**Feature-Based Safety Map of New York City**

Lakshmiah Sabitha Boyini

Big Data Essentials: AIT614 - DL3

Dr. Liao

George Mason University

04/24/2022

# 

## Abstract

New York City is one of the largest and most populous cities in the United States and in the world. Various methods and tactics have been implemented in combating the high crime rate in this city. Due to the city being a major tourist attraction, as well as having a massive population and a daily influx of commuters, there are such huge gaps in security that it is unfeasible to mitigate every danger. Our research aims to analyze a dataset of historical complaint data from the New York Police Department and attempts to identify feature-based trends within the given data. The primary focus of the project is on determining the risk level of an individual based on their location and their demographics. Leveraging machine learning using the Apache Spark MLlib system, as well as Python-based visualizations, we additionally created an interactive map showing risk levels for an individual around the city. From a dataset containing 7.4 million records of crimes committed in the last two decades, we were able to generate a model capable of predicting a person’s risk level in zip codes around New York City with a high accuracy rate, and display the risk levels geographically as a heat map.

## Introduction

Due to New York City’s high degree of popularity with tourists, as well as the enormous number of residents and commuters who commute to the city on a daily basis, there are such huge gaps in security that it is unfeasible to mitigate every danger. Our research aims to analyze a dataset of historical complaint data from the New York Police Department (NYPD) [1], and attempts to identify feature-based trends within the given data. The primary focus of the project is on the possibility and probability of any individual likely to become a victim of a crime scene in the future or present, based on the geographical locations, time, and their demographics. The project not only analyzes the likelihood of an individual being a part of a crime scene but also aims to provide resources such as an interactive map which will show what locations can be avoided to stay safe and away from the crime scene by drawing conclusions regarding the most and least safe geographical areas.

With the proposed interactive map, an individual should be able to input the parameters of their demographics and obtain an understanding of their risk level in various areas of New York City. The project is based on a dataset consisting of NYPD historic complaint information. This dataset is available on New York City’s data site and includes information about all valid felony, misdemeanor, and violation crimes reported to the New York City Police Department (NYPD) from the year 2006 to the end of the year 2019. The NYPD Complaints data set consists of around 7.4 million records. Our initial goal was to limit the scope of information to five years (2014-2019), as well as use only around 9 out of the available 35 attributes for each complaint.

As the end goal of the system is to approach the recorded crimes from a victim's perspective and allow anyone to look at the possible crime rate in an area, we incorporated zip code information against each complaint. A geographical breakdown into zip codes was selected due to the specificity of using coordinate values, and the genericity of a breakdown into boroughs, given their size. The zip codes were obtained via a reverse geocoding script, using a reverse lookup algorithm [9], the coordinates of records in the dataset, and US zip code data from geonames.org.

## 

## 

## Goals

The main goals of the project are listed below:

* To clean, unify and standardize the data structure of the dataset consisting of millions of records.
* To draw meaningful and in-depth insight from the NYPD historical crime dataset
* To correlate and deduce relationships between the major attributes or features utilized from the dataset
* To analyze the possibility and probability of any individual becoming the victim of a crime based on the geographical location and his/her demographics.
* To create interactive visualizations as a guidance tool for end-users
* To draw resulting conclusions regarding the safety of geographical areas highlighting correlations between the victim’s demographics and unsafe geographical spaces

Our research could be used to determine the relative safety of an individual based on their location within New York City, the time of day, the individual’s age group, race, and sex, and would also use historical data to inform them about the age group, race, and sex of suspects in crimes with the same parameters. The information our model would provide could significantly increase individual safety by providing a method of planning safer pedestrian routes through the city, as well as increasing awareness of crime potential within various areas of the city. It is also important to review such information since there are many indicators that could be derived from these datasets about potential influences on the crime rate, which could be used to address and diminish it.

