An Analysis of Crime in Milwaukee

Anthony Rautmann
Marquette Department of Math and Statistical Sciences
Milwaukee, Wisconsin 53233
anthony.rautmann@marquette.edu

Donghyun Shin
Marquette Department of Computer Science
Milwaukee, Wisconsin 53233
donghyun.shin@marquette.edu

Ryan Ramirez
Marquette Department of Computer Science
Milwaukee, Wisconsin 53233
ryan.ramirez@marquette.edu

Abstract

Through our analysis of crime in Milwaukee, we examined two datasets, which included all occurrences of crime in Milwaukee, from 2005 until 2021. From these datasets, we made a geoplot of criminal activity separated by aldermanic city districts, to show which districts had the highest frequency of crime. Using the geoplot, it was evident where the majority of crime took place so we set up a prediction model based on previous occurrences of crime, to determine if it was possible to predict crime for two types of offense: theft and assault. Our prediction model was trained on the frequency of thefts over a period of time, which was then able to predict a trend in crime activity up to one day in advance. The prediction showed an overall reduction in crime, perhaps due to the seasonality, as Fall becomes Winter.

1. Introduction

With a crime rate of 40 per one thousand residents, Milwaukee has one of the highest crime rates in America, compared to all communities of all sizes - from the smallest towns to the very largest of cities. One's chance of becoming a victim of either violent crime of theft or property is one in 25. [Neighborhood Scout]. From the frequent police reports sent to our mobile devices, or the wailing of sirens from emergency vehicles, crime is often a topic of discussion and a significant concern for Marquette students and Milwaukee residents, alike. Our project goal became clear - to present an analysis on the current and recent historical data of crime in Milwaukee, from 2018 - 2021. To do this, we split the project goal into two parts: visualizations and predictions. First, we wanted to show a map of all the districts in Milwaukee and distinguish which district had the highest and lowest rate of crime. In addition, what potential factors could be influencing the given rates of crime, such as time, day, or seasonality. The data was processed and grouped by district and seasonality to elucidate any trends by region and season. The grouping of the data in this way allowed for the geoplot to map the data and automatically split it into district regions. Our next goal was to construct a prediction model using ARIMA forecasting, to determine if crime could be predicted up to one day in advance.

2. Methodology

2.1 DataSet

Once the idea of our project was determined, and the project goal was outlined, we began working on our datasets. Using the Wisconsin Incident Based Report (WIBR) - a government database - we obtained two sets of data. One containing the most current criminal activity of this year, and the other containing all criminal activity from 2005 until 2020.

_id	IncidentNum	ReportedDateTime	Location	WeaponUsed	ALD	NSP	POLICE	TRACT	WARD	ZIP	RoughX	RoughY	Arson	AssaultOffense	Burgla
1	213020025	2021-10-29 02:58:00	3317 W MICHIGAN ST	UNKNOWN	4	14	3	13400	197	53208	2546483.140668109059	384837.161534857005	0	1	0
2	213020020	2021-10-29 02:30:00	1812 W LINCOLN AV	PERSONAL WEAPON	12	17	2	17400	257	53215	2552063.161784766242	372377.882274452597	0	1	0
3	213020028	2021-10-29 02:20:00	5076 N 24TH PL	BLUNT OBJECT	1	3	7	2300	56	53209	2549399.736252210569	411373.961952932179	0	1	0
4	213020019	2021-10-29 01:57:09	1945 N OAKLAND AV		3		1	10800	179	53202	2563710.171055797953	392343.637745097280	0	1	0
5	213010200	2021-10-28 22:50:00	4945 N 39TH ST	PERSONAL WEAPON ASPHYXIATION	1	2	7	2500	51	53209	2544348.057531779632	410186.135352134705	0	1	0

Figure 1 Original Dataset

The figure above is a small snapshot of well over 700,000 data points present in our original dataset. It shows the raw dataset before any cleaning and processing was made.

2.2 Data Processing

From these datasets, the data was cleaned and processed to be fitted towards our project goal. To do so, the two datasets were first concatenated into one large dataset. From there, we cleaned the data by removing all observations prior to 2018, since the scope of our project is from 2018 to 2021. Next, all unused columns were removed, and the sum of each column was calculated by using a groupby function to find the total count of crime based on per aldermanic district. In the final step of processing the data location-coordinate values were reformatted to a new column ‘geometry’ in preparation for creating the geoplot visualization.

