordinate the cell is named 'cOutlier'. The resolution of the X-axis was calculated as $x_{res} = \frac{lon\_max - lon\_min}{20}$ and the resolution of the y-axis was computed as $y_{res} = \frac{lat\_max - lat\_min}{20}$. For the computation of the number of cell of each observation, given its longitude and latitude coordinates, first I calculated the x-coordinate (the number of column) and y-coordinate (the number of row) in the grid map as follows $x\_coordinatet = \frac{floor(longitude - lon_min}{x_{res}}$, $y\_coordinate = \frac{floor(latitude - lat_min)}{y_{res}}$, once I had both coordinates the number of cell was obtained by applying the following equation $cell\_num = 20 * y\_coordinate + x\_coordinate$.

In this manner, I created a new feature called 'Cell' in both training and testing datasets, containing the cell number of each observation.

2.3 Classification Models

Throughout this project, different models were used by the team members such as: Naive Bayes Classifier, Random Forest, Multinomial Logistic Regression and Deep Learning. However I only used the last two. Below I give a brief explanation of both models. It's important to emphasize that the deep learning model was entirely developed by Kienka using the Keras library of python.

Multinomial Logistic Regression

Logistic Regression is a regression model where the target variable is two-levels categorical (i.e. binary, like True/False, sick/no-sick, etc.). The extension version of Logistic Regression that allows multinominal target variables is Multinomial logistic regression in which the log odds of the outcomes are modeled as a linear combination of the predictor variables. This model can be used either to predict the categories of the target variable or the probability of category membership on the target variable based on multiple independent or predictor variables. In this model, the predictor variables have to be either dichotomous or continuous.

Deep Neural Networks (Deep Learning)

Roughly speaking, a deep neural network (DNN) is an artificial neural network (ANN) with multiple hidden layers of units between the input and output layers. DNNs are typically designed as feedforward networks. Basically a feedforward network has an input layer, one or more hidden layers and output layer as shown in figure 4.

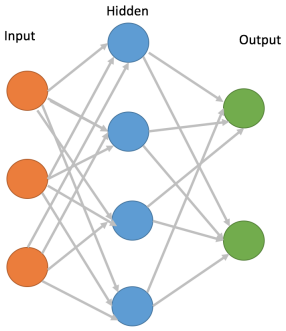

Figure 4: Feedforward network

2.4 Model Validation and Evaluation

In order to validate our models with the training dataset we performed the following two kinds of cross-validation:

Partition Cross-validation

In this technique a random split into the dataset is done, generating a new training set used to estimate and select the parameters of the model and a test set to evaluate its performance.

With this technique, we usuall split the data into 60% training and 40 % test sets.

Stratified fold Cross-Validation

This technique was particularly used for the Deep Neural Networks model. In stratified fold cross-validation, the aim is to return stratified randomized folds where the folds are made by preserving the percentage of samples for each class.

Logarithmic loss

In Kaggle, the submissions are evaluated using the multi-class logarithmic loss. Each incident of the testing set has been labeled with one true class by Kaggle. Therefore, for each incident in the testing set, we submitted a set of predicted probabilities (one for every class). [1].

The formula of logarithmic loss is given by:

[1] https://www.kaggle.com/c/sf-crime/details/evaluation

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

where N is the number of cases in the test set, M is the number of class labels, log is the natural logarithm, y_{ij} is 1 if observation i is in class j and 0 otherwise, and p_{ij} is the predicted probability that observation i belongs to class j.

3 Datasets

We used the training (train.csv) and testing (test.csv) datasets provided in the San Francisco Crime Classification competition of Kaggle [1]. These datasets contain incidents brought from SF Open Data[2]. The training-set's dates range from 6 January 2003 to 13 May 2015 with a total of 878,049 data points, meanwhile the testing set includes data from 1 January 2003 to 10 May 2015 with 884262 observations.

Each observation is described by 9 features as follows:

Dates: timestamp of the crime incident

Category: category of the crime incident (only in train.csv). This is the target variable we 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

The Category feature is the target variable and it contains 39 different crimes that are showed in figure 5.

4 Exploratory Data Analysis

In this section, I present a visual exploration about the behavior of the crimes with respect of the time and location, as well as their distribution, in order to delve into the dataset and obtain

4.1 Features Exploration

In this section, I aimed to analyze the features of the training set, in order to get insights and discover interrelations among the features and the target variable (Category).

With this purpose, the first thing I did was to observe the distribution of the crimes in the data set. One interesting fact discovered is that the crimes are not only unevenly distributed but also that their distribution follows the power law or long tail, as shown in figure 5, this fact will make the least common crimes more difficult to predict and further feature engineering must be

[2]https://data.sfgov.org

performed. Nevertheless, the ten most common crimes shown in table 1 cover more than 83% of the whole training dataset.

