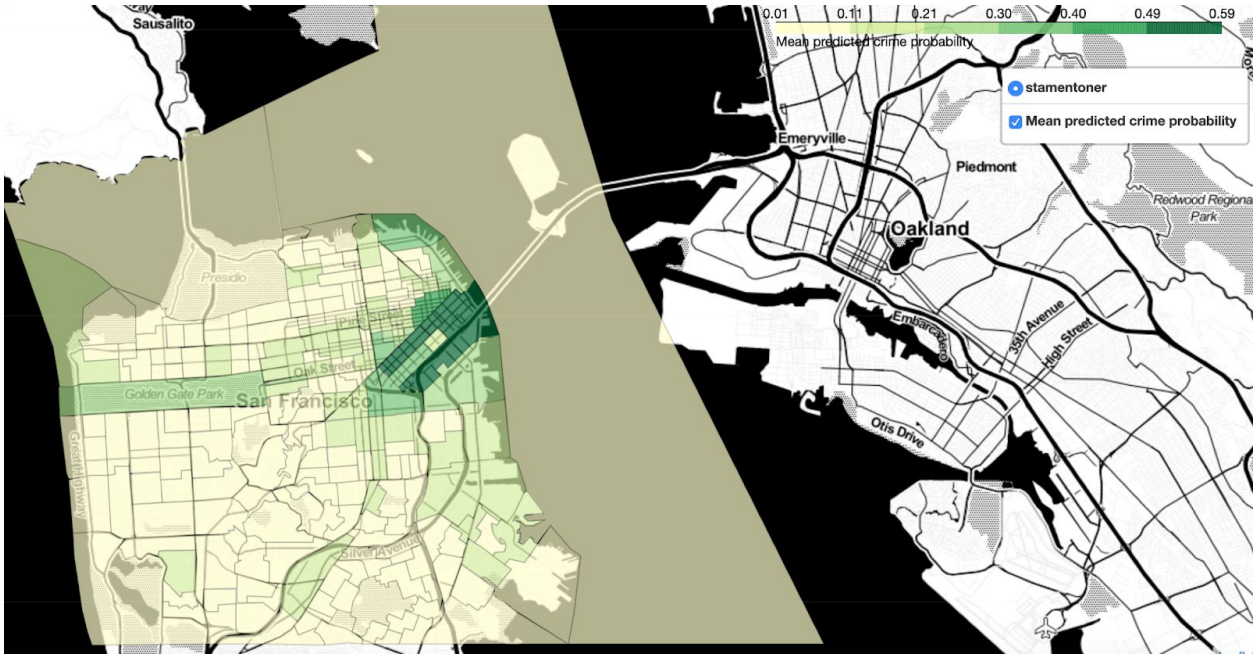
Predicting Crime in San Francisco

Using Open SF Data to Explore Issues with Algorithmic Crime Prediction



John Iselin, Takuma Kinoshita, and Naoyuki Komada, with Ted Kumagai

Due May 9th, 2019

INFO 254 - Data Mining and Analytics[1]

[1] See here for code, documentation, and results: https://github.com/johniselin/info254_sfo_police

INTRODUCTION

Over the past half decade, the development of machine learning techniques and the expansion of access to granular policing data has lead the way for the emergence of data-driven crime prediction programs. In the U.S, the Los Angeles Police Department and UCLA have collaborated to create crime prediction software called "PredPol"[2]. This software uses machine learning techniques to predict when and where specific crimes are most likely to occur.[3] It has proven effective in aiding certain departments in reducing some types of crime - the Jefferson County Sheriff Department saw a 24% reduction in robberies and a 13% reduction in burglaries after the adoption of this technique.

In Germany, IfmPt (the Institut für musterbasierte Prognosetechnik, Institute for pattern-based Prediction Technique) has developed Precobs (pre crime observation system). This system can forecast probabilities of future "near repeat" burglaries based on reports by residential burglary victims and police[4]. As opposed to PredPol, Precobs uses more detailed observation such as stolen goods, methods of entry and locality (type of house and neighborhood). These two programs chose not to reveal the details of their methodologies, though they are clear that their methods make use of machine learning techniques.

