Patterns and Predictions of Crime in Chicago

Matthew Yeh
Princeton University
matthewy@princeton.edu

May Jiang
Princeton University
mayjiang@princeton.edu

Russell Slighton
Princeton University
rts3@princeton.edu

Abstract

The objective of this work is to uncover exploitable patterns and latent structure in crime incidents in Chicago over the past few years. To this end, we analyze Chicago's public crime dataset using a range of techniques including classification, latent variable, and time series models. We then consider the consequences of our results in the context of policy and policing.

1 Introduction

While crime rates for the United States, on the whole, have fallen in the past few decades, the crime rate - particularly, the violent crime rate of Chicago, remains worryingly high, hovering around three times the national average [2]. As of yet, experts do not agree on the cause [3] which makes this problem rather difficult to address. The goal of this project, therefore, is to analyze Chicago's public crime dataset [7] to discover exploitable statistical structure capable of informing policing and policy counter-measures. To this end, we produced several models to predict whether crimes result in arrests and whether crimes are violent using a selection of classification techniques. We then move on to examine Chicago crimes' latent structure via LDA, PCA, KMeans, and GMMs. Next, we predict weekly crime rates using an autoregressive model. Finally, we examine the temporal relationships between types of crimes using Markov models.

2 Related Work

In an effort to prevent crime and maintain public safety, predictive policing tactics are being used increasingly often across the United States [4]. Consequently, the police are turning to big data to inform their decisions. Professors at Northwestern University utilized a combination network of Recurrent Neural Networks and Convolutional Neural Networks (Recurrent Convolutional Network) to predict one of twelve types of crime and achieved a 75.6% accuracy in Chicago and 65.3% accuracy in Portland [10]. They used RNNs as crime tends to influence the crime following it and used CNNs to reason about spatial data. Nathan Holt, a professor at Rochester Institute of Technology, utilized KMeans clustering in an attempt to predict crime [5]. Holt was unable to discern meaningful clusters from the data. Our work seeks to achieve many of these same goals. However, we do not use neural models and instead turn to more traditional statistical techniques.

2.1 Data

To facilitate this project, we are using the Kaggle Chicago Crime Dataset [7], which contains all crimes committed in Chicago from 2001 to the week prior to this week and is continuously updated. It contains information about the type, location (down to the block), and time (to the hour) of crimes committed, along with some other helpful details, including whether an arrest was made. However, the full dataset was extremely large, at 1.8 GB, so because of our limited computational power we used only the data since 2016 for the time-series regression models and the data since 2018 for the other models. The data contained 343,016 crimes since 2018 and 881,689 crimes since 2016.

3 Approach

3.1 Data Processing and Feature Selection

The dataset was very rich in features, with numerous columns describing the location of the crime - district, ward, community area, beat, block, x and y coordinates, and latitude and longitude (in increasing order of specificity), and the type of crime - primary type, secondary type, as well Illinois Uniform Crime Reporting codes (IUCR), which are four-digit codes that law enforcement agencies use to classify criminal incidents and also FBI code, a slightly less specific version of IUCR [7].

Because we had so many different models, we ran two different preprocessing pipelines. Our first pipeline prepared data for all models save for the Markov models. To begin this pipeline, we threw away the many different types of location except for community area. The rest were too specific and would not generalize. We also threw away unique identifiers such as case number. We also threw away repeat information such as the FBI code, primary type, and description as all of these were covered in the categorical variable of IUCR. Next, we manually reduced the location descriptions from 134 very specific descriptions, such as grocery food store and YMCA, down to 12 key categories, listed along with their counts in Table 7 in the Appendix. Furthermore, we loaded descriptions of the community areas' income levels and divided them into low, medium, and high based on groups of equal size [8]. Next, we manually determined which crimes were violent and nonviolent using IUCR. We also converted the date variable to time of day in 3-hour intervals. Finally, we one-hot encoded the categorical columns into a binary variable representation.

