University of Pennsylvania

CIS 520: Machine Learning

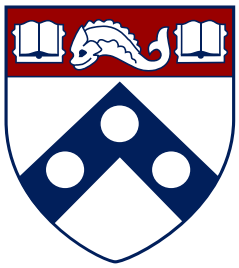
Yo Home to Bel-Air: Predicting Crime on The Streets of Philadelphia

Author:
Christian Tabedzki
Amruthesh Thirumalaiswamy
Paul van Vliet

Instructor:
Prof. Shivani Agarwal

Project Mentor:
Simeng Sun

April 24, 2018



Team Member	Contributions
Christian Tabedzki tabedzki@seas.upenn.edu	Data Cleaning (Crime/Weather), Data Stitching, Creating Crime Tabulation, Plotting, Report, Data Scraping
Amruthesh Thirumalaiswamy amru@seas.upenn.edu	Problem formulation, Solution approach, Data Cleaning, Machine Learning Algorithms and analysis, Report
Paul van Vliet paulvv@sas.upenn.edu	Data Cleaning (Zillow), Data Stitching, Machine Learning Algorithms, Report

Abstract

In this study, we use machine learning to predict crime related statistics in Philadelphia. We assembled a dataset with all relevant features, including weather and housing values, and divided the same into 3 main-datasets. The problem objective was divided into 3 parts: determining whether a crime occurs, the occurrences of crime, and the most likely crime. We use different algorithms and model to train these datasets and obtain detailed quantitative crime predictions with more physical significance. This report summarizes the methods, results and relevant analysis. We were able to predict whether a crime will happen with 69% accuracy, how many crimes, ranging from 1 to 32, with 47% accuracy and the type of crime out of 7 major classes with an F1 value of 0.258.

1 Introduction

One reality of urban life is the high frequency of crime. Our project was motivated by a desire to understand the frequency of such events, and the prevalence of each type of crime in various parts of Philadelphia and to identify how safe neighborhoods are around Penn. Identifying trends and causes could then be used with navigational applications to avoid neighborhoods that are riskier than others.

2 Related Work

In preparation for this project, we came across various articles tackling crime prediction in various cities across the country, but none focused on Philadelphia. Previous works in other cities such as San Francisco, predicted the types of crimes occurring in the city based on the assumption that a crime has occurred [1].

We read that the best classifiers for this type of datasets tended to be tree based methods, which allows for the decision based classifications [2, 3].

Additionally, we found other papers that corroborated the random forest model. They were only able to achieve 32% accuracy across 39 different categories [4]. The authors believed the random forest would be the best since the data was highly noisy. In other paper, Chandrasekar *et. al.* were able to achieve accuracy on the order of 80% after they had grouped the data into a series of binary crime classifications [1].

The lack of the quality of predictions in previous works, indicated a need for a deeper analysis in crime prediction and its machine learning perspective. We need a method that determines the probability of a crime happening and if so, its type.

3 Problem Formulation

We decided that, to tackle the problem more efficiently, we need to look at the intrinsic reasons or factors that contribute a crime. Apart from the geographic location and time, we hypothesized weather plays an important role in the occurrence of a crime, which had been ignored by previous works. Further, we believed the economic condition of a neighborhood over time affected the crime statistics of the place. It was examined and concluded that other factors don't significantly effect crime occurrence and thus were ignored and assumed constant over our training and prediction range. Thus we decided to look for features previously mentioned.

We decided to divide the time span of a day into 1-hour intervals and make relevant prediction for that time span. Available datasets indicated how multiple crime incidents were frequent within a time interval within a neighborhood [5]. Further, we saw the need to create synthetic data points to predict the occurrence of a crime. Thus, we developed the following work flow to tackle the problem.

Our method for tackling crime predication was to break down the problem into three, smaller problems. The first objective was to determine whether a crime will happen. To do this, we categorized our crime dataset up by districts and by hour, making it easier for the model to detect any incidents that might arise. We then took the data and calculated how many crimes existed for each unique district and time combination. This became the second dataset used to predict the amount of crimes for any time period from the first dataset that registered as positive.

