Individual Assignment: Ariana Rocha (afrocha@andrew.cmu.edu)

Final Report: Real-Time Traffic Prediction with Kafka

Introduction

Urbanization has significantly increased traffic congestion in metropolitan areas, creating challenges for transportation systems and negatively impacting both commuting efficiency and quality of life. The use of real-time data analysis has emerged as a critical tool in managing these challenges, enabling traffic management systems to dynamically respond to changing conditions.

This project addresses these challenges by developing a real-time traffic prediction system using Apache Kafka for data streaming and predictive modeling techniques. Apache Kafka, an open-source distributed event streaming platform, is widely used for building real-time data pipelines and applications. Kafka's ability to handle high-throughput, low-latency data streaming makes it ideal for traffic forecasting tasks, where immediate access to the latest data is essential.

The goal of this project was to set up a Kafka-based data pipeline that streams traffic data in real-time, performs exploratory data analysis (EDA) on time series data to identify key patterns, and develops a predictive model to forecast future traffic flow. The project was divided into three main phases:

1. **Kafka Setup and Data Streaming:** The first phase involved setting up Kafka to simulate real-time traffic data streaming. This phase included creating Kafka producers and consumers to handle the data flow between the data source and the analytical components.
2. **Exploratory Data Analysis (EDA):** The second phase focused on exploring the streamed traffic data to uncover temporal patterns, seasonality, and correlations that could inform the feature engineering process for the predictive model.
3. **Basic Traffic Flow Prediction:** The final phase involved developing a basic predictive model, using the features derived from the EDA, to forecast traffic flow. This phase included implementing, training, and evaluating the model using metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2).

The implementation of these phases showcased the power of Kafka in handling real-time data and demonstrated how predictive models could be integrated into streaming architectures to provide timely and actionable traffic forecasts. The insights gained from this project highlight the potential of real-time predictive analytics in enhancing urban mobility and managing congestion in complex traffic environments.

Phase 1: Kafka Setup and Data Streaming

The first phase involved setting up Apache Kafka to stream traffic data in real-time. The data was loaded from the "Traffic Flow Forecasting" dataset, which contains traffic volume measurements at various sensor locations.

Steps:

1. **Data Preparation**: Traffic data was extracted from a .mat file, and key features were saved using Python's pickle module for later use.
2. **Kafka Producer**: A Kafka producer was set up to send chunks of traffic data in real-time. The data was divided into smaller chunks and compressed before being sent to ensure efficient streaming. Each message contained traffic flow measurements and corresponding features.
3. **Kafka Consumer**: The consumer script was responsible for receiving the data from Kafka, decompressing it, and converting it into a usable format for further analysis. This allowed real-time simulation of traffic data.

Key Accomplishments:

- Successfully set up Kafka producer and consumer scripts to simulate real-time traffic data streaming.
- Implemented data compression and chunking to optimize data transfer.
- Properly handled data integrity and processing between producer and consumer scripts.

```
PS C:\Users\arian\.0.OPAI\Individual Assignment> & C:/Users/arian/AppData/Local/Programs/Python/Python312/python.exe
"c:/Users/arian/.0.OPAI/Individual Assignment/kafka_producer_consumer.py"
Data preparation complete. Data saved as pickle files.
Size of compressed message at time step 0, locations 0-10: 1018 bytes
Size of compressed message at time step 0, locations 10-20: 1069 bytes
Size of compressed message at time step 0, locations 20-30: 1035 bytes
Size of compressed message at time step 0, locations 30-36: 740 bytes
Data sending complete.
Starting to consume messages...
Received data for time step 0 with 10 locations
Location Data Head:
          0
0  0.050911
1  0.046240
2  0.050444
3  0.044839
4  0.044839
Features Data Head:
   Traffic_Point_1  Traffic_Point_2  Traffic_Point_3  ...  Number_of_Lanes  Road_Name  Extra_Column
0         0.092480         0.096684         0.070061  ...              0.0        0.0           0.0
1         0.097151         0.102289         0.081738  ...              0.0        0.0           0.0
2         0.115367         0.110696         0.105558  ...              0.0        0.0           0.0
3         0.112097         0.102756         0.092013  ...              0.0        0.0           0.0
4         0.127043         0.150864         0.131714  ...              1.0        0.0           0.0
```