For the Markov models, we used a slightly different process. We began by classifying the 22 primary_type codes into eight broader categories of crimes, including, violent crimes, property crimes, narcotic crimes, public order violations, etc. These 8 categories were then factorized (not one-hot-coded). Next, we sorted each crime ascending order of time. Finally, we introduced a spacial dimension to the data by grouping crimes according to community area which was found to preserve the most entropy in the eight crime categories of all measures of locality. This process yielded many sequences of time ordered crime types defining criminal behaviour in a given geographical area.

3.2 Methods

We evaluated several different supervised and unsupervised methods for this task, using the Scikit-learn [9] and hmmlearn [1] implementations and defaults unless otherwise specified.

Classification
1. Naive Bayes Classifier: Using multinomial implementation
2. Decision Tree Classifier: Using Gini impurity scores, best split at each node
3. Random Forest Classifier: Using Gini impurity scores, 100 trees
4. Support Vector Machine (SVM): Using linear kernel
5. Logistic Regression: Using stochastic gradient descent

Latent Variable Models and Dimensionality Reduction
6. KMeans: Using KMeans++ for initialization
7. Latent Dirichlet Allocation (LDA): Using Batch variational Bayes method
8. Principal Component Analysis (PCA): Using full Singular Value Decomposition
9. Gaussian Mixture Model (GMM): Using full covariance

Time Models
10. Linear Regression - Autoregressive Model (AR): Using ordinary least squares
11. Markov Chain: On sequences of consecutive crime types by community area
12. Hidden Markov Model (HMM): Multinomial emissions

3.3 Evaluation

Save for the hidden Markov and autoregressive models, we evaluated our models by hold-out validation under a random 80:20 train-test split for the classification and unsupervised learning

models. For the time-series regression, we trained the model using the data from 2016 to April 2018 and tested on held-out data over the last year, from April 2018 to April 2019. For the Markov models, we produce a 50-20-30 split after ordering crimes by time; thus, we trained the models on the earliest 50% of the data, selected hyper-parameters of the middle 20%, and tested the models with the most recent 30% of crimes.

We evaluated the performance of our classifiers using accuracy, precision, recall, F_1-score, and AUC. Denoting the number of false positives (FP), true positives (TP), false negatives (FN), true negatives (TN), we define these measures below. ROC curves plot the true positive rate against the false positive rate over varying discrimination thresholds, and the area under the ROC curve, AUC, measures how well a parameter can distinguish between two diagnostic groups. $accuracy = \frac{TP+TN}{TP+TN+FP+FN}$, $precision = \frac{TP}{TP+FP}$, $recall = \frac{TP}{TP+FN}$, $F_1 = 2\frac{precision*recall}{precision+recall}$

To evaluate across Latent Variable models and Dimensionality Reduction models we used Log likelihood. We also used reconstruction error on the test sets to compare across models. To ensure that we were not overfitting, we calculated the reconstruction error on the training set to see if we got similar numbers. Lastly, to pick the best K for KMeans, we used Silhouette Score which is:

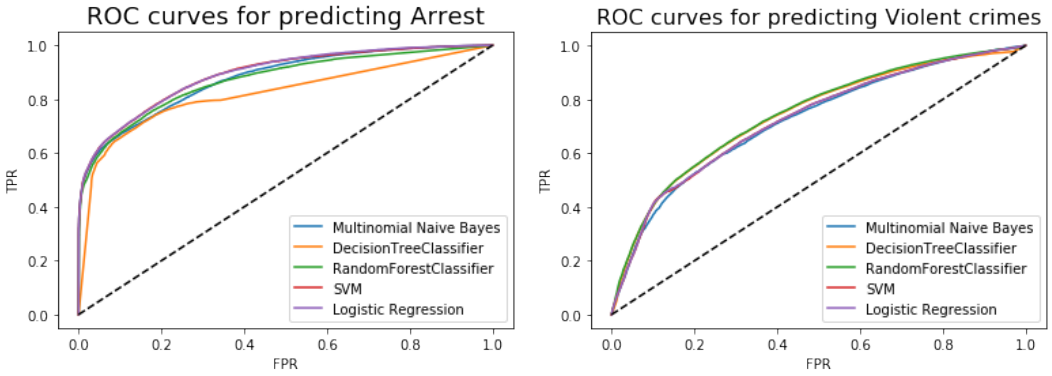
$$\frac{b-a}{max(a,b)}$$

where a denotes the mean intra-cluster distance and b denotes the mean nearest-cluster distance.

