Boston Crime Analysis

Presented by: Hunter Boles

For CSCI347 – Data Mining

Overview

Analysis of the Problem

Exploratory Data Analysis

Model Building

Conclusion

Analysis of the Problem

Goals

- Possible to predict crimes given its characteristics
  - Location?
  - Time of Day?
  - Day of the Week?

Why answer this question?

Allocate police officers efficiently, given the types of crimes in an area.

Exploratory Data Analysis

319,000 entries (60 unique classes)

Collected from June 2015 to September 2018

Provided by Analyze Boston

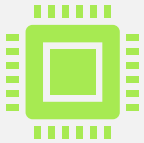
Data Overview

Variables Collected

Variable	Description
Incident Number	A unique identifier for the incident
Offense Code	A numerical code of the incident
Offense Code Group	A short name for the incident
Offense Code Desc.	A longer description of the incident
District	The city district that the crime was in
Reporting Area	A numerical location variable.
Shooting	Indicates whether there was a shooting.
Occurred on Date	Date and time of which the incident occurred
UCR Part	Partitions crimes by the Uniform Crime Reports (UCR)
Street	Street of the incident
Location	The latitude and longitude of the incident

Dealing with Null Values

Why are the null values in place?



Is information embedded within them?

Is it missing information?

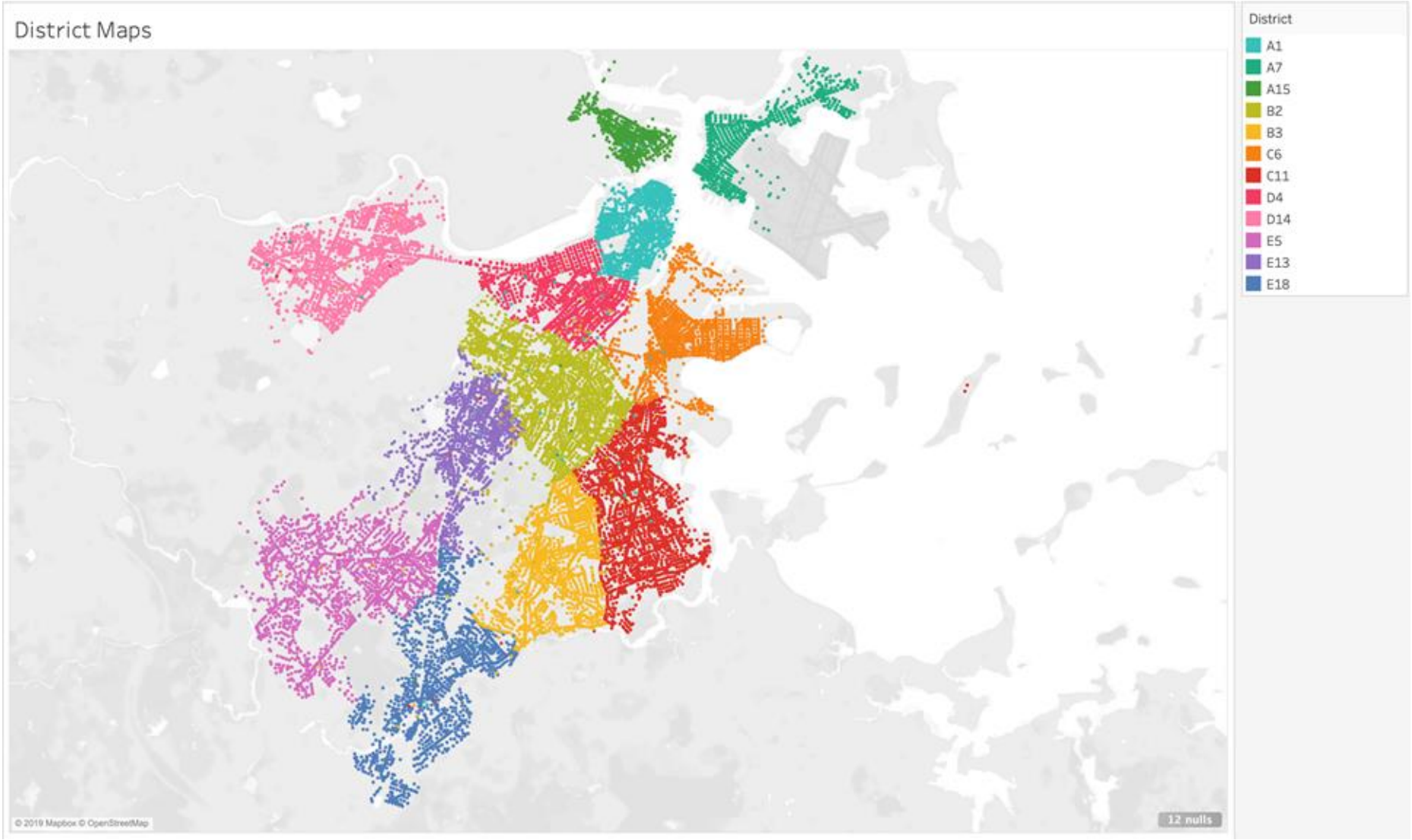
Null Values

Variable	Number of Null Values
District	1,765
Shooting	318,054
Street	10,871
Lat/Long	19,999
UCR Part	90

