Denver Crime Analytics

Nikita Albert
College of Engineering and Computer Science
Syracuse University
Syracuse, NY, USA
nialbert@syr.edu

***Abstract*— Crime has been and always will be a major issue for societies to face. Although law enforcement has made great strides in utilizing past cases and experience to be proactive against future crime, there are still challenges to overcome and concerns to address. This paper explores how machine learning principles and algorithms can be used to help understand the nature of crime in a municipality, draw meaningful information from it and build models that can used to make conclusions for any new data introduced. For the City of Denver, Colorado, crime dataset, we perform exploratory data analysis and explore how well different machine learning classification algorithms can predict what crime has occurred, given a time and location, under different scopes using R.**

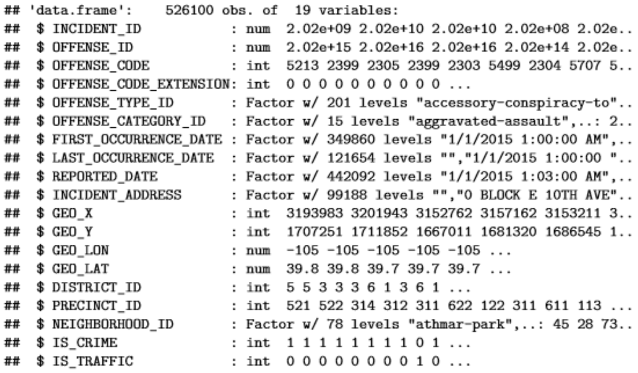
I. INTRODUCTION

It's a common trope in media, where law enforcement must deal with paperwork, whether it's for making an arrest, working on an investigation, or closing the case. We see the surly detective poring over past case files to find a crucial pattern in the middle of the night or filing a report of what happened just to make sure a case isn't lost. It may seem like a hyperbole, but such recordkeeping is the norm and is soon to be even more expansive. As law enforcement embraces technology, the need for easily accessible information and data is becoming ever more necessary and is necessitating the transition to store all records virtually on the "cloud."

With more information made digitally available, more powerful analyses may be performed. In this paper, we explore how machine learning principles and algorithms can be used on the open data catalog provided by the City of Denver, Colorado for all criminal offenses in the city and county for the previous five calendar years [1]. We will perform exploratory data analysis to better visual and understand trends in the data, split the dataset into training and testing sets to construct different machine learning classifier models and evaluate those models, for different scopes, at how well, given a time and location, they predict what crime occurred using R.

II. DATASET

The dataset provided by the City of Denver, Colorado is part of an open data catalog and is based on the National Incident Based Reporting System (NIBRS), which includes all victims of person crimes and all crimes within an incident. The data is dynamic, which allows for additions, deletions and modifications at any time to provide as accurate as possible information.



```
## 'data.frame':    526100 obs. of  19 variables:
##  $ INCIDENT_ID           : num  2.02e+09 2.02e+10 2.02e+10 2.02e+08 2.02e..
##  $ OFFENSE_ID            : num  2.02e+15 2.02e+16 2.02e+16 2.02e+14 2.02e..
##  $ OFFENSE_CODE          : int  5213 2399 2305 2399 2303 5499 2304 5707 5..
##  $ OFFENSE_CODE_EXTENSION: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ OFFENSE_TYPE_ID       : Factor w/ 201 levels "accessory-conspiracy-to"..
##  $ OFFENSE_CATEGORY_ID   : Factor w/ 15 levels "aggravated-assault",..: 2..
##  $ FIRST_OCCURRENCE_DATE : Factor w/ 349860 levels "1/1/2015 1:00:00 AM",..
##  $ LAST_OCCURRENCE_DATE  : Factor w/ 121654 levels "","1/1/2015 1:00:00 "..
##  $ REPORTED_DATE         : Factor w/ 442092 levels "1/1/2015 1:03:00 AM",..
##  $ INCIDENT_ADDRESS      : Factor w/ 99188 levels "","0 BLOCK E 10TH AVE"..
##  $ GEO_X                 : int  3193983 3201943 3152762 3157162 3153211 3..
##  $ GEO_Y                 : int  1707251 1711852 1667011 1681320 1686545 1..
##  $ GEO_LON               : num  -105 -105 -105 -105 -105 ...
##  $ GEO_LAT               : num  39.8 39.8 39.7 39.7 39.7 ...
##  $ DISTRICT_ID           : int  5 5 3 3 3 6 1 3 6 1 ...
##  $ PRECINCT_ID           : int  521 522 314 312 311 622 122 311 611 113 ...
##  $ NEIGHBORHOOD_ID       : Factor w/ 78 levels "athmar-park",..: 45 28 73..
##  $ IS_CRIME              : int  1 1 1 1 1 1 1 1 0 1 ...
##  $ IS_TRAFFIC            : int  0 0 0 0 0 0 0 0 1 0 ...
```