	IncidentNum	ReportedDateTime	ReportedYear	ReportedMonth	ALD	RoughX	RoughY	Arson	AssaultOffense	Burglary	CriminalDamage	Homicide	LockedVehicle	Robbery	SexOffense	Theft	VehicleTheft	geometry	count
0	213370032	2021-12-03 01:30:00	2021	12	11.0	2.545666e+06	367030.730972	0	1	0	0	0	0	0	0	0	0	POINT (2545665.878 367030.731)	20245.0
1	213370080	2021-12-03 00:05:00	2021	12	1.0	2.540891e+06	411450.962692	0	0	0	0	0	0	0	0	1	0	POINT (2540890.729 411450.963)	63606.0
2	213360166	2021-12-02 20:50:00	2021	12	15.0	2.544385e+06	391319.863441	0	1	0	0	0	0	0	0	0	0	POINT (2544384.796 391319.863)	83373.0
3	213360169	2021-12-02 18:30:00	2021	12	2.0	2.531349e+06	412680.931212	0	1	0	0	0	0	0	0	0	0	POINT (2531348.937 412680.931)	60738.0
4	213360137	2021-12-02 18:00:00	2021	12	5.0	2.523364e+06	405935.349436	0	1	0	0	0	0	0	0	0	0	POINT (2523363.890 405935.349)	35328.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
729934	213060042	2020-11-02 08:27:00	2020	11	3.0	2.561775e+06	389952.397000	0	0	0	0	0	1	0	0	0	0	POINT (2561774.569 389952.397)	45425.0
729935	213060082	2020-10-24 02:00:00	2020	10	8.0	2.547880e+06	372325.112000	0	1	0	0	0	0	0	0	0	0	POINT (2547879.658 372325.112)	42015.0
729936	213070035	2020-11-02 21:18:00	2020	11	3.0	2.561719e+06	395455.935000	0	0	0	0	0	0	0	0	0	1	POINT (2561719.325 395455.935)	45425.0
729937	213070196	2019-07-19 11:00:00	2019	7	6.0	2.558562e+06	396736.043000	0	0	0	0	0	0	0	0	1	0	POINT (2558562.352 396736.043)	72169.0
729938	213080111	2018-06-01 00:00:00	2018	6	12.0	2.555573e+06	374784.144000	0	0	0	0	0	0	0	1	0	0	POINT (2555572.840 374784.144)	60266.0

Figure 2 Final processed dataframe

2.3 GeoPlot Visualization

To transform and visualize the dataframe to map data, we plotted it by using geopandas library. Each observation was represented as a dot, and by the color we could easily discriminate whether which region had a higher crime rate. To visualize the location of Marquette university, we created a new dataframe, “df_marquette” and plotted it as a black dot in the map.

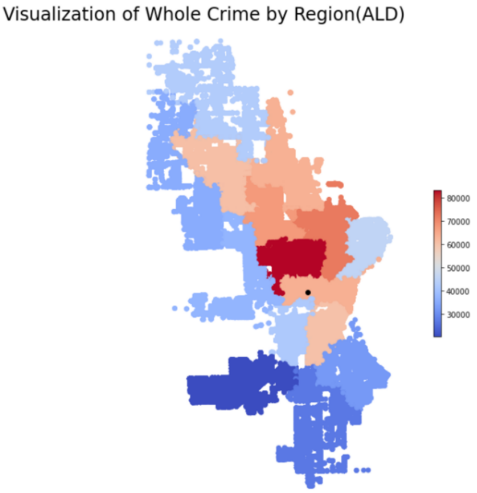


Figure 3. Visualization of Whole crime

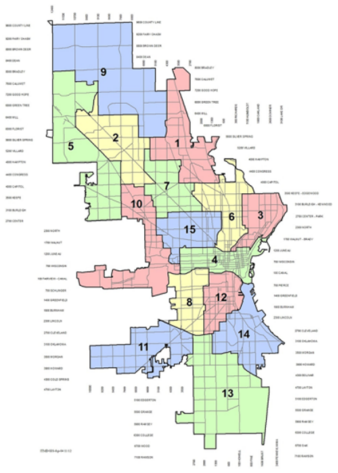


Figure 4. ALD - 15 districts of Milwaukee

The colormap indicates that from tan to dark red means a greater frequency of crime that has occured in that region, while light to dark blue means less crime, and thus a safer district. According to the geoplot, it can be concluded that District 15 has the highest crime rate at 83373 occurrences, and District 11 has the lowest crime, at 20245 occurrences.