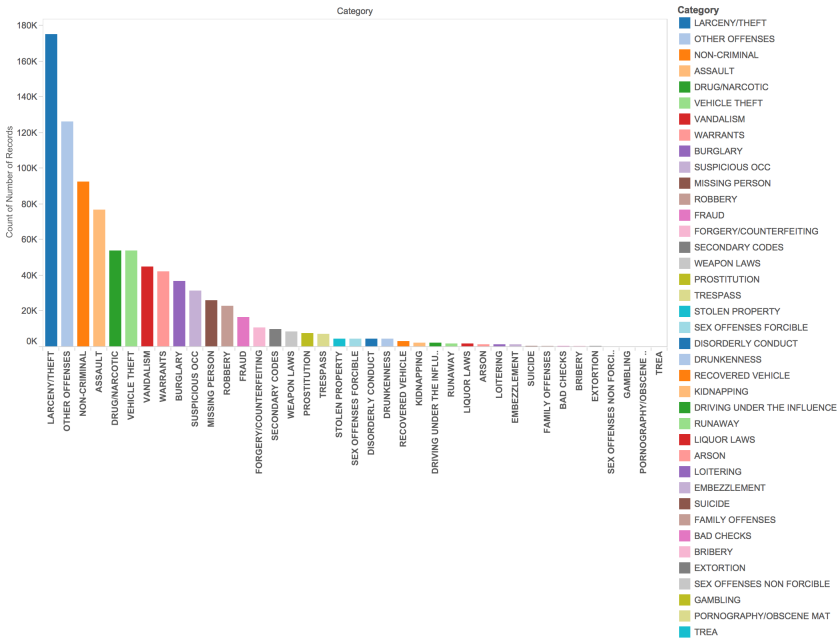

Figure 5: Distribution of the crimes in the training dataset

Table 1: 10 most frequent crimes

Category	**Number of Records**
Larceny/Theft	174,900
Other Offenses	126,182
Non-Criminal	92,304
Assault	76,876
Drug/Narcotic	53,971
Vehicle/Theft	53,781
Vandalism	44,725
Warrants	42,214
Burglary	36,755
Suspicious OCC	31,414

Table 2: 5 Least frequent crimes

Category	**Number of Records**
Trea	6
Pornography/ Obscene mat	22
Gambling	146
Sex Offenses Non Forcible	148
Extortion	256

Since the 10 most common crimes cover more than 83% of the whole dataset, along this section we will explore the variables by using these crimes.

Regarding the Pd District feature (Police Districts), we can see in figure 6 that the SOUTHERN district is the one with more crimes and many of them are LARCENY/THEFT, this crime also has higher frequency win the NORTHERN and CENTRAL districts whereas in the TENDERLOIN district the most frequent crime is 'DRUG/NARCOTIC'. We can also observe that the 'safest' districts are PARK and RICHMOND. As we can see this feature gives important information about the crimes and it must be included into the features to predict the crimes.

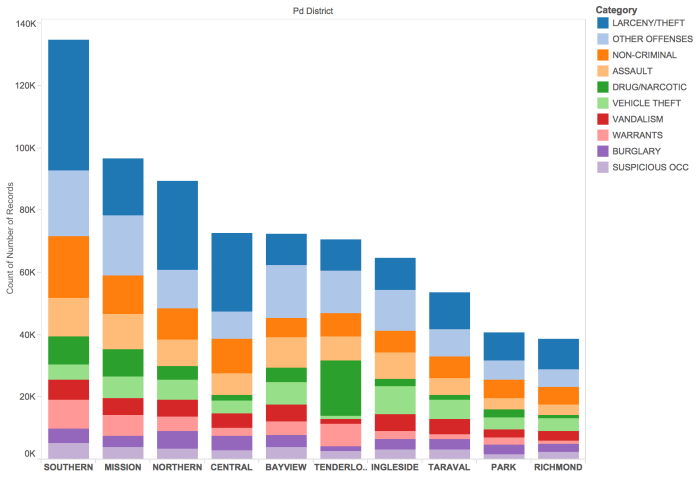

Figure 6: Stacked plot of crimes w.r.t. the Pd Districts

With respect to the day of the week, in figure 7 we can observe that the days in which more crimes occurred are Fridays, Wednesdays and Saturdays whereas the Mondays and Sundays less crimes occur. However, we do not see an important variance of 10 most common crimes with respect of the day of the week, therefore in the models we must check how much influence in fact, they have in order to improve the prediction results.

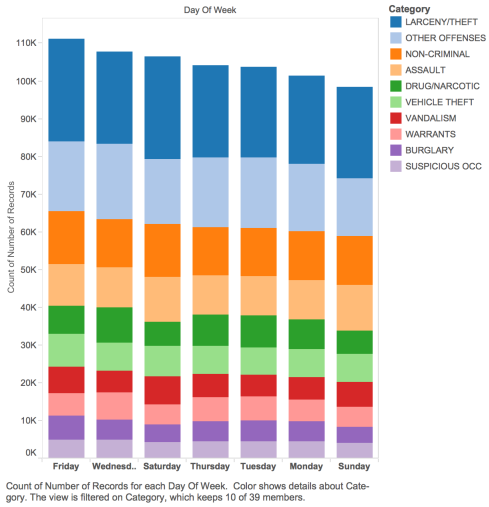

Figure 7: Stacked plot of crimes w.r.t. the Day of the Week

In order to discover, whether the day of the month is an important feature to add, I examined the distribution of the 10 most common crimes along the day of the months, as it is shown in figure 8, as we can see here, even though during the first day of each month more crimes happened and the 31st less crimes (because not all the months have 31st day), there are no substantially changes in proportion of the crimes happened each day, so it must be further investigated whether these days can give us crucial information for predicting.