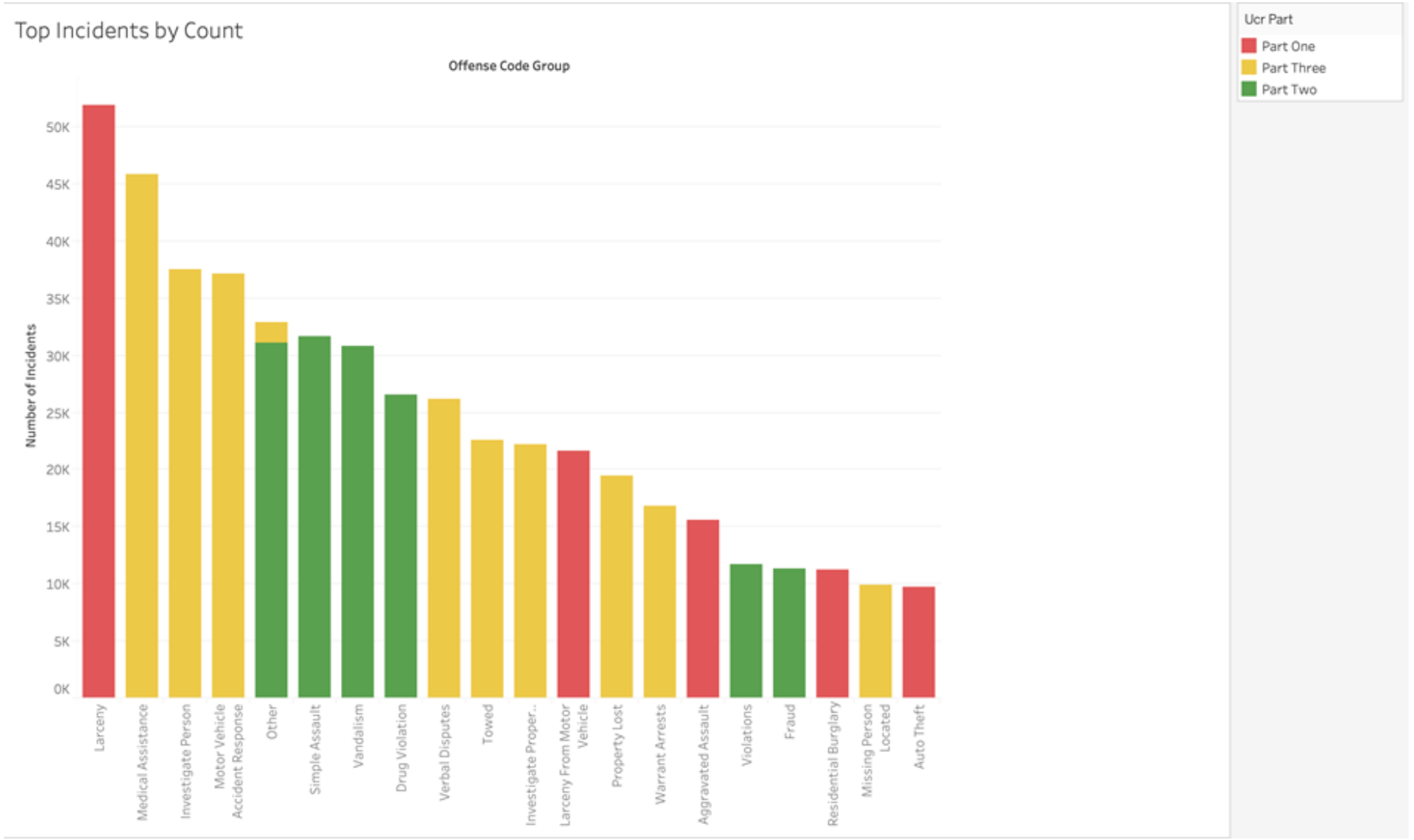
Exploratory Questions

- What does the district variable look like?
- What crimes are committed?
- What does the shooting variable look like?
- What crimes are connected to shootings?