Fig. 1. Denver Crime Dataset Structure

From the structure of the dataset, we make a few observations:

- The dataset contains 526100 observances of crimes and each observance is described by 19 attributes.
- Each crime observation has a unique *Incident ID* and *Offense ID*. The *Incident ID* is a unique numeric identifier for the crime and contains the year the crime is recorded under. The *Offense ID* is also unique numeric identifier and contains the crime's year and *Offense Code*.
- The *Offense Code* is a four-digit numeric identifier that, along with the *Offense Code Extension* attribute, can be used to determine what type of crime occurred.
- The *Offense Type ID* attribute is a nominal attribute with 201 levels. It is a label for what crime occurred, at the finest level, and is a plaintext translation for the meaning behind an observation's *Offense Code* and *Offense Code Extension.*
- The *Offense Category ID* attribute is a nominal attribute with 15 levels that describe what category a crime falls under. In this sense, *Offense Category ID* is an umbrella of different *Offense Type IDs* where the *Category* is the parent identifier and *Type* is the child identifier.
- Three different attributes exist for when the crime "occurs:" *First Occurrence Date, Last Occurrence Date, Reported Date*.

- Three sets of location attributes: *Incident Address*, *X* and *Y* coordinates, and *Lon* and *Lat* coordinates.
- Three sets of municipal jurisdiction areas: *District ID, Precinct ID* and *Neighborhood ID*.
- Two binary flags: *IS_CRIME* and *IS_TRAFFIC*.

We see that non-trivial information is provided for each observance of crime, such as time, date, location, jurisdiction and classification. There is, however, some information redundancy. For example, crime jurisdiction is repeated at the neighborhood, precinct and district levels for increasingly discretized grouping; We have the freedom to focus on the best attribute during model building, while disregarding the redundant ones. Before we make any such preprocessing, we perform exploratory data analysis.

III. EXPLORATORY DATA ANALYSIS

In our exploratory data analysis, we aim to discover trends in our dataset visually that we could not detect otherwise.

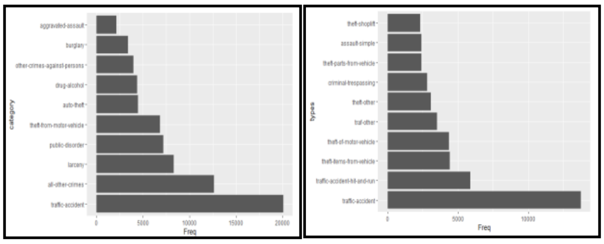


Fig. 2. Top 10 Crime Categories and Type

First, we visualize the top crime types and categories that Denver faces. In *Figure 2* we observe how the type crime categories are *Traffic Accident*, *All Other Crimes*, *Larceny*, *Public Disorder* and *Theft from Motor Vehicle*. We also see that there is a significant skew to the visualization, where *Traffic Accident* crimes occur 1.6 and 2.35 as many times as *All Other Crimes* and *Larceny* crimes, respectively. A similar skew exists within the top crime types, where the type *Traffic Accident* occurs 2.33 as many times as the next top crime type, *Traffic Accident – Hit and Run*.