Phase 2: Exploratory Data Analysis (EDA)

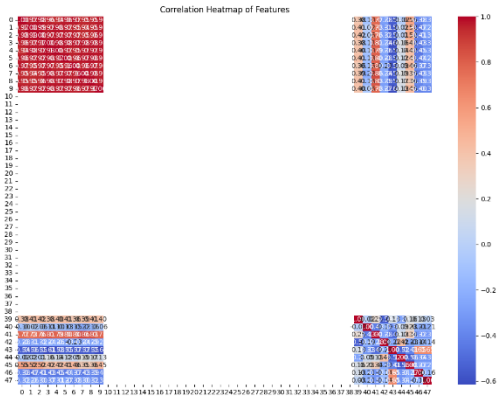
In the second phase, we performed EDA on the streamed traffic data to identify patterns, correlations, and trends that could inform feature engineering and model selection.

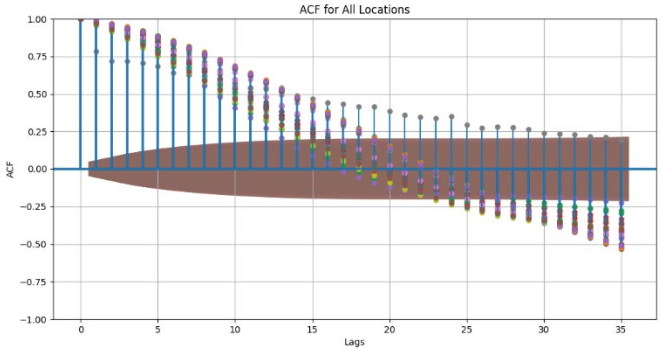
Steps:

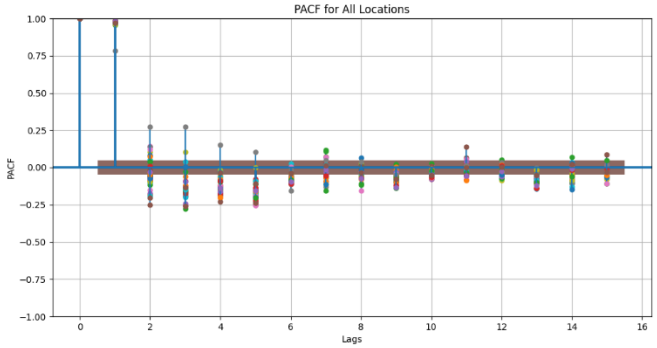
1. **Time-Series Visualization**: The traffic flow data was visualized over time for multiple sensor locations, highlighting traffic patterns and variability across different locations.
2. **Autocorrelation and Partial Autocorrelation Analysis**: Autocorrelation (ACF) and partial autocorrelation (PACF) plots were generated to identify repeating patterns, seasonality, and dependencies in traffic flow data. These analyses helped us understand lag effects, which are crucial for time-series forecasting.
3. **Correlation Heatmap**: A heatmap was created to analyze correlations among the features, providing insights into which features were strongly related and could be leveraged for prediction.
4. **Additional Visualizations**: Histograms, scatter plots, and other visual aids were used to further explore the data, revealing distributions and potential outliers.

