REAL-TIME TRAFFIC ANALYSIS & PREDICTION

TEAM NAME: DATA SEEKERS

TEAM MEMBERS

- SRI NIHITHA MANDADI
- SIVA PAVAN KUMAR CHAVA
- RAKESH KUSA
- BHARGAVA REDDY ERUGULA

INTRODUCTION

01 PROBLEM STATEMENT

MANAGING TRAFFIC CONGESTION REQUIRES ACCURATE PREDICTIONS OF TRAFFIC FLOW. CURRENT SYSTEMS LACK REAL-TIME PREDICTION USING HISTORICAL HOURLY DATA.

02 DATASET INFORMATION

WE USED THE MINNESOTA_TRAFFICDATA_2020–24.CSV, COMBINING 5 YEARS OF HOURLY TRAFFIC SENSOR DATA FROM MNDOT. THE DATA WAS CLEANED AND MERGED INTO A UNIFIED DATASET.
OFFICIAL SOURCE: MNDOT HOURLY TRAFFIC VOLUME REPORTS

03 APPROACH

WE APPLIED DATA PREPROCESSING, FEATURE EXTRACTION, AND MULTIPLE ML AND TIME-SERIES MODELS TO PREDICT TRAFFIC VOLUME ACCURATELY.

04 GOAL

TO FORECAST HOURLY TRAFFIC VOLUME, CLASSIFY INTENSITY LEVELS, AND SUPPORT CONGESTION PLANNING THROUGH INSIGHTFUL VISUALIZATIONS.

TOOLS & TECHNOLOGIES USED

- Google Colab (Python 3)

PLATFORM

LIBRARIES & FRAMEWORKS

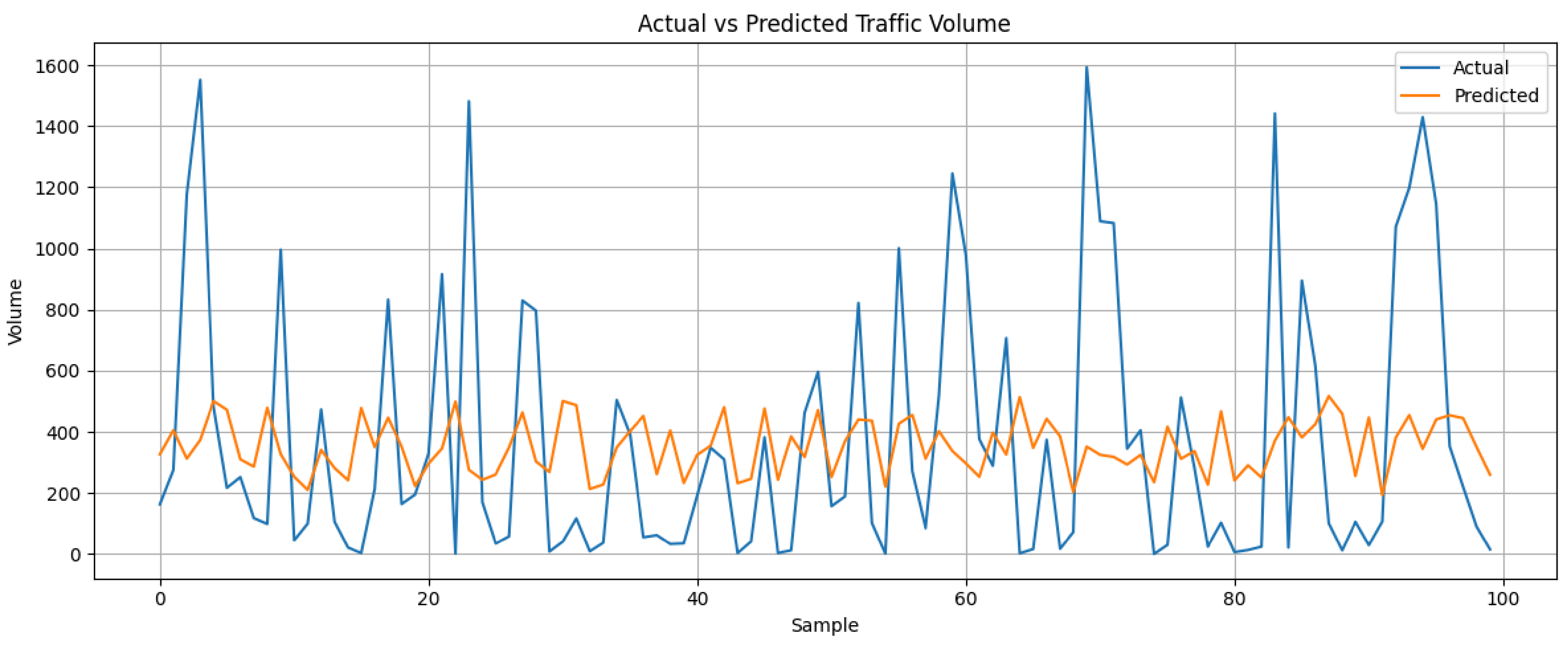
- Data Handling: Pandas, NumPy
- Machine Learning: scikit-learn, XGBoost, CatBoost
- Time Series Forecasting: Prophet
- Visualization: Matplotlib, Seaborn