We conducted the same visualization process based on types of crime, Assault Offense and Theft, which had the two highest crime rates.
If we compare three graphs (Figure 3, 5, 6), they look similar. Therefore, we can conclude that crimes occur the most in District 15, and the least in district 11.

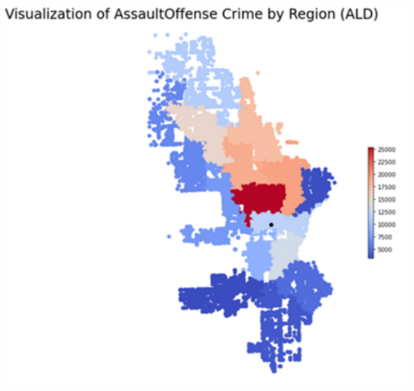


Figure 5. Visualization of AssaultOffense crime

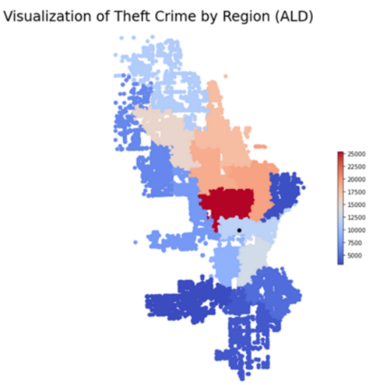
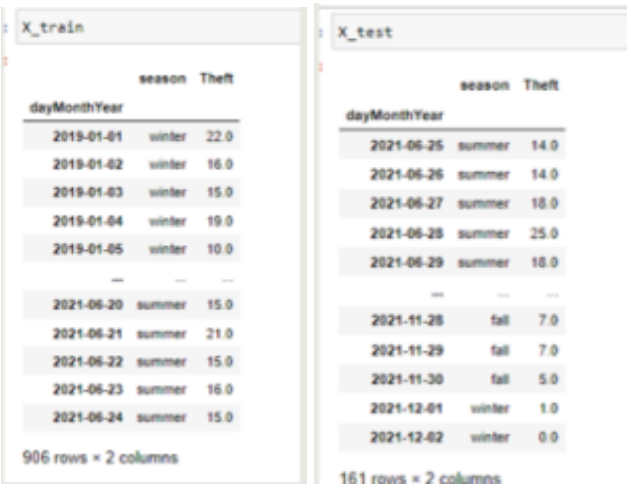


Figure 6. Visualization of Theft crime

3 ARIMA prediction

3.1 Data processing

The data processing section for the ARIMA model is different from the data processing implementation for the geoplot visualization section because for time series modeling we need to obtain the number of crimes per day rather than a count of crimes per ALD region. We plan to use 2019, 2020, and 2021 data to predict the number of thefts for days in 2022. Since we believe there may be some more variation in crime corresponding to different seasons, a seasonal column indicator needs to be created and mapped into the data frame by utilizing the date time column. We are only interested in theft for the type of crime column used to train our ARIMA model.



X_train

dayMonthYear	season	Theft
2019-01-01	winter	22.0
2019-01-02	winter	16.0
2019-01-03	winter	15.0
2019-01-04	winter	19.0
2019-01-05	winter	10.0
...	...	...
2021-06-20	summer	15.0
2021-06-21	summer	21.0
2021-06-22	summer	15.0
2021-06-23	summer	16.0
2021-06-24	summer	15.0

906 rows × 2 columns

X_test

dayMonthYear	season	Theft
2021-06-25	summer	14.0
2021-06-26	summer	14.0
2021-06-27	summer	18.0
2021-06-28	summer	25.0
2021-06-29	summer	18.0
...	...	...
2021-11-28	fall	7.0
2021-11-29	fall	7.0
2021-11-30	fall	5.0
2021-12-01	winter	1.0
2021-12-02	winter	0.0

161 rows × 2 columns

Figure 6. Finalized data frame used to train the ARIMA model, generated using Python.