The third dataset we created and used kept track of the most likely crime. This was created by taking the most frequent crime that occurred on any given day and district. In the instance of a tie, the first most likely crime was the one selected as the most likely.

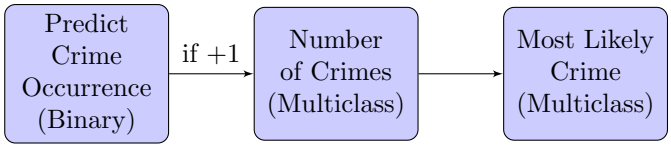


Figure 1: Workflow for predicting crime.

4 Data Set

Our major crime dataset came from OpenDataPhilly[5] storing each of the 2,378,559 crime with a unique identifier and other features including: latitude, longitude, block, category of crime (classifying feature), police district, dispatch date and time, and hour of the crime. The day of the week was extracted from the date of each incident. Further, the city was divided into 22 police districts.

The original range was from 2006 to present. However, only incidents between 2010 and 2017 were considered for data manipulation purposes, reducing the total amount of crimes down to 1.5 million.

Since the National Oceanic and Atmospheric Administration did not have weather records in 15 minute in-

tervals for the years in question, daily weather summaries were used instead and assumed to have extended to the whole day[6]. While allowing us to incorporate larger weather trends, it prevented us from measuring whether poor weather during the crime affected the crime itself, that is, we can capture weather it was raining any time during the day a crime occurred, but not if it was raining during the incident.

Other weather data features included daily precipitation level, snow fall, snow depth, min and max of daily temperature, water equivalent of snow depth. It also included the following binary features: fog and ice fog; heavy fog; thunder; ice pellets, sleet, snow pellets; hail; glaze or rime; dust; blowing or drifting snow; high or damaging winds; mist; drizzle; freezing drizzle; rain; freezing rain; snow, snow pellets, snow grains or ice crystals; ground fog; and ice fog. These features were reduced to fog; thunder; smoke/haze; blowing/drifting snow; high windows; drizzle; snow; hail; and rain. Additionally, sunrise and sunset data were scraped from the Department of Homeland Security's website and appended to the weather dataset [7] to determine whether the day/night cycle had an influence on crime.

To capture some demographic information based, average house value as determined by Zillow was introduced [8] based off of the assumption that poorer neighborhoods were more crime prone. For each crime, the coordinates were matched with a neighborhood and a corresponding housing value was appended.

5 Data Preprocessing

All relevant features from the different datasets were combined judiciously to obtain the three major datasets used in our approach. The following set of features were agreed upon based on their influence towards crime occurrence.

Table 1: A tabular view of all our features. The target values are not included.

Time	Weather	Neighborhood
month, day, sunrise, sunset, hour, day of the week	snow, rain, hail, thunder, fog, drizzle, high winds, min and max temperature, drifting snow, smoke/haze snow depth on ground, rain & snow (binary) water equivalent of snow	police district, house value

The first dataset constructed has all crime listings with the relevant classification code indicating the type of crime. Relevant points from this dataset were combined to form the second data set which has the number of crime incidents within a time interval of 1 hour with a relevant combination of other features. Synthetic data points were added to form the third dataset for every 'non-crime' time interval for every relevant combination of other features. Each data vector has a classification: crime (+1) or non-crime (-1). As discussed earlier, these datasets are inherently very noisy and thus, they were manually and laboriously cleaned up to obtain better predictions.

6 Methods

We use a variety of methods to learn a reasonable model from our different datasets. But as is evident, the prediction goal and thus the kind of models used in each case differs.