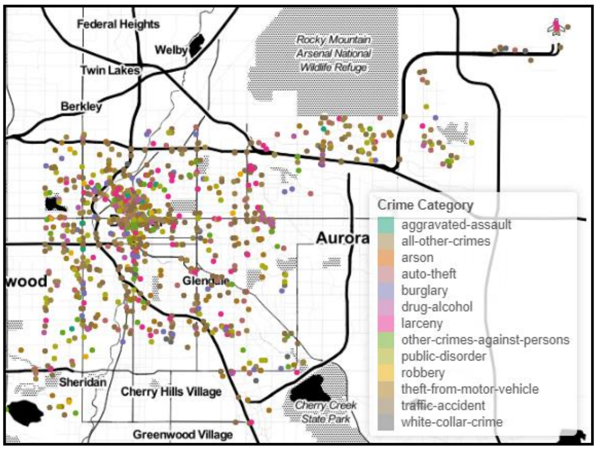


Fig. 3. Denver crime visualized on map, by category

To find any patterns in the dataset that indicate if certain categories of crime occur at certain locations, we plot a sample of 5,000 crime observations, by category, using their *Latitude* and *Longitude* attributes against a map of Denver, as shown in *Figure 3*. Although the visualization is cluttered, we quickly see that crimes cluster along major thruways, certain neighborhoods, and especially by the stadium.

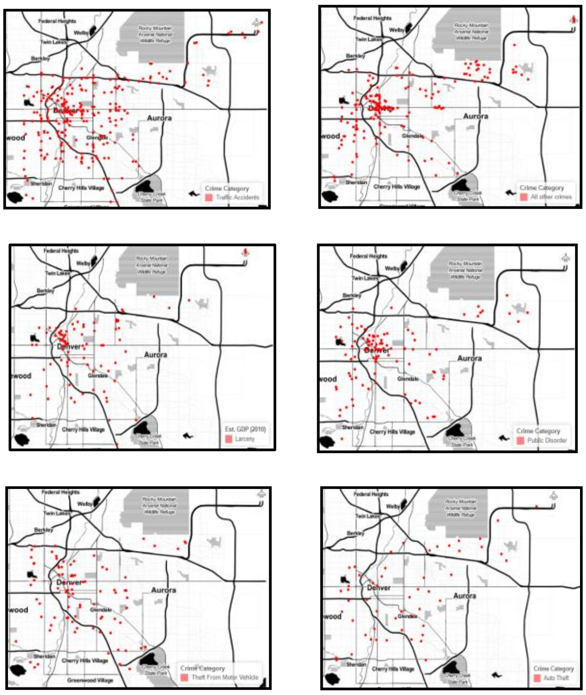


Fig. 4. Denver crime visualized on map, by individual category. (Left to Right: Traffic Accident, All Other Crimes, Larceny, Public Disorder, Theft from Motor Vehicle, Auto Theft)

By visualizing each crime category individually by location, we see certain patterns pop out. Crimes of *Traffic Accident*, *All Other Crimes* and *Public Disorder* categories follow a similar pattern and pop-up most frequently in the same areas, notably by the stadium and south of the Wildlife Refuge. *Theft From Motor Vehicle* and *Auto Theft* category crimes, however, are much more spread out and have their own unique flare-up areas. This discretized view shows us how each crime category has a unique occurrence pattern; Although such patterns may overlap, they are distinct enough within and outside the overlap that useful information can be found.