On the academic front, Stec and Klabjan (2018) use recurrent neural network methods[5] to aid in crime prediction. Taking advantage of deep neural networks, they tried to predict next day crime count predictions in a grid city partition of Chicago and Portland. They used not only these cities' crime data but also data on weather, census demographics, and public transportation. Their best accuracy score of crime count was 75.6% and 65.3% for Chicago and Portland. These approaches target not specific person who commits crime but specific place where crimes are more likely to occur. We took this approach, tried to create a model to crime prediction and discuss challenges and limitations of the model.

---

[2] https://www.predpol.com/
[3] This method uses data on crime type, crime location, crime date and time, and the produces state that they only use these features to protect the privacy and civil rights of residents.
[4] https://link.springer.com/article/10.1007/s41125-018-0033-0
[5] https://arxiv.org/pdf/1806.01486.pdf

While there is considerable excitement within both the criminal justice and machine learning communities on the potential for data tools to add to the efficacy of policing, there are some concerns about these programs raised by groups like the American Civil Liberties Union. The ACLU has publicly commented saying that algorithms that use historical data from police activities will reinforce attitudes about which area is "bad " or "good".[6] In other words, basing policy on crime predictions derived from arrest data that reflect biased historical police behavior could perpetuate biased police activities. Further, models that predict crime for the express purpose of allocating police resources to high-risk areas do not take into account potential negative externalities of additional police activity in certain communities, particularly communities of color.

The goal of our research is to create a model to crime prediction with open data from City government of San Francisco. Our model will predict whether or not - based on the last 24 hours of crime data - a particular census tract is likely to see criminal activity. Our goal is to use this model to better understand how criminal justice communities want to use machine learning to improve policing, and whether and how issues of bias and over policing may make machine learning techniques inappropriate for this task. The latter problem may be beyond the scope of this project, but it is important to state that we are unsure at the outset if these models are providing a net positive for the communities that will be affected by them.

Data

Our data primarily come from Data SF, the open data portal for San Francisco. From here, we accessed four datasets. First, we used data on the San Francisco Police Department Incident Reports, divided into two time periods, Police Department Incident Reports: Historical 2003 to May 2018, and Police Department Incident Reports: 2018 to Present. These two datasets were organized at the level of individual arrests, and contained the date, time, and location (in latitude and longitude) of the arrest. These data also contained information on the incident type (what was the incident that sparked the arrest) and the crime (what was the arrest for). These two are different, as an incident may result in an officer to checking to see if there are outstanding warrants for a suspect, meaning an individual may be arrested for an offence unrelated to a specific incident.[7]

---

6 https://www.smithsonianmag.com/innovation/artificial-intelligence-is-now-used-predict-crime-is-it-biased-180968337/

[7] Future work could also use Police Calls for Service as an input in the machine learning process.

Second, we used data on the geographic boundaries of 2010 Census Blocks and 2010 Census Tracts, also from Data SF. These boundaries were used to assign specific incidents to census tracts and blocks, so that we could predict if crime occurred in a specific geographic area. In the interest of time, we used census tracts for our model, but our model could easily be extended to the more granular census blocks.

We also use data on daily precipitation and temperature for the San Francisco Area from the NOAA's National Centers for Environmental Information.[8] These data are by day and by detection station. There are multiple stations in San Francisco, so our original plan was to assign census tracts to the closest station, but this turned out to be computationally intensive, so we simply averaged the daily precipitation and used that for all areas in our model. However, we included code that could carry out this matching process in our submission.

Our final dataset was created via the following process.[9] First, we used the Data SF API to download the crime data, merged the two time periods, and dropped unnecessary features. Second, we then cleaned it to prepare for manipulation using Geopandas. Third, using Geopandas, we mapped arrests onto census tracts. Fourth, we merged in the average precipitation data by day. Fifth, we updated the time and date formatting of our data, and collapsed the dataset by 1- and 2- hour chucks by census tract, so that we had data on the presence or absence of crime in a census tract during that time period, as well as a measure of the count of crime by different time slices examined in our Data Visualization section. At the end of this process, our data had 8 features per census-tract, 2-hour period pairs: the binary for the presence or absence of crime, as well as the year, month, week of year, day of week, dummy variables indicating weekend, hour, and precipitation

Introduction to Approaches

For this project, we chose to utilize a Recursive Neural Network (RNN) to predict the presence or absence of an arrest in a census tract during a two-hour period, based on the 12 previous 2-hour periods from the same census tract.