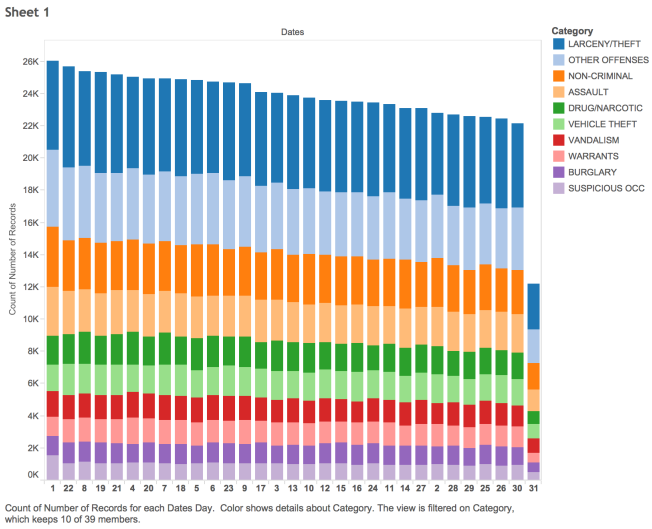

Figure 8: Stacked plot of crimes w.r.t. the Day of the month

By analyzing the months, in figure 9, we can see that during summer (June, July, August, September) less crimes occur, this is an interesting fact for doing feature engineering in this variable. We can also notice that the proportion of occurrence of each crime along the months does not substantially change.

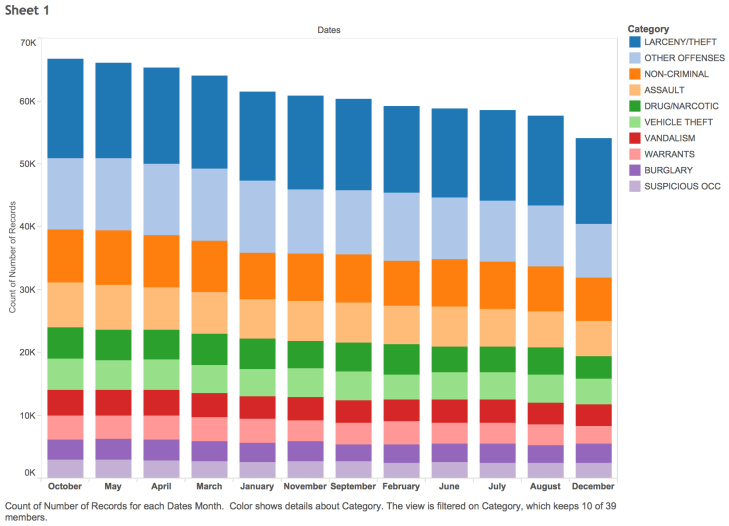

Figure 9: Stacked plot of crimes w.r.t. the months

On of the most interesting features was the year, as we can see in figure 10, in fact the crimes have different proportions throughout the years, one interesting fact discovered here is that the crime 'VEHICLE_THEFT' dramatically decreased after 2006. In order to investigate this case further, I checked the description of this crime and I found that the 'VEHICLE RECOVERED' description only appears in this crime during the years 2003-2005, and after 2005 this description appears in the crime 'VEHICLE_RECOVERED' as shown in figure 11. I also proved this fact, when I checked the distribution of both crimes along the years, and the occurrences of 'VEHICLE RECOVERED' appears after 2005 as shown in figure 12. Therefore, we can observe a change in the methodology structure of the crimes after 2005.