METHODOLOGY

- **Data Cleaning**: Restructured hourly columns using pd.melt() and created a unified datetime column.
- **Feature Engineering**: Extracted hour, weekday, and month from datetime.
- **Model Training**: Trained models like Linear Regression, Decision Tree, Random Forest, XGBoost, CatBoost, and Prophet.
- **Evaluation**: Used RMSE to compare prediction accuracy.
- **Classification**: Categorized traffic as Low (<300), Medium (300–700), and High (>700).
- **Visualization**: Used Prophet plots and Seaborn heatmaps for trends and traffic intensity.

LINEAR REGRESSION

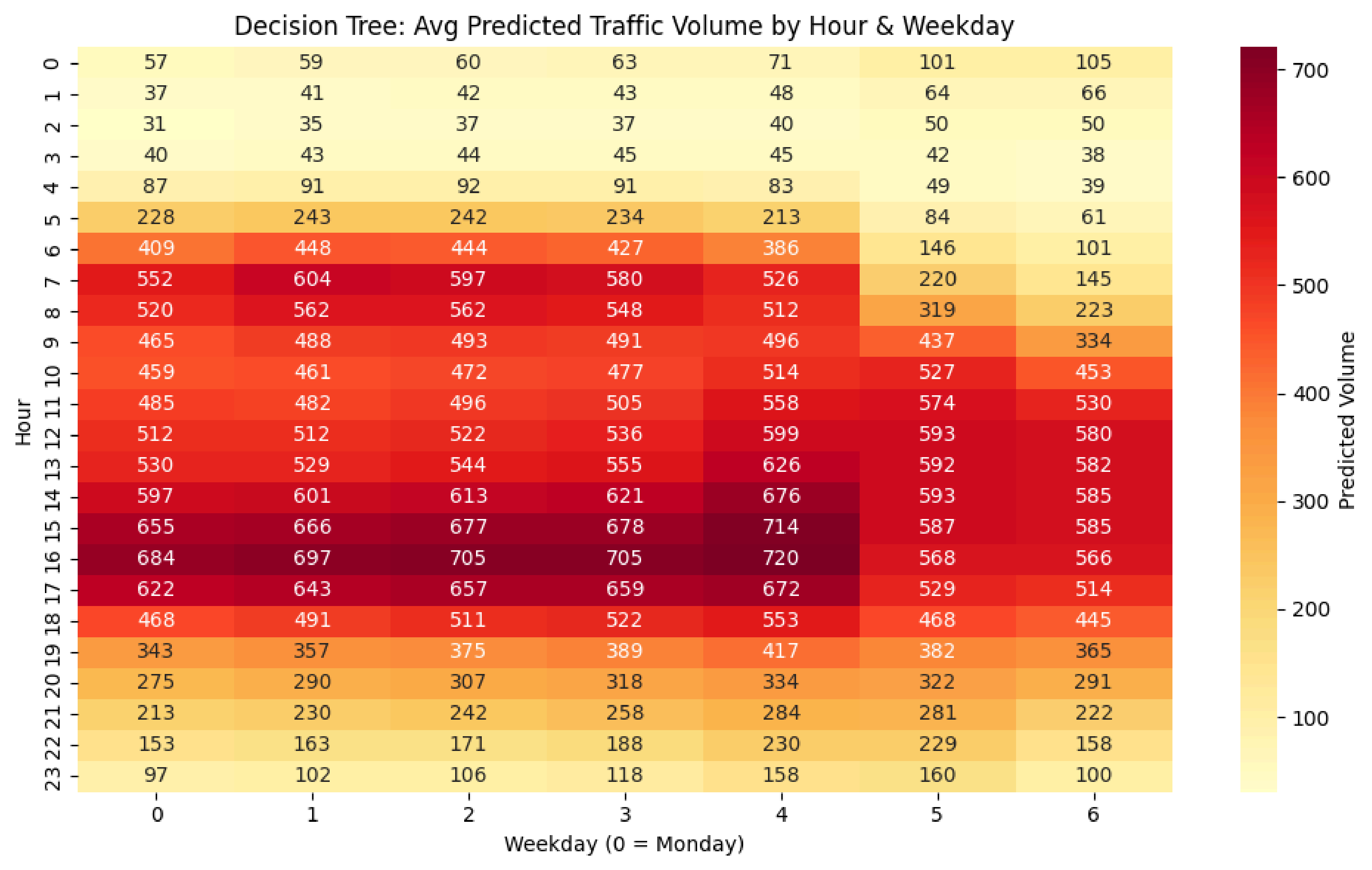
Purpose: To establish a simple baseline for predicting hourly traffic volume.
What We Did:
We trained a linear model using features like hour, weekday, and month to predict traffic flow.



Performance:
- RMSE: 417.32
- Accuracy was low due to the model's inability to capture non-linear traffic patterns.

Conclusion:
- Linear Regression served as our starting point but wasn’t suitable for capturing real-world traffic complexity.

DECISION TREE



Purpose: To build rule-based splits for more accurate predictions.

What We Did:

Implemented a decision tree to split traffic data based on feature values like hour and day.