## Requirements

The project requirements were provided as follows:

* Select a suitable dataset for analysis of victims of crimes in New York City
* Apply tools and methods of data analytics technologies for analysis as well as for data modeling or machine learning
* Develop skills in applying different data analytics techniques to particular tasks with a focus on the end goal
* Design and develop a preliminary data analysis system based on the requirements which could be used by the public to get an idea of their risk level of being a victim of a crime in New York City
* Leveraging different data analytics tools and techniques such as NoSQL databases and tools in the Spark ecosystem
* Learn how to write scientific/technical documentations
* Learn some project management techniques

## 

## Dataset

The dataset which is selected for the final project is that of NYPD historic complaint information [1]. This is available to download on NewYork cities data side and includes all valid felony, misdemeanor, and violation crimes reported to the New York City Police Department (NYPD) from 2006 to the end of 2019. The NYPD Complaints data set is relatively big and holds around 7.4 million records. For the purpose of this research, it is planned to limit the scope to the most recent 5 years, ie. from 2014-2019 that itself comes into millions of records. There are a total of 35 features recorded with each complaint. However after careful analysis, it is decided to use the below subset (15 features) for the purpose of this project since they relate to details of the crime that could be used for predictions related to crime victims. As the end goal of the system is to approach the recorded crimes from a victim's perspective and allow anyone to look at the possible crime rate in an area, we incorporated zip code information against each complaint. Zip code was selected as latitude and longitude are too specific and the borough name is very generic from a command man’s perspective. To get this zipcode, a reverse lookup algorithm using Latitude and Longitude information was used[9], using the US zip code data downloaded from geonames.org.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Dictionary - NYPD Complaint Incident Level Data (2006-2019) [1]** | | | |
| **Sl. No** | **Column** | **Description** | **Sample** |
| 1 | RPT\_DT | The date on which the complaint is reported. | 05/28/2015 |
| 2 | ADDR\_PCT\_CD | The precinct in which the incident occurred | 46 |
| 3 | BORO\_NM | The name of the borough in which the incident occurred | BRONX |
| 4 | CMPLNT\_FR\_DT | Exact date of occurrence for the reported event (or starting date of occurrence, if CMPLNT\_TO\_DT exists) | 05/28/2015 |
| 5 | CMPLNT\_FR\_TM | Exact time of occurrence for the reported event (or starting time of occurrence, if CMPLNT\_TO\_TM exists) | 15:00:00 |
| 6 | CRM\_ATPT\_CPTD\_CD | Indicator of whether crime was successfully completed or attempted, but failed or was interrupted prematurely | Completed |
| 7 | KY\_CD | Three digit offense classification code | 578 |
| 8 | LAW\_CAT\_CD | Level of offense: felony, misdemeanor, violation | VIOLATION |
| 9 | SUSP\_AGE\_GROUP | Suspect’s Age Group | 25-44 |
| 10 | SUSP\_RACE | Suspect’s Race Description | BLACK |
| 11 | SUSP\_SEX | Suspect’s Sex Description | M |
| 12 | VIC\_AGE\_GROUP | Victim’s Age Group | 25-44 |
| 13 | VIC\_RACE | Victim’s Race Description | WHITE HISPANIC |
| 14 | VIC\_SEX | Victim’s Sex Description | F |
| 15 | Latitude | Midblock Latitude coordinate for Global Coordinate System, WGS 1984, decimal degrees (EPSG 4326) | 40.84586773 |
| 16 | Longitude | Midblock Longitude coordinate for Global Coordinate System, WGS 1984, decimal degrees (EPSG 4326) | -73.915888033 |
| 17 | Zip Code | Derived field by implementing a reverse lookup algorithm[9] using longitude and latitude, using the US zip code information from geonames.org | 10313 |

A NoSQL model was selected to store this data. Although the original dataset is fairly structured, we wanted to ensure flexibility and scalability for our database. If in the future data needs to be added that is unstructured and will not fit perfectly into a predefined schema we want to ensure that our solution can handle storing and manipulating that data. We have chosen the tool MongoDB Atlas to store and manage our NoSQL database. Atlas was chosen since this is the cloud version of MongoDB which allowed for better collaboration for a team by working on a shared cluster.

**System**

## 

*Figure 2: Diagram of updated conceptual system framework*

### Architecture

At a high level the architecture for the designed system has four main components which are the raw data, data storage and cleaning, data processing, and visualizations.