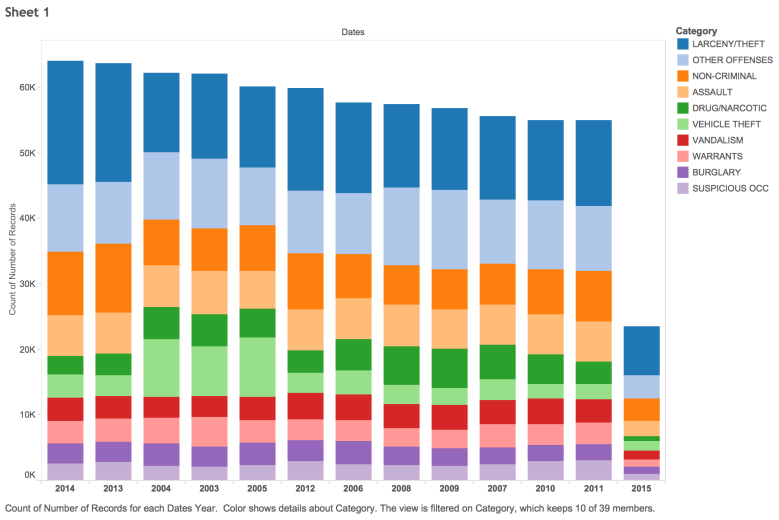

Figure 10: Stacked plot of crimes w.r.t. the years

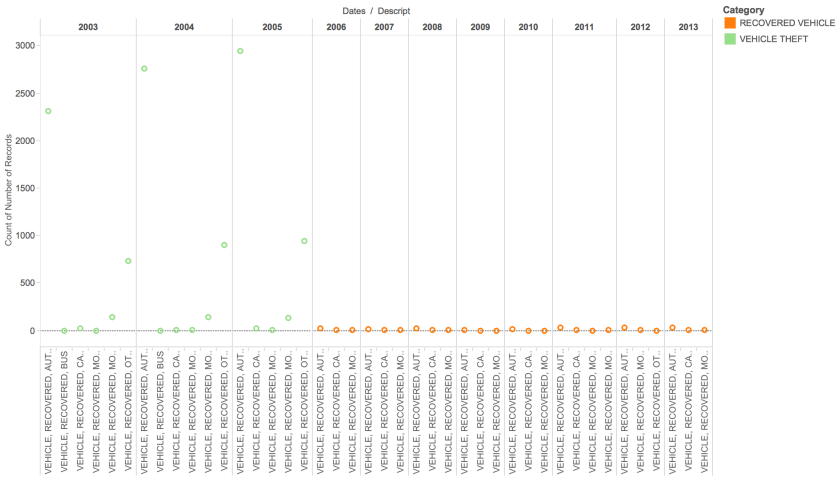

Figure 11: Distribution of the Vehicle Recovered Description among the years in the Vehicle Theft and Vehicle Recovered categories

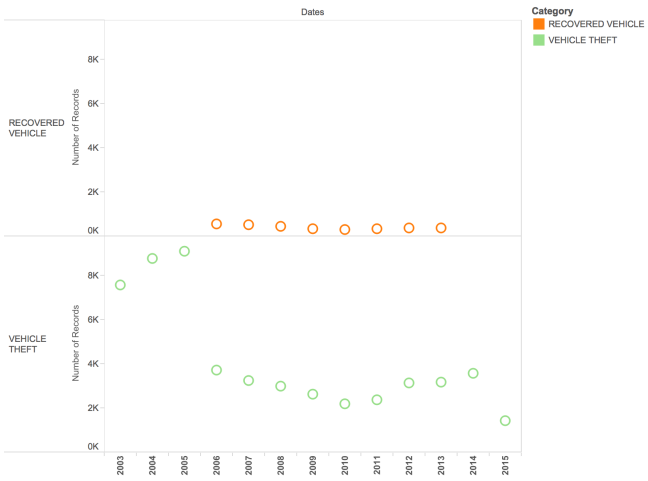

Figure 12: Distribution among the years of Vehicle Theft and Recovered Vehicle categories

Another interesting feature, which seems to give good information for predicting crimes are the hours. As we can see in figure 13, the total amount of crimes happening in each hour substantially differs from each other, moreover the proportion of each crime happening in each hour is also different.

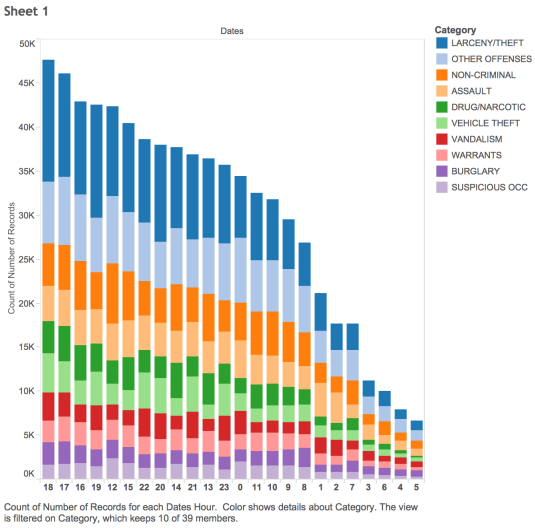

Figure 13: Stacked plot of crimes w.r.t. the hours

With respect to the Longitude and Latitude, as we previously expected, these features have high importance to predict crimes. One important thing discovered is that in both the training and testing data there are potential outliers in the latitude coordinate 90 as shown in figure 14, in the training dataset there are 62 observations in this coordinate whereas in the testing 76, since this represent less than 0.01% of the whole datasets, I just ignored them in the rest of the analysis of this feature.

Figure 14: Outliers in Longitude (X) and Latitude (Y)

Another interesting thing found in these features is the substantial amount of crimes happening in the coordinate -122.4 and 37.77 as shown respectively in figures 15 and 16. In the training dataset there are more than 26,000 crimes occurring in this point. In figure 17 we can see that in fact this point is in the city center and its address is 800 Block of BRYANT ST. By investigating this street in internet, I found that it is a long street and in fact is well-known for being dangerous.

