Mini Final Report

Sam Lee, Jeffry Troll

Introduction

Violent crime in Chicago remains a persistent challenge. The Chicago Police Department (CPD), particularly concerned about the situation in District 11, has partnered with Data Insight & Strategy Consultants (DISC) to explore innovative approaches. This project delves beyond traditional methods, incorporating human behavior, weather patterns, and even the lunar cycle, to analyze crime data and uncover potential predictors.

Our goal is to determine if and how these factors influence violent crime rates in Chicago. This knowledge will empower the CPD to proactively implement crime prevention strategies, such as public outreach and community events, during particularly vulnerable periods. By shedding light on the potential relationships between these factors and crime, we aim to make Chicago a safer place for all.

We start by analyzing the the number violent crimes that occur within District 11 over the past decade. We use the City of Chicago's definition (https://www.chicago.gov/city/en/sites/vrd/home/violence-victimization.html) to define violent crimes. To elicit the effect that weather has on crimes, we also divide out the crime by hour. We use historical weather data imported by a weather API to map hourly weather data to the hour the crime happened.

To analyze the predictive performance of these covariates, we employ several models. For more interpretation on the coefficients, we first train a generalized linear model adjusted with time-series components. For more predictive power, we employ more flexible models, including random forests, KNN, and clusters analysis.

We will first introduce the data used for this analysis, followed by some EDA we used to explore the data, the models we estimated, the results of our analysis, and then conclude.

Data Acquisition and Preprocessing

Primary Data Sources:

To conduct a comprehensive analysis of violent crime in Chicago's District 11, DISC acquired data from the following reputable sources:

- Chicago Crime Data: Detailed records of crimes reported in District 11 since 2001 were obtained from the official City of Chicago Data Portal https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-Present/ijzp-q8t2. We opted for this source to ensure data accuracy and consistency with official crime reporting procedures. Additionally, a valuable Kaggle resource https://www.kaggle.com/datasets/chicago/chicago-crime provided insights into the data structure and potential applications.
- Weather Data: Daily weather data for Chicago from 2010 to 2024, encompassing temperature, precipitation, wind speed, and other relevant meteorological variables, was originally retrieved from Visual Crossing (https://www.visualcrossing.com/). However, we asserted that to model the demand for violent crime more granually, as crimes were recorded by the hour; as such, we imported weather data from an historical weather API, https://open-meteo.com/.

Secondary Data Sources:

- Holiday Data: This data was obtained via ChatGPT for initial exploration, followed by cross-referencing with reliable online resources like national holiday calendars and local Chicago event listings.
- Full Moon Data: Dates of full moons since 2005 were sourced from Full Moon Info https://www.fullmoonology.com/full-moon-calendar-2015/, a website recognized for its comprehensive moon phase calendar. This data serves as a starting point for exploring potential lunar influences on crime rates.

Data Harmonization and Feature Engineering:

To ensure a consistent temporal scope across all data sets, a decision was made to utilize data from 2010 to 2024. While this approach sacrifices some crime data from the early 2000s, it allows for a more robust and comparable analysis with the available weather and holiday information.

Furthermore, monthly unemployment data for Chicago from 2010 to 2024 was incorporated as a socioeconomic indicator through using data from the Bureau of Labor Statistics.

Here's how we wrangled all the data files in R:

```
tryCatch({
  maps.api.key = read_file("Sam/maps_api.txt")
})

crimes <- read_csv("Sam/Data/Crimes.csv")
crimes$DateTime <- mdy_hms(crimes$Date)
crimes$Date <- as.Date(crimes$DateTime)
crimes$hour <- hour(crimes$DateTime)

moons <- read_csv("Sam/Data/full_moon.csv") %>%
```

```
  mutate(Date = dmy(FullMoonDates))

holidays <- read_csv("Sam/Data/holidays.csv") %>%
  mutate(Date = ymd(Date))

#Weather Data from API
weather.data <- fromJSON("Sam/Data/weather.json")["hourly"] %>%
  as.data.frame()
weather.data$hourly.time = weather.data$hourly.time %>%
  str_replace("T", " ")
weather.data$hourly.time = ymd_hm(weather.data$hourly.time)
weather.data <- weather.data %>%
  mutate(
    Date = as.Date(hourly.time),
    hour = hour(hourly.time) %>%
      as.numeric()
  )
weather.covariates <- colnames(weather.data)[2:(ncol(weather.data)-2)]

cutoff.date = max(
  c(min(crimes$Date), min(holidays$Date), min(weather.data$Date))
)
upper.cutoff = min(
  c(max(crimes$Date), max(holidays$Date), max(weather.data$Date))
)

crimes <- weather.data %>% left_join(crimes) %>%
  mutate(
    DateTime = hourly.time,
    Year = year(DateTime)
  )

week.days <- c("Mon", "Tues", "Wed", "Thurs", "Fri", "Sat", "Sun")

crimes.cleaned <- crimes %>%
  left_join(holidays) %>%
  left_join(moons) %>%
  mutate(
    Holiday = ifelse(is.na(Holiday), "", Holiday),
    DayofWeek = week.days[wday(Date, week_start=1)],
    FullMoon = ifelse(is.na(FullMoonDates),0,1)
  ) %>% filter(
    Date >= cutoff.date & Date <= upper.cutoff
  )
```

```r
factors = c("DateTime", "Date", "Primary Type", "Location Description",
            "Arrest", "Domestic", "Community Area", "Year", "Latitude",
            "Longitude", "FullMoon", "DayofWeek", "Holiday", "hour",
            weather.covariates)

crimes.cleaned <- crimes.cleaned[,factors]

#In the FBI's Uniform Crime Reporting (UCR) Program, violent crime is composed
#of four offenses: murder and nonnegligent manslaughter, forcible rape,
#robbery, and aggravated assault.

#https://www.chicago.gov/city/en/sites/vrd/home/violence-victimization.html
allcrimes = crimes.cleaned$`Primary Type` %>%
  unique() %>%
  sort()
violence.key <- c(0,1,1,1,0,1,0,1,0,0,0,1,1,0,1,1,0,0,0,0,
                  0,0,0,0,0,0,0,0,0,0,1,0,0,0,0)
mapping <- setNames(seq_along(unique(allcrimes)), unique(allcrimes))
crimes.cleaned$Violent <- violence.key[
  unname(mapping[crimes.cleaned$`Primary Type`])
]

crimes.cleaned <- crimes.cleaned %>%
  mutate(
    #These are the hours where there weren't any violent crimes
    Violent = ifelse(is.na(Violent), 0, Violent)
  )

crimes.cleaned <- crimes.cleaned %>% group_by(Violent, Date, hour) %>%
  mutate(
    NumViolentCrimes = Violent*n()
  ) %>% ungroup()

#Merge in unemployment data from BLS
unem <- read_csv("Sam/Data/chicago-unemployment.csv") %>%
  dplyr::select(Year, Label, Value) %>%
  mutate(
    Date = ym(Label),
    Month = month.name[month(Date)]
  ) %>% dplyr::select(-Label) %>% setNames(c(
    "Year", "Unemployment", "Date", "Month"
  )) %>% arrange(Date)
```

```r
crimes.cleaned$Month <- month.name[month(crimes.cleaned$Date)]

#Add in monthly unemployment
crimes.cleaned <- crimes.cleaned %>%
  left_join(
    unem %>% dplyr::select(-Date), by=join_by(Year, Month)
  )
```

