“FastTrack Forecast”: Predicting Driving Speed Violations with ARIMA

Shrey Shah

School of Information Sciences, University of Illinois at Urbana-Champaign

IS 597: Machine Learning Cloud

May 05, 2024

Introduction

In recent years, the global automotive landscape has witnessed a remarkable surge in both car production and vehicle ownership, driven by significant technological advancements. Projections indicated that by 2023, the global car population would surpass 1.47 billion. However, with this burgeoning vehicular presence comes the potential for a corresponding increase in traffic speed violations, accidents, and instances of road rage. In light of these concerns, fostering a safe environment for both pedestrians and drivers has become paramount.

The primary objective of this research project is to leverage predictive modeling techniques to anticipate instances of speeding infractions based on historical data. While my previous explorations in machine learning encompassed diverse domains such as classification, clustering, association rule mining, computer vision, and natural language processing, this endeavour marks my foray into the realm of time series analysis. Motivated by the pressing significance of addressing speed violation issues in contemporary society, I embarked on this project with the aim of exploring various algorithms and methodologies pertinent to time-series analysis. By delving into the speed violation data, I seek to uncover meaningful patterns that can inform decision-making processes pertaining to traffic management and urban planning initiatives.

There are some specific analytical questions that I would like to find answers to:

- The research aims to develop predictive models to forecast speed camera violations at specific intersections based on historical violation data.
- Spatial analysis techniques will be utilized to identify spatial patterns or clusters of speed camera violations across different intersections.
- Time series analysis will uncover temporal patterns in speed camera violation rates, including daily, weekly, and seasonal variations.

Literature Review

Amiruzzaman (2018) delves into the realm of data mining techniques for the prediction of traffic violations, shedding light on the contributing factors to accidents and the temporal patterns of such incidents. By employing algorithms like Naive Bayes, Support Vector Machines (SVMs), and decision trees, the study identifies specific times and days prone to violations, drawing insights from national traffic violation databases. The analysis underscores the importance of data-driven approaches in understanding and mitigating traffic offenses.

Shifting focus to driver behaviour and its impact on road conditions, Mozaffari et al. (2015) present a study focused on urban environments, particularly in San Francisco. Utilizing numerical tools and driving data, the authors introduce an evolutionary least learning machine (E-LLM) for forecasting vehicle speed sequences. Comparative analysis with established techniques such as auto-regressive (AR) models and neural networks highlights E-LLM's superior predictive capabilities. This advancement holds promise for automotive engineering, particularly in designing efficient powertrain controllers sensitive to real-time driving conditions.

Addressing the persistent issue of speeding violations, Shawky et al. (2017) investigate optimal locations for speed cameras, primarily on rural highways. Through meticulous data analysis and predictive modeling using negative binomial regression, the study identifies key variables such as traffic volume and posted speed limits influencing the frequency of speeding violations. Notably, the research underscores the significance of traffic-related variables and the distinction between weekdays and weekends in shaping enforcement strategies and policy interventions.

Salinas et al. (2020) introduce the DeepAR approach, a significant advancement in probabilistic forecasting utilizing auto-regressive recurrent network models. Emphasizing the value of probabilistic forecasting in optimizing business operations, DeepAR demonstrates substantial accuracy gains over traditional methods through empirical tests on real-world datasets. This innovative approach offers promise for sectors seeking to enhance their predictive capacities, particularly in dynamic and uncertain environments.

In furthering the discussion on traffic prediction and management, Wang et al. (2022) propose a method based on adaptive Kalman filtering within the Autoregressive Moving Average (ARMA) framework for vehicle speed prediction. Integrating multi-source traffic data fusion and interval speed prediction, the approach achieves commendable accuracy, thereby offering potential enhancements to traffic management strategies and vehicle safety protocols.

A modified auto-regressive integrated moving average (ARIMA) modeling procedure is presented in the paper by (Yunus et al., 2016), specifically designed for wind-speed time-series data, especially in the Baltic Sea region. To capture time correlation and probability distribution, the approach combines frequency decomposition, shifting, limiting, differencing, and power transformation. The study also emphasizes how models can be applied to other nearby locations, suggesting useful applications for wind-speed forecasting and modeling in maritime environments.

In order to identify critical traffic road segments in urban environments—those with prolonged spatial and elevated risks of traffic accidents—the research paper by (Košanin et al., 2023) presents a parameter-free clustering-based methodology. It suggests a domain-specific evaluation criterion that penalizes large cluster sizes and highlights the consistency of clustering results over time and the spatial collocation of accidents between different periods. Real-world data from three years' worth of traffic accidents in three major Serbian cities that resulted in injuries or fatalities has been used to illustrate the approach. Using this methodology, the study offers insights into the identification and evaluation of high-risk road segments, thereby supporting traffic safety protocols.

Data

A. Data Collection

The dataset used for this was acquired from the Chicago Data Portal, and it shows the daily total of infractions that have occurred in Children's Safety Zones for each camera. The dataset is updated every day and can be exported in CSV format or retrieved via the API. There are currently 378,726 rows in the dataset, which includes data from July 2014 to the present. The dataset can be found at the link https://data.cityofchicago.org/Transportation/Speed-Camera-Violations/hhkd-xvj4/data_preview. The dataset consists of nine attributes that include information on the location of speeding incidents, the camera that recorded them, the date and time of the incident, and the total number of violations. A thorough explanation of each attribute can be found in the table below.

Attribute Name	Description
Address	Address of the location of the speed enforcement camera(s). There may be more than one camera at each address.
Camera ID	A unique ID associated with the physical camera at each location. There may be more than one camera at a physical address.
Violation Date	The date of when the violations occurred. NOTE: The citation may be issued on a different date.
Violations	Number of violations for each camera on a particular day.
X Coordinate	The X Coordinate, measured in feet, of the location of the camera. Geocoded using Illinois State Plane East
Y Coordinate	The Y Coordinate, measured in feet, of the location of the camera. Geocoded using Illinois State Plane East
Latitude	The latitude of the physical location of the camera(s) based on the ADDRESS column. Geocoded using the WGS84.
Longitude	The longitude of the physical location of the camera(s) based on the ADDRESS column. Geocoded using the WGS84.