For our time series regression models, we evaluated our models on the test set based off of R-squared and Root Mean Square Forecast Error (RMSFE). RMSFE is defined as: $\sqrt{E[(Y_{T+1} - \hat{Y}_{T+1|T,T-1...})^2]}$

4 Results

4.1 Classification



We built 5 different classification models to predict whether a crime was violent and to predict for a given crime whether an arrest was made. To predict whether an arrest was made, we trained on a feature set including the time, type of location, community area, and IUCR code for type of crime. To predict whether a crime was violent, we trained on the same feature set, including the time, type of location, and community area, but excluding the IUCR code. We found that all the models performed relatively well, with logistic regression and the random forest classifier the best predictors of whether an arrest was made and the random forest classifier and SVM the best predictors of whether a crime was violent. We attribute the strong performance of random forests to their flexibility in accommodating large numbers of predictors and their ability to handle heterogeneity and to capture interactions among predictors [6]. Further, the random forest model enabled us to extract the features that contributed most to decreasing Gini impurity scores. Inspecting the top features, we found that the most informative features for predicting whether an

arrest was made were the IUCR codes denoting a weapons violation for unlawful possession of handguns, possession of heroin, cannabis, and crack, and theft of $500 or less. The most predictive features for whether a committed crime was violent were whether the crime was domestic, occurred at a residential location, on a street or sidewalk, and the indicator for time of day for the hours around midnight.

Model	Accuracy	Precision	Recall	F1	AUC
Multinomial Naive Bayes	0.889	0.829	0.563	0.670	0.877
DecisionTreeClassifier	0.877	0.896	0.656	0.758	0.821
RandomForestClassifier	0.883	0.889	**0.668**	**0.763**	**0.891**
SVM (Linear kernel)	0.891	**0.897**	0.516	0.655	0.890
Logistic Regression	**0.892**	0.891	0.523	0.659	**0.891**

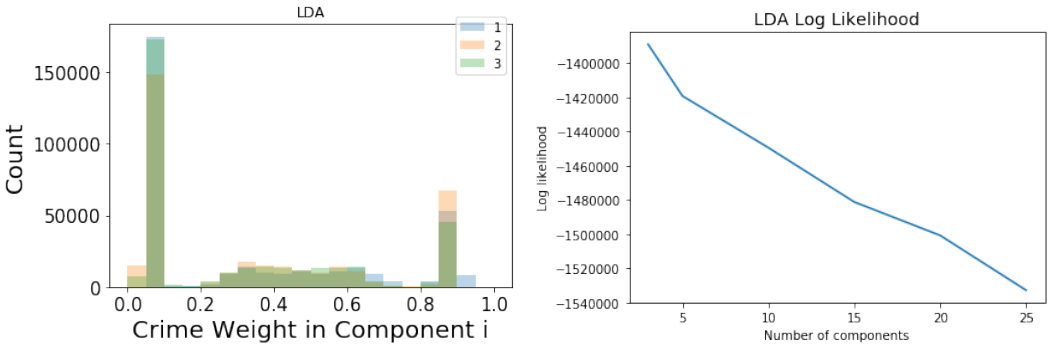
Table 1: Results for Predicting Arrest

Model	Accuracy	Precision	Recall	F1	AUC
Multinomial Naive Bayes	0.711	0.648	0.433	0.519	0.720
DecisionTreeClassifier	0.723	0.676	0.440	0.533	0.739
RandomForestClassifier	0.724	0.671	**0.456**	**0.543**	**0.745**
SVM (Linear kernel)	**0.725**	**0.691**	0.426	0.527	0.726
Logistic Regression	0.724	0.688	0.426	0.526	0.726