---

https://data.sfgov.org/Public-Safety/Police-Department-Calls-for-Service/hz9m-tj6z

[8] See here: https://www.ncdc.noaa.gov/cdo-web/

[9] This process takes place in three notebooks: dataset_creation.ipynb, dataset_modification.ipynb, and dataset_feature_engineering.ipynb. There are some elements of those files that are not utilized in the binary crime estimation model, because we tried to develop functionality for either a binary or multi-category model.

Our project started with designing a process to utilize the SF Data API to download our data, as is described in John's section below. We then had to clean and merge our datasets together, after which we when through a process of feature engineering (John, Takuma, and Naoyuki). We then conducted some exploratory data analysis to make sure our features would be useful in the prediction process (Ted). We then transformed out data into a format that we could feed into a RNN (Takuma and Naoyuki), and ran our model through Optuna to prune unpromising versions of our model (John). [10] Finally, we ran our preferred RNN (Takuma and Naoyuki), and described the results (John and Ted).

Individual Work

John Iselin

Our first task was to create our dataset, which - as described above - was constructed by combining arrest data from the San Francisco Police Department with weather data from the NOAA and geographic data on census tracts from the US Census bureau. I constructed our first notebook - dataset_creation.ipynb - that used the Socrata Open Data API (SODA) capacity of Open SF to download the arrest records. This was necessary because the individual csv files were too large to store on github, and we want to allow researchers or data scientists the ability to replicate our work with only the resources on github. Because of data throttling issues, I had to set up a loop to pull in our data in small sections. This notebook also cleaned and merged the two datasets, reconciling some technical differences between the two time periods (pre and post 2018) in the process.

I also worked with Takuma, Naoyuki, and Ted to construct dataset_modification.ipynb, which collapsed our data so that we had the count of arrests - for property, violent, and other crimes - for each census tract during two-hour periods. The full process is more fully described in the data section above.

Once Takuma and Naoyuki put together the framework for our LSTM model, I constructed model_tuning.ipynb, where I used Optuna to select parameters for our model - namely, the learning rate (lr) for the adam optimizer, and the dropout rate. We used Optuna because it is designed to optimize selected hyperparameters - in our case the learning rate and the dropout rate - and because it has the capacity to prune unpromising trials. This later fact was important for our project, given the long run time of our model.

[10] See here for a description of Optuna:
https://optuna.readthedocs.io/en/latest/tutorial/pruning.html

In our tuning process, we tried to minimize our model's log-loss over 100 trials, where each trial was an iteration of our model over 200 epochs. This study (Optuna's framework is based around a study) resulted in ten finished trials and ninety pruned trails. It is worth noting that without the pruning, we would not have been able to finish the tuning process within the time allotted. This tuning process gave us our hyperparameters - namely, the adam learning rate and our dropout rate.

After completing our work on the model, I also constructed the maps that overlayed our results with US census statistics to show how the results of our model would affect specific communities in San Francisco.

Naoyuki Komada, Takuma Kinoshita

Modeling

We use long short-term memory (LSTM) model to predict crimes. This is because LSTM has the strength for predicting future events by learning longer trend of time series data than ordinary RNN. In our basic model, individual input is the number of census truck (195) *8 features. Features are year, month, week of year, day of week, dummy variables indicating weekend, hour and precipitation. We set single layer LSTM model with 100 LSTM units and 12 window sizes (= 1 day). Our output is 2 hours ahead and binary crime prediction. Activation function is sigmoid, so our prediction will be expressed by probability. Thus, our loss function is binary cross entropy. When splitting into test and train set, we also use cross validation preventing the future data leakage in order to test our model by time series. Next, we tuned this basic model by taking advantage of Optuna methods, which is a hyperparameter optimization framework based on the bayesian optimization. After using this, learning rate of adam optimizer is 1.77547438185e-05; dropout rate is 0.159961594635. Finally, we set early stopping methods with 5 patience, which means that after 5 consecutive epochs with no improvement of binary cross entropy on validation dataset, training will be stopped.

Evaluation Metrics

The prediction power of our model was evaluated by two metrics: Accuracy and Binary Cross-entropy (log-loss).