Figure 6 shows the final dataframe after data processing. The original date time column was used to filter out for 2019 and onward, provided a grouping column for aggregate sum functions on the type of crime columns, and map a new season column into our data frame so that we could explore variation in crime by season. Finally, the original date time column was converted into the dayMonthYear column so that we could see the number of crimes per day rather than number of crimes per timestamp.

After we finalized the data frame, we split the data into separate train and test sets. We opted to use an 85 - 15 split between train and test. This is because we wanted a large amount of training data, as we are only going to be predicting days into the future. Notice that, in Figure 6, the training data set has 906 rows and the testing set has only 161 rows. The aggregations and filtering significantly reduced the size of our final data frame from the original data frames, but this final data frame is still based on 700,000 plus observations.

3.2 ARIMA parameters

The ARIMA model is used to fit and forecast time series data that exhibits a pattern and is not purely random data points.
(AR) - Auto Regressive component: Forecasted Y(t) is a function of (p) time lags
(I) - Integrated component: realized as the number of differences to make the series stationary
(MA) - Moving Average component: Forecasted Y(t) is a function of the lagged forecast errors

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p} \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + .. + \phi_q \epsilon_{t-q}$$

Figure 7. The ARIMA model function

The ARIMA model function shows the betas (B), which are the estimators corresponding to the time lags. It also shows the parameters on the forecasted errors, (E), which are the estimators corresponding to the lagged forecast errors.

ARIMA(p,d,q) - the parameters p, d, and q are to be set corresponding to the AR, I, and MA terms.
p = number of time lags used
d = minimum number of differences (new data = data point - previous data point) to make the time series stationary (see Figure 8 for stationary time series)
q = number of time lag error predictors to use

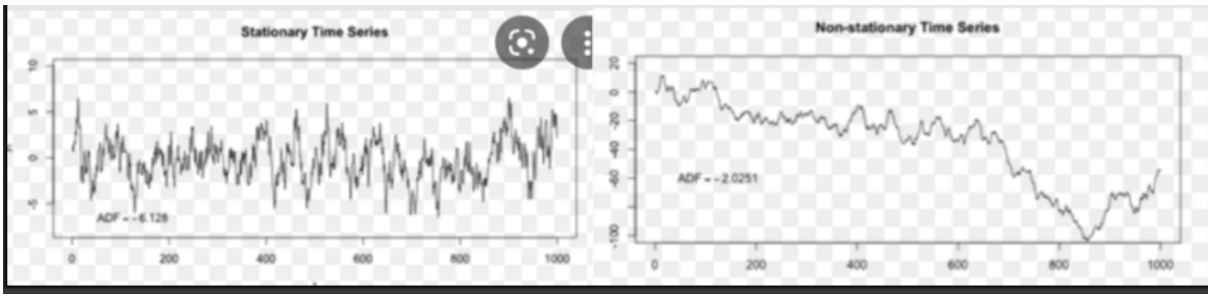


Figure 8. Representation of a stationary versus non-stationary time series

First, before we start determining what these parameters should be, look at a plot of the time series itself to see if it is stationary (see Figure 9).

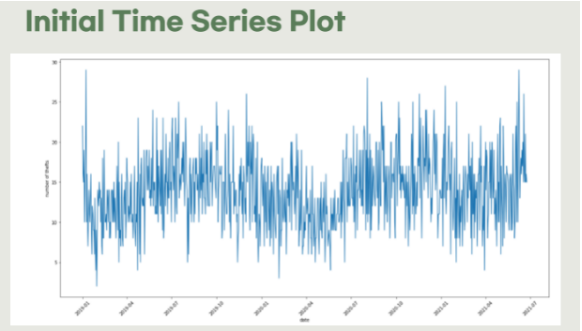


Figure 9. An initial plotting of the time series, generated with Python

We can see that the mean of the data is changing throughout the years and this is not just random white noise, so an ARIMA model will be appropriate here.