Note that we define `NumViolentCrimes` as the number of violent crimes that happen within a given hour. This is one of our response variables.

EDA

Summary Statistics

Table 1: Summary Statistics of Location Variables

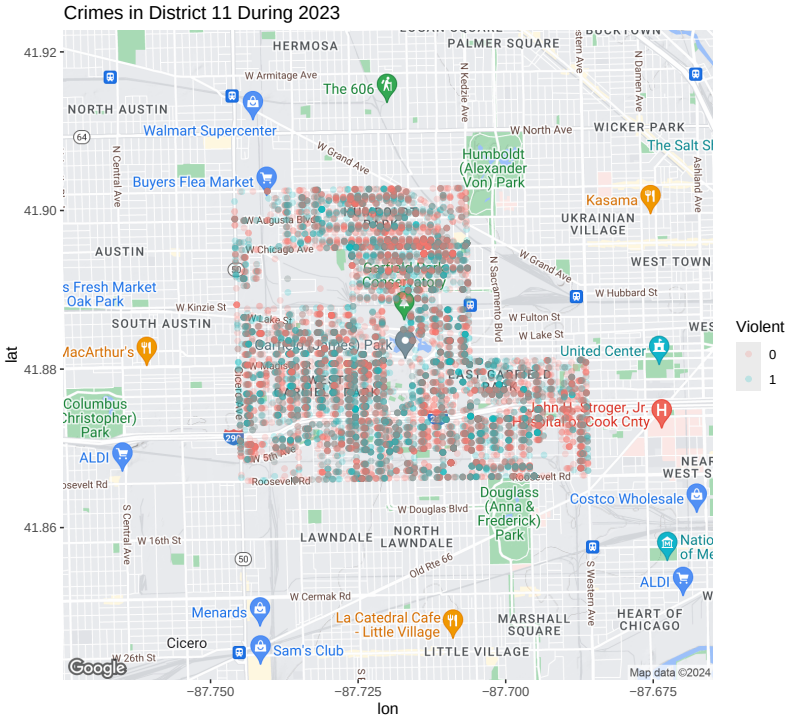
	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
Community Area	23.0	23.0	26.0	25.7	27.0	76.0	23768
Latitude	36.6	41.9	41.9	41.9	41.9	41.9	27740
Longitude	-91.7	-87.7	-87.7	-87.7	-87.7	-87.7	27740

Table 2: Summary Statistics of Numerical Variables

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Temperature	-33.2	2.4	12.0	11.30000	20.8	37.10
Humidity	11.0	59.0	71.0	70.40000	83.0	100.00
Apparent Temperature	-39.3	-2.3	8.9	9.01000	21.0	43.60
Rain	0.0	0.0	0.0	0.10600	0.0	47.70
Snowfall	0.0	0.0	0.0	0.00708	0.0	2.73
Snow Depth	0.0	0.0	0.0	0.01800	0.0	0.46
Cloud Cover	0.0	8.0	36.0	46.20000	89.0	100.00
Wind Speed	0.0	9.4	13.8	14.80000	19.2	54.50
Wind Gusts	1.8	19.4	27.7	29.00000	36.7	97.90
Is Day Time	0.0	0.0	1.0	0.57600	1.0	1.00
Shortwave Radiation	0.0	0.0	49.0	201.00000	361.0	1015.00
Direct Radiation	0.0	0.0	3.3	129.80000	200.2	910.70
Is Violent Crime	0.0	0.0	0.0	0.31900	1.0	1.00
Hourly Violent Crimes	0.0	0.0	0.0	0.59800	1.0	9.00
Unemployment	3.7	4.9	6.1	6.80000	8.7	16.80