Location	The coordinates of the camera(s) based on the LATITUDE and LONGITUDE columns. Geocoded using the WGS84.

Methodology

Data preprocessing represents a critical preliminary step in machine learning projects, pivotal for ensuring data quality and consistency. Employing standard Python libraries like NumPy, Pandas, Matplotlib, and scikit-learn, a comprehensive data preprocessing approach was undertaken. This encompassed handling missing values, managing outliers, eliminating duplicates, and standardizing data formats to establish a clean and uniform dataset. Through exploratory data analysis leveraging Matplotlib, valuable insights into data distributions, trends, and potential anomalies were garnered. Furthermore, the utilization of scikit-learn's preprocessing modules facilitated essential tasks such as feature scaling, normalization, and categorical variable encoding.

A. Data Ingestion

The dataset was exported in a CSV format from the Chicago Data Portal. The data is available at the link https://data.cityofchicago.org/Transportation/Speed-Camera-Violations/hhkd-xvj4/data_preview. The data is then read using the pandas library functions and stored in the form of a dataframe.

B. Data Exploration

I performed the initial exploration of the data to get details related to the dimensions of the dataset in the form of numbers or rows and columns. I explored a few data instances of the dataset to see the values in the different columns and guess the data types. There are some numeric columns in the data, and I got the summary statistics of those columns, including the data about the mean, median, standard deviation, minimum, and maximum values. I extracted the details about the columns, their data types, the number of null values, and the different columns in the dataset.

C. Data Preprocessing

Upon examining the data, it was noted that certain rows contained details regarding the camera, violation date, and counts, yet lacked geographical coordinates. Despite this, it was deemed that these rows contained valuable information, and thus, they were not discarded. To streamline processing and maintain consistency, the decision was made to convert the date from a string format to a datetime format. Additionally, there were instances where violation counts were missing. In such cases, a value of 0 was assigned for the violation count when data for a given camera on a specific date was not available. Duplicate instances, if encountered, were removed, retaining only the first occurrence. Since it is a forecasting problem, there is no specific label or target column available in the dataset.

D. Data Visualization

To understand the data more, I developed visualizations about the different aspects of the dataset with respect to the speed violations. I started with a histogram to show the maximum and minimum speed violations. To understand the difference between the number of speed violations based on the year and month, I plotted boxplot and bar plots. I wanted to find the distribution of the violations based on the address column. So, I extracted the unique addresses in the dataset and created temporary data frames for each address. To inspect the trends and seasonality in the dataset, I used a rolling mean and rolling standard deviation for the violations column.

E. Dataset Splitting

The dataset splitting involves partitioning the data into training and testing subsets using a simple index-based approach. Here, the dataset is divided such that the first 80% of the observations constitute the training

data, while the remaining 20% are reserved to the testing data. This splitting strategy is generally used for time series data, where temporal ordering is necessary for modeling accuracy.

In time series modeling, traditional train-test splitting techniques, such as those used in typical supervised learning tasks, are not directly applicable. This is because time series data lacks the presence of a target column that can be arbitrarily split into training and testing sets. Instead, the data must be split in a sequential manner to preserve the temporal relationship between observations.

F. Methods and Algorithms

To achieve the goal of predicting speed camera violations, a structured approach was followed, involving several steps. Firstly, the dataset was preprocessed, and relevant features such as time of day and day of the week were extracted. For spatial analysis, latitude and longitude coordinates were utilized to explore spatial patterns of violations across different locations. The majority amount of time working on the project was spend in this phase, understanding the different algorithms for time-series modelling, which one would suit the needs of the problem, and working with all of the models and trying to find the ways to address the issues while developing the code for the model training.

To perform the forecasting of the speed camera violations, the initial idea was to make use of the DeepAR algorithm from AWS SageMaker. I researched about the algorithm and its working to understand the different parameters that are used to model. DeepAR can capture intricate temporal patterns and dependencies in the data and it seemed a good fit for the task. To train the DeepAR algorithm and produce a model for the forecasting approach, it needs an Estimator and this requires an AWS IAM role for the execution. The AWS Learner Lab has restrictions and does not allow us to create an AWS IAM role. Since, the role is a mandatory argument for the DeepAR Estimator and the lab has restrictions on the creation of AWS roles, the DeepAR algorithm cannot run in the AWS Learner Lab. To resolve the issue, I tried to make use of some other libraries available in the Python scikit-learn package for the DeepAR model. I tried working with the gluonts and the mxnet libraries that offer the functionality to work with the DeepAR estimator. I was able to modify the dataset to suit the needs of the gluonts requirements, but the issue with the gluonts library is that it needs a specific version with which the numpy and pandas library are not compatible. The gluonts library needs a version of 0.7.6 and the code for training the model encountered certain errors with respect to the date column. Whether the date column is of the timestamp data type or the string type, it produced the same error and even after trying both the methods, I was unable to resolve the issue. If I would have some more time to conduct the research, I would have tried to work a solution to read the complete documentation of the gluonts library, the algorithm inputs and outputs and then come up with the code to perform the modelling.

As a final solution, I decided to go with the conventional time-series models like the ARIMA and the SARIMA models. To answer the research questions, I have used the different components of time-series modelling including the forecasting using the ARIMA model and decomposition to analyse the trends. Also, for the different hotspots of the speed violations, I have used the clustering algorithms like the K - Means++.