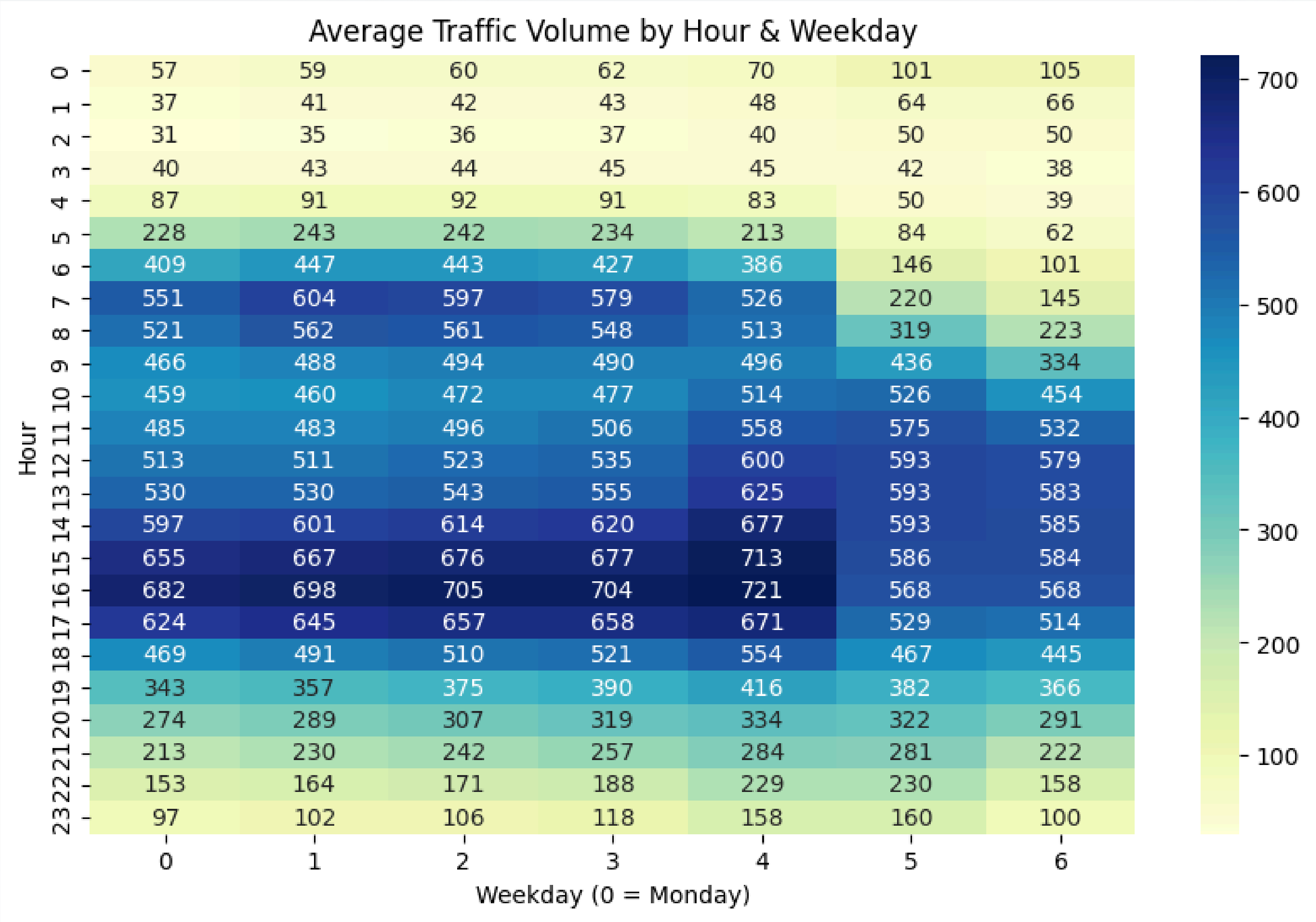
Performance:

- RMSE: 363.48
- Captured patterns better than linear regression.

Conclusion:

- Simple yet more powerful; handled fluctuations in traffic better but had slight overfitting tendencies.

RANDOM FOREST



Purpose: To improve prediction stability and reduce overfitting.

What We Did:

Used an ensemble of multiple decision trees and averaged the results.