The binary classification assignment requires us to minimize the classification error. The multiclass classification to predict the occurrences of crime intuitively calls for a absolute mean loss. Absolute mean loss penalizes incorrect classifications that are further away from the true label so this loss is ideal for predicting the amount of occurrences. However, a 0-1 loss would be a very a harsh metric since you require for each sample that each label set be correctly predicted, thus we lax our requirement for a higher test accuracy. Further, we find a case of class imbalance in our latter multiclass classification, which is generally tackled using a ROC analysis or 'grouping'. We choose to perform the latter.

7 Algorithms

All our machine learning algorithms were implemented via the Scikit-learn package [9], which is used in the Python language. Data cleaning was done in Matlab and figures were plotted using ggplot2 in R [10], except for the overlay on the map, which was done in Microsoft Excel. Further we used the ordinal regression algorithm by Pedregosa to implement ordinal regression for our second dataset[11].

Our report uses various methods for each type of classification: binary classification for prediction of occurrence; multiclass for amount of crime and for the likeliest crime to occur. For each process, we used multiple methods: for the binary classification, we used logistic regression, support vector machines, boosted trees and k-nearest neighbors; for detecting how many crimes happened in any period, we used ordinal logistic regression and regular regression; to detect the most likely crime for each period, we used random forests and boosted trees. Here, we provide a brief discussion of these methods.

7.1 Logistic Regression

Logistic regression is a discriminative probabilistic model that is used to categorize data and is well suited for binary problems because of the shape of the sigmoid expression. We do so by minimizing a loss function of the form:

$$\ell_{\log}(y, f) = \log_2(1 + e^{-yf}) \tag{1}$$

Further, we use L-2 regularisation to avoid overfitting and determine the relevant parameters for the regularisation using cross-validation. The same can be extended for a multiclass setting using a 'one versus rest' approach.

7.2 k-Nearest Neighbors

The k nearest neighbor method is a very simple method: the training dataset is stored and each new instance is measured against the entire training set to determine which training point is closest and categorize the new instance with the same label.

7.3 Ordinal Regression

Ordinal regression is used to predict variable orderings where there might not be a rigid scale but where the order matters, such as a ranking of customer service on a scale from 1–5. For this report, ordinal regression was used to predict the number of crimes happening in any given time and police district. It provided a natural choice since misclassified points further away were penalized more than closer misclassified points. This, using this intuitive loss gives us a model, that gives practically more reasonable predictions.

7.4 Tree Methods

In the multiclass classification of individual crime instances, we use random forests and boosted trees. The problem of classifying individual crime instances lends itself quite well to tree-based methods, since our dataset is a mixture of numerical features (ex. Zillow index, temperature, precipitation) and binary features (ex. weather indicators, time of day dummy variables). In addition, tree methods scale quite nicely to multiclass prediction, as they don't have to rely on an all-vs-one approach or a series of one-vs-one classifiers unlike methods such as SVM. The main difference between random forests and boosted trees is how the trees are constructed. For random forest, fully formed decision trees built independently to vote on a classification. Boosted trees utilizing the AdaBoost algorithm repeatedly classifies using weak learners, increasing the weight on the misclassified points for the learner in the next period.

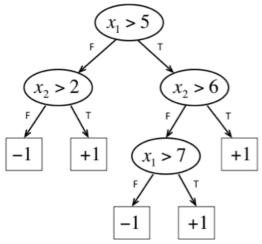


Figure 2: This is an example forest used for classification. This report's forests would focus on time and weather. Adapted from lecture notes of random forests[12].

8 Experimental Design and Results

We designated dataset from 2010 to 2016 as our training set and used 2017 as our testing set.

Our project breaks up the prediction into three distinct parts: a binary classification to determine if a crime will happen given conditions, the amount of crimes that will happen in a given time, and the likeliest crime to happen given information about a particular instance.

8.1 Crime prediction: Binary Classification

We use a variety of binary classification algorithms for the above task namely: Decision trees, Random Forest classifier, Logistic Regression with L-2 regularisation Cross validation, k-Nearest Neighbour classifier, etc. We summarise a few results from our analysis of these different algorithms and their classifiers.