Table 2: Results for Predicting Violent crimes

4.2 Latent Structure

4.2.1 LDA

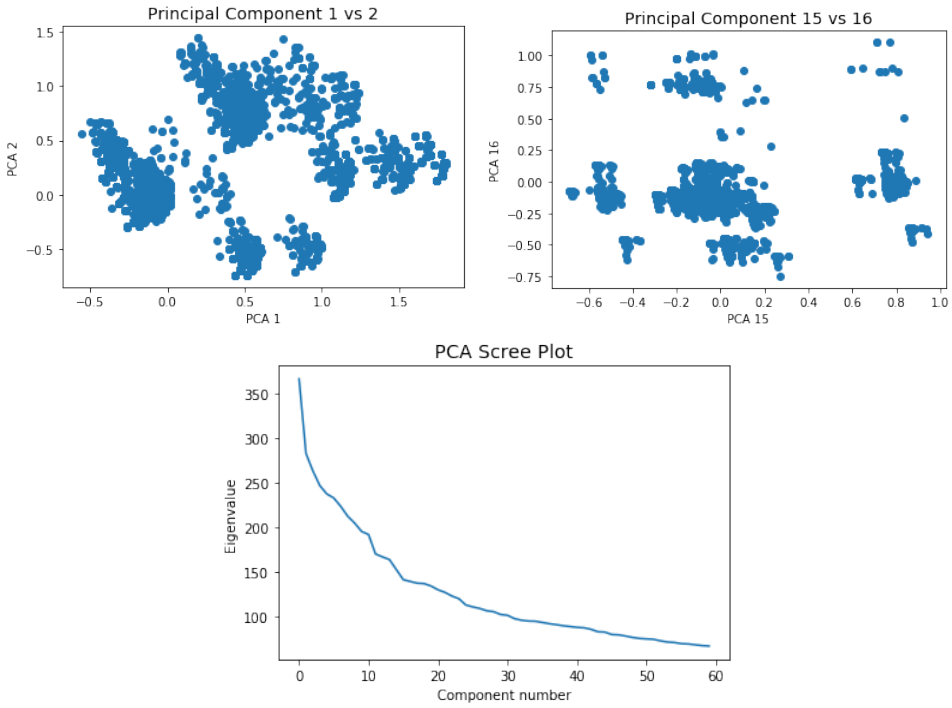


Component 1	Component 2	Component 3
Residential	Street	Store
Violent	Transportation	Sidewalk
Domestic	Location_sketchy	Violent
Domestic battery	Theft	Battery
Hours 6am-12pm	Night time	Night time
Property damage	Arrest likely	Arrest likely

Table 3: Top Features in LDA Components

We found that the optimal number of components for LDA was 3, as increasing the number of components beyond 3 only decreased the log likelihood. Nonetheless, the 3 components produced by LDA were semantically meaningful and easy to interpret. The top features by feature weight in each component are listed in table 3; we see that the first latent crime represents residential, violent crimes such as domestic battery, the second latent crime represents non-violent crimes committed on the street or related to transportation, such as motor vehicle theft, and the third latent crime represents violent crimes committed in stores or sidewalks at night such as battery, that are likely to result in an arrest.