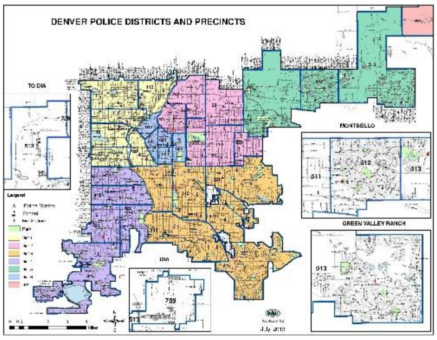


Fig. 5. Denver Police Districts and Precincts

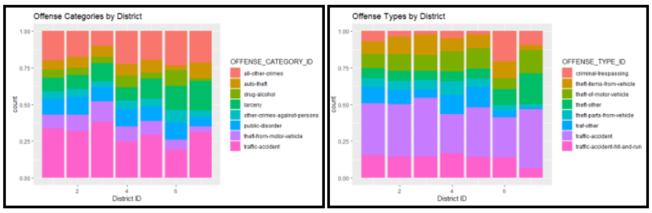


Fig. 6. Denver Crime Type and Category Frequency, by District

In *Figure 6* above, we see how crimes mostly follow the same pattern and relative frequency across districts. Some exceptions exist, such as criminal trespassing that occurs most frequently in District 6 and drug/alcohol crimes that occur the least frequently in District 7.

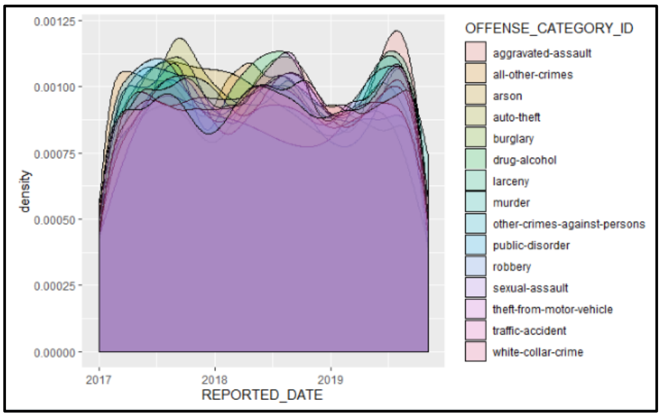


Fig. 7. Top crime categories over time

Interestingly when we visualize the frequency of the top crime categories over time, as in *Figure 7*, we see that all of them follow the same pattern. Regardless of category, crimes peak during the summertime (around July) and retract during winter (around January). Interestingly, we see how at certain times, certain categories break from the overall pattern by either peaking/retracting at different intervals or by different amounts. So, we believe that a “time series” attribute carries a significant of information and provides an understanding of crime’s temporal behavior.

IV. Experiments

A. *Overview*

The exploratory data analysis provided some valuable information on the nature of crime in Denver and can be expanded on to provide even more in-depth metrics. Another approach can be taken, however, where machine learning methods can be applied to the dataset to build models that can make predictions on subsequent previously “unseen” data.

In our dataset, we are provided information on the time, location, jurisdiction and type of crime that occurred in the city. We take inspiration from the *San Francisco Crime Prediction* Kaggle competition [2] and build machine learning models that will predict what a crime’s category or type is, given the crime’s time and location. To better understand the capabilities and limits of our models, we evaluate them under three experiments that cover different scopes of model building and prediction. The experiments we perform are the following:

Experiment A: In this experiment, we will use the “least” preprocessed dataset to predict the *Offense Category ID* of crimes that occurred, given a time and location anywhere in the municipality. We will construct and evaluate the predictive accuracy of five models: Naïve Bayes, RPART Classification Tree, Random Forest, BCART, and Gradient Boosting Machine classifiers. This experiment is meant to be exploratory as to which models perform the best and which models we should utilize for the subsequent experiments.

Experiment B: This experiment is very similar to *Experiment A*, expect that we will construct and evaluate two models that will predict the *Offense Category ID* of crimes that occurred, given a time and location, in a specific district. We will construct and evaluate said models for each district. The models we construct for this experiment will be determined by which two models are the most accurate in *Experiment A*. This experiment is meant to explore whether limiting the scope of prediction to a finer district-by-district basis will yield better or worse predictive performance.

Experiment C: With *Experiment C*, we attempt to see how well predictive classifier models perform under a more restricted scope. Using the consistently best performing model type found in *Experiment B*, we construct and evaluate a model that will predict what type of crime occurred given the time and location of a crime of a certain category. We do this for crimes across the municipality and crimes in a certain district, for each district. For this exercise, we will focus only on predicting the *Offense Type ID* of crimes of the *Drug-Alcohol* category. This experiment is meant to explore how well machine learning models can make predictions for a specific task. For example, law enforcement has a set budget for tackling drug related crimes; Given a time and location, what crime will have occurred and what resources should be allocated to address it, whether it is a small time bust or a coordinated raid?

