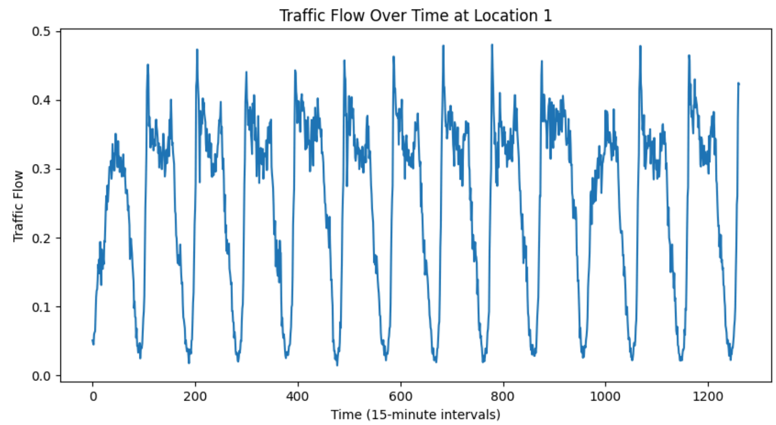
Real-Time Traffic Prediction with Kafka

Phase 1

1. **Kafka Producer and Consumer Implementation:**
    a. Kafka producer Script - https://github.com/pushkar-saraf/traffic_flow_prediction/blob/master/src/producer.py
    b. Kafka Consumer Script - https://github.com/pushkar-saraf/traffic_flow_prediction/blob/master/src/consumer.py
2. **Stream data from the producer to Kafka and consume it using the consumer.**
    a. In Producer - function *stream_data()*
        i. Reads data from pickle file
        ii. Processes and encodes to json
        iii. Streams it with 1 second delay
    b. In Consumer - function *subscribe()*
        i. Subscribes to the topic, and decodes back to json
        ii. Sends it for prediction
3. **Ensure real-time simulation by introducing appropriate delays (e.g., 1-second intervals).**
    a. Function *stream_data()* has sleep(1)

Phase 2

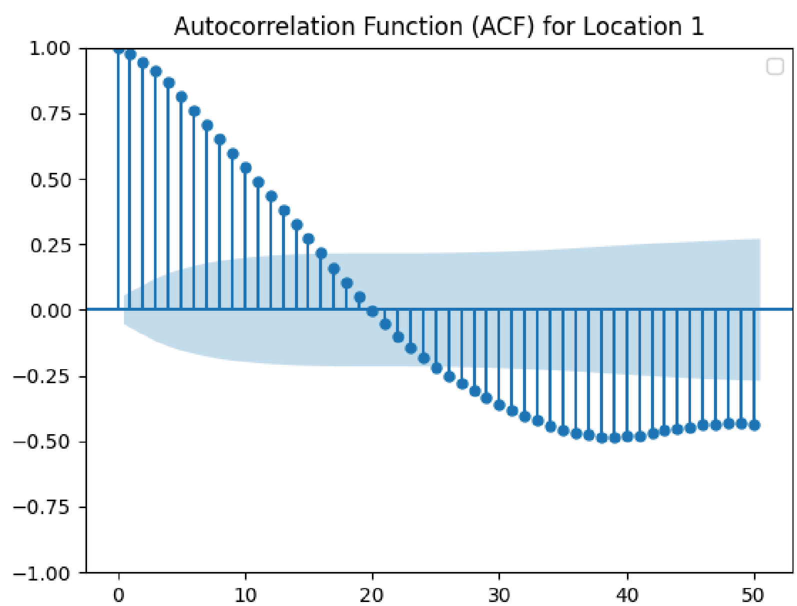
1. **Visualizations**
    a. **Time-Series Plots (5 Points): Create clear and well-labeled time-series plots showing traffic flow over time, including at least:**
        i. **Traffic Flow vs Time**



b. **Autocorrelation and Partial Autocorrelation Plots (5 Points): Generate and interpret ACF and PACF plots to identify patterns, seasonality, and trends in the traffic flow data.**