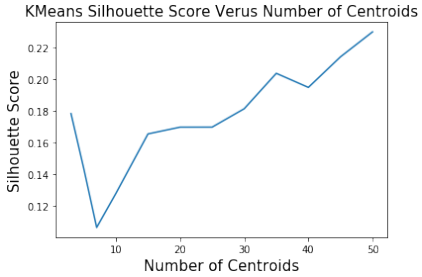
4.2.2 PCA



PCA was successful at creating semantically meaningful components. From the elbow in the scree plot, we determined that 16 was the optimal number of components to retain. Inspecting these principal components by projecting the data onto them, we found that many of the components separated crimes by time of day. We attribute this to the tendency of crimes committed during a certain time of day to be drastically different from one another. Additionally, we found that principal component 1 separated residential crimes such as domestic abuse from crimes located on the street or related to transportation such as vehicle trespass, principal component 15 separated crimes related to drugs and weapons that happen at night from crimes such as fraud and electronic harassment that occur in the morning, and principal component 16 separated violent crimes such as assault that result in arrests from financial theft and crimes in stores.

4.2.3 KMeans

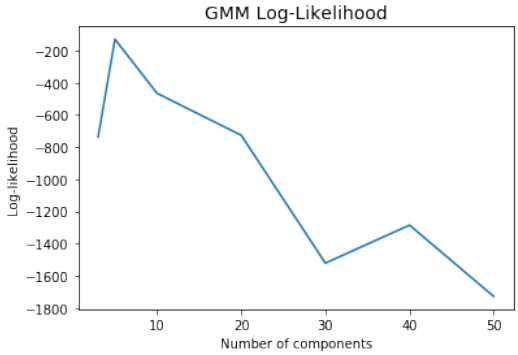
We selected the optimal number of centroids based on silhouette score and found that KMeans performed best with 50 centroids. Manually inspecting the centroids, we found that several were semantically meaningful as they differentiated some of the most common types of crime, as listed in Table 4. Unfortunately, this is likely due to the large number of centroids and not the ability to capture meaningful structure overall.



Centroid Number	Characteristics
3	store crimes
4	arrest, street, late night, drugs
5	violent, residential, non-domestic
7	violent, domestic, residential, late night

Table 4: Top Features in KMeans Centroids

4.2.4 GMM



We found that using a GMM with 5 components maximized log-likelihood. However, by manual inspection we found no meaningful structure in the components produced.

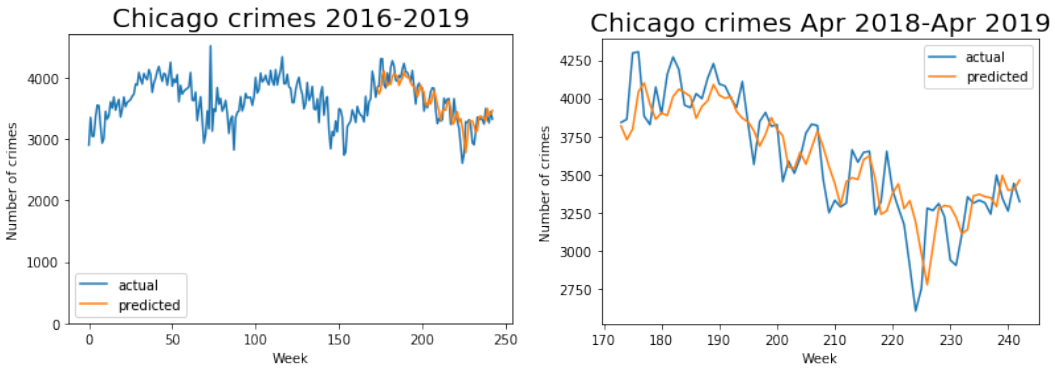
4.2.5 Comparison of Models

Model	Components	Train Error	Test Error	Average Log-Likelihood	Accuracy
LDA	3	5.765	5.767	-11.477	0.6385
PCA	16	**0.158**	**0.158**	**-0.819**	0.6860
KMeans	50	1.046	1.045	N/A	0.6385
GMM	5	3.596	3.598	-128.85	0.6385

Table 5: Comparison of Unsupervised Learning Models

Comparing our latent variable and dimensionality reduction methods, we found that PCA performed the best in terms of train and test error, as well as average log-likelihood. All of our unsupervised models besides GMM were able to produce meaningful and interpretable latent components. Across all the models we found comparable train and test reconstruction errors, which indicates that our models are not overfitting to the training data. We also wanted to see if any of these latent models would be able to produce meaningful results for the prediction of violent crimes. Unfortunately, all of these models performed significantly worse than just fitting classifiers to the dataset in terms of accuracy. All of the models except for PCA predicted the most common results (not-violent) for everything. PCA performed only slightly better. Consequently, we decided not to proceed further and graph the ROC curves or measure AUC scores.