Firstly, accuracy is easy to understand intuitively, which can be advantage to introduce the model quality to non-expert of quantitative analysis (e.g. police department, citizen). Another reason of accuracy is that past comparative research by Stec and Klabjan (2018)

[11] evaluated their model by accuracy too. The output of our model can be comparable with the past research. On the other hand, the downside of accuracy is the potential misleading of high accuracy on unbalanced input data. In fact, the major category of the input data (i.e. no crime) occupies greater than 90% of total input data. To prevent this misleading, the accuracy of baseline model (i.e. return 0 (no crime) for all predictions) is also shown in addition to the accuracy of the LSTM model.

Secondly, binary cross-entropy (log-loss) can detect the potential improvement or depreciation of predicted probability. Accuracy evaluates only the corresponding of the predicted binary value and the test value. However, the predicted value 1 based on 0.99 should be penalized less than same prediction based on 0.51, vice versa. Binary-Cross entropy defined by the following formula can take this probability difference into consideration. The disadvantage of binary cross-entropy is non-intuitive and non-compatible with models based on different dataset. To cover these disadvantages, the model is evaluated both by accuracy and binary cross-entropy.

$$-\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

Where:

y_i: Binary label (1 for crime and 0 for no crime)
p(y_i): Predicted probability of observing y_i event
N: The number of all combinations of timesteps and locations

EDA

When creating a dataset to predict on, we did not want to just use geographical crime incident data. There were other “common sense” factors we discussed beforehand, as how weather and time of year may or may not affect crime activity. From these assumptions, we found datasets that allowed us to add data, such as weather, to our dataset. After merging and cleaning outside datasets, we then added features: datetime and exponentially weighted mean. Datetime allowed us easier access to temporal data while exponentially weighted mean smoothed short-term and highlighted longer-term trends in crime data occurrences.

Ted Kumagai

Data Visualization

We wanted to be able to spot any potential outliers manually. Using datetime, pyplot and

[11] https://arxiv.org/pdf/1806.01486.pdf

seaborn, we created various graphs. These graphs gave us visual indicators to explore. Starting from a general overview of our dataset, we found there were clear trends that correlated to real world events and backed by appropriate news articles. From there, we delved deeper, looking at the incidents per day, weekday vs weekend activity and finally a general overview with a violin plot. These aids allowed us to get a visual feel for our data.