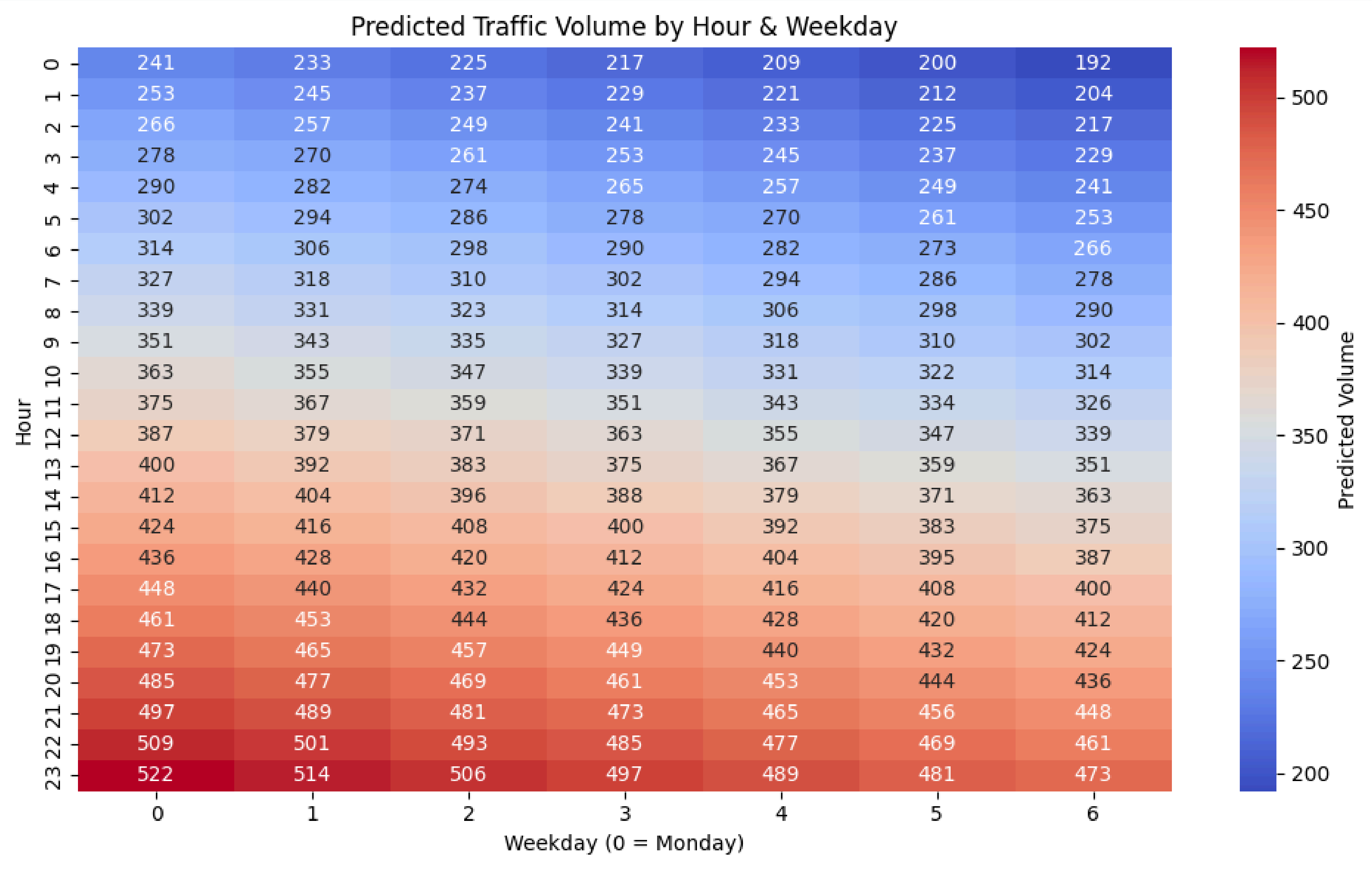
Performance:

- RMSE: 363.48
- Gave consistent results across training and test data.

Conclusion:

- A reliable and balanced model with improved prediction quality over a single tree.

XGBOOST



Purpose: To fine-tune predictions by reducing errors in successive iterations.

What We Did:

Trained XGBoost on engineered time features to capture complex patterns in traffic.