4.3 Time Series Regression for Crime Rate



To predict the next week's crime rate from past crimes, we use the autoregressive model, which regresses a time series on previous values from that same time series. This allowed us to take advantage of the temporal structure of crime frequency, which we can clearly see in the graph of crimes reported in Chicago from 2016 to present. In particular, we noticed that there was a visible

periodic pattern in crime rate over the years. As a result, we used autoregressive models that would predict the next week's crime rate using this week's crime rate, last week's crime rate, the crime rate from this week a year ago, and the crime rate from a week after this week a year ago. The models we used were:
AR 1. $Y_t = \beta_0 + \beta_1 * Y_{t-1}$
AR 2. $Y_t = \beta_0 + \beta_1 * Y_{t-1} + \beta_2 * Y_{t-2}$
AR 3. $Y_t = \beta_0 + \beta_1 * Y_{t-1} + \beta_2 * Y_{t-2} + \beta_3 * Y_{t-year}$
AR 4. $Y_t = \beta_0 + \beta_1 * Y_{t-1} + \beta_2 * Y_{t-2} + \beta_3 * Y_{t-year} + \beta_4 * Y_{t+1-year}$

Evaluating our models by training on data from 2016 to April 2018, and testing on the last year's data from April 2018 to April 2019, we found that all the autoregressive models greatly outperformed a baseline of predicting the mean of 3611 crimes, for both R^2 and RMSFE.

Further, to determine whether the models would better predict crime rate of a specific area, or violent crimes in particular rather than crimes of all kinds, we built separate models to predict the crime rate for each district, for each community area and for each ward. We found that these more specific models did not perform as well as the general model for all types of crimes reported in all of Chicago. In particular, the maximum R-squared for the models by district is 0.507, by ward 0.487, and by community area even lower, at only 0.478. The R-squared for violent crimes was 0.700, slightly lower than that for all crimes. These results suggest that the structure of crimes in Chicago is better characterized with respect to the city of Chicago as a whole, rather than by specific local area.

Model	R^2	RMSFE
Mean	0.000	393.91
AR 1	0.703	214.56
AR 2	0.714	210.56
AR 3	0.726	206.26
AR 4	0.745	198.76

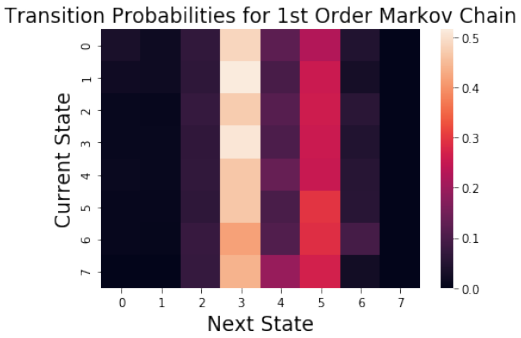
Table 6: Results for Predicting Crime Rate

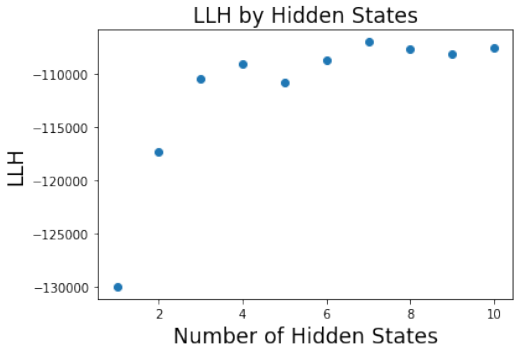
4.4 Markov Models and Temporal Structure of Crime Types

One of our hypotheses was that some crimes might be temporally related to other crimes. For instance, it might be plausible that narcotic crimes could lead to violent crimes, or perhaps that violent crimes might lead to more violent crimes via retribution attacks. To test this hypothesis, we modeled Chicago's crimes as multiple time series - one for each community area - which we then used as training sequences for two Markov models; if the models displayed interesting structure, this would tend to support the hypothesis that some crime types might be temporally related.