B. *Data Preprocessing*

Before we construct any models, we first perform some preprocessing on the dataset to make it more suitable to work with. Said preprocessing is part general housekeeping, to ensure the dataset is of good quality, and part implementation of certain design choices for our experiments. To bring the dataset into a baseline form for our experiments, we do the following:

- Remove any observance with any missing attribute value. This is to ensure that when we split the dataset into training and testing sets, we have as much information as possible and do not have to account for missing information during training.
- Drop the *First Occurrence Date* and *Last Occurrence Date* attributes. We found these two attributes to be inconsistently populated before we dropped occurrences with empty values. As each

observance is a unique crime occurrence, the idea of said crime occurring in the past is moot. With this uncertainty, we decide to only use the *Reported Date* attribute as our observance timestamp.

- Drop the *Incident ID* and *Offense ID* attributes, as these are nominal attributes and each observance has its own unique value. Such attributes are not useful for us in this application and we choose to disregard them.
- Drop the *Offense Code* and *Offense Code Extension* attributes. These two are directly tied to the *Offense Type ID* attribute and, subsequently, to the *Offense Category ID* attributes; attributes that we hope to predict with our models. Such correlation would make model training rely on the wrong information and any predictions invalid.
- Drop the *Incident Address* and *Neighborhood ID* attributes. These two are nominal attributes that serve as a level of abstraction for the *Geo X*, *Geo Y*, *Geo Lat*, and *Geo Lon* attributes. We elect to use the latter attributes for specific locations and hold onto the *District ID* and *Precinct ID* attributes for abstracted locations.
- Break up the *Report Date* attribute into *Month, Day, Year, Day of Week*, and *Hour* attributes. We do this to extract meaningful information from a date/time into a more useable form.

At this point we have a baseline dataset that we will work with for our subsequent experiments. *Table 1*, below, provides a structural overview of said baseline dataset.

Attribute Name	**Type**
OFFENSE_TYPE_ID	Categorical: 201 levels
OFFENSE_CATEGORY_ID	Categorical: 15 levels
GEO_X	Integer
GEO_Y	Integer
GEO_LON	Numeric
GEO_LAT	Numeric
DISTRICT_ID	Integer: 1:7
PRECINCT_ID	Integer
IS_CRIME	Integer: 0:1
IS_TRAFFIC	Integer: 0:1
MONTH	Integer: 1:12
DAY	Integer: 1:31
DAYOFWEEK	Categorical: 7 levels
YEAR	Integer: 2015:2019
HOUR	Integer

Table 1. Baseline Preprocessed Dataset Structure.

For *Experiment A*, we will use a sample of 20,000 observances from the above dataset. This subset will be structurally identical to the baseline, with exception to a subset clone used for our Naïve Bayes model, where we will encode *DAYOFWEEK* as a numeric attribute.

For *Experiment B*, we will split the sampled dataset by district to create district specific subsets. We will drop the *DISTRICT_ID* attribute in the resulting subsets, as it will have no variation within a set.

For *Experiment C*, we clone the baseline preprocessed dataset and filter out any crimes that are not part of the *drug-alcohol* category. We then split this filtered set into subsets by district and use these subsets for district specific model building and evaluation. In this experiment, we drop the *DISTRICT_ID, OFFENSE_CATEGORY_ID, IS_CRIME* and *IS_TRAFFIC* attributes as they all have no variation within a set for this scope.

For all datasets we create for model building, we split them into training and testing sets. We will randomly sample 75% of the experiment specific datasets to create the respective training sets. The remaining 25% will be used as the training sets.

C. Machine Learning Methods

The model types we evaluate in *Experiment A*, and potentially use in *Experiments B* and *C*, perform predictive classification using different principles and methodologies.