To work on the question related to the forecast, the ARIMA model is trained on the training dataset using the violation count as the endogenous variable. An endogenous variable is a variable whose value is determined within the system being analyzed. In our case, the violation count is an endogenous variable as it is affected by the previous counts due to temporal variations, and it remains a central variable of interest that captures the interest within the system. There are some parameters related to the time-series modeling like the autoregressive terms, differencing terms, and the moving averages terms. I drew some graphs to find the optimal values for these terms. The autocorrelation and partial autocorrelation curves of the original, first and second differences can give us some clues as to the optimal values. Using these values, I have trained the ARIMA model and used that for forecasting on the test dataset.

For the question as to which locations can be the centres of speed camera violations, the clustering technique is used to find hotspots of such violations. I have worked with the K-Means++ algorithm to find the clusters. For the clustering process, I created a subset of the data which had only those locations with non-zero values for the latitude and longitude features. I filtered the rows with 0 as the values as they do not add any meaningful data. To add more context, I have used the address column as well for plotting the clusters. To find the optimal number of clusters before actually clustering, I have used the elbow method. I plotted the elbow graph and used the knee locator to find the cluster count. Using this optimal count, I applied the K-Means++ clustering on the filtered dataset and created a column in the dataset to allocate a

data point to the cluster. I plotted the clusters on a map of Chicago city from the geopandas and geodatasets libraries.

Examining the trend and seasonality in the time-series data, I have used the decomposition modeling techniques within the time-series analysis. The decomposing technique decomposes the time series data into its constituent components: trend, seasonal, and residual. The decomposition is done using an additive model and we have used a periodicity of 365 days suggesting yearly seasonality. It displays the original time series data; the trend component that represents the long-term behaviour or direction of the data; the seasonality which depicts the recurring patterns or fluctuations that occur at fixed intervals; and the residual component. The residual component highlight the errors after removing the trend and seasonal components which shows the random noise or fluctuations.

In evaluating the performance of the models, I considered metrics such as mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE) to calculate the accuracy of predictions.

Results

The process and the steps taken to answer the research questions followed the simple and structured approach. It started with the data collection from the online website, followed by the data ingestion, exploratory data analysis and then data preprocessing to clean the data. After this the data was split into train and test dataset and the modelling techniques were applied. Based on the above process followed, the results are described as below:

A. Research Question 1: Can we predict the likelihood of speed camera violations at specific intersections based on historical violation data?

The ARIMA model is trained on the training dataset with the violation count as the endogenous variable in order to address the forecast-related question. Moving averages, differencing, and autoregressive terms are some of the parameters associated with time-series modeling. To determine the ideal values for these terms, I created a few graphs. The original, first, and second differences' autocorrelation and partial autocorrelation curves can provide the information as to how we can get the good enough values for this model.

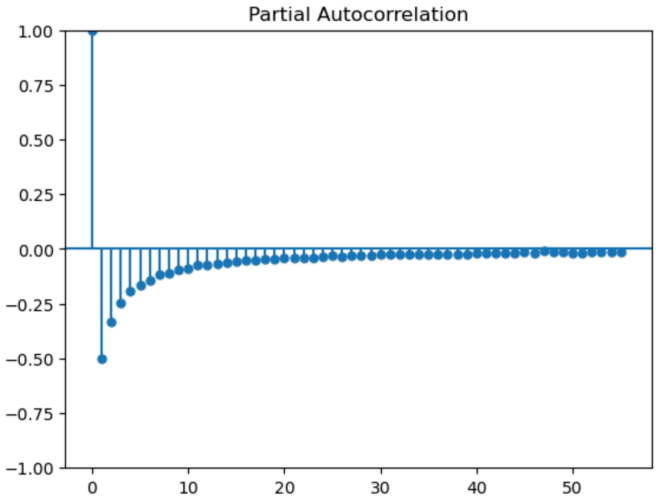

Fig 1. Partial Autocorrelation plot for the first difference of Violations Count feature

As seen in Fig. 1, we see that the first lag is way significantly out of the limit, so we can select the order of the autoregressive parameter 'p' as 1 less from the number of lags outside the limit. From Fig. 2, we see that the first difference (2nd plot) has the second lag negative as compared to the first lag. This represents that in the first difference, the series has become over the difference, and hence we can use the first

difference series for the ‘d’ parameter, which means that we can have ‘d’ as 1. Based on Fig. 3, we understand that the first 2 lags of the series data are very far from the significance limit and the other lags. Hence, we can use the value of 2 for the moving averages parameter 'q'.

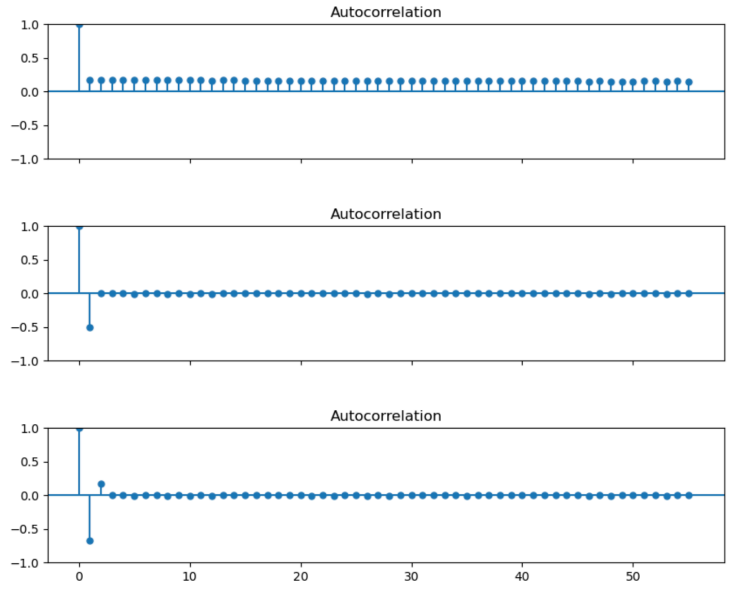

Fig 2. Original, First Difference, and Second Difference of Violations Count feature