* Raw Data: The raw data component of the system has two parts. One is the crime data that is discussed in detail in the “Dataset” portion of this report. The second raw data component is a table created that maps the different zip codes in New York City to their corresponding neighborhood in correspondence with a published list of neighborhoods and zip codes [12].
* Data Storage and Cleaning:The first component of the data storage and cleaning is MongoDB Atlas which is where the raw data was first uploaded. From here, data was injected into a Python based ecosystem. In this ecosystem, data was stored in Pandas dataframes for data preprocessing without long term plans of using this stored data and in the Spark Core RDDs for machine learning algorithms and more long term usage of the data.
* Data Processing: This component of the ecosystem is where the majority of the analysis and modeling occurred. Packages from SparkSQL, SparkMLib, and Pandas were used for tasks such as frequency analysis and creating models for classification and prediction. The specifics of the models will be expanded on in the “Data Analytics Methods” portion of this section of the report.
* Data Visualization: The visualizations were created on a Python based ecosystem using libraries such as Folium and Bokeh. The visualizations were created and displayed in Google Colaboratory notebooks.

### Data Pre-Processing

Data preprocessing included many steps such as limiting the dataset to that of the most recent 5 yrs of crime data. Since the available location fields were either too broad(BOROUGH\_NAME) or too narrow(Latitude and Longitude of the location) a reverse zip code lookup was done and added zip code information against each crime record. The total number of unique zip codes posed an issue of sparsity when using this attribute. For this reason, zip codes were matched to a more condensed list of neighborhoods to reduce the number of values that are present in this attribute. This gave us the raw data that we could begin working with for modeling and for visualizations.

The raw data file which included zip codes needed to be imported into a shared MongoDB cluster so that it would be available for the team in a collaborative manner for further pre-processing. To accomplish this, a cluster was set up in MongoDB Atlas from which data in our local machines could manage and manipulate the data that was stored. The data stored in the shared cluster could then be moved to Spark or Pandas storage structures. To work with the data locally, local versions of MongoDB Compass were connected to the shared Atlas cluster. Data was inserted and deleted on the local MongoDB Compass versions which were automatically pushed up to the shared MongoDB Atlas cloud cluster for the team to access.

The raw dataset that was used at the center of this project had all unlabeled data while it was in MongoDB. In order to create a decision tree or random forest model which are both types of supervised machine learning, we needed to turn the unlabeled data into labeled data which required additional data preprocessing. It was determined that risk levels would be given based on the frequency of crimes that a member of a demographic at a specific zip code experienced. Pandas libraries were used to calculate frequencies, get summary statistics on the frequencies, and Pandas dataframes were used to append these frequencies to the dataset of crimes. Matplotlib and Pandas tools were leveraged to construct an elbow graph to see the optimal number of “K” clusters. Once this value had been shown to be 3, a K-Means clustering algorithm would be run to put the frequency values into clusters. The resulting clusters were analyzed to determine which clusters correspond to the “Low”, “Medium”, and “High” risk based on the value of the frequencies in the clusters. Once the clusters were complete and the crime samples were labeled with “Low”, “Medium”, and “High” risk the supervised machine learning could begin.

### 

### Data Analytics Methods

#### Existing Methods

Most methods that are used focus on creating a profile of perpetrators as they are involved in crimes. They employ machine learning tools such as association rule mining to create a profile or demographics most at risk of becoming perpetrators in a crime. Other models look at using a variation of linear regression to come up with a percentage of the likelihood for a person to commit or recommit a crime. The visualizations are typically charts or graphs that show the demographics or past history of perpetrators of crimes. Our solution looks at using decision trees and random forests to pay more close attention to the victims. As the main form of visualization we propose using maps that respond to input from the user regarding their demographic information. This helps in overall crime prevention by reducing the exposure of at risk populations to these situations or locations.

#### New Methods

The method that was developed for this solution uses a combination of K-Means clustering, decision tree models, random forest models, and interactive maps.