- Is there a difference between different districts' crime profiles?
- Does the day of the week affect crime?

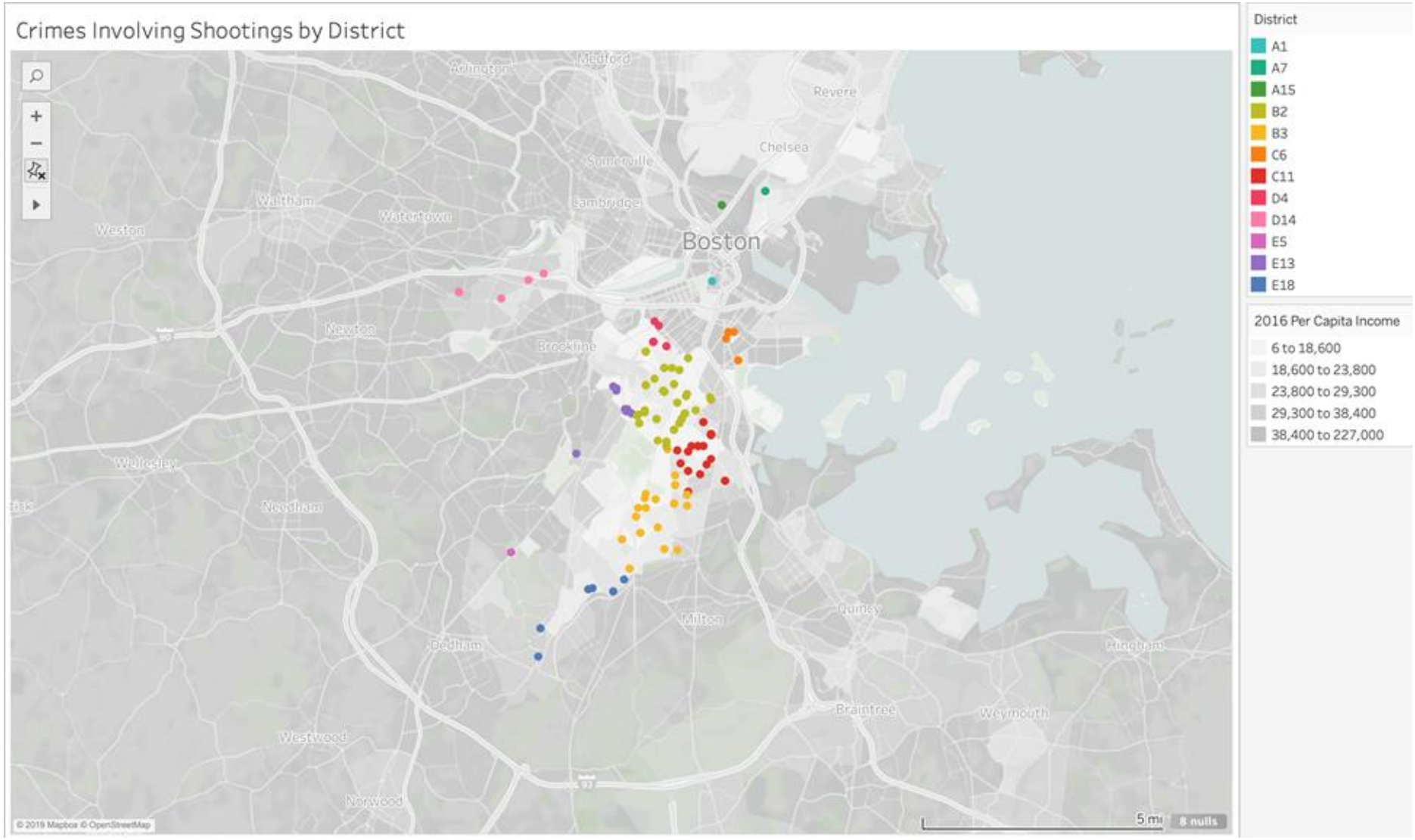
What does the district variable look like?