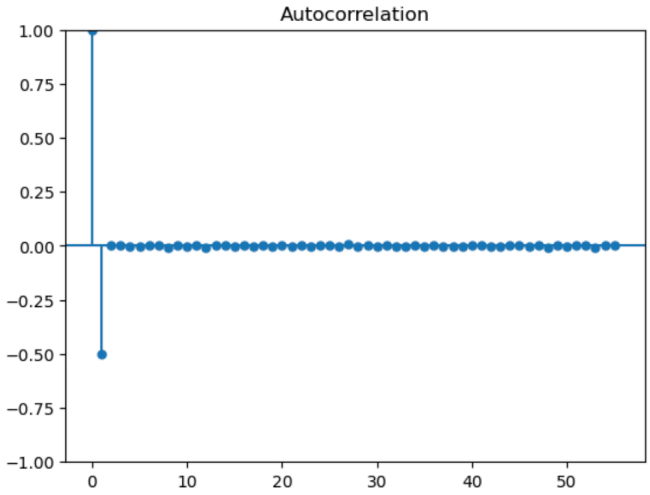

Fig 3. Autocorrelation plot for the first difference of Violations Count feature

Using the values of 1, 0, 2 for the ‘p’, ‘q’ and ‘d’ parameters, we train the ARIMA model on the training data and get the summary of the results as shown in Table 1.

SARIMAX Results

Dep. Variable:	VIOLATIONS	No. Observations:	302980
Model:	ARIMA(0, 1, 2)	Log Likelihood	-1603730.668
Date:	Sun, 05 May 2024	AIC	3207467.335
Time:	21:51:58	BIC	3207499.200
Sample:	0	HQIC	3207476.548

			- 302980			
Covariance Type:			opg			

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-0.9679	0.001	-763.124	0.000	-0.970	-0.965
ma.L2	-0.0128	0.001	-10.056	0.000	-0.015	-0.010
sigma2	2317.0929	1.798	1288.450	0.000	2313.568	2320.618

Ljung-Box (L1) (Q):	0.02	Jarque-Bera (JB):	6033075.67
Prob(Q):	0.88	Prob(JB):	0.00
Heteroskedasticity (H):	0.60	Skew:	3.42
Prob(H) (two-sided):	0.00	Kurtosis:	23.76

Table 1. Summary details of the ARIMA model on training dataset

- **Dependent Variable and Observations**: The model analyzes the VIOLATIONS variable with 302980 observations.
- **Model Specification**: Utilizes SARIMAX with ARIMA(1, 0, 2), representing autoregressive, differencing, and moving average components.
- **Coefficients**: Coefficients include a constant term of 37.3974, indicating the average violations when other predictors are zero, and significant autoregressive and moving average terms.
- **Standard Error and Significance**: Coefficients exhibit statistical significance ($p < 0.05$) with associated z-statistics.
- **Residuals and Heteroskedasticity**: Residual variance (sigma2) is 2317.5389, and tests suggest no significant autocorrelation or deviation from normality in the residuals.
- **Heteroskedasticity**: The probability of heteroskedasticity (Prob(H)) is low, indicating the presence of varying residual variance with the series level.

The ARIMA(1, 0, 2) model sis a time series with significant autoregressive and moving average terms, enhancing predictive capability. The model captures data patterns well, with stable residual variance and minimal autocorrelation, indicating adequate performance. However, high skewness and kurtosis suggest deviations from normality, indicating potential areas for improvement. Overall, the model adequately captures temporal dynamics.

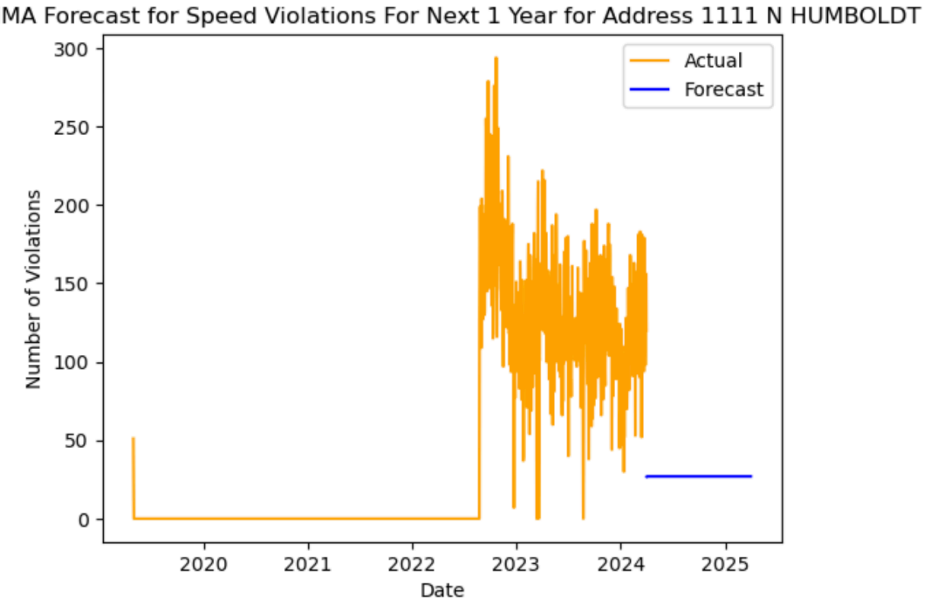

Fig 4. ARIMA model forecast for the speed violations for next one year with optimal parameters

We want to predict the speed violations for the next one year. Fig 4. shows the forecasting results using the above model for this. This used the optimal values of the parameters. We wanted to predict the speed violations at a specific address. For this, I have used the '1111 N HUMBOLDT' address. From the graph, we can see that the orange lines represents the actual forecasts for the ARIMA model and the average number of violations come from 2014 to 2024 (till date) comes out to 75 to 100 (considering the 0s and 300s). Based on this, the model predicts that for the next year the number of speed violations for this address comes out close to 45 to 50, which is a good estimation.