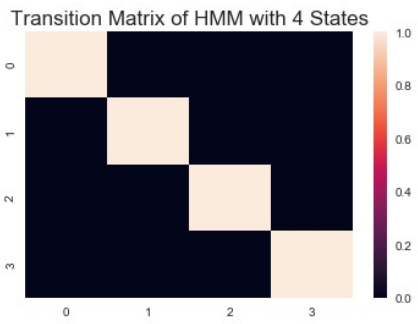
Our first model was a Markov chain. Examining its transition matrix, some structure is fairly evident. Specifically, there are clear bars of similar probabilities (running vertically) for each next state. This means that all crime types tend to transition to other crime types with similar distributions. Unfortunately, this structure is not so much evidence for temporal relationships between crime types as much as it reflects imbalances in the dataset. For instance, crime type 3 (property crimes), was the most frequent type of crime in the dataset, and also the most likely to be transitioned to by all other crime types. A similar relationship held for the second most common crime type (5 - violent crimes), which was transitioned to second most often by all crime types.

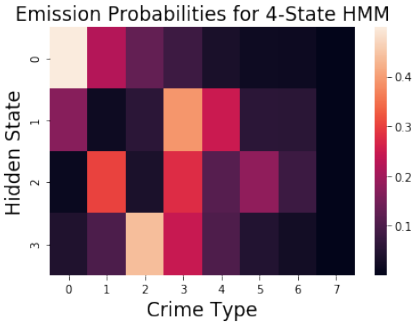
Seeing no interesting manifest structure, we wondered whether Markov models could pick up any latent structure. Thus, we also ran the same training sequences through a hidden Markov model. Using a validation set, we determined that performance (measured by log likelihood (LLH)) plateaued with four or more hidden states, and so, to avoid over-fitting, we initially decided to test our hypothesis using an HMM limited to four states. This HMM's performance was not very impressive. When used to predict next crime types, it exhibited an $F1$ score of 0.31, which incidentally was worse than always predicting the most frequent type of crime.





Unsurprisingly then, the HMM also did not reveal any interesting latent structure. Examining its transition matrix, we found only a near diagonal matrix, implying that all hidden states were self transitioning. It is not immediately clear why this would be the case, so we looked at the HMM's emission probabilities matrix for hints. Unfortunately, this step was not very informative. Since the transition matrix was essentially the identity matrix, we would expect the emission matrix to in some way approximate the distribution of crime types in the dataset (that is, look a lot like the transition matrix for the Markov chain). However, this clearly did not turn out to be the case. We could reason about why hidden states might have their given emission vectors, but we could not explain why the transition matrix might be so diagonal. We suspected the odd transition matrix might simply be an example of underfitting, and so we also constructed models with more hidden states. However, even up to 20 states (which was the most we tried), we found that the HMM always produced a (near) identity transition matrix. Thus, without evidence for temporal relationships between crime types, we must tentatively conclude against them.





5 Discussion and Conclusion

In this paper, we found that many classifiers were successful at predicting violence and arrests. Additionally, we found that LDA, PCA, and to some extent KMeans were successful at creating semantically meaningful components. Unfortunately, none of these were close to achieving the predictive power of the non-latent variable models. We were further able to create a predictive time-series model to determine the number of crimes in the next week. We also attempted to understand causal relationships between the types of crimes and were unable to find interesting transitions or latent variables in the model. In summary, we were able to find some strong predictive models and some meaningful components but not from the same model. Our most robust model was the AR, which accurately predicts crime rates. This could be used to better allocate resources and planning.