* K-Means Clustering: K-Means clustering occurs at the pre-processing stage to create labels for the unlabeled data. At first, a data frame containing frequency information is used with the elbow method to decide the optimal number of clusters. For our dataset, the diagram revealed 3 as an optimal value. A K-Means clustering algorithm with a K value of 3 is created using Spark MLib along with the appropriate pipeline to cluster frequencies and evaluate their correspondence to “Low”, “Medium”, and “High” risk then these frequency clusters are used to label the crime data. The Silhouette with squared euclidean distance is used to get an understanding of the effectiveness of the clustering.

|  |  |
| --- | --- |
| *Figure 3: Elbow method showing an “Elbow” at 3 for optimal K value.* | *Figure 4: Silhouette value close to 1 indicates consistency within the clusters that were generated.* |

* Decision Tree: A decision tree model is generated to predict based on demographic attributes and location what the risk level of becoming a victim of a crime is for a person. The labeled data is split into an 85 / 15 train and test data split. Next the pipeline is created which transforms data into the required vector and trains a SparkMLib decision tree model. The created decision tree model is then tested for accuracy. The decision tree model that our algorithm and crime data generated had an accuracy level of approximately 87.1%.

|  |
| --- |
| *Figure 5: Results from running test data through the trained decision tree model.* |

* Random Forest: To overcome issues inherent to decision trees such as the potential for overfitting, a random forest model is also created using SparkMLib as the final step in our algorithm to model the data. The random forest uses the same 85 / 15 train and test sets as the decision tree. This is kept constant to better assess if there are any improvements in the accuracy of the model generated from this method. After running the test data through the train model, it is apparent that the random forest model does improve on the accuracy of the decision tree model by approximately 5.7%. For this reason, the random forest model is presented as the model for making predictions based on victim demographics and it is also used for creating visualizations.

|  |
| --- |
| *Figure 6: Results from running test data through the trained random forest model.* |

* Visualization: The first portion of the visualization requires getting input from the user regarding their demographic information (age group, race, gender). A Python script was developed to get input from the user and turn that input into the values that our random forest model is expecting. Then, these inputs are put into the random forest algorithm in a loop for all zip codes in New York City and the predictions for risk level are returned. Lastly, the zip code and predictions are mapped.

|  |  |
| --- | --- |
| *Figure 7: Prompts for input from user regarding their demographic information* | |
| *Figure 8: Output from random forest model for the same demographic on every zip code* | *Figure 9: Visualization created from the risk predictions made by the random forest model* |

### Software and Hardware Development Platforms

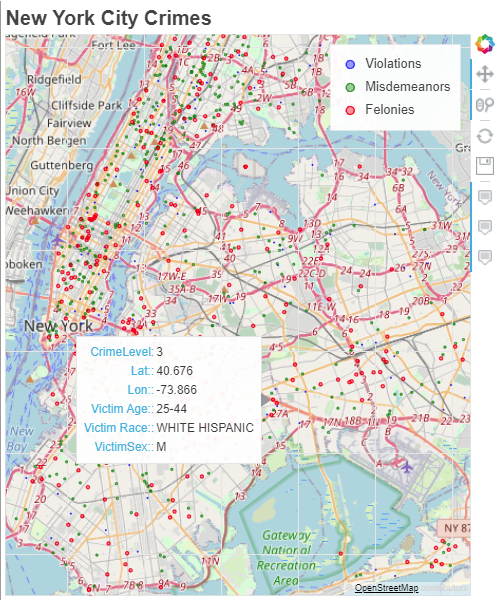
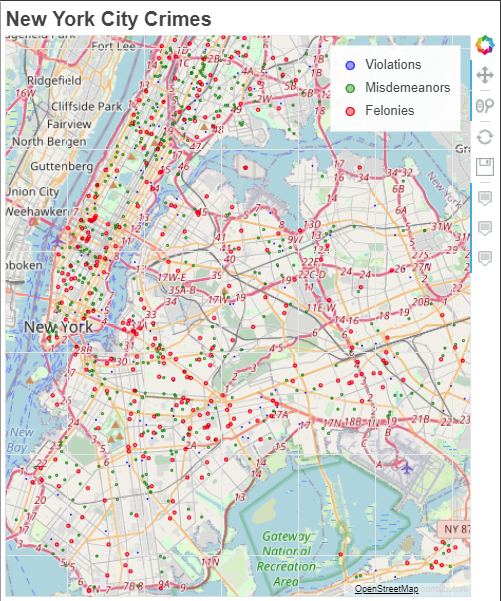
The software requirements for our functioning system are outlined in the “Architecture” subsection above, and are all accessed through a set of virtualization platforms. MongoDB was utilized through the use of MongoDB Atlas, an online platform allowing easy collaboration on data analytics. The Apache Spark platform, as well as all other portions of our system, were accessed using Google Collaboratory, another virtual tool permitting group-based efforts in code-based modeling. Neither of the above platforms have any hardware requirements, since they can be accessed via any standard Internet browser, and are limited only by a user’s connection.