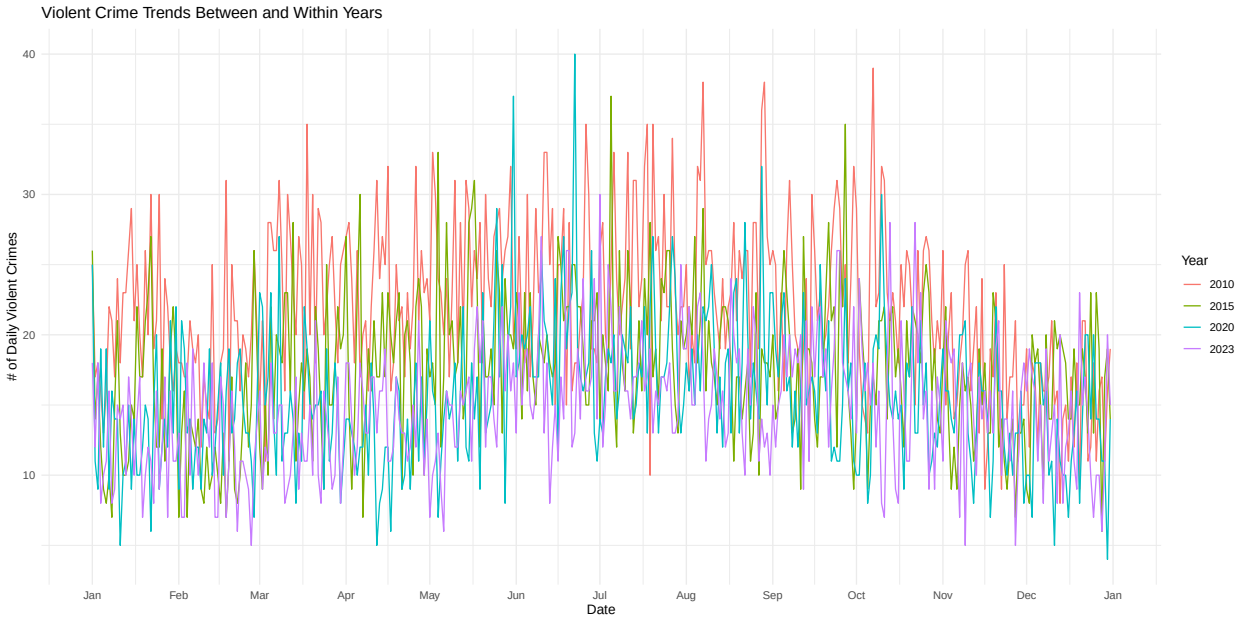
Note that summary statistics come from hourly records of crimes. The NA values on the location summary statistics are the hours where no crimes were observed.

Is there any spatial correlation between where crimes are occurring?



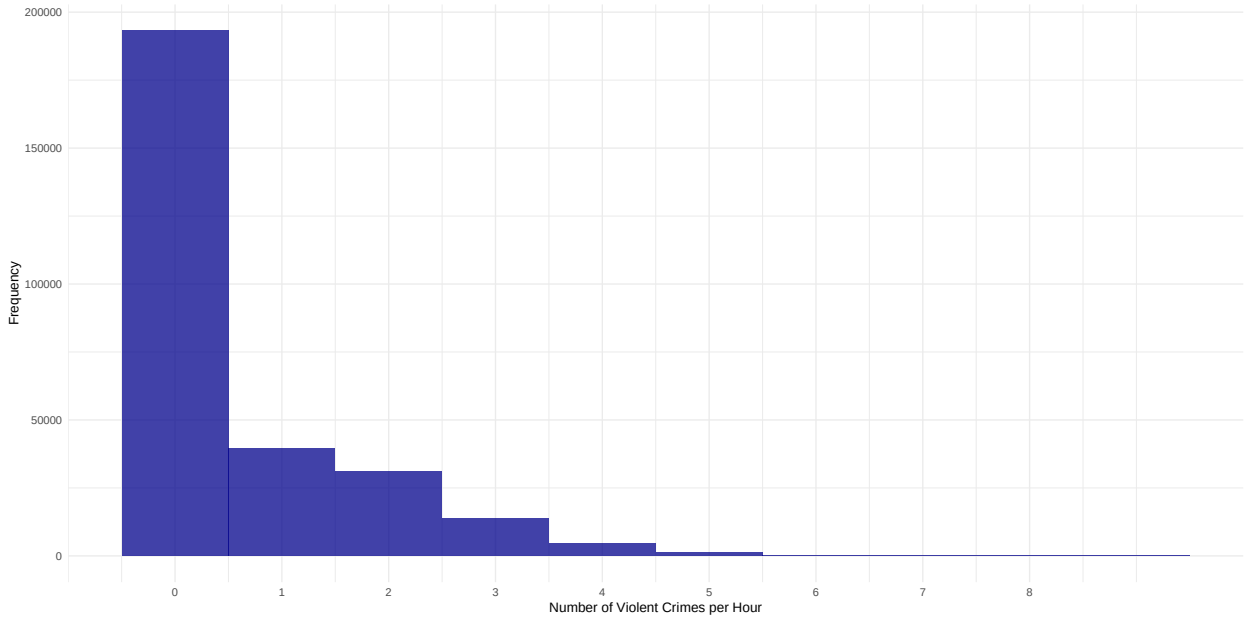
Based on the map above, since District 11 is such a small area, and since we don't have other data on the where crime is occurring in other districts, from this preliminary visual assessment, it doesn't look like there are any strong hotspots for violent crime throughout the year.

How do the number of violent crimes generally fluctuate throughout the year?