3.3 Model Training

To start model training, we will first determine the order of differencing, or the “d” term. To do so, we need to look at the autocorrelation plot (Figure 10)

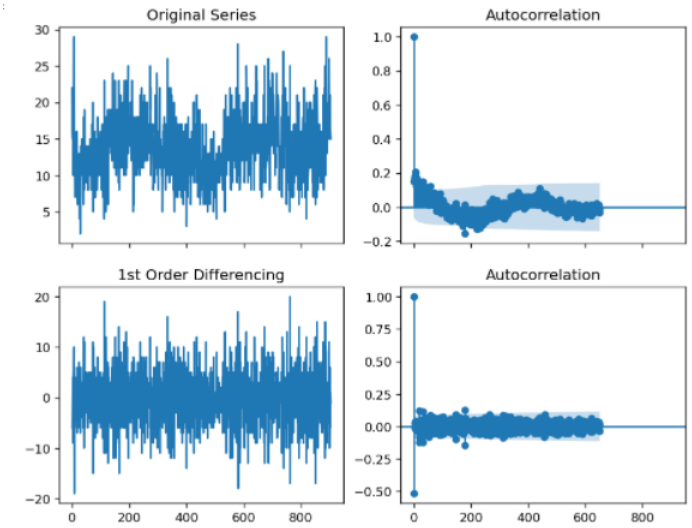


Figure 10. Autocorrelation plot, generated with Python.

The plot tells us how similar the time series is after x amount of lags. For example, lag = 2 is the correlation between the time series and the time series two lags (or days, in our case) before. Since our series is semi-stationary, meaning there are no highly significant lags, we used a differencing of 0. We could have used 1, which would create a very stationary time series; however, the best ARIMA model was trained with a differencing degree of 0.

Next, we need to determine "p," the order of the AR term. To do so, we need to examine the partial autocorrelation plot (see Figure 11).

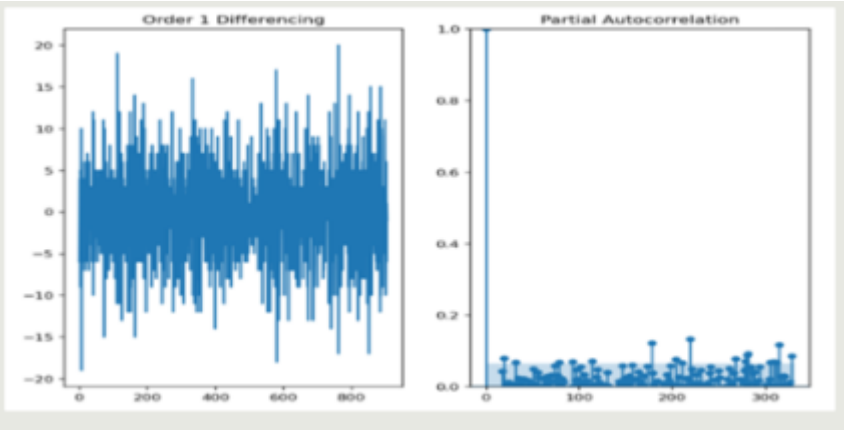


Figure 11. Partial autocorrelation plot, generated with Python.

The partial autocorrelation plot is the correlation between the first lag and the resulting time series model at that lag. Since there are no lags other than the first that are significantly correlated (outside of the blue shaded region in Figure 11), we will set p = 1 in our model.

Lastly, we need to determine "q," the order of the MA term. This will be the number of lagged forecast error terms (E) to incorporate into our model. To do so, refer again to Figure 10. The goal for determining this term is to remove any significant autocorrelation at every lag. Since a differencing of 1 shows the autocorrelation plot with no significant lags, we will set the q term equal to 1.

