1-Introduction

The New York Police Department has been recordkeeping and publishing detailed crime data for NYC for many years. We believe a scientific analysis of this data could bring an insight into the type of crime an individual might be a victim of, given the time of the day, location, and the demographics of the individual. It could also help law enforcements predict the likelihood of a crime at a given location and a time of the year, such as the list of likely crimes on December 31st at midnight in Manhattan for a particular demographic.

NYC Open Data is free public data published by New York City Agencies and other partners. This data has been maintained by them and updated annually. The dataset spans the years 2006 through 2020 and contains around 7 million records. For practicality reasons, we limited our time scope to January 2019 through February 2020 which is around 350,000 records. The data itself is well maintained however, it still contains missing values and mixed data types in some columns. It will require minimal cleaning, but some variables will have to be either translated, transformed, grouped, sterilized, or mapped in order to be usable. In particular, many of the features are qualitative (crime type, demographic characteristic) and gaps in the data will need to be accounted for, though the size of the dataset will allow for some instance selection.

The project was initially divided into three parts for each of the team members to work on. The first part was cleaning and imputation of the data which was Samuel's responsibility, followed by modeling which was my responsibility and finally the GUI, which was Omar's responsibility. This was a well thought out plan until we all started doing a bit of everything. Since my intentions were to try multiple models, I too started cleaning and encoding the data in different ways for a range of models. After finishing the initial portion of GUI, Omar began doing the same and we all eventually started modeling. The advantage of this unanticipated strategy was the variety of models as well as the dependent variables that was tested. Since there were possible dependent variables and they were all categorical, I utilized supervised machine learning algorithms like Support Vector Machine, Random Forest, and Multinomial Regression. Leaving XGBoost and Decision Tree for Omar and Sam.

2-Backgroung of Algorithm Development

The dataset contained different types of data like numerical (discrete and continues), categorical (ordinal and nominal), and date/time. After gaining some insight of the variables, I narrowed down the useful and usable variables. The next step was to get an idea of the value distribution of these variables which helped me decide between imputing or deleting entries with missing values. Because of the sheer size of the dataset, I was able to afford deleting all the missing data without losing the minimum sample required to do such analysis. Once all the possible target variables and usable features were finalized, I began grouping, encoding, and standardizing the variables. Some of these were easy to encode but others, especially discrete variables like "premise_type" that can have a hundred different location names, required manual work and personal opinion to group smaller aggregates so that the model can perform better.

3-Details of Algorithms

Part of the challenge was to find/reveal a relationship between some of the features and the target because of their high number of unique values of some of dependent and independent variables. Looking at the distribution of these variables against the dependent variable suggested a grouping scheme for premise description, month, and the time of the day. These were some of the variables I encoded in a variety of ways to improve the model. For instance, in one logistic model, I encoded time in milliseconds and converted date into a three-digit number to have continues numerical values as well as categorical (I also standardized them so that the magnitude of one variable wouldn't dominate). In another model, I divided hours of the day into three and months into two categories where the crimes were mostly taking place.

Once the variables were encoded, the modeling phase began with Random Forest (RF). The dependent variable was the crime code and the independent variables were time, date, premise type, premise description, borough, victim's age and sex. After splitting the data into test and train, RF with bootstrap gave disappointing results. Crime description, two different crime codes, and a crime category (felony, misdemeanor, violation) were tested as target variables while the features only varied in the type of encoding, like numerical vs categorical. Trying different combination of dependent and independent variables, only improved the model slightly but not as expected.

Crime_Code1	**precision**	**recall**	**score**
RandomForest	0.33	0.22	0.35
Regression	0.32	0.25	0.34
SVM	0.34	0.27	0.34

The next method was Multinomial Logistic Regression. A slightly different approach, recursive feature elimination (RFE), was applied to the entire data to see what features it would deem important to use in the model. I tried logistic regression with and without using the RFE and had similar results as RF. While I kept getting undesirable results, I was both improving and converging to a disappointing accuracy with a consistency which could also be an indicator.

Crime_Code2	**precision**	**recall**	**score**
RandomForest	0.39	0.44	0.41
Regression	0.40	0.45	0.41
SVM	0.38	0.46	0.41

The last method was Support Vector Machine. Similar results repeated themselves here too where the most basic model gave the worst results and as I changed the groupings of the features and fine-tuned parameters of the model the results got better and better until converging around 49%.

OFFNS_Type (V,F,M)	**precision**	**recall**	**score**
RandomForest	0.44	0.45	0.42
Regression	0.45	0.48	0.37
SVM	0.47	0.49	0.45

4-Results

After trying a numerous combination of models and target/feature pairs, I was unable to exceed the desired precision, recall and f1-value all at once. Even though the target variables were all multinomial, the expectation was to build a complex enough model to be able to find a relationship between the dependent and independent variables. The biggest wrench thrown into the process was the unbalance of the data. For instance, for one of the target variables, crime type (3), there were as many misdemeanors as felony and violation combined. This was also reflected in the crime codes which was another target, even though I further aggregated them into smaller number of groups. There were also missing crimes, like murder, that would have been expected from NYC within a year. One of the improvements I would implement in the future is the size of the dataset. There were approximately seven million records where we only extracted one year's worth of data since it was already around 350,000 units to begin with. Although the immense size of the data made it more difficult to conduct timely analysis and maneuver around, the intuition dictates that adding more data into our frame until class balance as well as a minimum variability of feature values are obtained would have been better for the model. Also using models that penalize for the imperfections of your data is advisable in this case.

5-References

Random Forest Example
(https://towardsdatascience.com/an-implementation-and-explanation-of-the-random-forest-in-python-77bf308a9b76)

SVM Example
https://www.kaggle.com/nirajvermafcb/support-vector-machine-detail-analysis

ROC Curve, Precision-Recall Curve
https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/

Decision Regions
http://rasbt.github.io/mlxtend/user_guide/plotting/plot_decision_regions/

(The code is 100% authentic)