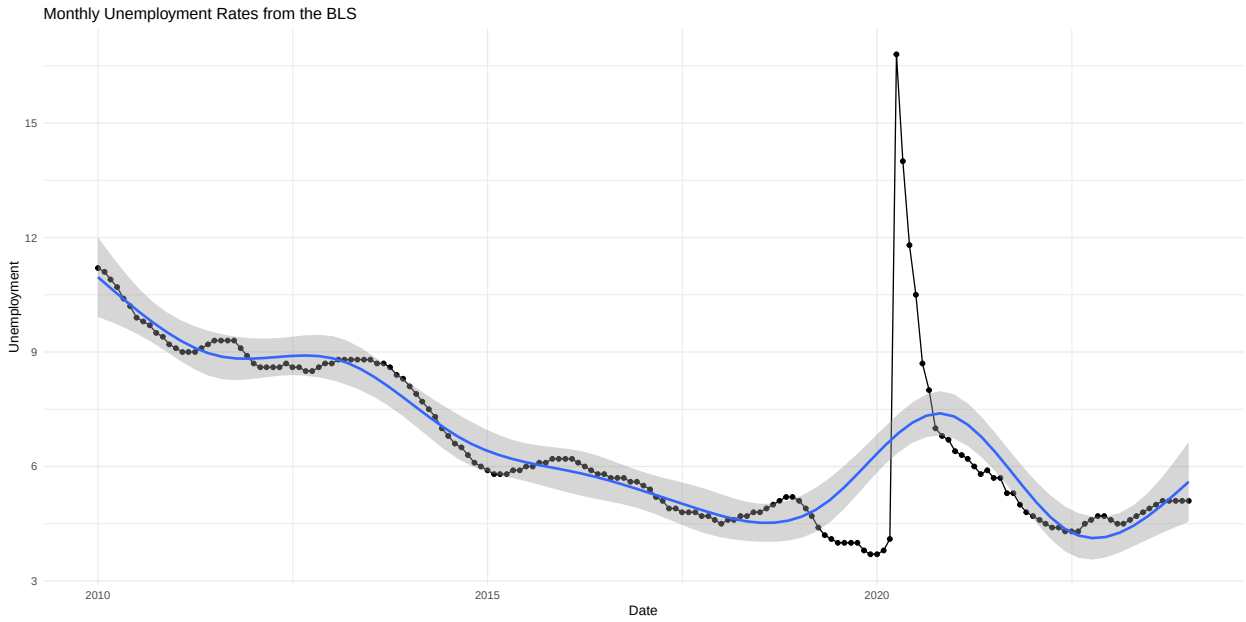
While it's hard to read in between the lines (literally), this graph shows that crime generally peaks during the summer months, while at the same time, crime has generally been decreasing over the pass decade.

What does the distribution of hourly violent crimes look like?



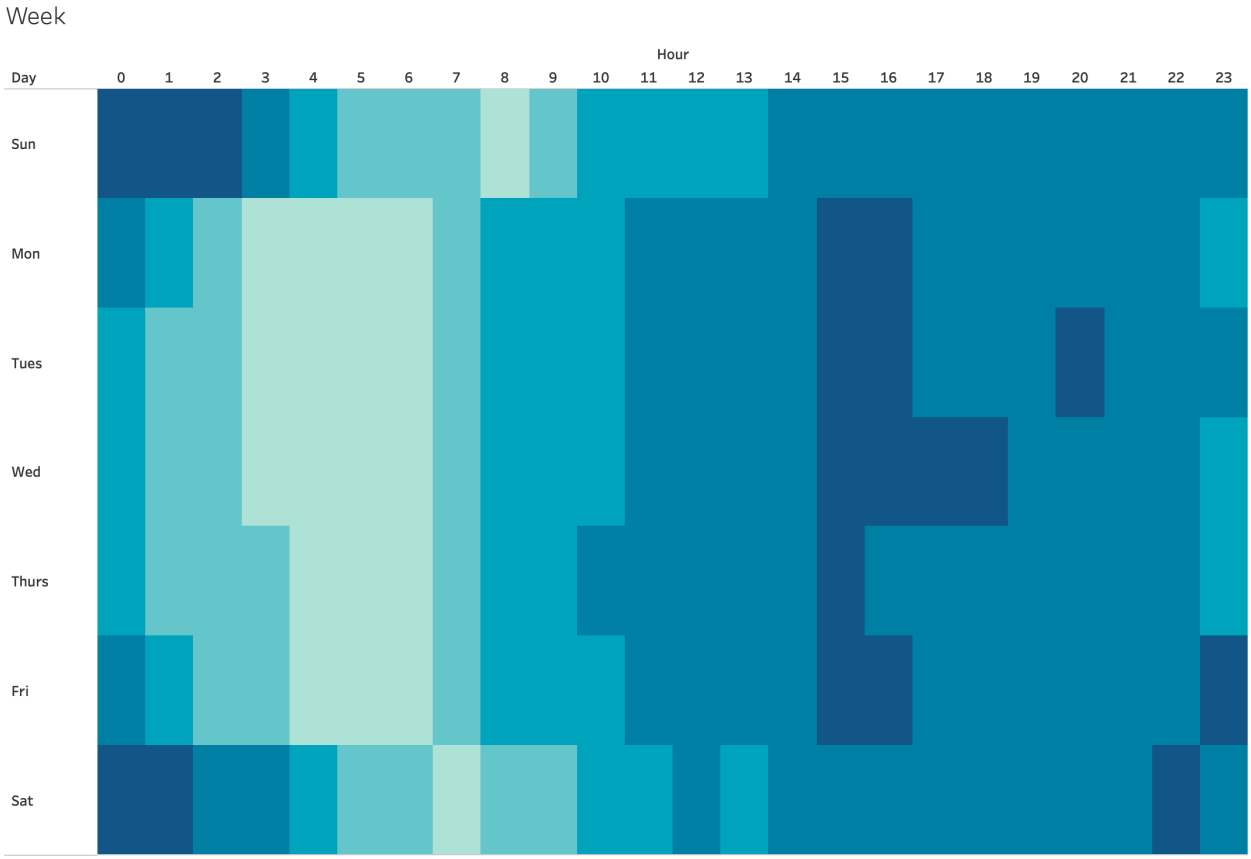
This appears that the distribution of the hourly violent crimes is Poisson-distributed. Specifically, since there's a (greater than expected) concentration of counts at 0, we first thought about modeling hourly violent number of crimes via a zero-inflated Poisson model (ZIP).

How does unemployment fluctuate throughout the observed decade in Chicago?