A second model is trained on the training dataset with some random parameters of (5, 1, 0). The model when used for forecasting gives the results in Fig 5. It shows the forecasting results using the model for this. To highlight the difference between the 2 models, I have used the '1111 N HUMBOLDT' address. From the graph, we can see that the orange lines represents the actual forecasts for the ARIMA model and the average number of violations come from 2014 to 2024 (till date) comes out to 75 to 100 (considering the 0s and 300s). Based on this, the model predicts that for the next year the number of speed violations for this address comes out close to 0, which is not a very good estimation.

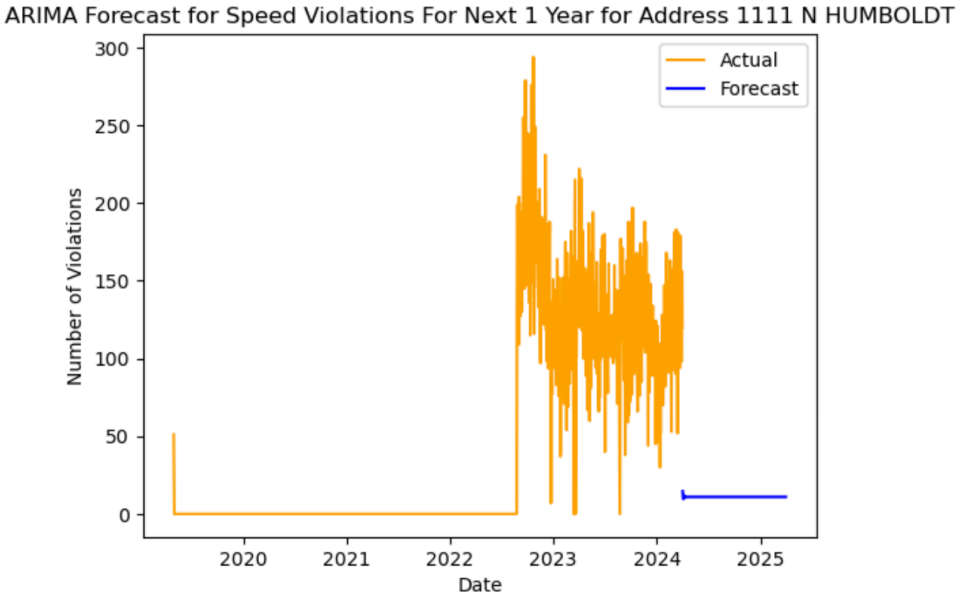

Fig 5. ARIMA model forecast for the speed violations for next one year with random parameters

To measure and identify which of the two models is better at forecasting the predictions, we have used the metrics for model evaluation. For time-series data the normal metrics used for evaluation include the Mean Square Error (MSE), Root Mean Square Error (RMSE), and the Mean Absolute Error (MAE). We will discuss the results of both the models.

Model with optimal parameters:

```
arima_model_evaluation(test_data, endogenous_feature, arima_model_good)
Mean Squared Error (MSE): 4086.1317633451717

Root Mean Squared Error (RMSE): 63.922857909711546

Mean Absolute Error (MAE): 36.33472865466731
```

Model with random parameters:

```
arima_model_evaluation(test_data, endogenous_feature, arima_model_bad)
Mean Squared Error (MSE): 5115.83410294748

Root Mean Squared Error (RMSE): 71.52505926559921

Mean Absolute Error (MAE): 42.23747740608929
```

The model with optimal parameters:

- **Mean Squared Error (MSE):** The MSE value of 4086.13 indicates that, on average, the squared differences between the predicted and actual speed camera violation values amount to approximately 4086.13. This suggests that there is some variability or dispersion in the errors made by the model in predicting violations.
- **Root Mean Squared Error (RMSE):** The RMSE value of 63.92 indicates the average magnitude of errors in the predicted speed camera violation values. It suggests that, on average, the model's predictions are off by approximately 63.92 ~ 64 violations.
- **Mean Absolute Error (MAE):** The MAE value of 36.33 represents the average magnitude of errors in the predicted speed camera violation values. It depicts that, on average, the model's predictions deviate from the actual values by approximately 36.33 ~ 33 violations.

Overall, these metrics suggest that while the ARIMA model may have some level of error in predicting speed camera violations, the errors are within a reasonable range.

The model with random parameters:

- **Mean Squared Error (MSE):** The MSE value of 5115.83 indicates that, on average, the squared differences between the predicted and actual speed camera violation values amount to approximately 5115.83. This suggests that there is more errors made by the model in predicting violations.
- **Root Mean Squared Error (RMSE):** The RMSE value of 71.53 indicates the average magnitude of errors in the predicted speed camera violation values. It suggests that, on average, the model's predictions are off by approximately 71.53 ~ 72 violations.
- **Mean Absolute Error (MAE):** The MAE value of 42.24 represents the average magnitude of errors in the predicted speed camera violation values. It depicts that, on average, the model's predictions deviate from the actual values by approximately 42.24 ~ 42 violations.

The model with optimal parameters exhibits lower values across all three evaluation metrics compared to the second model. Specifically, the first model has an MSE of 4086.13, RMSE of 63.92, and MAE of 36.33, while the second model yields higher values with an MSE of 5115.83, RMSE of 71.53, and MAE of 42.24. These results indicate that the first model generally provides more accurate predictions, as evidenced by its lower error metrics. Therefore, based on the evaluation criteria, the first model outperforms the second model in forecasting performance.

B. Are there spatial patterns or clusters of speed camera violations across different addresses?

To locate the clusters of speed camera violations based on the address, we have used the latitude, longitude and the address features from the dataset. The training dataset is filtered to obtain the rows that have non-zero values for the latitude and longitude features as only such data points can be used for plotting the points on the map of Chicago.