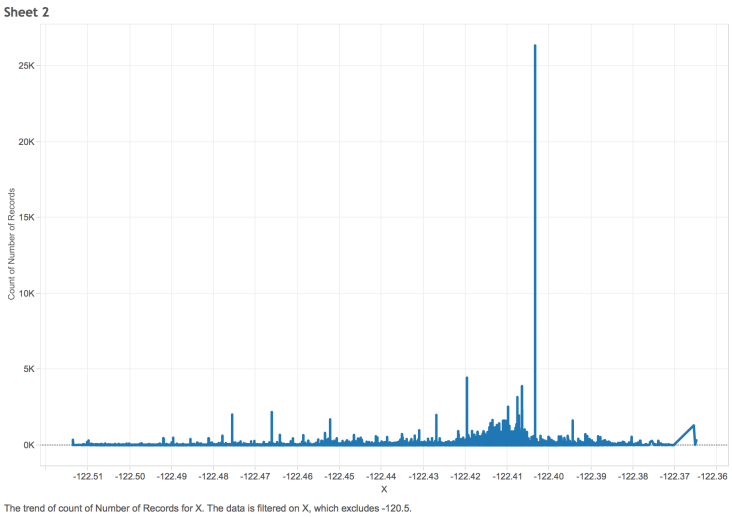

Figure 15: Distribution of the Longitude

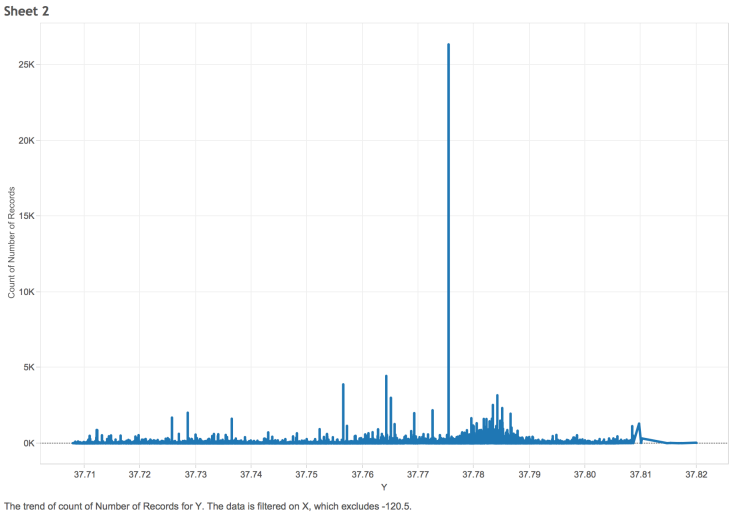

Figure 16: Distribution of the Latitude

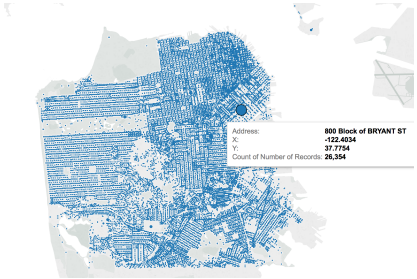

Figure 17: Distribution of Longitude and Latitude in San Francisco map

As we previously have seen there is an important amount of crimes happening in the address 800 Block of BRYANT ST, to further investigate this, I filtered the 10 most frequent address as shown in figure 18 and it turns out that in fact this address is by far the most frequent among the entire dataset. Therefore, I looked for the distribution of the crimes in this address as shown in figure 19, and I found again a power-law distribution, making the prediction tasks of the least frequent crimes in this address more difficult. As we can see, these features are very important for predicting but they do require a big effort in the feature engineering process.