Unemployment seems to generally decrease throughout the decade (accounting for the seasonal fluctuations). There is a large spike around the COVID months. This could account for the spike we see in crime in the summer months in 2020. We include unemployment as a factor in our models.

How does the number of violent crime fluctuate throughout the hour of day and day of the week?



It appears that the early hours of the morning are only prone to more violent crime during the weekend, while violent crimes typically occur in the afternoon on weekdays. We want to account for this in our models.

Data Analysis, Model Building, and Interpretation

Generalized Linear Models

We first attempted to understand how the different coefficients were affecting our response variable, the number of violent crimes per hour. We employed several generalized linear models:

1) Poisson Regression
2) Negative Binomial Regression
3) Lasso Regression

4) Zero-Inflated Poisson Regression

The main linear

Main Model

$y \sim MVN(\mathbf{X}\square, \sigma^2\mathbf{R}(\phi,\omega))$, $\mathbf{y} = \mathbf{X}\square + \epsilon$, where $\epsilon \sim N(0, \sigma^2 R(\phi,\omega))$

$$\mathbf{y} = \begin{bmatrix} WHC_1 \\ WHC_2 \\ \vdots \\ WHC_{528} \end{bmatrix},$$

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ \vdots & \vdots & \vdots \\ 1 & x_{5281} & x_{5282} \end{bmatrix}, \beta = \begin{bmatrix} \beta_0 \\ \beta_{Yield} \\ \beta_{EC} \end{bmatrix},$$

$$\mathbf{R} = \begin{bmatrix} 1 & \rho(s_1,s_2) & \rho(s_1,s_3) & \dots & \rho(s_1,s_{528}) \\ \rho(s_2,s_1) & 1 & \rho(s_2,s_3) & \dots & \rho(s_2,s_{528}) \\ \vdots & \vdots & \ddots & 1 & \rho(s_{527},s_{528}) \\ \rho(s_{528},s_1) & \rho(s_{528},s_2) & \rho(s_{528},s_3) & \dots & 1 \end{bmatrix}, \text{ where } \rho(s_i,s_j) = exp\{-\frac{||s_i - s_j||}{\phi}\}.$$

Within our model, we have 5 parameters.

- β_0 represents the intercept, which is the expected value when all other coefficients within our model are zero.

- β_{Yield} and β_{EC} represent the estimated change in WHC when Yield and EC are increased by 1. They are the coefficients of our model.
- σ^2 represents the unbiased maximum likelihood for the variance.
- **R** represents the correlation matrix between observations from the data, where we have chosen an exponential correlation function.
- ϕ represents the parameter used in the exponential correlation calculated from the data.
- ω represents the variance "nugget" to capture same-location variability.

[Can you add the model that we use and the parameters and stuff here]

#Predictive model

#Implementation of at least four standard supervised machine learning models We tried a variety of model to get try to get better predictions and compare those results. Indeed, we fit a Random Forest model, Decision Tree, Gradient boosting and KNN model and already mentioned Poisson model.

#Significant or innovative feature engineering

Since we noticed that that time was an important factor, we considered to add generate more variables that can handle the seasonality presented in the data.

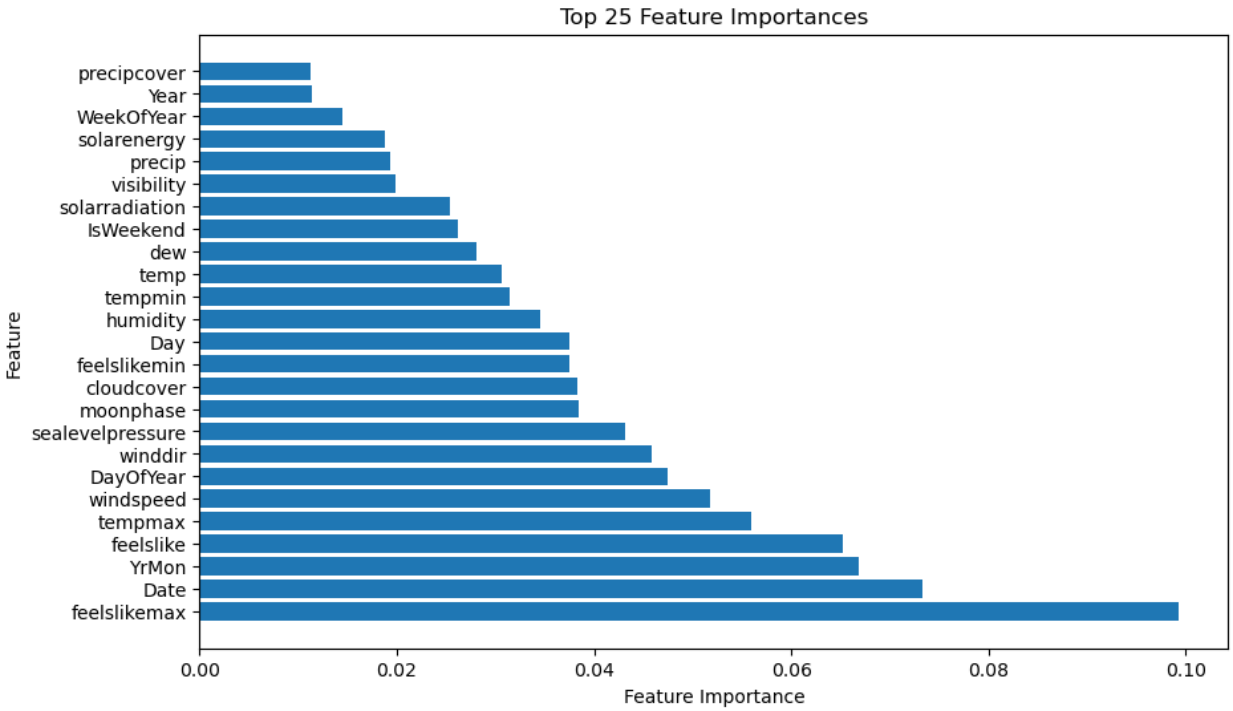
```
data['Month_Sin'] = np.sin(2 * np.pi * data['Month']/12)
data['Month_Cos'] = np.cos(2 * np.pi * data['Month']/12)
data['Day_Sin'] = np.sin(2 * np.pi * data['Day']/data['Date'].dt.days_in_month)
data['Day_Cos'] = np.cos(2 * np.pi * data['Day']/data['Date'].dt.days_in_month)
```

This feature engineering lies in capturing temporal patterns in a way that is more suitable for machine learning models, like this Random Forest. By encoding months and days as cyclical features, the model can understand their cyclic nature and learn patterns effectively, such as monthly or daily seasonality which is present in the crime data because of weather and another factors.