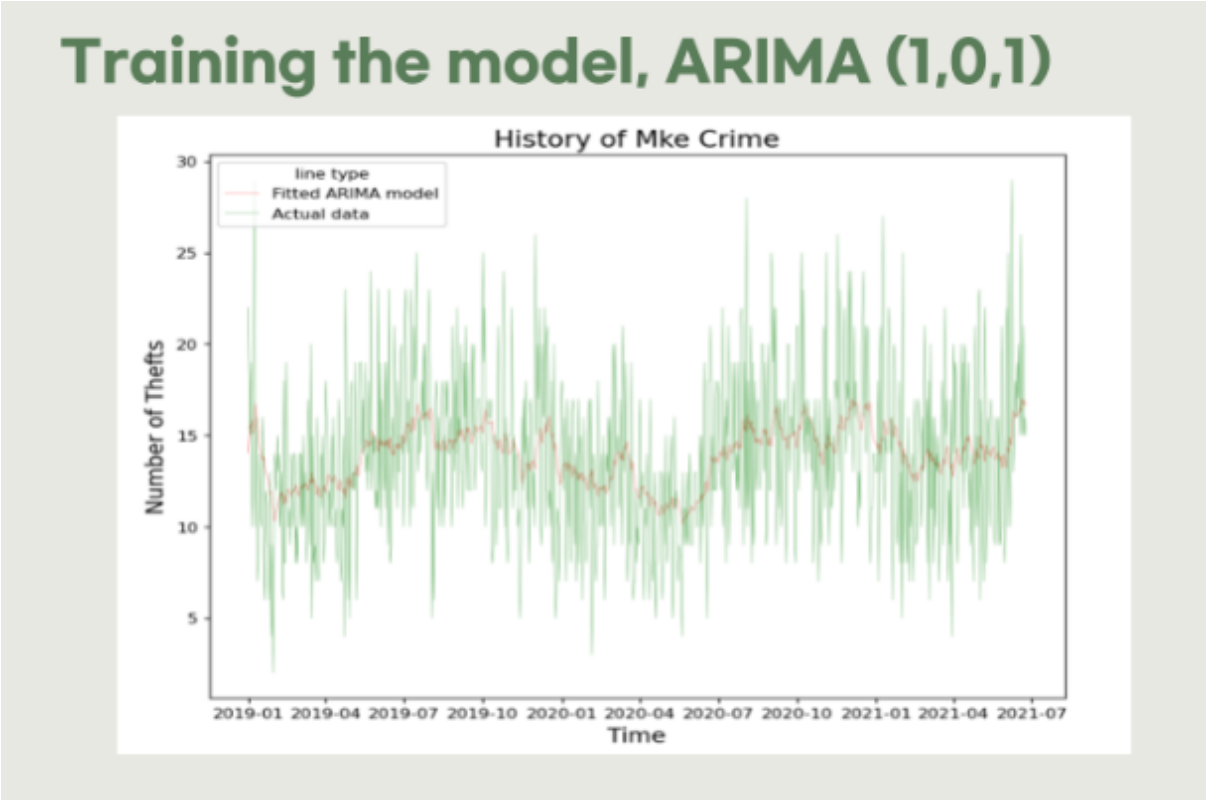


Figure 12. The ARIMA model fit

Figure 12 shows the resulting ARIMA model fit on top of the original time series graph. We can see that this is a good fit, as the model captured much of the variation in the time series.

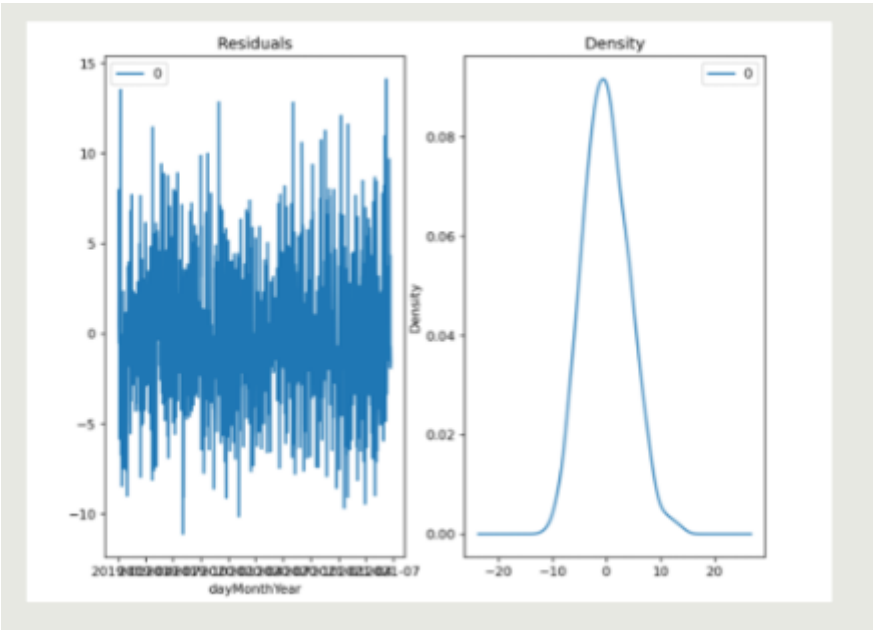


Figure 13. Residual plots

Figure 13 shows the distributions of the residuals of the ARIMA model fit. No assumptions of the model are violated when looking at these plots. The residuals follow a normal distribution, and there is no specific pattern or function applied to them.