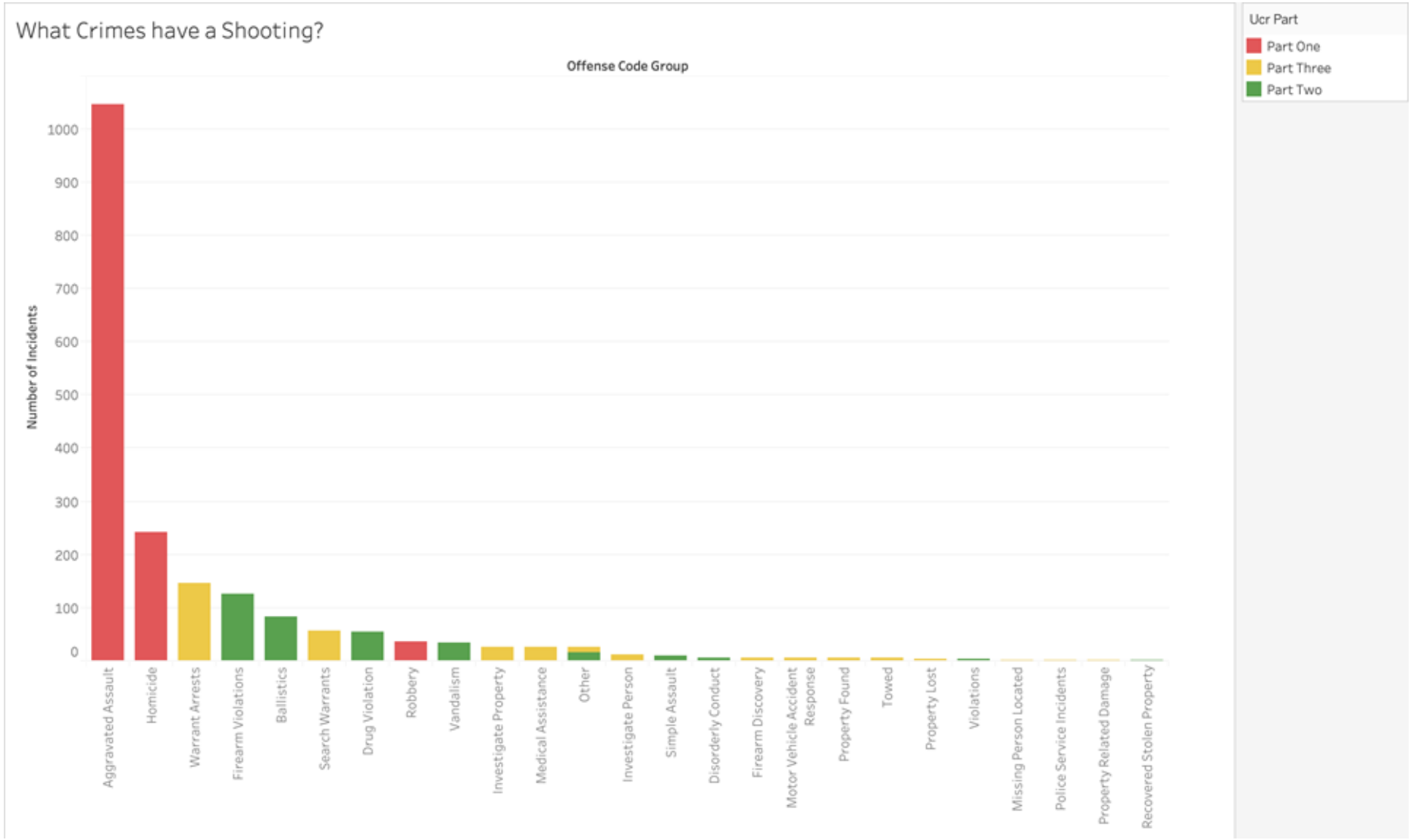
What crimes are committed?