#An exploration of variable importance using SHAP

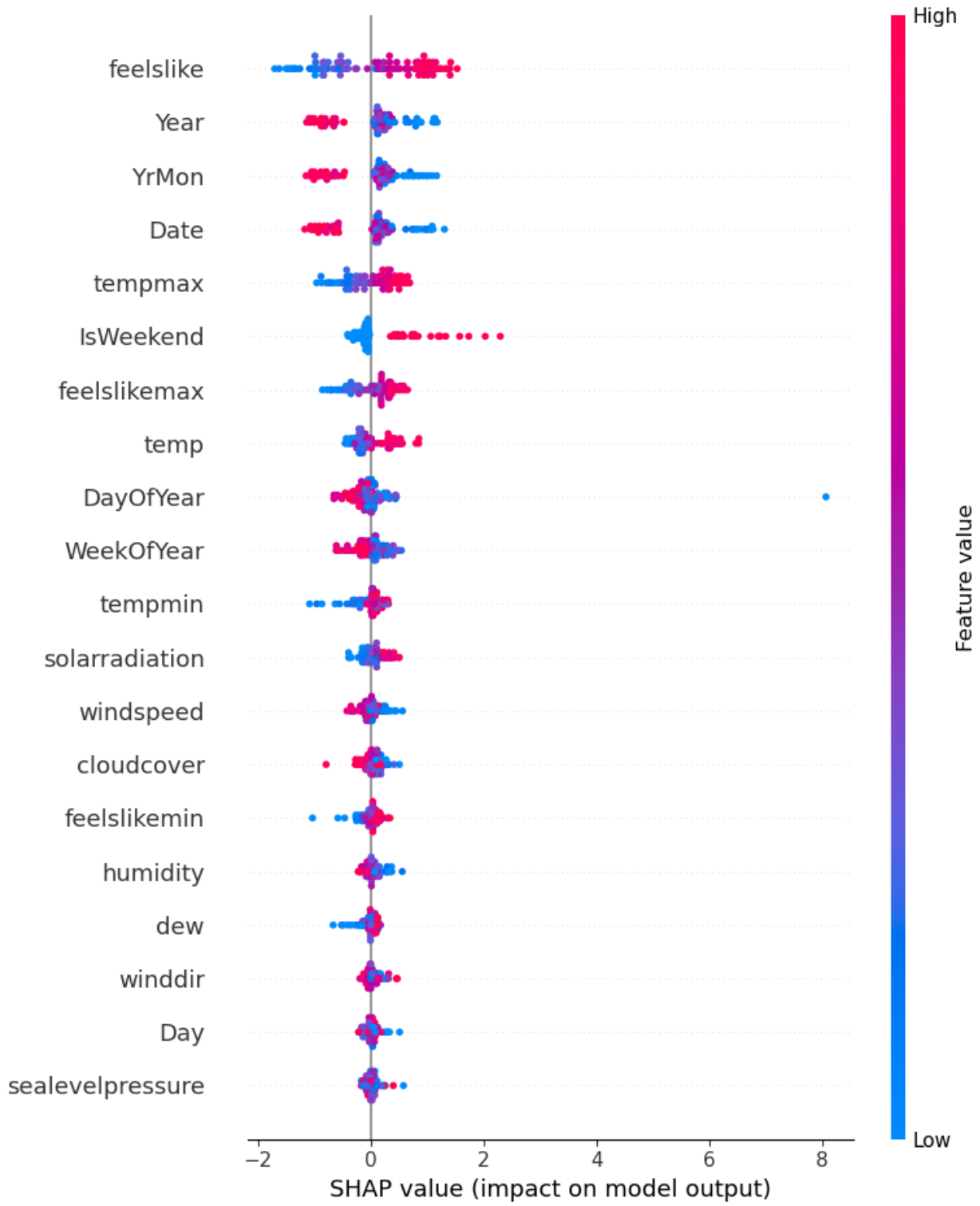
As we mentioned before, we used a Decision Tree to understand better the factors that impact the count of violent crime.

```
rf_model = RandomForestRegressor(n_estimators=20, random_state=42)

# Fit the model
rf_model.fit(X, y)

# Get feature importances
feature_importances = rf_model.feature_importances_

# Map feature names to their respective importances
feature_importance_map = dict(zip(X.columns, feature_importances))

# Sort the features by importance
sorted_features = sorted(feature_importance_map.items(), key=lambda x: x[1], reverse=T

# Extract top 25 features
top_features = sorted_features[:25]
```



We kind of expecting that that temperature will be an important factor, so we will check with SHAP as well to understand the impact of features on model predictions.