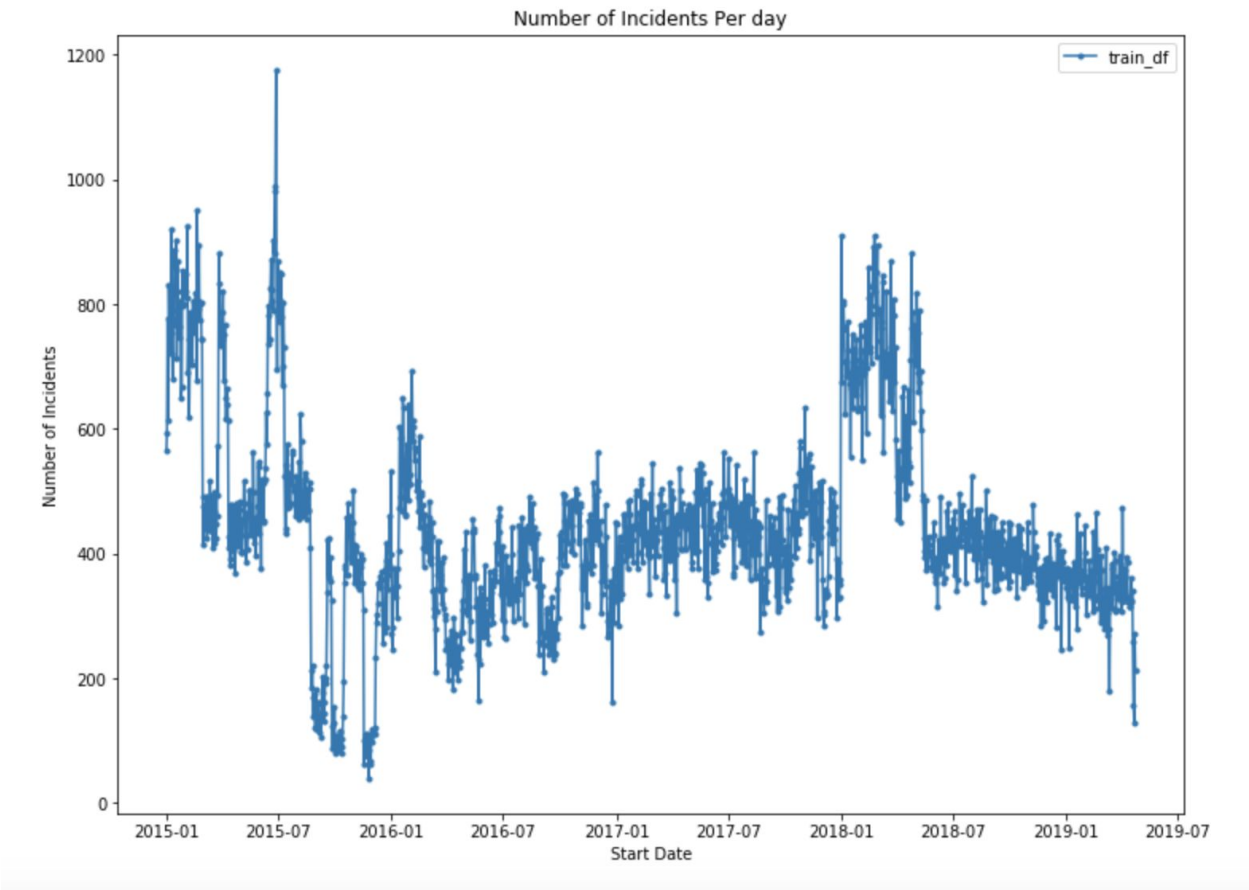


Figure 1: Date vs. Number of Incident

Looking at the graph above of the Number of Incidents per day in San Francisco. We noticed that there was a large spike in crime activity in June of 2015. This spike correspond directly with a crime wave in that year (tinyurl.com/y6zchx5n). The drop in crimes thereafter were choppy, swaying up and down. Instead of following a general crime increase, this event however, followed a crime drop country wide in the US (tinyurl.com/y6zd425y). Some other trends we noticed we that crime seemed to follow somewhat sinusoidal motion within the weeks of the year. We then investigated the crime activity of individual days.

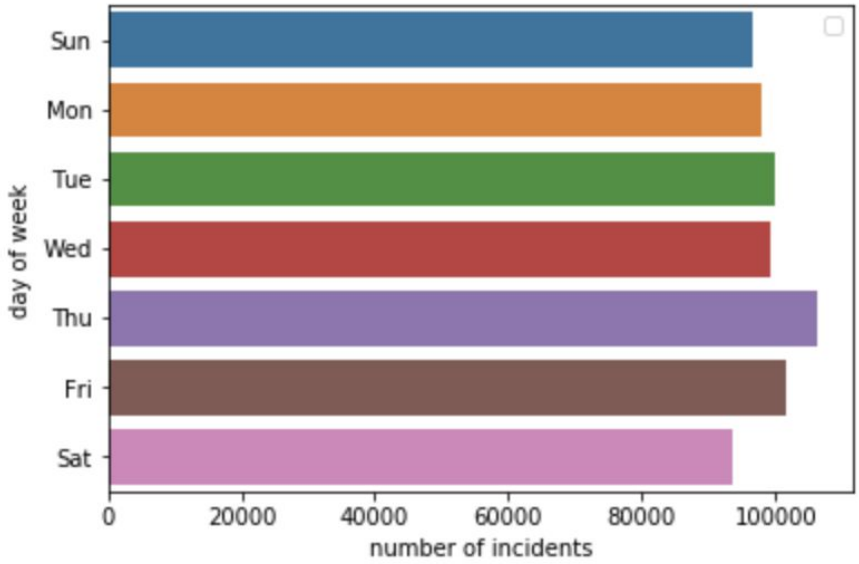


Figure 2: Number of Incidents vs. Day of Week

From the bar graph above, there is indeed sinusoidal/period motion trends. Every week, on average, crime increases from Sunday to Thursday then drops through Friday and Saturday. We then started to question if there is any correlation with crime activity on weekdays (MTWThF) versus weekends (SatSun):