What does the shooting variable look like?

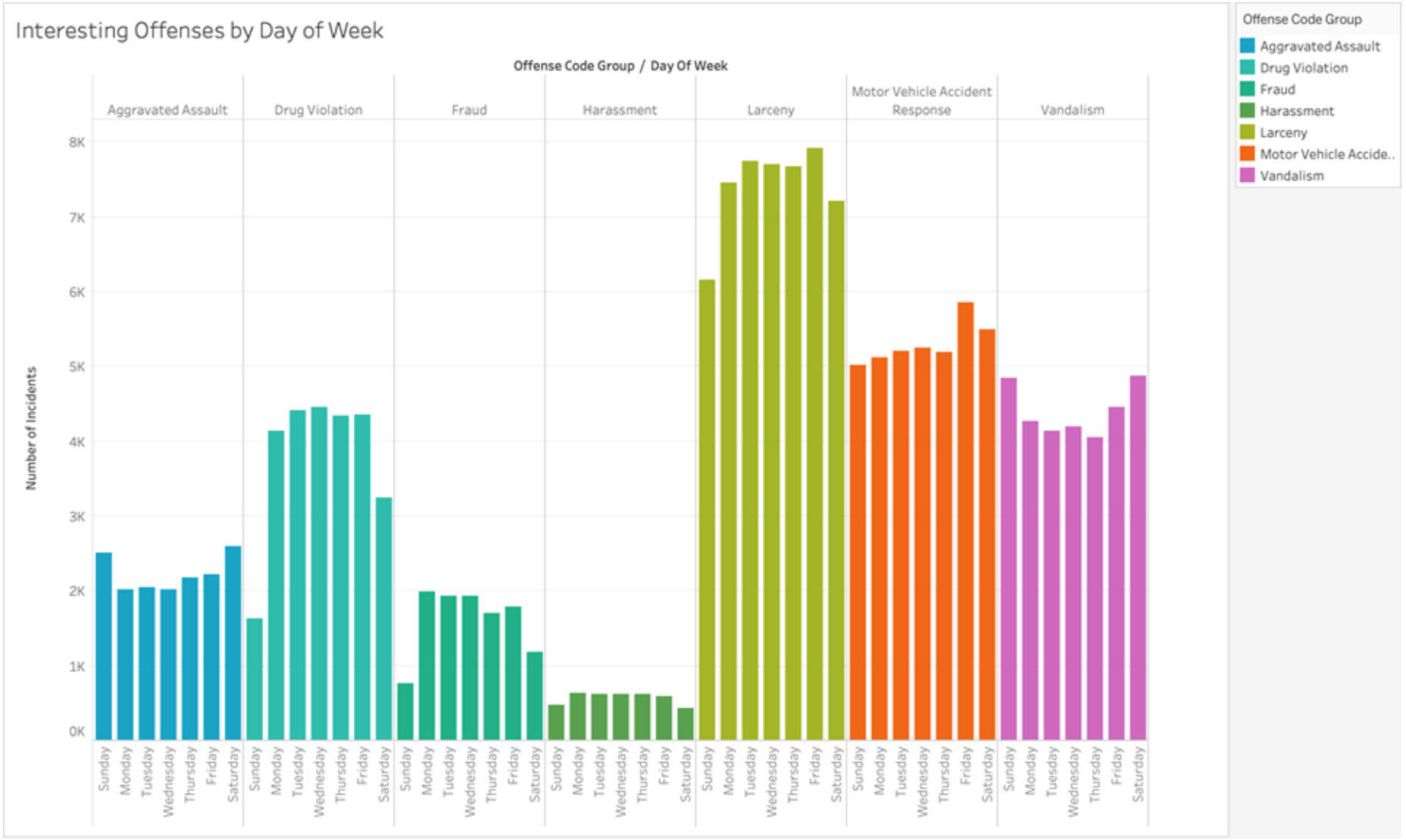


What crimes are connected to shootings?



Is there a difference between different districts' crime profiles?

Does the day of the week affect crime?