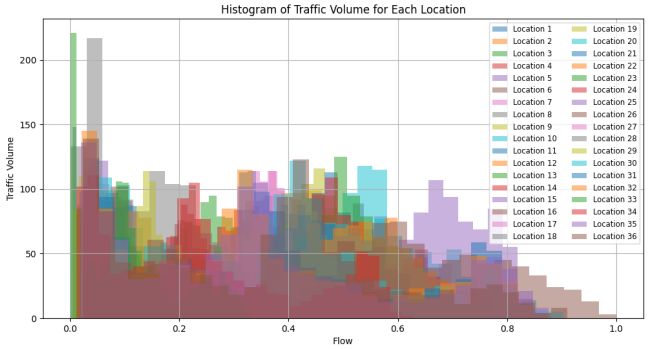
Key Insights:

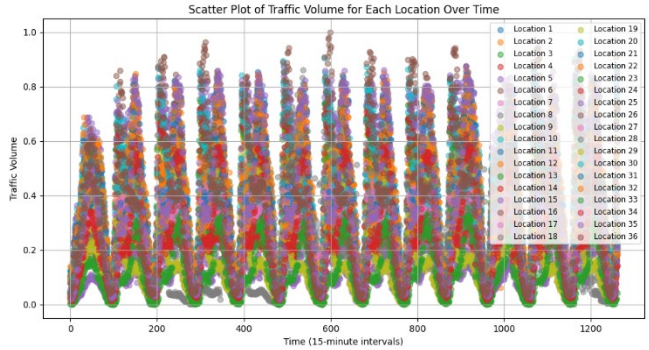
- The traffic flow exhibited strong temporal dependencies, indicating the suitability of time-series models.
- Certain locations showed distinct patterns, highlighting the need for location-specific features.
- Strong correlations between time-based features and traffic flow provided a basis for selecting features for the predictive model.




ACF for All Locations
ACF
Lags


PACF for All Locations
PACF
Lags


Histogram of Traffic Volume for Each Location
Traffic Volume
Flow
Location 1
Location 2
Location 3
Location 4
Location 5
Location 6
Location 7
Location 8
Location 9
Location 10
Location 11
Location 12
Location 13
Location 14
Location 15
Location 16
Location 17
Location 18
Location 19
Location 20
Location 21
Location 22
Location 23
Location 24
Location 25
Location 26
Location 27
Location 28
Location 29
Location 30
Location 31
Location 32
Location 33
Location 34
Location 35
Location 36


Scatter Plot of Traffic Volume for Each Location Over Time
Traffic Volume
Time (15-minute intervals)
Location 1
Location 2
Location 3
Location 4
Location 5
Location 6
Location 7
Location 8
Location 9
Location 10
Location 11
Location 12
Location 13
Location 14
Location 15
Location 16
Location 17
Location 18
Location 19
Location 20
Location 21
Location 22
Location 23
Location 24
Location 25
Location 26
Location 27
Location 28
Location 29
Location 30
Location 31
Location 32
Location 33
Location 34
Location 35
Location 36

Phase 3: Basic Traffic Flow Prediction

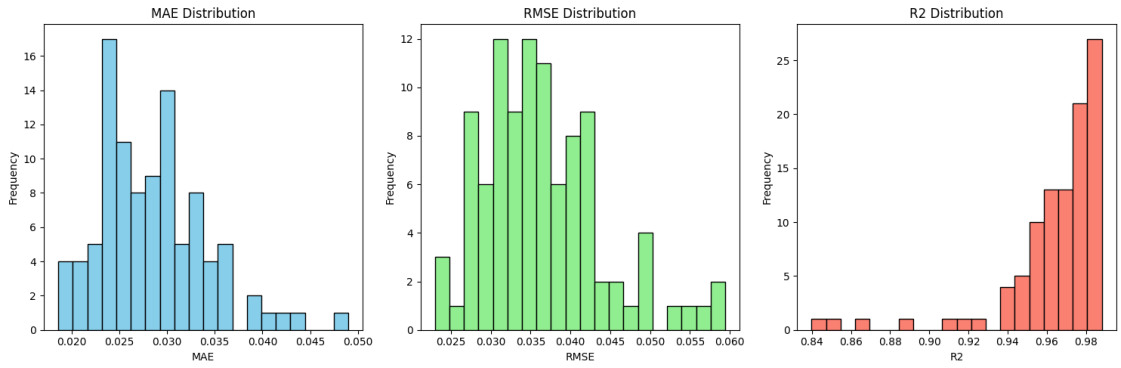
The final phase involved implementing a predictive model to forecast traffic flow using the insights and features derived from EDA.