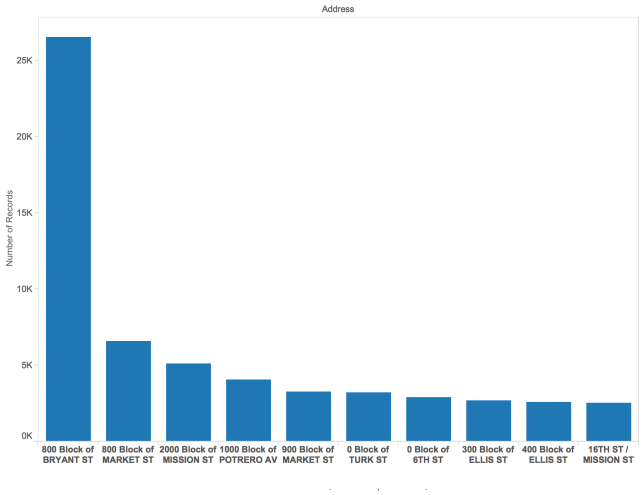

Figure 18: 10 most frequent addresses

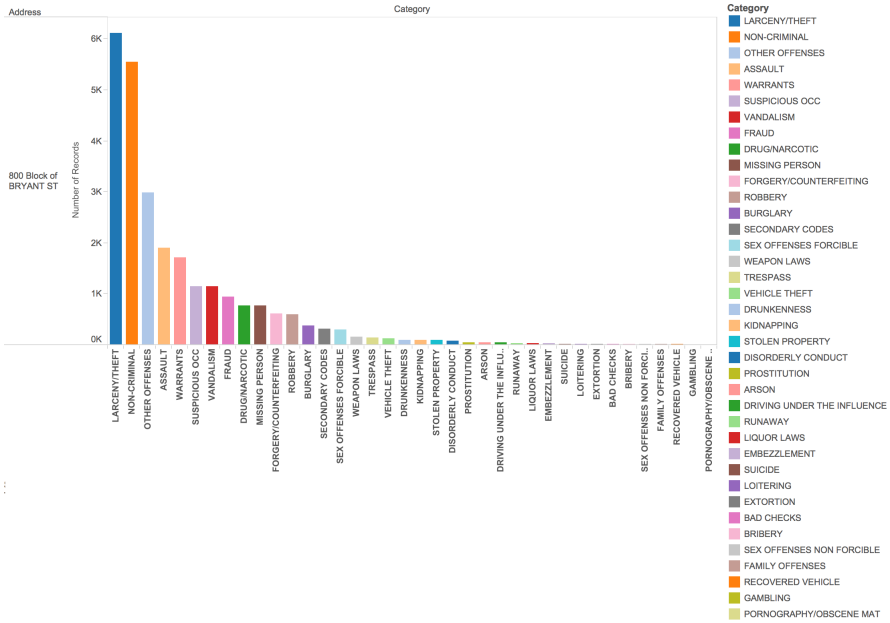

Figure 19: Crimes distribution in the address 800 Block of Bryant St

Finally, for exploring the Grid Map and cells, I created heatmaps for each crime. Each cell of the heatmap represents a cell of the Grid Map. For each crime I created a matrix where I stored in each element the number of records of the specific crime that happened in the cell divided by the total number of crimes happening in the cell, then I normalized the matrix and with this I created the heatmap. Figures 20, 21, 22, 23, 24, 25 show the heatmaps of the 6 first more common crimes, the rest of the heatmaps can be found in the following link `https://onedrive.live.com/redir?resid=E05A622B8A3C40BC!23575&authkey=!AFEAzvYwaO1glC4&v=3&ithint=photo%2cpng`. As we can see in these maps, for each crime we can in fact find crucial zones or more specifically 'hot-spots' of the crimes