The Naïve Bayes classifier model, for example, follows the Bayes' Theorem to find the probability of an event occurring given the probability of another event that has already occurred. It is a wholistic probabilistic classifier that essentially follows below representation of class probability:

$$\begin{aligned} P(c_i|x_0,\ldots,x_n) &\propto P(x_0,\ldots,x_n|c_i)P(c_i) \\ &\propto P(c_i)\prod_{j=1}^{n} P(x_j|c_i) \end{aligned}$$

Fig. 7. Naïve Bayes Class Probability representation [3]

We implement the Naïve Bayes classifier model in our experiments with the *NaïveBayes()* function from the *klaR* library [4].

The decision tree classifier, however, is a supervised learning predictive model that uses a set of binary rules to calculate a target value. Model building is done algorithmically as the data is split into subsets that are as homogeneous as possible. If a new observation falls into any of the subsets, it would be classified by the majority of observations in that subset. In *Experiment A* we run two iterations of decision tree classification; Once with the base tree, and once with the pruned tree. As we found that a tuned model outperforms a base one, we move forward with using tuned models of the remaining model types. To grow, prune and make predictions with out tree, we use the *rpart()* function from the *rpart* library [5].

Our random forest classifier behaves similarly to our decision tree classifier, however, consists of many individual decision trees that operate as an ensemble. We hope that a large number of relatively uncorrelated models, that operate as a committee, will outperform any of the individual constituent models that make up it. We construct our random forest using *tuneRF()* function from the *randomForest* library [6].

The gradient boosting machine works similarly to the random forest model, where it achieves better performance by using multiple "sub-models." Specifically, it trains many individual models in a gradual, additive and sequential manner where each new tree is a fit on a modified version of a prior tree. In a sense, it attempts to convert weak learners into strong learners. We construct our GBM model using the *gbm()* function from the *gbm* library [7].

To explore how well other boosting algorithms perform, we use the *bagging()* function from the ipred library [8]. It behaves similarly to GBM and RF classifiers, where averaging the ensemble of predictions can help reduce variance and minimize overfitting.

We will use the above-mentioned libraries and apply their respective functions on our training sets to construct and train our machine learning models. Then we will apply our models on the respective test sets and have the models make predictions of which crime category or type occurred. We evaluate the performance by observing the confusion matrix of the true results and predicted results.

V. RESULTS

A. *Experiment A*

Model	Accuracy	Kappa
Naïve Bayes	0.2478	0.0389
RPart Classification Tree	0.4104	0.3142
Pruned Rpart Classification Tree	0.4587	0.3485
Tuned Random Forest	0.4815	0.3757
Gradient Boosting Machines	0.4705	0.3634
BCART	0.4651	0.3696

Table 2. Experiment A Models and their Performance

Table 2, above, summarizes the performance of the models we constructed, trained and predicted with in *Experiment A*. The Naïve Bayes model was the worst performing one with an accuracy of 24.78% and Kappa of 0.0389. This is most likely because this model relies on independent attributes, which our dataset is lacking. For example, district is reliant on precinct, which is reliant on geographic location.

Our best performing models, however, are all the ensemble classification models. Of these models, the Tuned Random Forest and GBM models achieved the best performance with respective accuracies of 48.15% and 47.05%, and respective Kappas of 0.3757 and 0.3634. As such, for *Experiment B* we use Random Forest and GBM classifier models.

B. *Experiment B*

District	Tuned RF		GBM	
	Accuracy	Kappa	Accuracy	Kappa
1	0.5281	0.3985	0.5160	0.4001
2	0.4614	0.3541	0.4499	0.3489
3	0.5116	0.3951	0.4972	0.3819
4	0.4380	0.3245	0.4261	0.3095
5	0.4919	0.3812	0.5121	0.4177
6	0.4110	0.2882	0.4027	0.2919
7	0.7424	0.6731	0.6970	0.6220
Average	0.5120	0.4021	0.5001	0.3960

Table 3. Experiment B Models and Results

In *Experiment B*, we quickly see that as we construct models and make predictions on discretized segments of the overall dataset, the metrics of our results improve in most cases. For instance, we were able to predict the category of crimes that occurred in District 7 with a 74.24% accuracy via Tuned Random Forest model; We know that such performance is not a fluke, as the Kappa is 0.6731 and indicates that such performance is intrinsic to the model.