Before we start with the actual clustering process, we need to identify a good value for the number of clusters. I have used the 'Elbow Plot' to determine the optimal number of clusters. To start with the random points for cluster initialization may not be a good approach as in the K - Means algorithm. This issue is resolved in the K - Means++ algorithm as it specifies a procedure to initialize the cluster centers before moving forward with the standard k-means clustering algorithm. I have defined a function to plot the 'Within Cluster Sum-of-Squares' (WCSS) for the cluster count from 1 to 10 and this forms the elbow plot. The knee locator library helps us find the optimal number of clusters as shown in Fig 6.

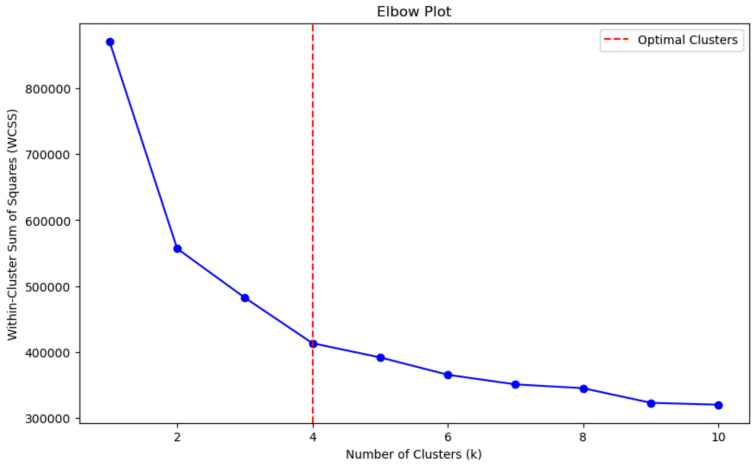

Fig 6. Elbow plot to determine the optimal number of clusters for speed violations

Based on this, we have 4 optimal clusters for our speed violations use case. To perform the actual clustering, I have scaled the filtered data and used the K - Means++ algorithm. The cluster points are then plotted on the map of Chicago as seen in Fig 7.

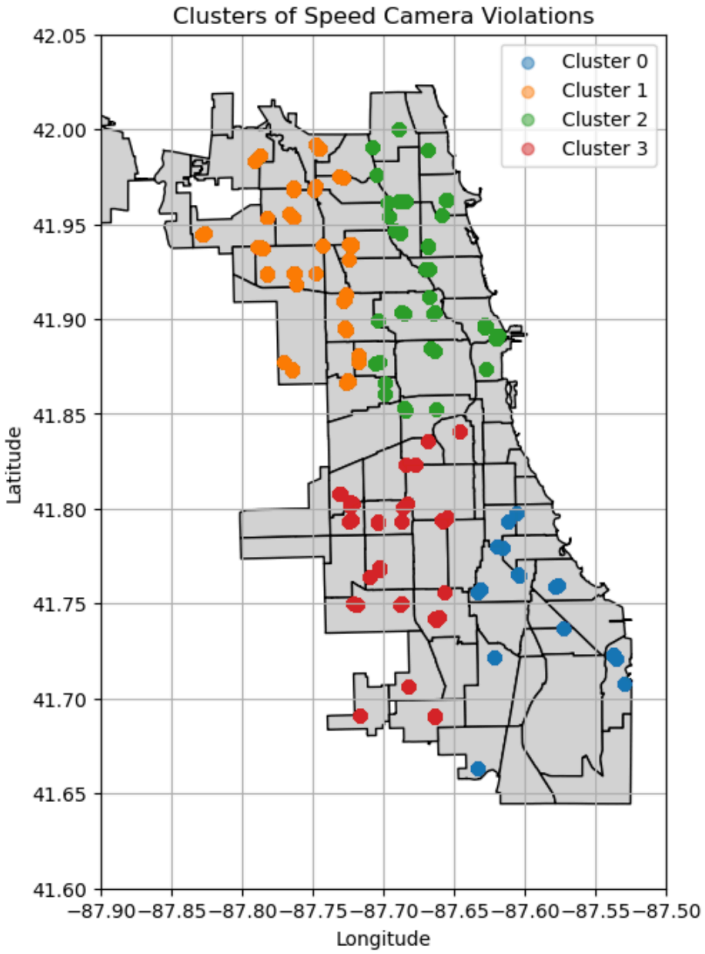

Fig 7. The hotspots of speed violations for Chicago city

C. How do speed camera violation rates vary over time, such as daily, weekly, or seasonal patterns?

To analyze any temporal variations in the time-series data, I have used the decomposition modeling technique. This gives us 4 major components for any time-series data: original, trend, seasonality, and residuals.

The 4 components are plotted as as shown in the Fig 8. The X-axis represents the time using the total length of the dataset as a representative feature and the Y-axis represents the maximum number of violations.

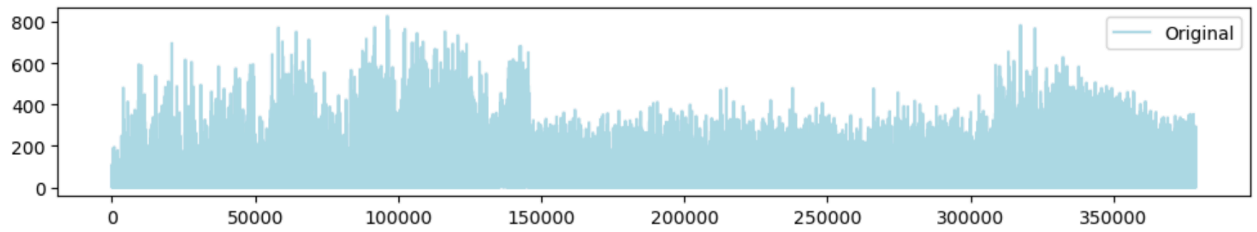

Fig 8(a). Original component

Fig 8(a) shows the 'Original' component and it shows us the actual time-series data. Here the variable under consideration is the number of speed violations.