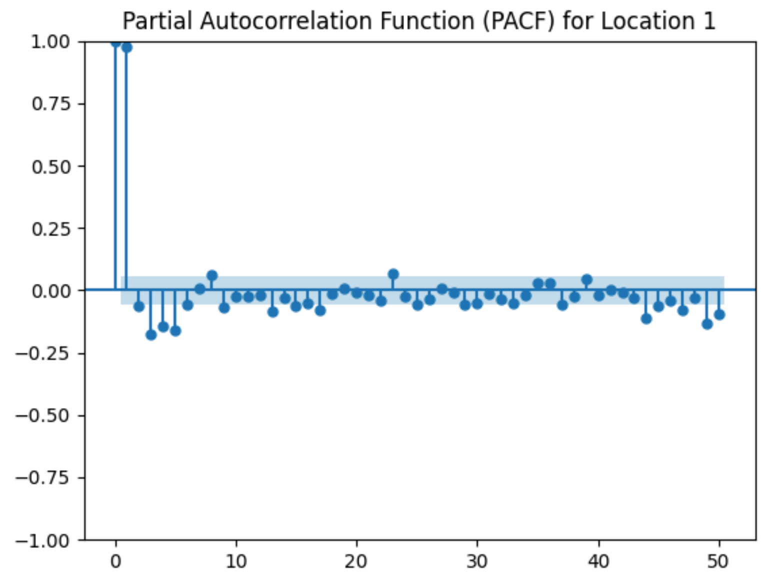
   i. **Autocorrelation**

      - Plot below clearly indicates seasonality. Notice how for lags of 1 to 10 there is a high correlation, and how it decreases as we go further back.



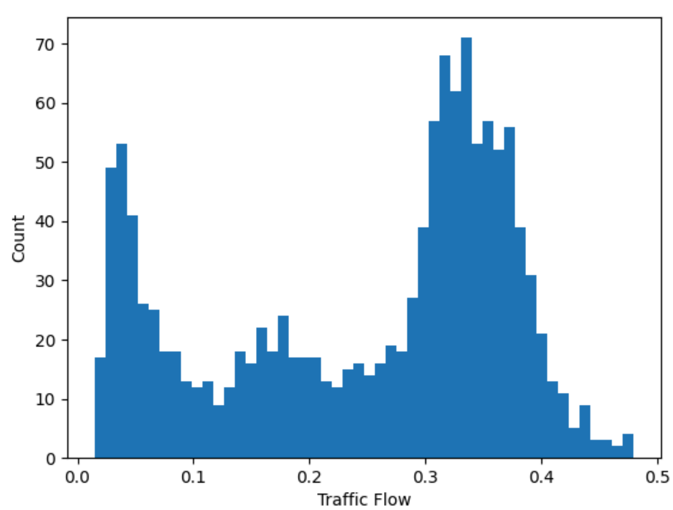
   ii. **Partial Autocorrelation**

- Confirms our understanding of the autoregressive model. Current value is highly dependent of previous value. Traffic is not truly random. It can be predicted!



c. Additional Visualizations (5 Points): Include any additional visualizations that help in understanding the data, such as histograms, scatter plots, or heatmaps.

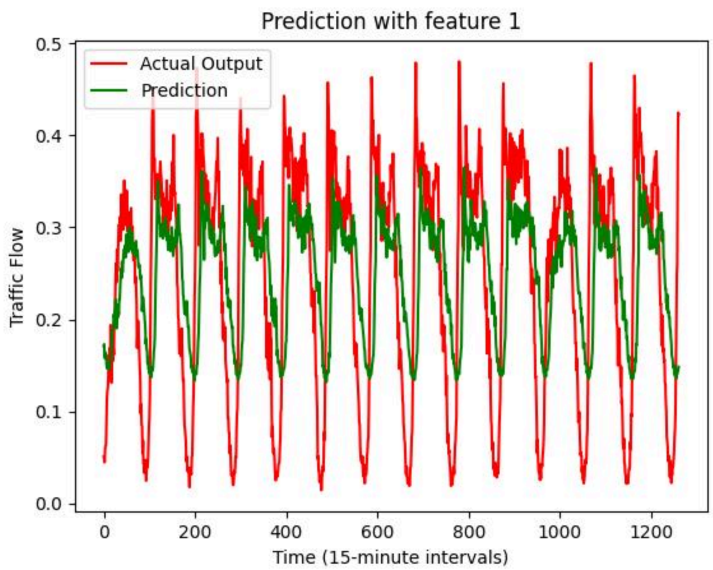
i. Histogram



Histogram shows that there is a high probability of very high and very low traffic.

