Katie Burgess
Operationalizing AI

Real-Time Traffic Prediction with Kafka

The Urban Mobility Challenge explains how urbanization has increased demands on city infrastructure, especially in transportation. This project helps simulate real-time data analysis, essential for transportation systems. Kafka will be used to send and receive information, after which exploratory data analysis will be performed and a model will be created.

Phase 1: Kafka

Kafka is used to simulate real-time data. A producer sends the data over a topic while a consumer collects those data packets sent. This data is displayed within the terminal, but in a system where all of the project steps are done at once, the data would be taken straight from the topic and fed into the data analysis scripts.

Phase 2: Exploratory Data Analysis (EDA)

After the data is collected from Kafka, the data is put into Python scripts for exploratory data analysis through graphs and other visualizations. Graphs such as time series plots, ACF, and PACF are used to get a better understanding of the data.

The traffic over time graphs showed that the level of traffic significantly changes between different hours. I will not be able to use any kind of stationary model; the one I choose will have to work well with time series data. Additionally, the changes in traffic can increase and decrease very quickly, meaning that it may be best to compare the same time of day on different days rather than on the same day predicting the same level of traffic forward. This data suggests that I should use a model such as ARIMA to predict traffic levels.

My ACF and PACF graphs display that at some times the period before can predict the one after it, but at some points, it is completely unpredictable. Additionally, negative values on these graphs show that there is definitive seasonality in this dataset. This data only solidifies what I learned in my other graphs, that I should be using an ARIMA model that will best capture this kind of data.

Phase 3: Traffic Flow Prediction Model

After I finished EDA, I created my model. Due to the time-series nature of the data, alongside the fact that the data has hour and minute seasonality, I went with an ARIMA model. This model allowed me to get a good prediction on traffic without overturning my model. I originally tried a linear regression model but ran into issues with either having an overtuned model or a model that did not capture the data effectively enough.

I created features of the first 3 locations, giving me a solid prediction of all the data over a period of time. This data encapsulates the different periods of these locations as well, ensuring that the hourly seasonality is not ignored. I tried working with several other features, such as weekend days, peak traffic hours, and work days, but these features either contributed to model overturning or were otherwise unhelpful compared to the features I ended up using for this project.

I calculated four metrics for my model: mean absolute error (MAE), root mean square error (RMSE), mean squared error (MSE), and r-squared. Here are my computed values and their interpretations:

MAE: 0.0199, this is a good MAE as it is close to 0.

RMSE: 0.0285, as this is small compared to the traffic values I am looking at, this is a good RMSE.

MSE: 0.0008, this is a good MSE value, as the smaller the MSE the better.

R-squared: 0.4046, this value says that around 40% of the variance of traffic is explained by my model.

These values are a significant improvement from the first ML model I completed before the exploratory data analysis while avoiding the overfitting that I ran into in some of my first attempts at improving my features.

Results:

I was able to successfully create an ML prediction model for traffic utilizing Kafka and Python scripting for EDA and ML model creation. My model has good model evaluation metrics while also avoiding overfitting. This project taught me a lot about Kafka, as well as the model creation and evaluation process, as this is something that I have never done before this class.

Conclusion:

In conclusion, this project successfully demonstrates the use of Kafka and time-series modeling to predict real-time traffic patterns. The ARIMA model I chose after performing EDA proved to be a good choice, being accurate accuracy while also avoiding overfitting.