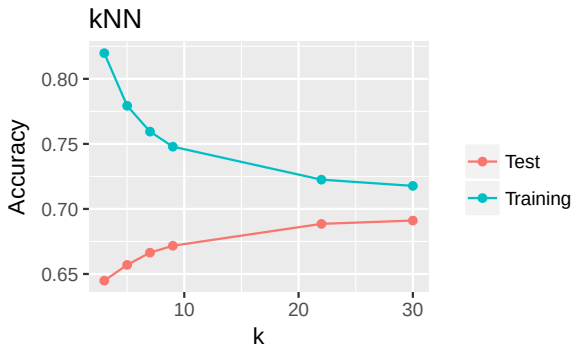


Figure 3: k-nearest neighbor used for binary classification.

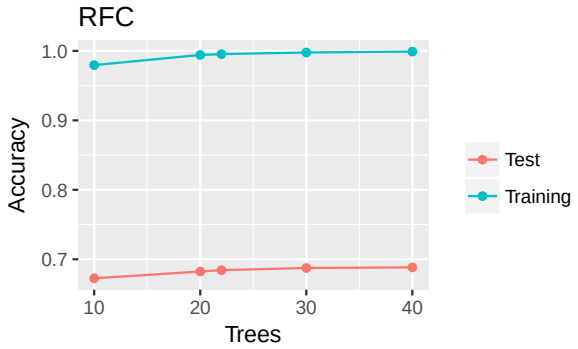


Figure 4: Random forest classifier used for binary classification.

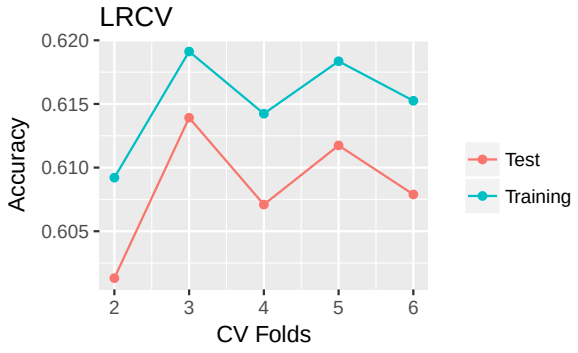


Figure 5: Logistic Regression Cross Validation used for binary classification.

Table 2: Comparison of methods for the binary classification

Method	Training Accuracy	Test Accuracy
LRCV (3 fold)	0.61912	0.61392
RFC (40 trees)	0.99899	**0.68830**
DTC	1.00000	0.61343
k-NN (30)	0.717730	**0.691075**

We get an accuracy as high as ~ 0.7 which given the noisy data setting is a reasonable quantitative value to achieve.

We can see how logistic regression fails compared to the tree and neighbor methods. The same can be attributed to the 'discrete' nature of the dataset. Thus, we avoid using classifiers like Neural Networks and SVMs which work better with a dataset of a 'continuous' nature. The same can be reasoned on following lines: We can see how crime generally occurs within a feature range i.e say crime occurs in West Philly at night. The 'discrete' nature of the dataset allows a Tree classifier and Nearest neighbor classifier to adhere to its details and predict a discrete feature boundary within which the classification is $+1$. Further, it is evident how the k-NN and Random Forest methods asymptotically tend to a value as k or the number of folds $\sim O(n)$ where n being the number of features. Thus each feature is sufficiently captured when the relevant parameters $\sim O(n)$.

8.2 Occurrence of crime prediction: Multiclass classification

Here we basically try to predict the number of occurrences, which range from $1-32$ of crime corresponding to a certain data vector. We tried to do a multiclass classification to obtain the same. However, as discussed earlier an ordinal classifier with an absolute error should be more intuitive and practical to use. We thus use an ordinal logistic regression and compare it with a simple logistic regression. The results are summarized below.

Table 3: Comparison of two logistic models for the number of crimes per day.