## Experimental Results

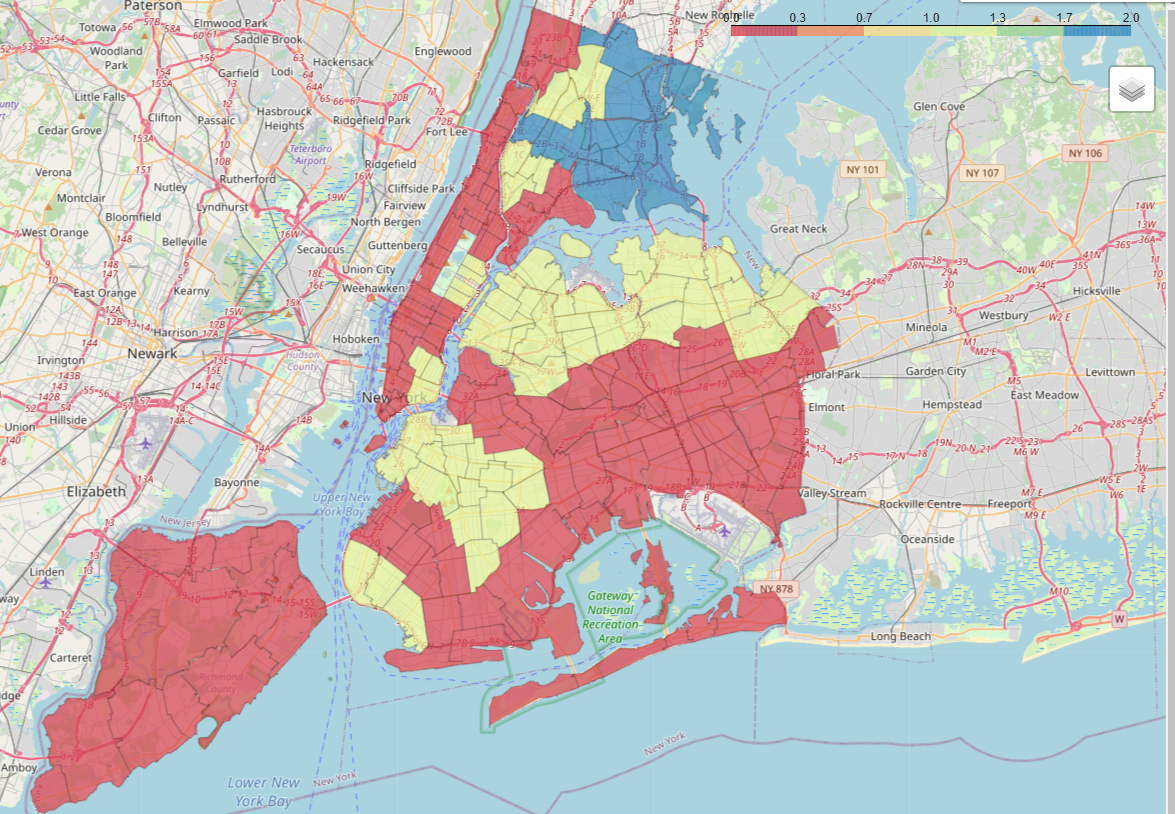
Through our efforts with a repository of custom code and machine learning models, we have created a system that generates the data necessary for analysis and loads it into MongoDB, at which point it can be injected into the Apache Spark ecosystem into an RDD. Various modeling processes are applied, and the results are output as visualizations based on a set of Python libraries. We have tested our system with various datasets ranging from 1,000-30,000 records, and have obtained data indicating the accuracy and limitations of our system. Through the development of our machine learning algorithm, we went through four iterations of a model before identifying one that had the best performance coupled with the most effective data output:

* The first iteration used a decision tree based on the five boroughs of New York City which was developed as a proof of concept for the model that we had planned. This model had a 92% accuracy in predicting risk level at the borough level which gave us the confidence to move forward with this idea. However, the breadth of the data zones was determined by our team to be too great, meaning that determining a risk level for such broad areas of NYC would provide little value so we looked into a more granular breakdown of areas in New York City.
* The second iteration was based on over 150 zip codes for all of New York City. We quickly encountered issues with the sparsity of data in zip codes. To test the idea of using zip codes to create models and create visualizations, a decision tree model was created for only Staten Island which had a 79.2% accuracy based on our test set, due to sparsity issues with zip codes. Although our model’s accuracy was not where we had hoped, our visualization worked how we had hoped which helped us see the value in using zip codes throughout the proposed solution.
* The third iteration took care of the issue of sparsity in zip codes by putting the zip codes into their corresponding neighborhood then using the neighborhoods in the machine learning algorithms. It was during this iteration that we also brought in the concept of using a random forest to compare its accuracy to decision tree models. This greatly increased the accuracy of the model to approximately 87% which we deemed as more acceptable. However, the neighborhood approach did not produce the type of visualization that we were going for in this approach.
* The fourth iteration, which is what we are presenting as our solution, combines the best aspects of iterations two and three. In this iteration, the machine learning model is provided with neighborhood information rather than zip codes for training and testing. For the visualizations, there is an intermediary step added that takes a zip code, converts the zip code to a neighborhood as input in the machine learning model, then the machine learning model output neighborhood is converted back to a zip code to use for visualizations. This model was scaled up from being used for only Staten Island to being used to make predictions and visualizations for all of New York City with impressive results.

Using Folium and Bokeh, we were able to create two interactive, dynamic, map-based visualizations for the data within our system, fulfilling our initial project goal. The Python library Bokeh was used to create a bubble map over New York to demonstrate historical crime locations, differentiated in color and bubble radius by crime level (violations, misdemeanors, and felonies). This map can be used to get an idea of where crimes are most likely to occur which adds value to our model showing how likely a person is to be a victim of a crime in those areas. Figures 10 to 11 show what this visualization looks like:

*Figures 10-11: Bubble map of 1000 crimes, by crime category; single hovered crime with displayed details*