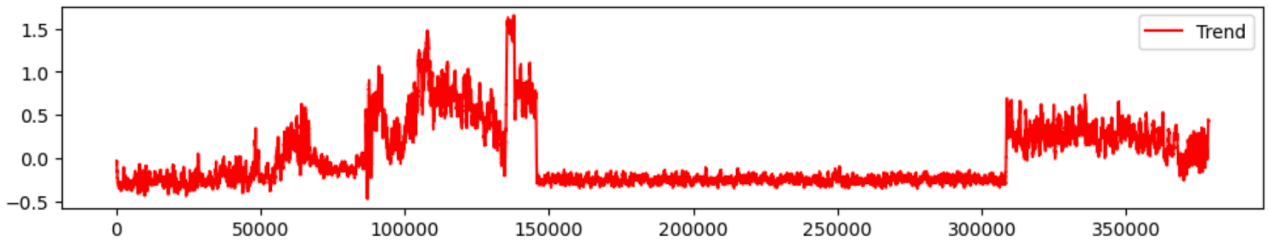

Fig 8(b). Trend component

Fig 8(b) shows the 'Trend' component. This represents the long-term behaviour or direction of the data. As we can see from the graphs (a) and (b), there is a very close resemblance. In the initial stages, it shows a positive trend and then it drops and flattens during the mid time period. This may be during the years 2020 and 2021, when there was Covid-19 pandemic and people were not allowed to roam on the streets.

This shows a slight positive trend in the time-series data and a chance of increase in the number of speed violations count as time progresses in the future.

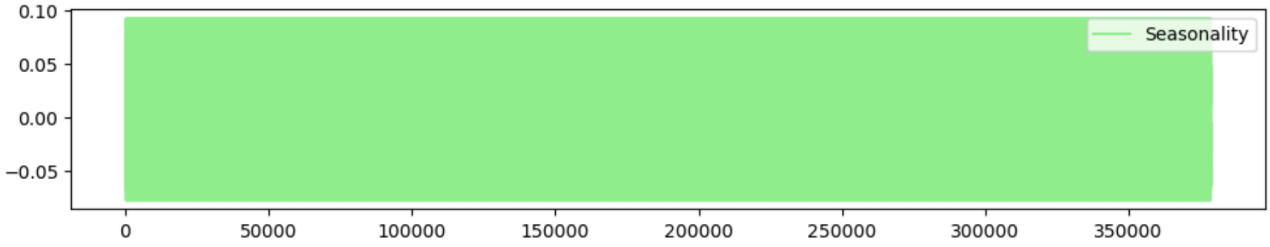

Fig 8(c). Seasonality component

Fig 8(c) shows the 'Seasonality' component. This displays the seasonal component, which captures recurring patterns or fluctuations that occur at fixed intervals (e.g., daily, weekly, or yearly). As we can see from the graph (c), it is complete green block without any variations.

This shows that there are no lows and highs in the dataset and there is no pattern that repeats itself at periodic intervals. This means that there is no seasonality and the speed camera violations is a random event that can happen without any influence of any season or time of the year

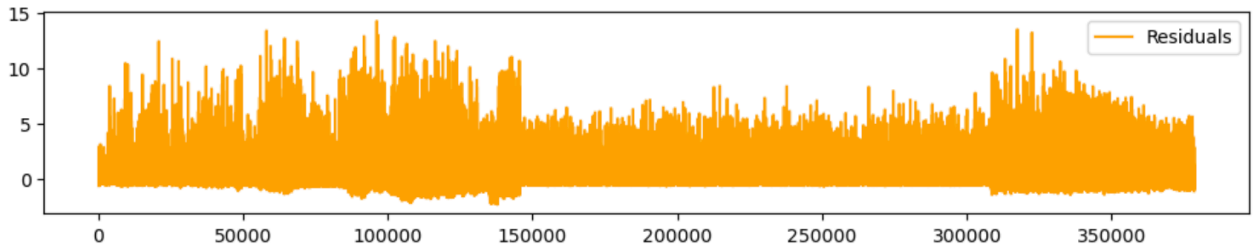

Fig 8(d). Residuals component

Fig 8(d) shows the 'Residuals' component. It shows the errors after removing the trend and seasonal components. These residuals ideally should be close to 0, indicating that the trend and seasonal patterns have been successfully extracted.

In our graph, the residuals range from 0 to 15 which means that there are some errors observed in the training and prediction of the model. It also indicates that there is a room for improvement in the existing model.

Discussion

Using the best parameter configurations found through thorough analysis, the ARIMA modeling approach provides a framework for predicting speeding violations in cities. The model exhibits decent predictive performance when configured as ARIMA(1, 0, 2), capturing significant autoregressive and moving average effects while preserving residual variance stability. The results of the forecasting provide insights into a complex understanding of temporal dynamics, which is helpful in developing strategies for traffic control and enforcement. Additionally, the comparison between the ARIMA(1, 0, 2) and randomly initialized (5, 1, 0) models highlights the significance of parameter selection through comparative analysis necessary in time-series forecasting use cases. All things considered, the study emphasizes how effective ARIMA modeling is at resolving traffic-related issues.

Spatial analysis of speed camera violations across different addresses reveals distinct clusters within the city of Chicago. By leveraging latitude, longitude, and address features from the dataset, we identified spatial patterns using the K - Means++ clustering algorithm. The determination of the optimal number of clusters was facilitated through the Elbow Plot method, which indicated four as the ideal cluster count. Subsequently, the K - Means++ algorithm was applied to the scaled data to perform clustering, resulting in the visualization of cluster points on the map of Chicago. This approach enables an understanding of the spatial distribution of speed camera violations, providing valuable insights for enforcement in violation-prone areas.