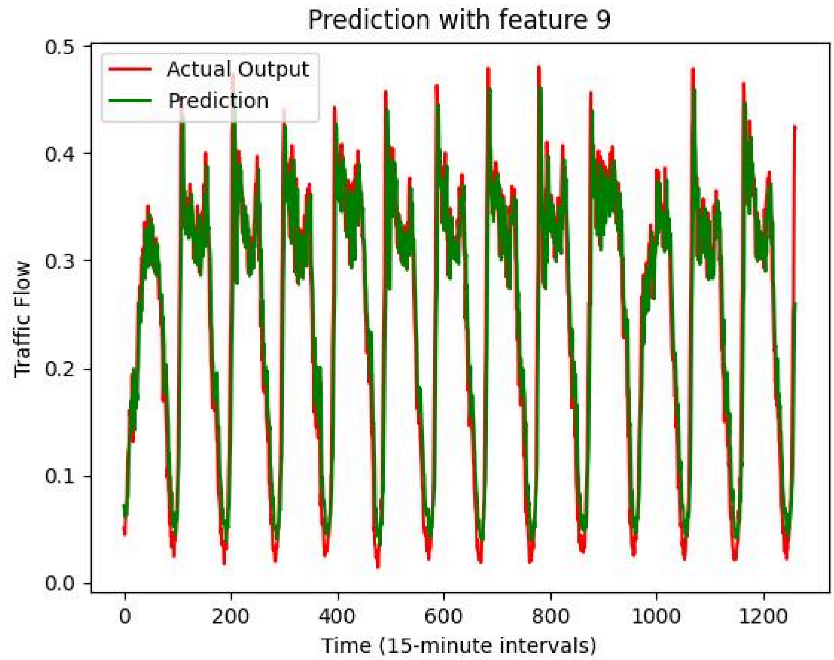
ii. **Prediction with Lag 10:** Confirms our prediction that that data is not truly random and can be predicted

- Training MAE: 0.08731196348703892
- Training RMSE: 0.10894698839724803



iii. **Prediction with Lag 1:**

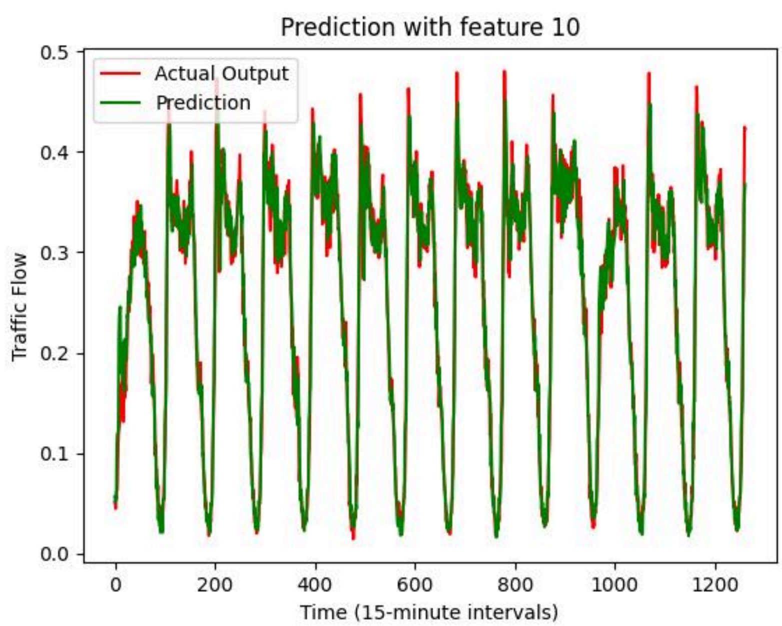
- Highly correlated. We can use this!



- Training MAE: 0.03605470311660593
- Training RMSE: 0.05078659336312084

iv. Prediction with time vectors

- There is some overfitting.



- Training MAE: 0.019540443928494636

- Training RMSE: 0.02630296300970578

2. **Analysis and Interpretation:**
    a. **Provide insightful analysis based on the visualizations.**
    b. **Identify key patterns, trends, and any anomalies in the data.**

    Based on above, its clear that we can use following things easily

    i. **Past data** high correlation with less lag
    ii. **Time of day**: Higher traffic in the middle of the day. Periodical.
    iii. Higher probability of **extreme traffic conditions**. Empty roads or high traffic.

    c. Discuss how these findings will influence your model selection and feature engineering.
        i. Model will use previous data as input
        ii. Linear regression provides a good baseline.
        iii. Additional features can be designed.
            - Example: Rolling average: To reduce no of vectors
            - Weighted average: As data is highly correlated with lesser lag
            - Reverse weighted average: For comparison
            - Time of day on linear scale and Day of week on linear scale.