In cases where there is degradation in performance, the degradation is not too significant. The exception is with District 1, where accuracy is at 41.10% and the kappa of the model is 0.2882.

If we average the metrics of the district specific tuned RF models and GBM models, we find that they together outperform the singular municipal models in *Experiment A*. Our averaged tuned RF models for example see a 3.05% increase in accuracy to 51.20% and a 7.02% increase in Kappa to 0.4021. With our averaged GBM models, we see a 2.96% increase in accuracy to 50.01% and a 8.97% increase in Kappa to 0.3960.

As the Tuned Random Forest model consistently produced the best performance metrics in accuracy and model fidelity in *Experiments A* and *B*, we will use it as our model for *Experiment C*.

C. *Experiment C*

District	Tuned RF	
	Accuracy	Kappa
All	0.4151	0.2734
1	0.3401	0.1842
2	0.4254	0.2495
3	0.3407	0.1793
4	0.2946	0.0799
5	0.2830	0.0945
6	0.5528	0.4109
7	0.2857	-0.0294
Average	0.3603	0.1770

Table 4. Experiment C Result Metrics

Interestingly trying to predict what type of crime occurred for crimes of the *Drug-Alcohol* category using Tuned Random Forest models yielded us the worst performing initial results. For instance, with the District 5 model, we saw an accuracy of 28.30% and a model kappa of 0.0945. Our District 7 model, however, achieved a negative value kappa of -0.0294. The only true improvement we saw was with our District 7 model, where we achieved an accuracy of 55.28% and model kappa of 0.4109.

This degraded performance seems to be abnormal. We hypothesize that such degradation has to do with a mis-definition of what our models should accomplish, specifically with how the models are expected to predict finely defined crime types. For example, we have specific crime type values for the manufacture, sale, possession and purchase of a given drug. This poses the question: Do we really care about whether a crime observance had to do with the sale or purchase of a drug, or rather that crime was just related to a given drug?

To explore this hypothesis, we simplify the crime type labels by consolidating them to what drug they relate to and repeat *Experiment C*. *Table 5* below maps our new *Offense Type ID* values to the original values.

New Type	Original Type
Drug-barbiturate	Drug-barbiturate-possess Drug-barbiturate-sell
Drug-cocaine	Drug-cocaine-possess Drug-cocaine-sell
Drug-hallucinogen	Drug-hallucinogen-possess Drug-hallucinogen-sell
Drug-marijuana	Drug-marijuana-possess Drug-marijuana-sell
Drug-meth	Drug-meth-possess Drug-meth-sell
Drug-opium	Drug-opium-possess Drug-opium-sell
Drug-other	Drug-fraud-to-obtain Drug-make-sell-other Drug-pcs-other-drug Drug-poss-paraphernalia
Drug-synth	Drug-synth-narcotic-possess Drug-synth-narcotic-sell
Liquor	Liquor-possession Liquor-sell

Table 5. Experiment C Simplified Types Result Metrics

District	Tuned RF	
	Accuracy	Kappa
All	0.4882	0.3018
1	0.4534	0.2677
2	0.4683	0.2759
3	0.3878	0.2046
4	0.3614	0.1713
5	0.3459	0.1459
6	0.5774	0.4335
7	0.2857	-0.0294
Average	0.4114	0.2530

Table 6. Experiment C Simplified Types Result Metrics

As we see in *Table* 6, with simplified *Offense Type ID* labels and retrained models, we were able to improve our predictive metrics significantly in every case. For example, we saw as much as a 11.33% increase in accuracy of our predictions and 45.33% increase in our model Kappa for our District 1 model. Interestingly, we saw no improvement for our District 7. This is most likely due to not have enough drug related activity in this district and subsequently poorly trained models.