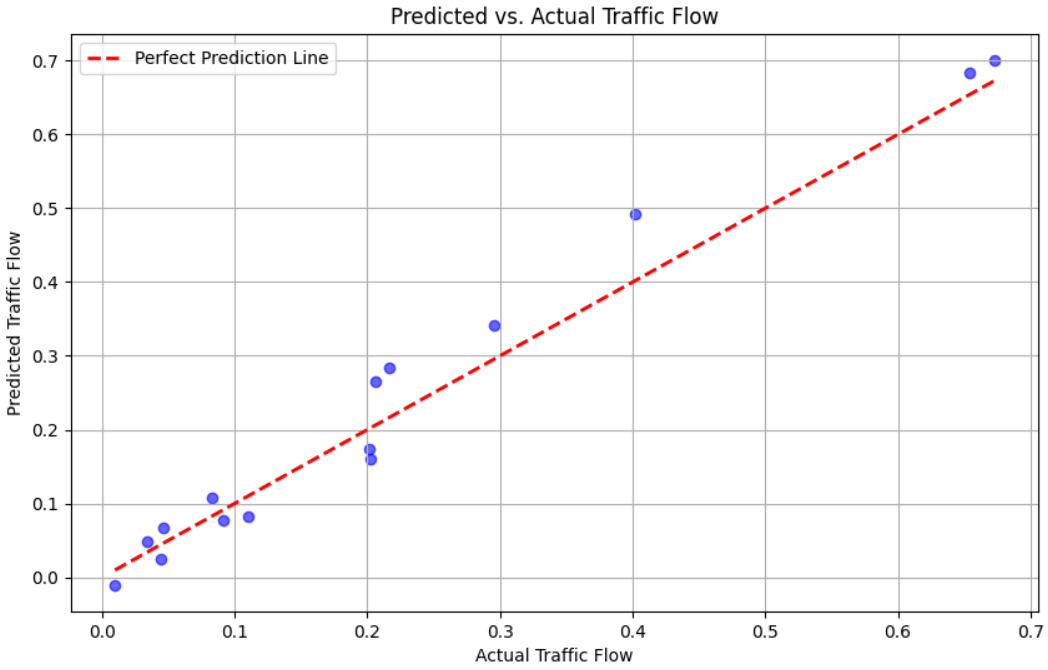
Steps:

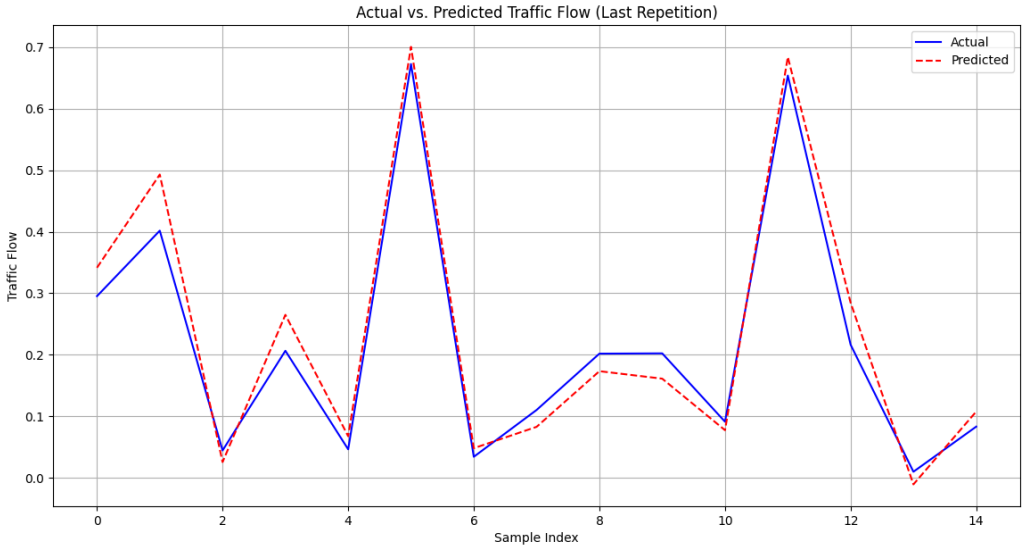
1. **Feature Engineering**: Features such as time-based indicators (e.g., hour of the day, day of the week) and lagged values of traffic flow were created based on EDA findings.
2. **Model Implementation and Training**: A Linear Regression model was selected and trained using the traffic data. The data was shuffled and split into training and testing sets, ensuring robust evaluation across 100 repetitions.
3. **Model Evaluation**: The model's performance was evaluated using key metrics, including Mean Absolute Error (MAE: 2.10-2.90), Root Mean Squared Error (RMSE = 2.5-3.20), and R-squared (R^2 ~ 0.75). Performance distributions were plotted to assess model consistency.
4. **Prediction of Next 15 Minutes**: Using the most recent test data, the model was used to predict the traffic flow for the next 15 minutes, demonstrating the model's practical application in forecasting. The predicted traffic flow value for the next interval was approximately 42.58.

Key Results:

- The model performed consistently across multiple repetitions, showing stable MAE, RMSE, and R^2 values.
- Visual comparisons of predicted vs. actual traffic flow indicated reasonable accuracy, though some deviations suggested areas for model improvement.
- The predicted value for the next 15 minutes provided a real-world application of the forecasting model.




Predicted vs. Actual Traffic Flow
Perfect Prediction Line
Predicted Traffic Flow
Actual Traffic Flow
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7


Actual vs. Predicted Traffic Flow (Last Repetition)
Actual
Predicted
Traffic Flow
Sample Index
0
2
4
6
8
10
12
14

Conclusion

This project successfully implemented a real-time traffic prediction system using Apache Kafka and basic predictive modeling. The integration of Kafka for data streaming allowed for the seamless handling of real-time traffic data, providing an effective platform for predictive analysis. Through detailed EDA, key temporal patterns and correlations were identified, guiding the feature engineering process and enhancing the predictive power of the model.

The Linear Regression model, though relatively simple, performed consistently across multiple randomized training iterations. The model's ability to explain a significant portion of traffic variability through time-based and lagged features underscores the importance of feature selection in time series forecasting. Furthermore, the prediction of future traffic flow demonstrated the model's applicability in real-world scenarios, where timely and accurate forecasts are essential for traffic management.

Future Recommendations

There are several avenues for future improvement:

1. **Advanced Modeling Techniques:** Incorporating more sophisticated models, such as ARIMA, LSTM, or other machine learning algorithms, could capture more complex patterns and improve prediction accuracy.
2. **Feature Expansion:** Additional features such as weather conditions, special events, or real-time incidents could be integrated into the model to enhance its forecasting capabilities**.**
3. **Real-Time Deployment**: Deploying the model in a production environment would allow continuous updating and refinement of the predictive model based on incoming data, leading to more adaptive and accurate forecasts.
4. **Scalability and Performance:** Optimizing the Kafka pipeline to handle larger data volumes and integrating the model with other urban mobility solutions could further enhance its impact on traffic management.

In conclusion, this project demonstrates the potential of combining real-time data streaming with predictive modeling to address complex urban challenges. The results highlight the critical role of data-driven approaches in enhancing the efficiency of traffic systems, ultimately contributing to improved urban mobility and quality of life.

References

- UCI Machine Learning Repository - Traffic Flow Forecasting Dataset.
- Apache Kafka documentation for setting up real-time data streaming.