	Type	MAE	Accuracy
Ordinal logistic regression	Training	**0.8039**	0.4591
	Testing	**0.7371**	0.4740
Logistic regression	Training	0.85326	0.52447
	Testing	0.7617	0.55626

As we can see, a logistic regression has a stronger accuracy of classification but practically having a better absolute error would make sense, and thus going with the ordinal logistic regression would be the way to go. Thus we forgo on the accuracy but go for a lower absolute error. Further, it can be noticed that these values are reasonable for a classification of $K = 32$.

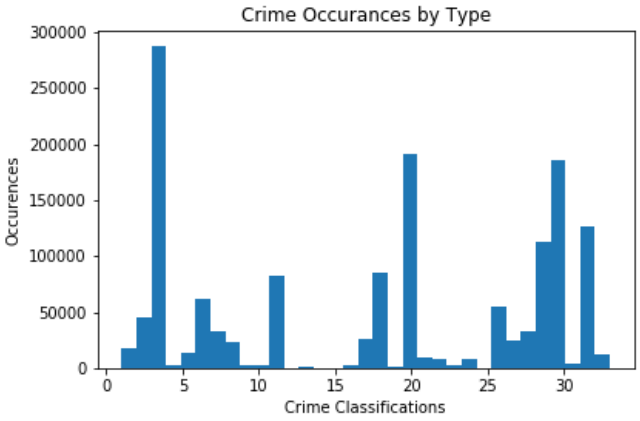


Figure 6: The frequency of the various crime classes. The most frequent crimes in the dataset were the 'miscellaneous.

8.3 Type of crime prediction: Multiclass classification

Here, we try to predict the most likely class of crime occurring for a data vector. The problem involves a multiclass classification with 32 crime types. However, in predicting the type of crime given a particular crime instance, we had to deal with the large class imbalances inherent in the data as shown in Figure 6. There were several pre-defined categories in our data that were much larger as a result of being a catch-all for crimes that didn't get a more specific classification. For example, one of the largest groups is "Thefts," even though there are five other classes that would appear to cover all possible thefts. In some related studies, the authors have aggregated to broader classifications, generally to the binary level [1, 4]. Our goal was a bit more ambitious, so we classified into seven different groups, based around the type of crime. The groups roughly correspond to: "All Other Offenses," "Non-violent Misdemeanor," "Theft," "Unclassified Thefts," "Assault," "White Collar," and "Homicide". The abundance of thefts necessitated the addition of the "Unclassified Thefts" category.

In both our random forest and boosted trees implementations we used cross validation to tune the maximum depth of the trees and the number of classifiers. We also used balanced class weights (inverse of class share) to mitigate the tendency to predict all one class. In predicting the likeliest crime, a standard accuracy measure isn't sufficient to capture the performance due to class imbalance even in the aggregated crime classes. We therefore used a weighted F1 score when evaluating the multiclass classification models.

Table 4: Comparison of different tree techniques with best parameters.

Model	F1 Score	Estimators	Depth
Random Forest	.258	100	20
Boosted Trees	.223	100	5

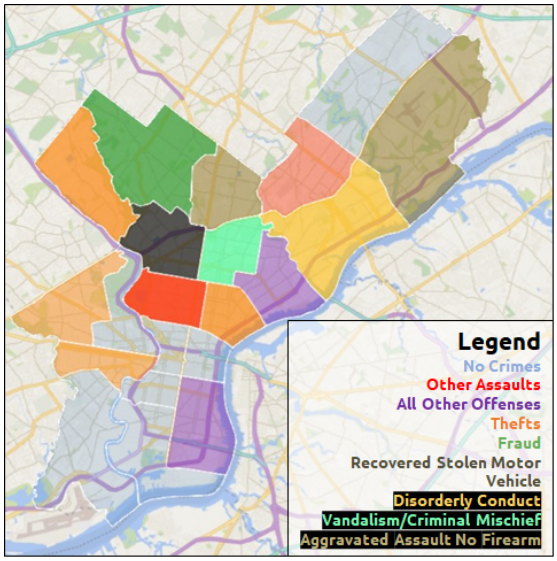


Figure 7: Crime prediction map for during the presentation. More opaque colors means we predict more crimes. The most opaque represents 3 crimes while the lightest represents 0.