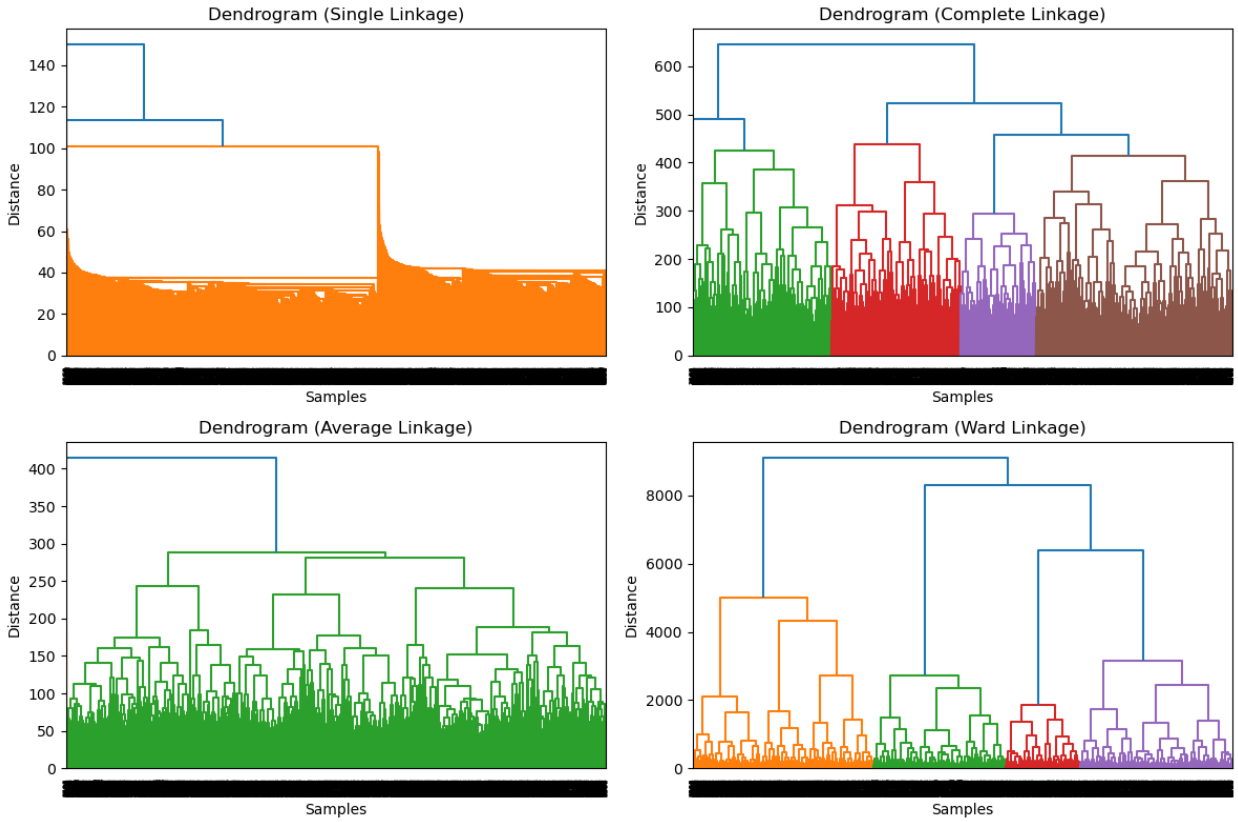
```
#Fitting the model
rf = RandomForestRegressor(n_estimators=200, random_state=42)
rf.fit(X_train, y_train)

#Getting SHAP values and plotting them
explainer = shap.Explainer(rf, X_train)
shap_values = explainer.shap_values(X_test[:100])
shap.summary_plot(shap_values, features=X_test[:100], feature_names=X_train.columns)
```



In summary, thanks to the SHAP values analysis, it seems that the model seems to be influenced significantly by perceived temperature (feelslike) and actual temperature measures, with time-related features also playing a key role. Lower on the importance scale are some other weather-related features and atmospheric conditions.

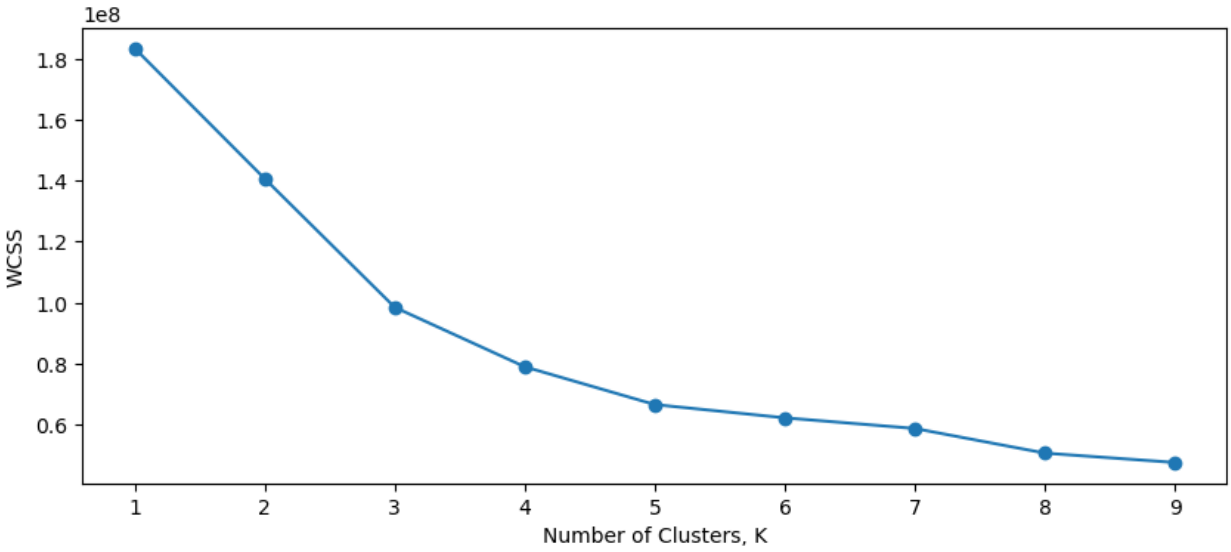
#Cluster analysis OR anomaly detection Also we wanted to apply cluster analysis to understand patterns or groups in how these factors correlate with crime rates

```
#Starting clustering analysis
kmeans = KMeans(n_clusters=3, n_init='auto')
kmeans.fit(X)
y_kmeans = kmeans.predict(X)

hc = AgglomerativeClustering(distance_threshold=None, n_clusters=3)
hc.fit(X)
linkage_methods = ['single', 'complete', 'average', 'ward']

# Plotting
plt.figure(figsize=(12, 8))
for i, method in enumerate(linkage_methods, 1):
    plt.subplot(2, 2, i)
    Z = linkage(X, method=method)
    dendrogram(Z)
    plt.title(f'Dendrogram ({method.capitalize()} Linkage)')
    plt.xlabel('Samples')
    plt.ylabel('Distance')

plt.tight_layout()
plt.show()
```