3.4 Model Prediction

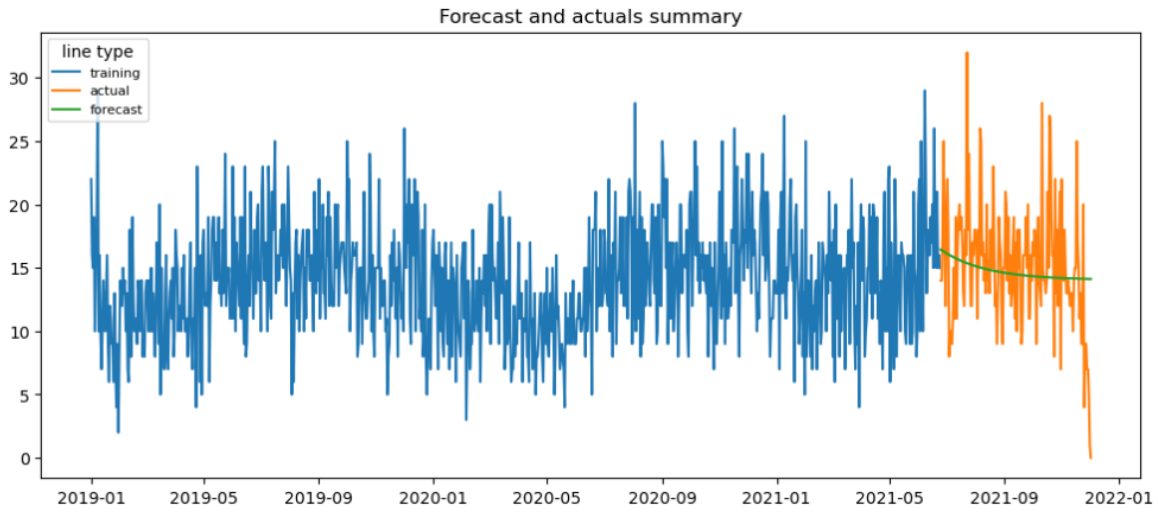


Figure 14. Final plot of our model: prediction

Figure 14 displays the actual history in orange and blue corresponding to train and test. The green line shows the predictions the ARIMA model made. We expected the number of thefts to decrease when the season changes from fall to winter, which is what our model is predicting. We predict that on January 1st of 2022, there will be 14.12 theft crimes committed across the entire county of Milwaukee.

4. Conclusion

In summary, our project goal was to conduct an analysis on the crime in Milwaukee by creating a geoplot to map data of criminal activity into the aldermanic city districts. It was determined that District 15 had the highest rate of crime, and District 11 had the lowest. Our second goal was creating an ARIMA prediction model to forecast if AssaultOffense and Theft could be predicted up to one day in advance. From our results, it showed that crime would progressively decline as seasonality had an effect on its frequency during the Winter season.

References

- Jason Brownlee. “ARIMA Model for Time Series Forecasting in Python.” *Machine Learning Mastery*, 8 Jan. 2017, machinelearningmastery.com/arima-for-time-series-forecasting-with-python/.
- “WIBR Crime Data (Current) - CSV - City of Milwaukee Open Data Portal.” *Data.milwaukee.gov*, data.milwaukee.gov/dataset/wibr/resource/87843297-a6fa-46d4-ba5d-cb342fb2d3bb?view_id=7d2f1fac-0c35-4d4f-a3d0-e1f365ab6945.
- “WIBR Crime Data (Historical) - CSV - City of Milwaukee Open Data Portal.” *Data.milwaukee.gov*, data.milwaukee.gov/dataset/wibrarchive/resource/395db729-a30a-4e53-ab66-faeb5e1899c8. Accessed 7 Dec. 2021.
- “Crime Maps & Statistics.” *City.milwaukee.gov*, city.milwaukee.gov/police/Information-Services/Crime-Maps-and-Statistics.
- Dr. Andrew Schiller, “Milwaukee, WI Crime Rates,” Neighborhoodscout.com, Jun. 10, 2019. https://www.neighborhoodscout.com/wi/milwaukee/crime.
- https://github.com/RyanR1019/An-Analysis-of-Crime-in-Milwaukee/upload/main
- https://github.com/18arautm/18arautm/blob/main/Data%20Science%20Project.ipynb