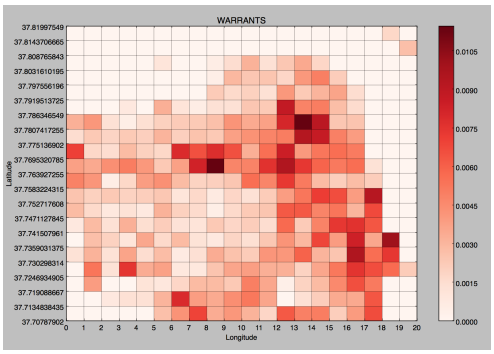

Figure 20: Heatmap for the crime WARRANTS

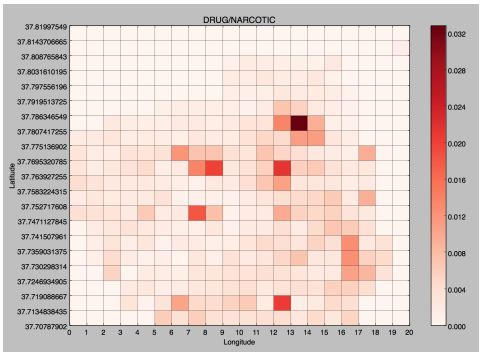

Figure 21: Heatmap for the crime DRUG/NARCOTIC

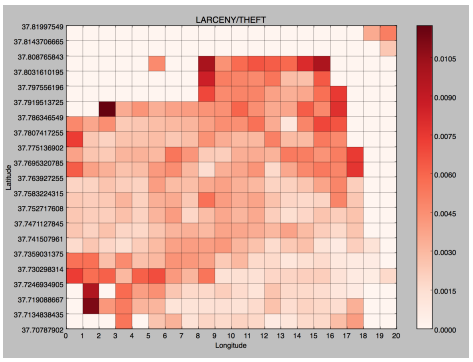

Figure 22: Heatmap for the crime LARCENY/THEFT

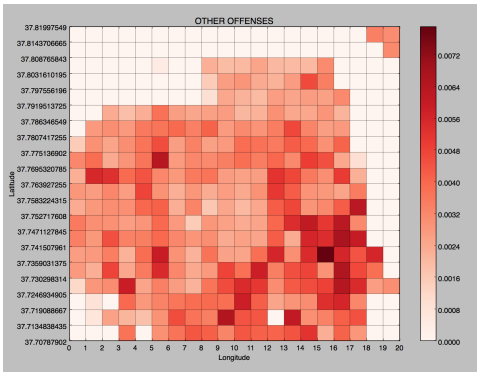

Figure 23: Heatmap for the crime OTHER OFENSES

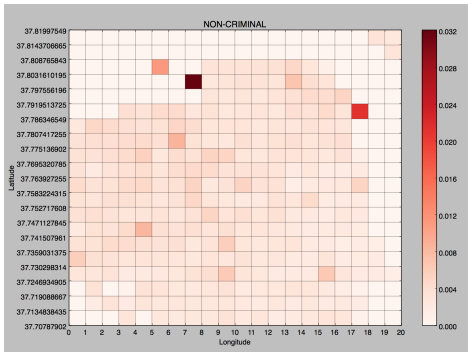

Figure 24: Heatmap for the crime NON-CRIMINAL

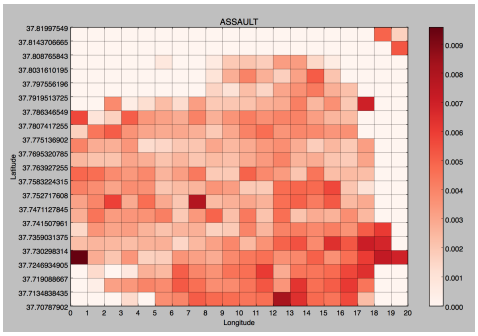

Figure 25: Heatmap for the crime ASSAULT

5 Data Preprocessing

The data cleaning together with the data transformation play an essential and crucial role in any data mining project. In this section, I present the main preprocessing techniques I carried out during this project

5.1 Data Cleaning

As we have seen in section 4, that in fact there are features that might not be relevant at all or need further feature engineering.

Before implementing machine learning algorithms on our data, I cleaned the dataset as follows:

1. I dropped the 'days of month' feature, since it does not add much marginal value.

2. I also dropped the observations containing outliers in Latitude feature (i.e. Y=90). Due to less than 0.001% in the data has this problem, the cost of missclassifying these data points in the testing dataset is lower than adding them to train our model.

5.2 Feature Enrichment

For feature engineering, many approaches have been taken into consideration by the team members. Below I summarize the main ones that I used and were carried out by them:

Aleksandra created the following three new columns based on the "Address":

- Corner: Specifies whether the stree is a corner or not and it has two labels: 'Corner' and

'No_Corner'

- Street Type: Specifies the type of street with the following labels:'Street', 'Avenue', 'Way', 'Boulevard', 'Drive', 'Court', 'Terrace', 'Highway', 'Road', 'Plaza', 'Place', 'Lane', 'Alley', 'Circle', 'Unknown', 'Walk', 'Expressway' and 'Park'.
- Repeated: Specifies whether a crime already have occurred in an address or not.

Maira has worked mostly with the time data creating the following two new features:

- Season_year: This feature specifies the seasons of the year with the following labels: 'Spring', 'Winter', 'Fall', 'Summer'.
- Period_of_month: For emphasizing if a crime happened in the beginning, middle of end of a month with the following labels: 'middle_of_month', 'begin_of_month', 'end_of_month'.

In my case, I used two different approaches for enriching the 'Cell' column that I created with the latitude and longitude variables (explained in section 2). .

The first approach consists on only using the cell names and creating one hot encoding variables. The pitfall of this approach is that it adds 200 columns more.

In the second approach, I used a 39-d vector to represent each cell. Each vector represents the absolute frequency distribution over 39 crime categories at the corresponding cell. The absolute frequency distribution of a $crime_j$ in a $cell_i$ was computed as follows:

$$Fabs_{i,j} = \frac{\#\ of\ \ records\ \ of\ \ crime_j\ \ in\ \ cell_i}{\#\ of\ \ records\ \ of\ \ crime_j\ \ in\ \ the\ \ entire\ \ dataset}$$

It's important to mention that the cell 'cOutlier' was represented by a 39-d vector with the absolute probability of each crime, it means, the frequency of the crime in the dataset divided by the total amount of crimes in the dataset.

Afterwards, I replaced the 'Cell' column in both training and testing datasets by these 39 new features containing for each observation the absolute frequencies of each crime at its corresponding cell.

5.3 Feature Selection

For selecting the best features, I carried out several tests in the Logistic Regression Model with different combination of features, however I decided to keep only the following two different combinations of features, that gave the best scores:

Dataset 1: Containing the following features: 'Hour', 'Day of the Week', 'month', 'year', 'PdDistrict', 'Season_year', 'Corners', 'Street_type', 'Repetition', and the 39-d features representing the Cells.

Dataset 2: Containing the following features: 'Hour', 'Day of the Week', 'month', 'year', 'PdDistrict', 'Season_year', 'Corners', 'Street_type', 'Repetition' and 'Cell'.

5.4 Data Transformation

Without taking into account the 39-d features representing the Cells, the rest of the features can be seen as categorical, even the hour, month and year. However, the best classification models require continuous data. With this aim, we transformed each categorical feature into one hot encoding array by using the LabelEncoder() function implemented in the module 'preprocessing' of sklearn. At the end, with the features I selected I ended up with two datasets, one containing 131 different features for the 39-d vector, and other

6 Models

Once the data was cleaned and the features were enriched, selected and transformed. I used the following two different models, in order to classify and predict the crimes.