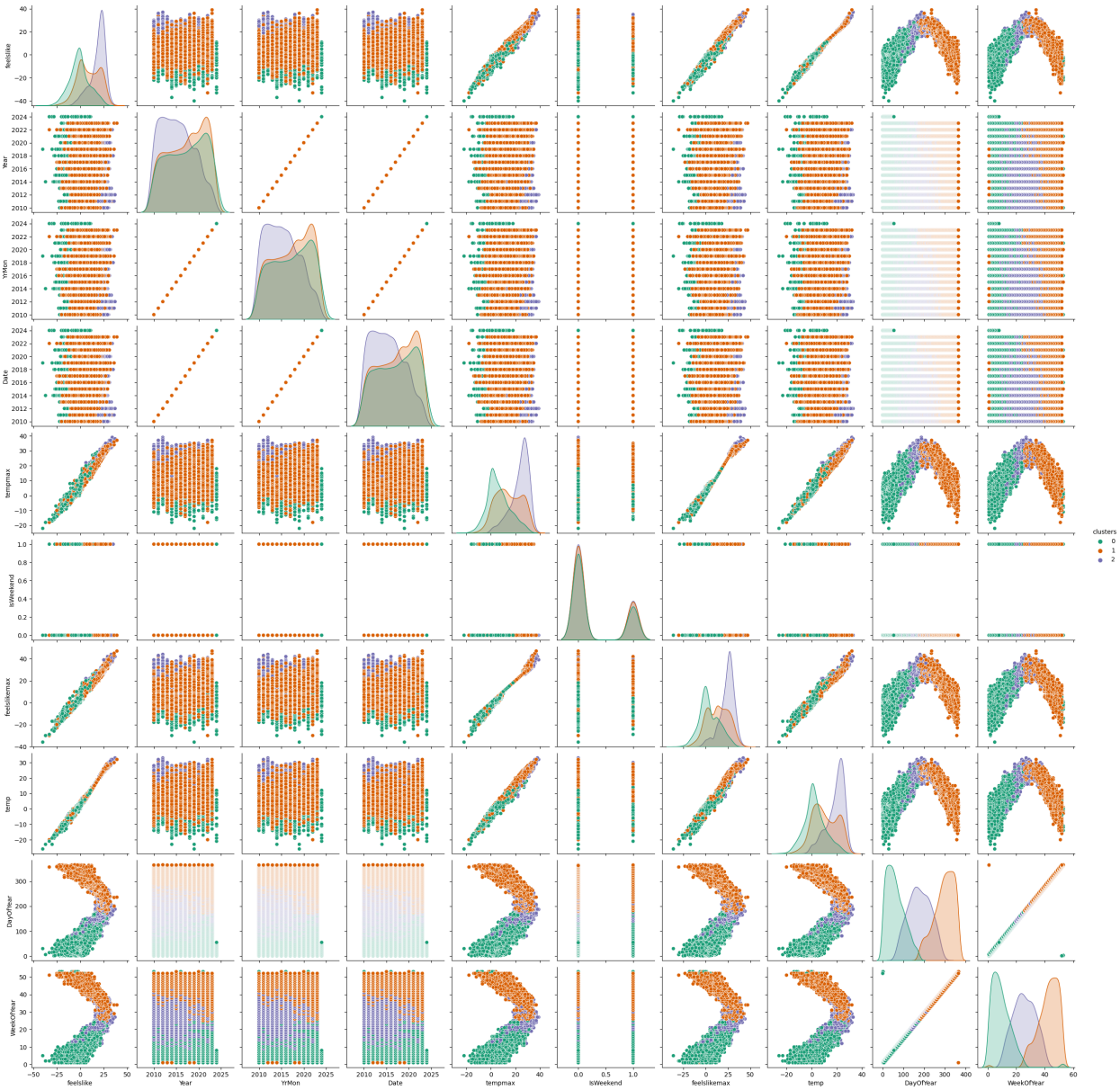
It seems that they're 3 main clusters in the data. This is confirm with the "elbow method" technique, since around 3 or 4 it seems to be the place where the rate of decrease in WCSS slows down significantly.

```
# Calculating WCSS by K number of clusters for plot
kmeans_per_k = [KMeans(n_clusters=k, n_init='auto', random_state=42).fit(X)
                for k in range(1, 10)]
inertias = [model.inertia_ for model in kmeans_per_k]


#Plotting WCSS by number of clusters
plt.figure(figsize=(10,4))
plt.plot(np.arange(len(inertias))+1,inertias,marker="o")
plt.xlabel('Number of Clusters, K')
plt.ylabel('WCSS')
```



After defining the number of clusters, we can start analyzing the major factors with these 3 clusters.

```
#Selecting factors to see
plot_df = pd.DataFrame(X[['feelslike', 'Year', 'YrMon', 'Date', 'tempmax', 'IsWeekend

#Plotting
plot_df['clusters'] = y_kmeansf Clusters, K')
plt.ylabel('WCSS')
sns.pairplot(plot_df, hue='clusters', palette='Dark2')
```



From the clustering analysis, we can learn that the feature relationships, distribution, and seasonality.