9 Conclusion and Discussion

Our above analysis allowed us to make detailed quantitative predictions for crime in Philadelphia. The accuracy values obtained across the three datasets were reasonable within the noisy nature of the dataset and comparable with other similar studies. This allows us to deem the method and approach a reasonably successful one. Thus, we were able to make reasonable successful predictions about whether a crime would occur, if so how many within an hour range and which type of crime would be the most likely one.

Above, we present a map for the crime prediction in the different districts of Philadelphia for the 25th April, 2018 at 1:00 PM, corresponding to the time of the presentation. The color represents the type or category of crime and opacity indicates the occurrence of the same. The gray scale regions are crime free. The predictions seem reasonable as they include all different types of crimes ranging from assaults to cyber frauds and match the general pattern of recorded crime in Philadelphia in that time interval. We look forward to seeing this type of crime predictiono integrated into navigational applications.

10 Recommendations

Further refinement of the dataset could potentially lead to a more accurate model; some recommended features to add are smaller weather intervals (hourly or every 15 minutes), racial composition, per capita income, median age, high school dropout rate, average daily attendance, and SAT and ACT scores. While this information may be correlated to other features, these recommendations are believed to help capture the data and look forward to any future results utilizing these recommendations. We would also suggest using the temporal correlation.

Acknowledgments

We would also like to acknowledge the Prof Shivani for teaching this course; Nikos, for all the late night help; Simeng for helping us with the project; Professors Riggleman and Crocker for recommending this course to us.

References

[1] Addarsh Chandrasekar, Abhilash Sunder Raj, and Poorna Kumar. Crime Prediction and Classification in San Francisco City. URL http://cs229.stanford.edu/proj2015/228{_}report.pdf.

[2] Andrey Bogomolov, Bruno Lepri, Jacopo Staiano, Nuria Oliver, Fabio Pianesi, and Alex Pentland. Once Upon a Crime: Towards Crime Prediction from Demographics and Mobile Data. sep 2014. URL https://arxiv.org/pdf/1409.2983.pdfhttp://arxiv.org/abs/1409.2983.

[3] Aziz Nasridinov, Sun Young Ihm, and Young Ho Park. A decision tree-based classification model for crime prediction. In *Lecture Notes in Electrical Engineering*, volume 253 LNEE, pages 531–538, 2013. ISBN 9789400769953. doi: 10.1007/978-94-007-6996-0-56.

[4] John Cherian and Mitchell Dawson. RoboCop: Crime Classification and Prediction in San Francisco. http://cs229.stanford.edu/proj2015/254_report.pdf, 2015.

[5] City of Philadelphia. Crime Incidents - Crime Incidents (CSV) - OpenDataPhilly. URL https://www.opendataphilly.org/dataset/crime-incidents/resource/c57a9de2-e300-468a-9a20-3e64e5b9b2da.

[6] NOAA. Datasets | Climate Data Online (CDO) | National Climatic Data Center (NCDC). URL https://www.ncdc.noaa.gov/cdo-web/datasets#GHCND.

[7] Department Of Homeland Security. MyTSA API Documentation | Homeland Security, Nov 2017. URL https://www.dhs.gov/mytsa-api-documentation.

[8] Zillow Inc. Data - Zillow Research . URL https://www.zillow.com/research/data.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[10] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2009. ISBN 978-0-387-98140-6. URL http://ggplot2.org.

[11] Fabian Pedregosa. Logistic ordinal regression. http://fa.bianp.net/blog/tag/ordinal-regression.html, Website: I say things - ordinal regression.

[12] Shvani Agarwal. Decision trees and nearest neighbor methods, Feb 2018. URL http://www.shivani-agarwal.net/Teaching/CIS-520/Spring-2018/Lectures/Reading/decision-trees-nearest-neighbor.pdf.