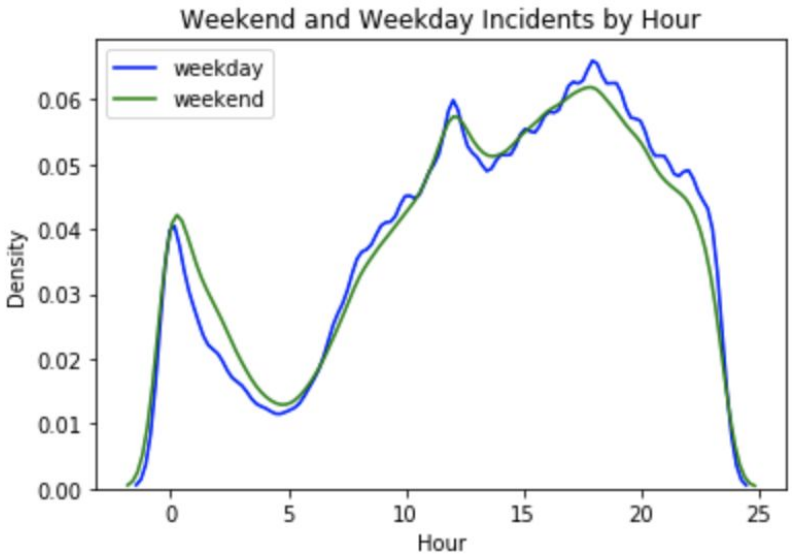


Figure 3: Weekend and Weekday Incident Density Vs. Hour

Shown directly above, there does seem to be some differences in crime activity. Although slight, crime on weekends seem to be more common during the early morning (00:00:00 -

05:00:00) compared to weekdays. However, for the rest of the day (05:00:00 - 23:59:59), crime on weekdays generally exceed weekend incidents.

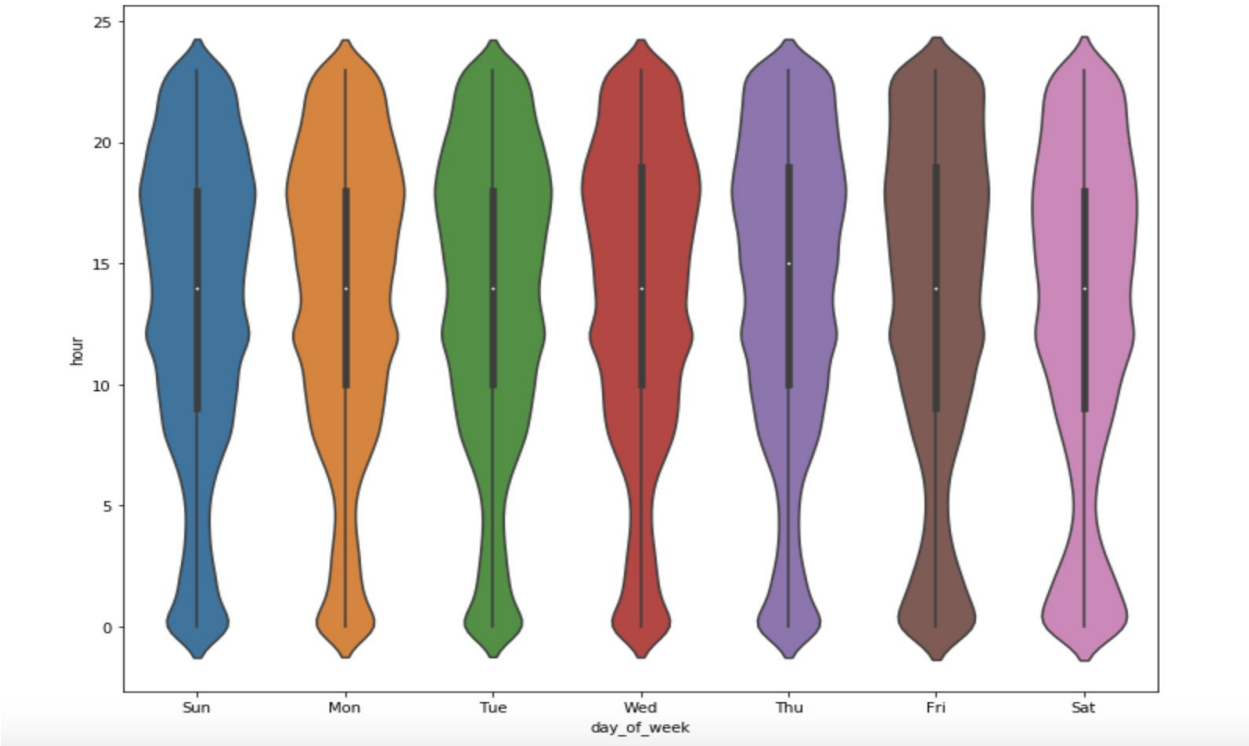


Figure 4: Incident vs Hour per Day

To further illustrate all findings, this violin chart contains, for every day, standard deviations of crimes, amount of crime and the hours to which crimes occur.