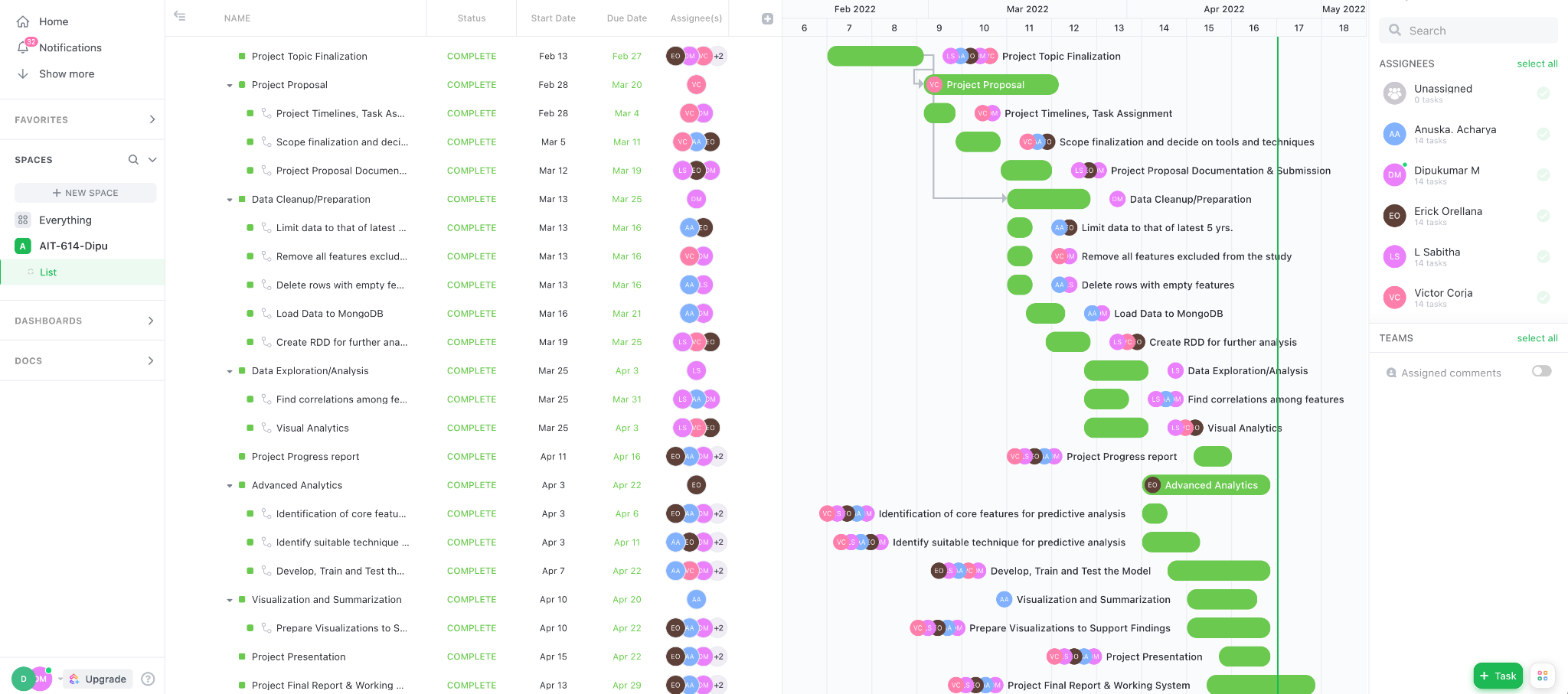
The Folium library was used to create a heatmap of NYC zip codes, containing color values based on the predictions of risk from our random forest model. The visualization in Figure 5 was made using the input demographic of a 25-45 year-old white female, as derived through our machine learning algorithm. The current model outputs are risk levels categorized as “Low”, “Medium”, or “High” based on clustering analysis on the frequency of different demographics being victims of a crime. In the visualization, the lower numbers on the scale represent low risk while the higher numbers represent higher risk.

**

*Figure 12: Heat map of New York City zip codes by risk level for a white female 25-45 years of age.*

## Project Timeline

Overall project development was divided into different sections and one team member was identified as responsible for the successful completion of that section. All team members contribute to these sections, based on their expertise with various tools and techniques. The project tasks and timelines are managed through the free version of online productivity tool ClickUp. The task list created and associated timeline with any updates is shown below. As expected, the bulk of the project time was spent on data exploration and data cleanup as well as researching different machine learning algorithms to select the most appropriate to accomplish the goals that we set forth in this project. The final product, including documentation, was completed on time.



## Conclusions

A significant conclusion we arrived at as a result of our research and experimental results was that increasing the granularity of a model does not always increase that model’s usability. Initially, we implemented a three-category risk level model for the zip codes found within Staten Island, developing this as a proof of concept model to later expand upon it. Our goal was to enhance the model’s accuracy, as well as to increase the granularity by expanding the geographical scope to the entirety of New York City and increasing the number of risk levels to at least 10, and preferably 20, representing 5-10% quartiles of risk probability. However, through experimentation, we determined that the two were mutually exclusive - an increase in granularity led to a significant loss of accuracy in our machine learning model, introducing an array of variables that caused the model’s predictive capabilities to drastically decrease. While we were able to increase the geographical scope of our model by using neighborhood clustering without causing any significant accuracy loss, and to increase the model’s accuracy by substituting the algorithmic methods used, the attempt to increase the number of risk category levels resulted in a 49% loss of accuracy. Despite the simplicity of the code changes needed to increase the number of categories into which our predictions could be sorted, which would have resulted in a more compelling and informative visualization, we concluded that in this case, the model’s accuracy was far more important than the quality of the visualization, meaning that the benefit brought by the change did not merit the accuracy loss.