Model Building

Data Cleaning

- Offense Code Group: Removed duplicates by providing consistent formatting
- Day of the Week: Mapped to integer values
- Latitude / Longitude: Recalculated values using Location
- Shooting: Mapped to 0/1 instead of Y/Null
- Ignored Street and District.

Sampling Strategy

Important to split datasets into training and validation

Used an 80/20 split

Stratified Random Sampling

Keeps class value proportions roughly equal to entire dataset

No-Skill Test

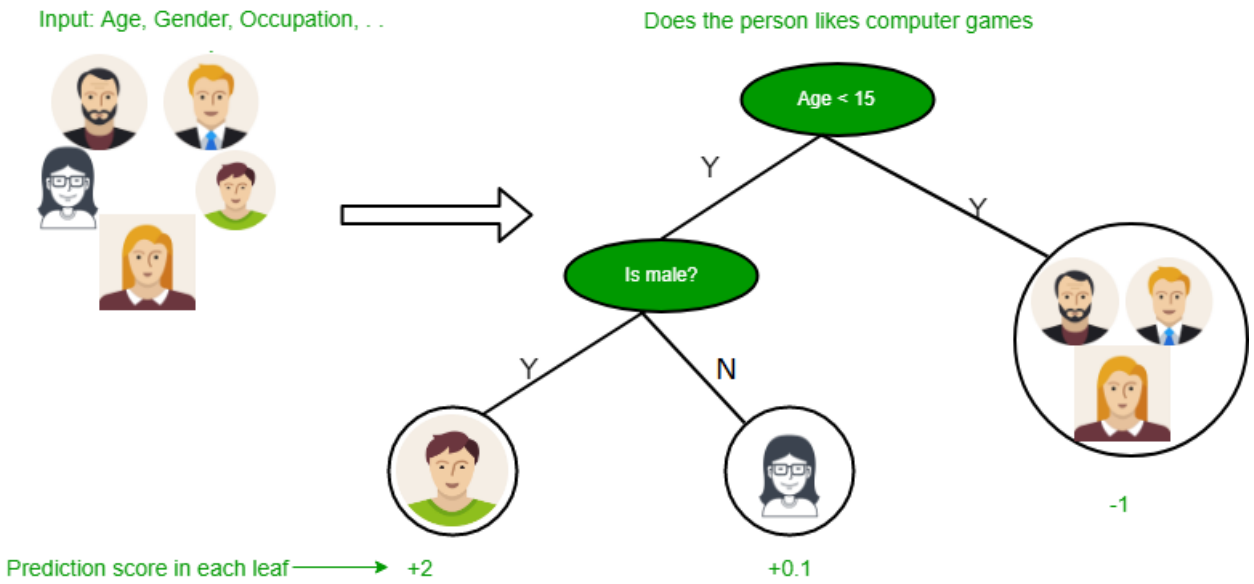
What class can be guessed for the highest accuracy?

Makes a great baseline case. If a model does worse, then it should be thrown out.

Decision Tree

Hyperparameters:

- Criterion: Gini / Entropy (Function to measure quality of split)
- Max Depth
- Splitter: Best, Random (Best split, best random split)



Random Forest

Makes multiple decision trees

- Averages results

Hyperparameters:

- Criterion: Entropy / Gini
- Max Depth
- N Estimators (How many trees in the forest)