Overall, this visualization process helped us get a deeper look into our data. Visual keys were useful in questioning and spotting trends outliers that were considered before applying our final model.

RESULTS AND CONCLUSION

Our results came in the form of predicted probabilities of crime by census tract for each two-hour period. There are 195 census tracts in San Francisco, and in our dataset we had 18,600 two-hour periods - approximately 4.25 years of data. Given that - for most of census tracts - there was no crime in a specific two-hour period, we used as a baseline the case where we assumed there was no crime in any tract during any time. This produced a baseline accuracy of 0.9082 and a log-loss of 3.170. Our model, by contrast, had a test accuracy of 0.9083 and a log-loss of 0.273. This is a negligible change in

accuracy and a substantial improvement of the log-loss. Our accuracy measure was calculated transforming any estimated probability of crime over 0.5 into a 1, and any estimated probability of below 0.5 into a 0, then comparing these predictions with the actual observed values. It therefore makes some sense that our accuracy measure is relatively unchanged and our log-loss is much improved. Our model rarely estimated a probability of crime over 0.5, meaning there is not a lot of distance between our baseline - predicting no crime - and our model - predicting almost no crime. However, our log-loss shows that our predicted probabilities are converging towards the actual labels - indicating that our model is predicting that crime is more likely in places where crime does indeed occur. This makes it potentially useful to police departments or other managers of city resources, who will find it helpful to know - for any time of day - where the probability of crime is highest.

This modeling effort is far from perfect. First, we reduce our data to a binary - did crime occur or not. We had the data - if not the time or computing power - to predict the intensity of crime (how many arrests per unit) or the type of crime (violent, property, or other). We also had the data to assign each census tract weather data from the nearest weather station, as opposed to giving all census tracts the mean San Francisco weather, but this was computationally very intensive, given the need to calculate the nearest distance between polygons. We included code to calculate this in the future.

More importantly, the data we use has some known problems. We only see crimes and arrests that resulted in a police report filed by an officer or a citizen (for some cases). Therefore, we don't see other crimes or interactions that were not recorded. So we are inherently predicting crime conditional on police presence or interest, which should produce skewed results. This means that the data includes bias - some neighborhoods might already be policed heavily, meaning that arrests are more likely in those neighborhoods, meaning our model will predict that those neighborhoods have a higher volume of crime. It addition, police might have incentives to over or under report crime, depending on the police department.

This model also does not include a measure of the potential benefits or costs of police activity. There could be harms caused by police appearing more in certain areas or for certain crimes that this model does not capture. This mean that even if we think this model does a good job estimating the likelihood of actual crime - as opposed to reported crime - there are still concerns with using these models to assign police as is currently being done because we may not be accounting for the harm police could do in particular communities of color.

We tried to examine some of the concerns about over policing by comparing the results of our model to existing data on the population of census tracts by race from the US Census Bureau's American Community Survey 5-year estimates (2013-2017).