We see several ways that further research could enhance our analysis. For instance, future work could investigate how violent crimes and narcotics crimes influence future crimes within a community area and a neighboring community area, if at all. Another factor to investigate would be the current allocation of police resources, which could be confounding why arrests are more likely and why small narcotics crimes are more likely in certain areas. Therefore, we currently do not have the information necessary to make recommendations in terms of over or underpolicing or how to better allocate resources. In addition, we were able to work with a rich dataset. Unfortunately, we did not have the computational power to work with more than 10% of the dataset. It would be interesting to see how utilizing data from previous years impacts the predictive power of the models.

Acknowledgments

Kaggle's data collection was critical to our research. We would like to thank the course staff and Professor Engelhardt for helping us and teaching us all throughout the semester.

References

[1] hmmlearn. https://hmmlearn.readthedocs.io/en/latest/tutorial.html.

[2] Crime in chicago. https://en.wikipedia.org/wiki/Crime_in_Chicago, Apr 2019.

[3] Matt Ford. What's actually causing chicago's homicide spike? https://www.theatlantic.com/politics/archive/2017/01/chicago-homicide-spike-2016/514331/, Jan 2017.

[4] Caroline Haskins. Dozens of cities have secretly experimented with predictive policing software. 2019.

[5] Nathan Holt. Analyzing crime in chicago through machine learning. 2017.

[6] Kathryn L. Lunetta. Screening large-scale association study data: exploiting interactions using random forests. *BMC genetics*, 2004.

[7] City of Chicago. Crimes - 2001 to present. https://data.cityofchicago.org/Health-Human-Services/Per-Capita-Income/r6ad-wvtk.

[8] City of Chicago. Per capita income. https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2.

[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] Alexander Stec and Diego Klabjan. Forecasting crime with deep learning. *stat.ML*, 2018.

Appendix

Location Description	Count
Airport	1430
Bar	4263
Hospital	1912
Other	16961
Parking Lot	913
Residential	111516
School	7344
Sidewalk	26372
Sketchy	21418
Store	43774
Street	76147
Transportation	30966

Table 7: Location Description Categories and Frequencies

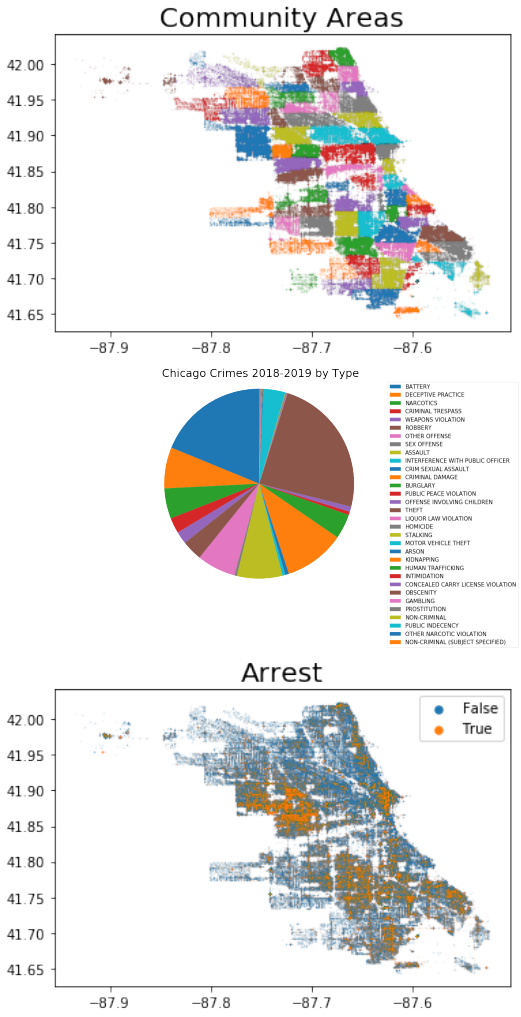


Figure 1: Crimes by community areas and arrests