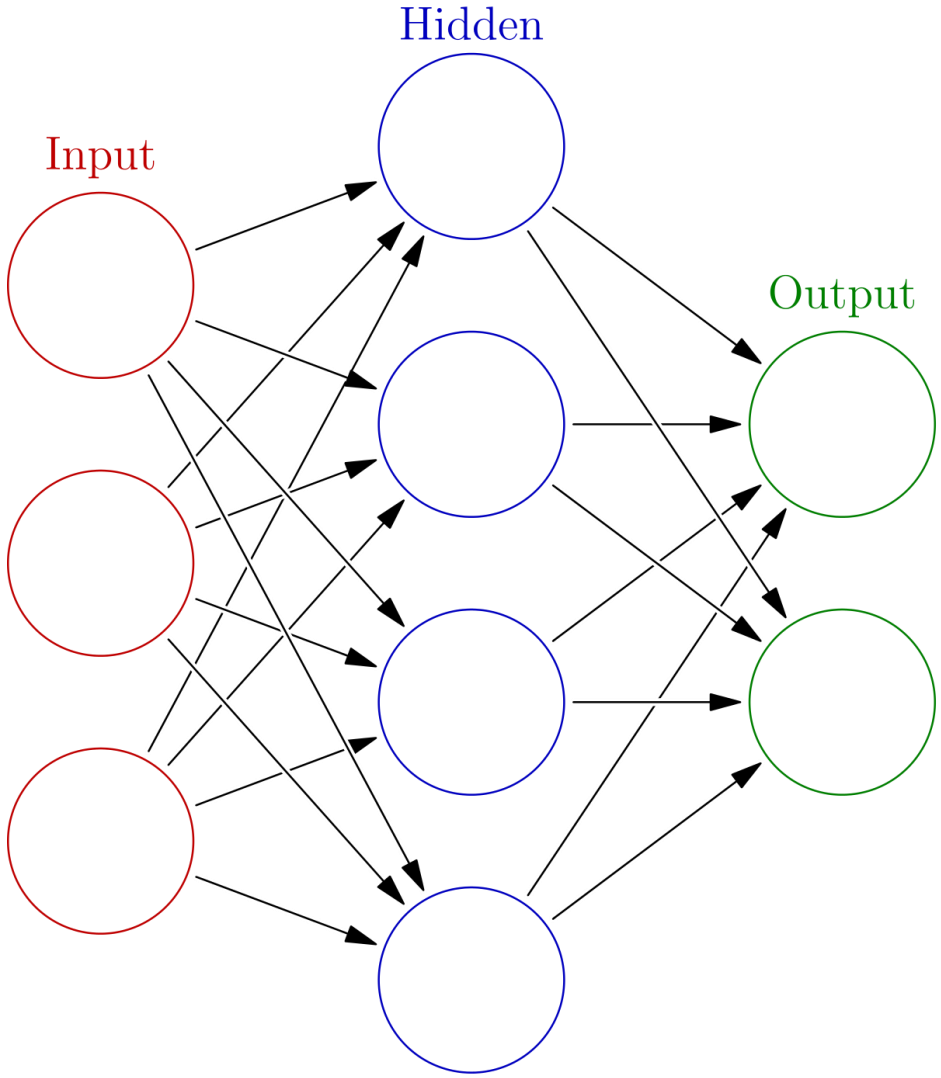
Multilayer Perceptron Neural Network (MLP-NN)

Hyperparameters:

hidden_layer_sizes

activation: What criteria is needed for the hidden neuron to fire. (ReLU, Tanh, …)

solver: lbfgs, sgd, adam (weight optimization)



Phase 1 Summary

- Ran No-Skill Test
  - 11.6% accuracy by guessing that there was a car accident
- Built Decision Tree and Random Forest
  - Out of the box hyperparameters
  - Overfit with 75% accuracy
- Built Multi-Layered Perceptron Neural Network
  - 11.6% Accuracy
- Realization that I was doing something completely wrong

Phase 1 Flaws

No hyperparameter tuning

- Maximum depth of tree, learning rate, etc.

Sloppy data cleaning

- Day of the week should have been one-hot encoded (As it is a categorical variable)
- Certain class values were being predicted with < 10 instances

Phase 1 Fixes (For Phase 2)

Set up a consistent framework for creating models

sampling
visualizations
metrics

Tune hyperparameters automatically

GridSearchCV with Scikit-Learn

Sped up the process significantly

XGBoost

A gradient boosting (Weights what it missed) algorithm.

- Reduces bias
- Reduces variance in supervised learning (Predicting a class value)

A decision tree on steroids.

No-Skill Test	Decision Tree	Random Forest	MLP Neural Network	XGBoost
• 11.6% -> 11.6%	• 74.0% -> 16.6%	• 74.0% -> 16.9%	• 11.6% -> 12.3%	• 0.0% -> 18.9%

Phase 2 Model Accuracy

Conclusion

- XGBoost performed the best (18.9% Accuracy)
- Lack of variables to classify 60 unique values
  - Income would be useful

GitHub / Source Code

https://github.com/Hunterb9101/Analysis-Crime-Dataset

Questions?