The speed camera violation time-series data can be used to gain important insights into temporal variations through the use of the decomposition modeling technique. The original, trend, seasonality, and residuals—the four main components—offer a thorough comprehension of the dataset's dynamics. The trend component shows a long-term behaviour, reflecting fluctuations that may have been influenced by outside factors like the Covid-19 pandemic, whereas the original component depicts the actual violations count over time. It is interesting to note that the lack of patterns in the seasonality component implies that speed camera infractions happen at random and are not influenced by seasonal fluctuations. On the other hand, the residuals component indicates possible areas for model refinement by highlighting the existence of errors in model training and prediction. In general, this analysis clarifies the underlying trends and uncertainties present in the data on speed camera violations.

Apart from this, if I would have some extra time, I would have invested the time to work with the DeepAR model and examining the different ways to work with the gluonts library and making it compatible with different versions of numpy, pandas and other libraries. Additionally, I would have applied more additional techniques to perform a better cleaning of the data to find anomalies or outliers that could have made the model generalise better to the data and reduce the different errors like MSE, RMSE, and MAE.

Conclusion

To sum up, the project shows how useful it is to use spatial analysis, decomposition, and ARIMA modeling to comprehend and forecast speed camera infractions in urban settings. By means of meticulous parameter adjustment and comparative evaluation, we have determined the ideal ARIMA model that effectively captures noteworthy temporal fluctuations, thereby offering critical details for predicting subsequent speed violations. Targeted enforcement tactics can be made easier by the spatial analysis that also identifies unique clusters of violations among various addresses. By emphasizing long-term trends, seasonal patterns, and residual errors in the data, the decomposition modeling approach further improved our understanding of temporal variations. Overall, the research shows how time-series modeling plays a crucial role in detecting speed camera violations for better traffic control and enforcement strategies in urban areas.

Through working on the project, I have gained a deeper understanding of time-series forecasting and spatial analysis methodologies. DeepAR, with its recurrent neural network architecture, provides a way for capturing temporal dependencies in data, which is particularly useful for long-term forecasting. Unfortunately, due to the limitations of the AWS Learner Lab, I could not use the DeepAR model for forecasting. ARIMA, on the other hand, offers a classical approach, leveraging autoregressive, differencing, and moving average components to model time-series data. Exploring clustering techniques has revealed spatial patterns and clusters within datasets, enabling the identification of critical regions or hotspots for targeted interventions. Understanding seasonality in time-series data has shed light on recurring patterns and fluctuations, allowing for better interpretation and forecasting. Lastly, error metrics such as MSE, RMSE, and MAE have provided quantitative measures of model performance, facilitating model evaluation and refinement. Overall, this journey has equipped me with tools and techniques for analyzing and forecasting time-series and spatial data, contributing to a more comprehensive understanding of data analytics in real-world applications.

GitHub Repository

Link to the GitHub Repository: https://github.com/shreyss99/IS597-Machine-Learning-Cloud

The project is organised as follows:

```
|
|── Data/
|   ├── Speed_Camera_Violations.csv
|
|── Code/
|   └── Import Modules
|   |      ├── data_ingestion.py
|   |      ├── data_exploration.py
|   |      ├── data_preprocessing.py
|   |      ├── data_visualization.py
|   |      ├── data_splitting.py
|   |      ├── arima_optimal_parameters.py
|   |      ├── arima_model_training.py
|   |      ├── arima_model_forecasting.py
|   |      ├── arima_model_evaluation.py
|   |      ├── spatial_variations.py
|   |      └── temporal_variations.py
|   |
|   └── Jupyter Notebooks
|          ├── Instructions.ipynb
|          ├── DeepAR_Code.ipynb
|          ├── MLC_Project_Final_Code.ipynb
```

The Instructions.ipynb notebook contains the instructions to run the notebooks. Make sure to run the notebooks, you need to have all the modules and the data in the same directory as the notebook itself, otherwise it will not execute.

References

Chicago, C. of. (2024, April 14). Speed camera violations: City of Chicago: Data Portal. Chicago Data Portal. https://data.cityofchicago.org/Transportation/Speed-Camera-Violations/hhkd-xvj4/data_preview

Amiruzzaman, M. (2018). Prediction of traffic-violation using data mining techniques. *Proceedings of the Future Technologies Conference (FTC) 2018*, 283-297. https://doi.org/10.1007/978-3-030-02686-8_23

Košanin, I., Gnjatović, M., Maček, N., & Joksimović, D. (2023). A clustering-based approach to detecting critical traffic road segments in urban areas. *Axioms*, *12*(6), 509. https://doi.org/10.3390/axioms12060509

Mozaffari, L., Mozaffari, A., & Azad, N. L. (2015). Vehicle speed prediction via a sliding-window time series analysis and an evolutionary least learning machine: A case study on san francisco urban roads. *Engineering Science and Technology, an International Journal*, *18*(2), 150-162. https://doi.org/10.1016/j.jestch.2014.11.002

Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, *36*(3), 1181â€“1191. https://doi.org/10.1016/j.ijforecast.2019.07.001

Shawky, M., Sahnoon, I., & Al-Zaidy, A. (2017). Predicting speed-related traffic violations on Rural Highways. *Proceedings of the 2nd World Congress on Civil, Structural, and Environmental Engineering*. https://doi.org/10.11159/icte17.117

Wang, Y., Yu, C., Hou, J., Chu, S., Zhang, Y., & Zhu, Y. (2022). Arima model and few-shot learning for vehicle speed time series analysis and prediction. *Computational Intelligence and Neuroscience*, *2022*, 1â€“9. https://doi.org/10.1155/2022/2526821

Yunus, K., Thiringer, T., & Chen, P. (2016a). Arima-based frequency-decomposed modeling of wind speed time series. *IEEE Transactions on Power Systems*, *31*(4), 2546–2556. https://doi.org/10.1109/tpwrs.2015.2468586