While our model meets our initial goals and provides the desired functionality, there are still future enhancements that could be added to increase its usability and applicability. Despite our own decision not to increase the number of risk level categories, it could be worthwhile to identify a method such that this increase would not result in a significant accuracy decrease, as this would provide a much more usable visualization. Additionally, the implementation of additional variables such as suspect data, the crime’s time of day, season, or year, or other such data points, while increasing the complexity of the system and the models used, would boost the informational value of the system’s outputs significantly.

Through the process of creating our project system, there were many valuable lessons learned, as well as conclusions derived and decisions made as to the final structure of the system. We identified the tools and methodologies that would best allow us to perform a data analysis of a large dataset, and implemented those tools into a working system converting the raw data into a machine learning algorithm, with interactive visualizations.

## 

## 

## 

## 

## 

## 

## References

1. NYPD Complaint Data Historic | NYC Open Data. (n.d.). Retrieved February 26, 2022, from<https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Historic/qgea-i56i>
2. Rosenfeld, & Fornango, R. (2017). The Relationship Between Crime and Stop, Question, and Frisk Rates in New York City Neighborhoods. *Justice Quarterly, 34*(6), 931–951. <https://doi.org/10.1080/07418825.2016.1275748>
3. Herrmann, Maroko, A. R., & Taniguchi, T. A. (2021). Subway Station Closures and Robbery Hot Spots in New York City—Understanding Mobility Factors and Crime Reduction. *European Journal on Criminal Policy and Research, 27*(3), 415–432. <https://doi.org/10.1007/s10610-020-09476-x>
4. Almuhanna, A. A., Alrehili, M. M., Alsubhi, S. H., & Syed, L. (2021, April). *Prediction of crime in neighbourhoods of New York City using spatial data analysis*. In 2021 1st International conference on artificial intelligence and data analytics (CAIDA) (pp. 23-30). IEEE.
5. Pymongo 4.0.2 documentation. PyMongo 4.0.2 Documentation - PyMongo 4.0.2 documentation. (n.d.). Retrieved March 20, 2022, from <https://pymongo.readthedocs.io/en/stable/>
6. Botelho, B., & Vaughan, J. (2020, August 28). What is mongodb? A definition from whatis.com. *SearchDataManagement*. Retrieved March 20, 2022, from <https://www.techtarget.com/searchdatamanagement/definition/MongoDB>
7. RDD Programming Guide. RDD Programming Guide - Spark 3.2.1 Documentation. (n.d.). Retrieved March 20, 2022, from <https://spark.apache.org/docs/latest/rdd-programming-guide.html>
8. Machine Learning Library (mllib) guide. MLlib: Main Guide - Spark 3.2.1 Documentation. (n.d.). Retrieved March 20, 2022, from <https://spark.apache.org/docs/latest/ml-guide.html>
9. GitHub—Playdust/rgeocode: Offline reverse geocoder in Python using sqlite3. (n.d.). Retrieved April 23, 2022, from <https://github.com/playdust/rgeocode>
10. GitHub-thecraigd/BokehAvocado: Using Bokeh to create an interactive visualization of avocado prices. (n.d.). Retrieved April 23, 2022, from <https://github.com/thecraigd/BokehAvocado>
11. Sharma, Abhishek (2020, June 10). “Your Guide to Getting Started with Geospatial Analysis using Folium (with multiple case studies)”, Analytics Vidhya. Retrieved April 24, 2022, from <https://www.analyticsvidhya.com/blog/2020/06/guide-geospatial-analysis-folium-python/>
12. United Hospital. (2004, June 3). UHF codes - welcome to nyc.gov | city of New York. Retrieved May 1, 2022, from <https://www1.nyc.gov/assets/doh/downloads/pdf/ah/zipcodetable.pdf>
13. GitHub-fedhere/PUI2015\_EC: nyc-zip-code-tabulation-areas-polygons.geojson. (n.d.) Retrieved April 25, 2022, from <https://github.com/fedhere/PUI2015_EC>