6.1 Multinomial Logistic Regression

For implementing this model, we used the function 'LogisticRegression()' of the sklearn library. The parameter 'multiclass' controls whether to use Logistic Regression or its version for Multinomial target variable. In this case, we specified set 'multinomial' to this parameter. I used the both datasets specified in 5.3, I transformed the categorical features (except the 39-d features representing the 'Cell') in the 'Dataset1' as one-hot label encoding, and for the 'Dataset2' all the features were transformed as one-hot label encoding. The results of both datasets in cross validation and in the submission file are summarized in table 3

Table 3: Results in the Multinomial Logistic Regression

Dataset	**Log -loss Result in Crossvalidation**	**Log-loss Result in Final submission**
Dataset 1	2.3655	2.3729
Dataset 2	2.4722	2.4834

6.2 Deep Neural Networks with Keras

This model was completely implemented by Kienka KIO, using the Keras library of Python and the stratified fold crossvalidation of sklearn. Therefore, the explanation of its implementation is out o the scope of this report. It's worth to mention that the only things I modified in this model were the features used, i.e. the selection of the features and I also removed scaling of the variables and PCA.

For this model, I only used the 'Dataset 1' with the combination of features explained in the section 5.3. In this case I also 'binarized' the categorical features (except the 39-d features representing the 'Cell').

The overall log-loss result in the cross validation with this model and this training dataset was **2.32358**. And the log-loss result of the final submission file in Kaggle was **2.33329**. Nowadays, this is the best result the team has obtained, in the competition, and it positioned us in the 100 position out of 1215 (February 7, 2016).

7 Results

Table 4, summarizes the best results obtained in the submission files by the DMKM team. As we can see, the best results were always obtained by using the 'Deep Neural Network' (DNN) model, proving the power of this novel technique in predicting crimes.

Table 4: Summary of the best results obtained in the Kaggle competition by the DMKM team

Description	**Log-loss Result in Final submission**
Prediction with DNN with the 39-d features describing the cell	2.33329
Prediction with DNN Using Hotspots with respect to crime frequency given an address.	2.35182
Prediction with DNN using the repetition of time and addres	2.35583
Prediction with DNN and the clustered centroids	2.35949
Prediction with Deep Neural Networks (DNN)	2.3729
Prediction with Logistic Regression and with the 39-d features describing the cell	2.37294

8 Conclusions and Future Work

In this project, we aimed at classifying and predicting the crimes that occurred from 2003 to 2015 in the city of San Francisco, California. With this goal, we conducted several approaches either for feature engineering or model implementation, in order to accomplish this task. Even though that at the beginning this problem appeared to be a normal classification task, when we inspected the dataset in fact, we found specific characteristics that made the crime prediction harder, such as the power-law distribution of the crimes among the dataset, specific points (address, Longitude and Latitude) with a significant am ount of crimes with respect to the others and a possible change in the methodology of storing the Vehicle_Theft and Vehicle_Recovered categories after 2005, for mentioning some of them. We are aware that there can be even more things and noise hidden in the dataset that can give us insights of how to model and enrich the features, so as a future work I would like to unearth more trends and patterns in this dataset.

Moreover, the provided datasets for predicting the crimes are basically conformed only by time and location features, two of the most difficult features to model, to overcome this we have exhaustevly working with the location features, nevertheless I would like to do the same with the time features by trying to model them and understanding the main changes among the years or possible correlations with specific time-series models.

As we have seen throughout this project, predicting crimes is not a trivial task and there exist many factors that make this effort more difficult. In the existent literature of this area, we have noticed that many authors claim that the crimes are 'Predictable' whereas others believe they are not. As per my experience with this work I can conclude that the crimes might be predictable but of course we can never expect an absolute true or very high accuracy. Indeed, there are some strong patterns that can make the classification models work reasonably good, but in this area there will always be an important amount of outliers.

In my personal opinion, I have really enjoyed working on this dataset and enrolled a Kaggle competion. During this work I could face the main difficulties of working with a real dataset, I have also learned new techniques for handling spatial and time data. Furthermore, I have not only learned useful techniques in Python with the libraries Pandas, NumPy and Sklearn but also I have experienced their power and I have noticed their importance in the machine learning and data mining field.

References

[1] "Kaggle: San francisco crime classification," https://www.kaggle.com/c/sf-crime, last access: 06-02-2016.

[2] W. L. Perry, *Predictive policing: The role of crime forecasting in law enforcement operations*. Rand Corporation, 2013.

[3] "Python," https://www.python.org/, last access: 30-12-2015.

[4] "Pandas library," http://pandas.pydata.org, last access: 02-02-2016.

[5] "Numpy library," http://www.numpy.org, last access: 12-12-2015.

[6] "Scikit-learn library," http://scikit-learn.org/stable/, last access: 05-02-2016.

[7] "Matplotlib library," http://matplotlib.org, last access: 05-02-2016.

[8] "Keras library," http://keras.io, last access: 05-02-2016.

[9] "Tableau software," http://www.tableau.com, last access: 05-02-2016.