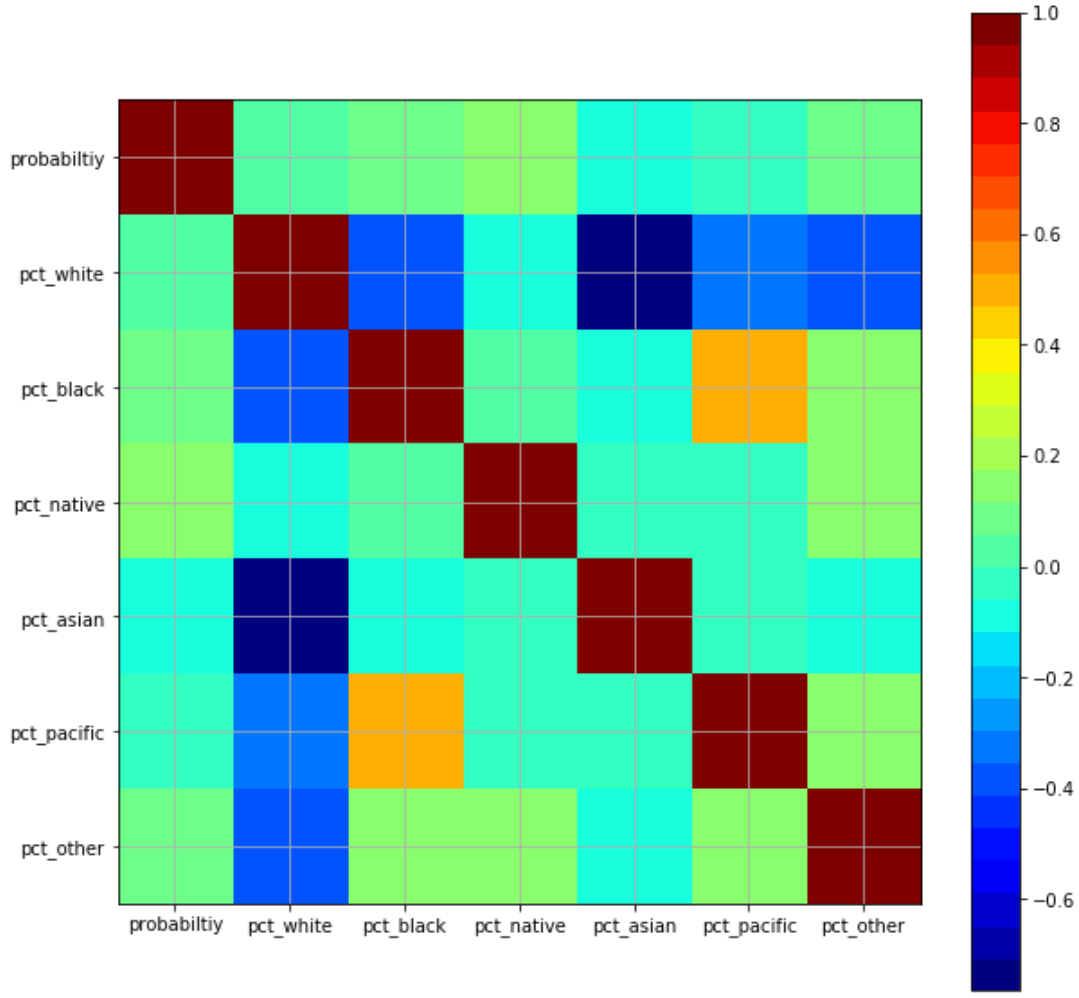


Figure 5: Correlation between percent of population by race and probability of crime

We did not find any correlation between the census tracts with the highest predicted probabilities of crime and tracts with higher share of white, black, asian, or other racial groups. This could be because our model is produced estimates that are unbiased - meaning that they are not predicting crime in a way that differentially impacts specific racial groups. However, there are three other explanations. First, most research[12] has shown that black communities are generally subject to over policing, but in San

[12] As an example, see this work from the Stanford open policing project on traffic stops. https://openpolicing.stanford.edu/findings/

Francisco the median share of black residents per census tract is 2.8 percent, verses 49.3 percent white and 29.6 percent asian. The lack of correlation between our crime predictions and race could be because the communities traditionally over-policed have left San Francisco for other parts of the Bay Area - and future should try to measure if there is a relationship between race and predicted crime in these communities. Second, we could be active policing of certain areas that targets homeless individuals who are likely to be people of color. However, because these people are homeless, they would not show up in the census data, meaning that while the communities where we predict that crime is more likely have a low-share of black residents, the model we are building could still result in police targeting areas with more homeless individuals. Third, there is some research[13] that black communities are both under and over policed. If black residents are stopped more often by officers and during traffic stops, but at the same time there are fewer police around their homes, we could see a model that does not predict high crime in black neighborhoods (because there are few police there to begin with) and predicts crimes in places where black residents may be stopped on there way to work.

All told, we think that it is possible for cities to develop models similar to ours that accurately predict crime given refinements to the features that we created here. However, we think future work should be cautious about using models that could replicate existing biases in policing that exist in the available data. The use of these models to allocate police resources could perpetuate existing harms done by police to communities of color. However, if algorithms could be built to predict crime, and were used in concert with a more deliberative way of thinking about the costs and benefits of police, and where communities could have other resources to send out into communities, these models could be a powerful tool for local governments.

[13] See here: https://www.vox.com/2015/4/14/8411733/black-community-policing-crime