VI. CONCLUSION

Our exploratory data analysis allowed us to visualize a raw dataset in an elegant and enlightening manner. We were able to quickly determine if any geographic outliers existed, which crimes occurred where, and how densely they occur in certain areas. We were able to quickly chart and observe how crime behaves over time and how specific crime types and categories behave against each other. Of course, such visualizations could be done via other methods, but with R and RStudio we were able to do said visuals and subsequent data manipulation and model building using one tool.

With *Experiment A* we saw how certain models are able to outperform the rest and provide an idea of what techniques subsequent attempts should take advantage off. Although our best model was able to classify previously unseen data with a 48.15% accuracy and with a model fidelity kappa of 0.3757 which seems unimpressive, it shows that machine learning models can be applied to such a dataset and application to yield decent results.

Our results from *Experiments B* and *C* actually show us how tweaking the scope of what we want to accomplish can yield us better performing models and results. In *Experiment B*, we constructed models that only focused on the fixed geographic areas of Denver’s police district. This discretization and filtering made it so that, during training, our models were able to better understand and learn from the minutiae of data patterns that could be overlooked when learning from the entirety of municipal crime.

With *Experiment C* we narrowed our scope even further by building models for every district, however, training them to predict the *Offense Type ID* of a crime of a given category, date and time. For this experiment, we focused on crimes of the *Drug-Alcohol* category. Although initially we received low performing metrics and results, we found that we were too stringent on how we evaluated model performance. For example, was it more valuable to us to know specifically whether a crime was a possession, sale or purchase of a specific drug or rather that it was related to a specific drug? When adjusted our final labeling schema and retrained our models to predict them, we had significant increases in performance metrics.

So, although the metrics for our classification models were far from perfect, this foray into applying machine learning principles and methods to a crime dataset provides valuable lessons. For example with *Experiment A*, we initially had decent results but nontrivial improvements in *Experiment B*; This shows us how building an omnipotent model is not necessary, and an ensemble of models handling discrete responsibilities can be more advantageous and better to use. *Experiment C* shows us how machine learning models do have certain limitations and how adjusting to mitigate those limitations may provide us more information. For example, there are not that many *Drug-Alcohol* crimes in Denver compared to others, so training them to accurately predict a sale, purchase, or manufacturing of a drug would be difficult without more information. However, is that truly necessary? We found that when we simplified the attribute values to just being related to the drug, model performance improved and provided more information, especially now such models can be used to show whether such crimes are endemic within a specific area.

We believe that as better data is collected, with more information, such application of machine learning to crime data would yield better performing models, clearer understanding of the issues and lead to more adept, nuanced methods of handling the underlying issues that lead to crime.

REFERENCES

[1] P. Mooney, "Denver Crime Data," Accessed on: Nov. 12, 2019. [Online] https://www.kaggle.com/paultimothymooney/denver-crime-data

[2] SF OpenData, "San Francisco Crime Classification," June, 2015. Accessed on: Nov. 12, 2019 [Online]: https://www.kaggle.com/c/sf-crime/overview

[3] D. Soni, "Introduction to Naïve Bayes Classification," May 16, 2018. Accessed on Nov. 18, 2019 [Online] https://towardsdatascience.com/introduction-to-naive-bayes-classification-4cffabb1ae54

[4] RDocumentation, "NaiveBayes," Accessed on Nov. 18, 2019 [Online] https://www.rdocumentation.org/packages/klaR/versions/0.6-14/topics/NaiveBayes

[5] RDocumentation, "rpart," Accessed on Nov. 18, 2019 [Online] https://www.rdocumentation.org/packages/rpart/versions/4.1-15/topics/rpart

[6] RDocumentation, "tuneRF," Accessed on Nov. 18, 2019 [Online] https://www.rdocumentation.org/packages/randomForest/versions/4.6-14/topics/tuneRF

[7] RDocumentation, "gbm," Accessed on Nov. 18, 2019 [Online] https://www.rdocumentation.org/packages/gbm/versions/2.1.5/topics/gbm

[8] RDocumentation, "bagging," Accessed on Nov. 18, 2019 [Online] https://www.rdocumentation.org/packages/ipred/versions/0.4-0/topics/bagging