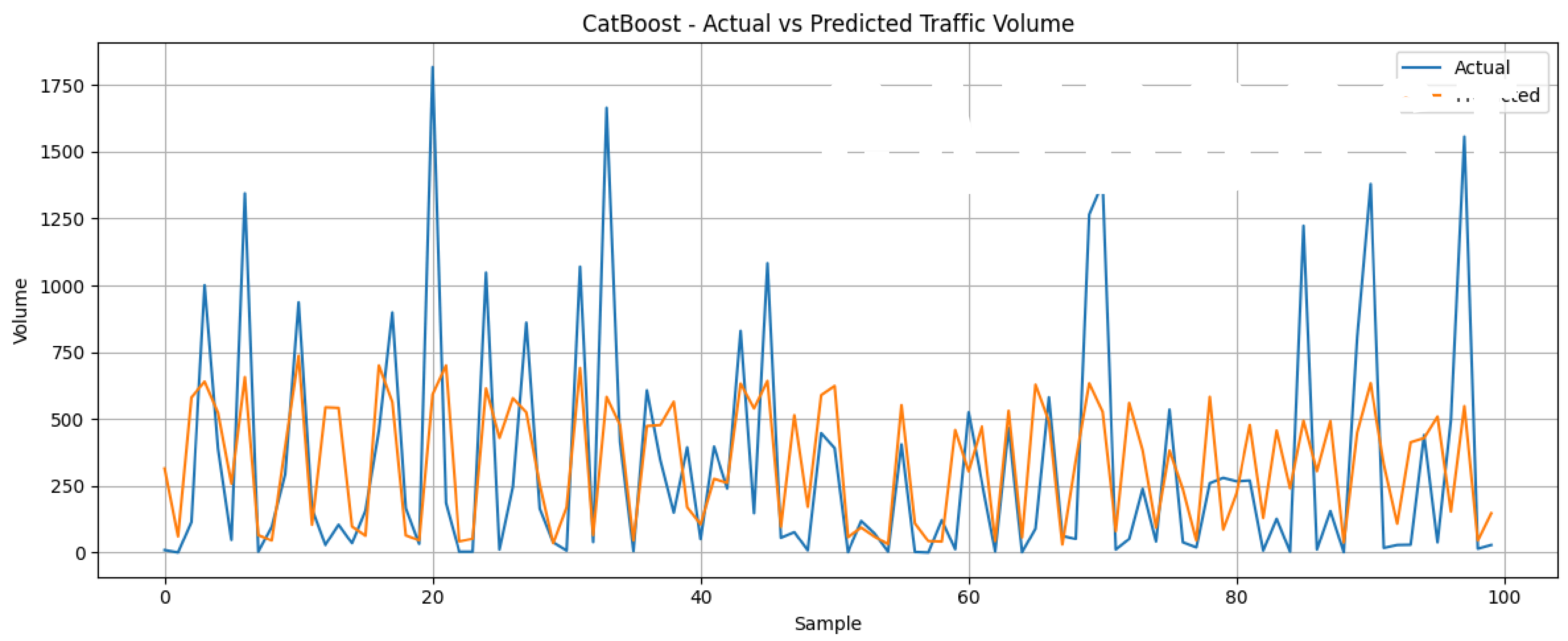
Performance:

- RMSE: 363.69
- Slightly better than Random Forest.

Conclusion:

- Powerful model for structured data; improved learning through boosting.

CATBOOST



Purpose: To improve performance on categorical time features.

What We Did:

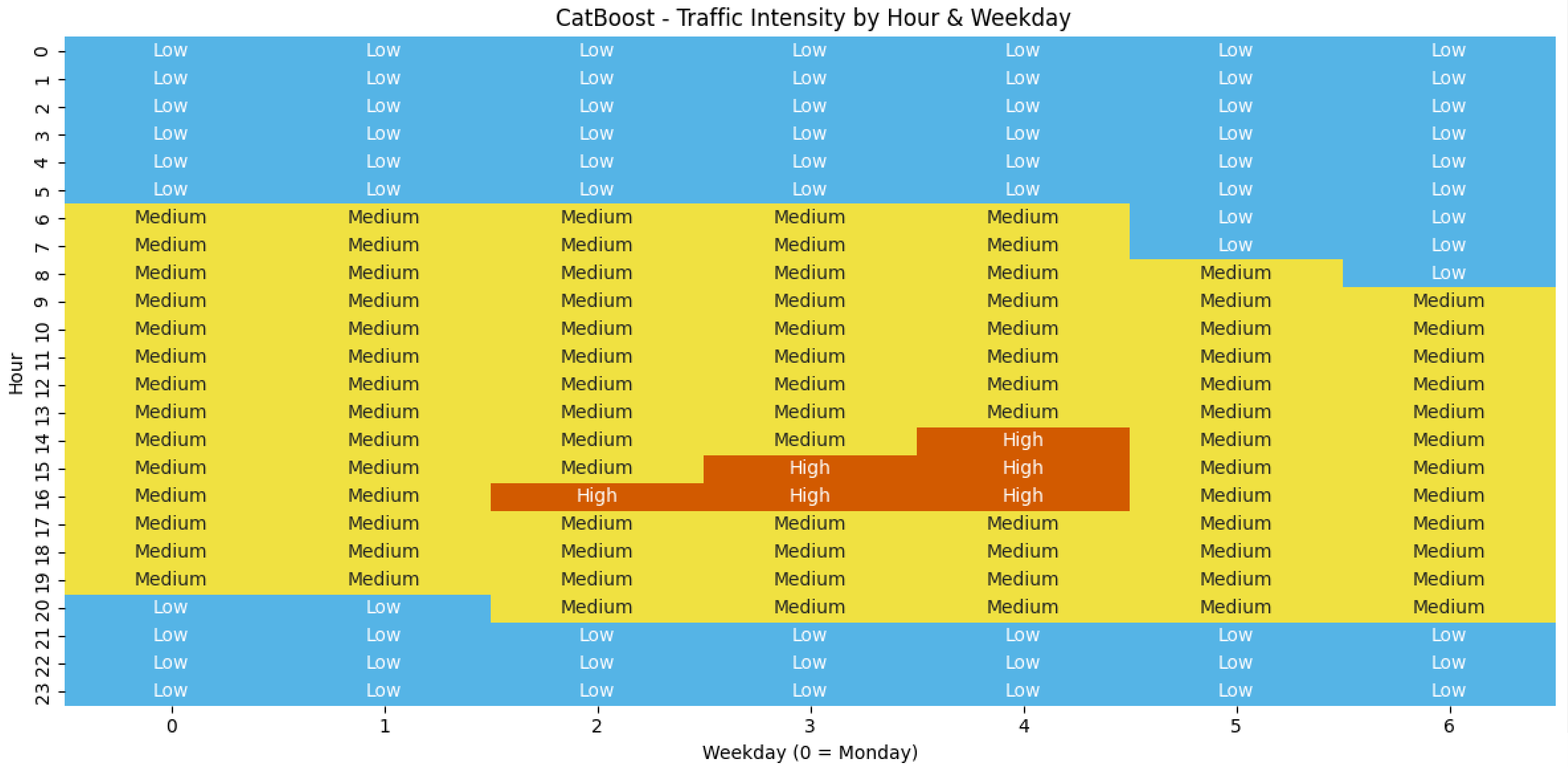
Leveraged CatBoost to efficiently learn from features like weekday and hour.

Performance:

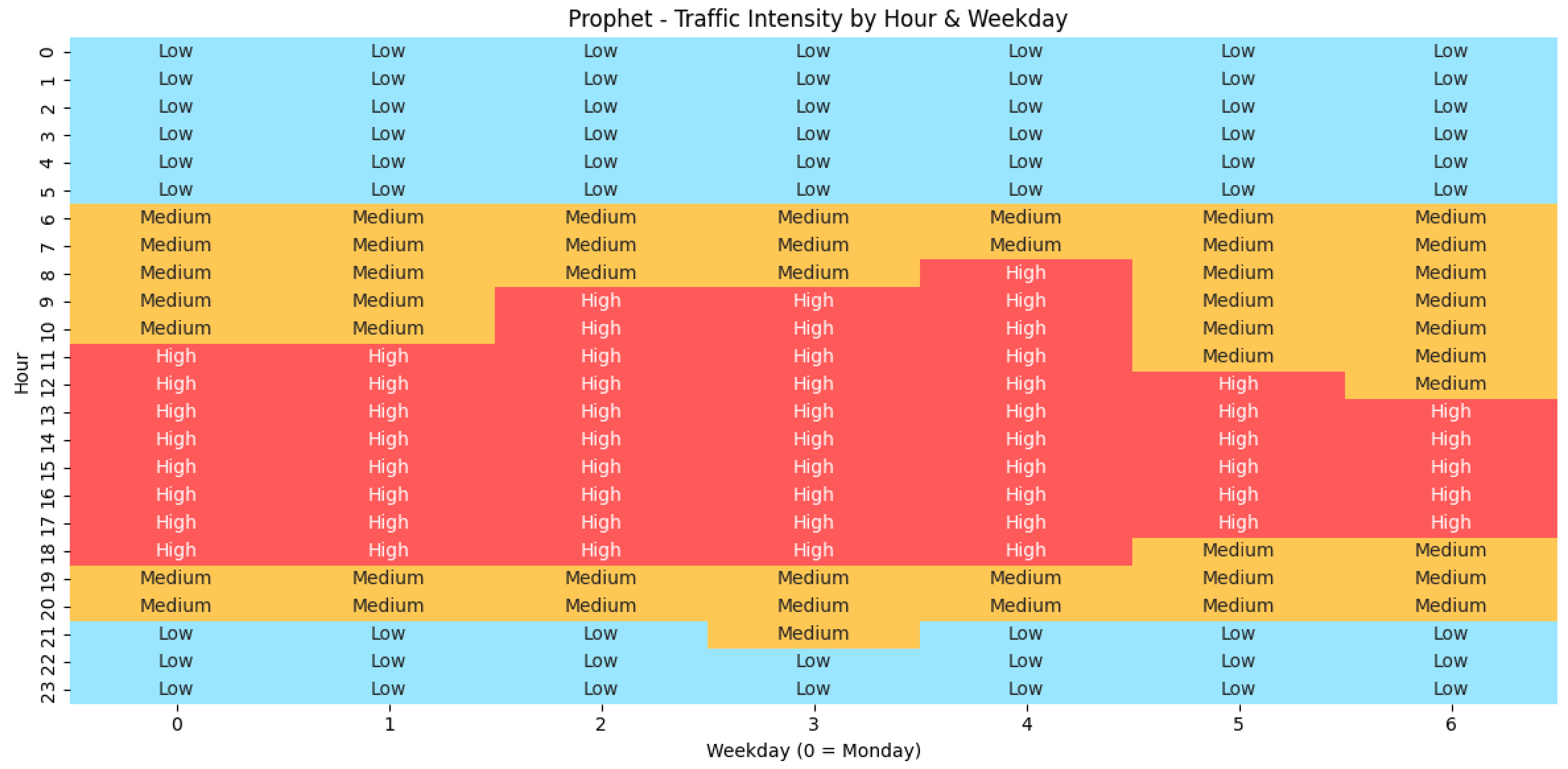
- RMSE: 363.47 (Best among all tree-based models)

Conclusion:

- Offered smooth training, handled categorical features automatically, and showed best results among all ensemble methods.



PROPHET



Purpose: To capture daily, weekly, and seasonal trends in traffic data.

What We Did:
Used Prophet to model long-term and periodic traffic patterns across 5 years of data.

Performance:

- RMSE: 80.57 (Best among all models)
- Generated trend, seasonal, and weekly plots.

Conclusion:

- Highly effective in identifying temporal trends. Despite slightly longer computation time, Prophet gave the most accurate predictions in our project.

MODEL COMPARISON

Model	**RMSE (Lower is Better)**
Linear Regression	417.32
Decision Tree	363.47
Random Forest	363.47
XGBoost	363.69
CatBoost	363.47
Prophet	80.57

CONCLUSION

Prophet gave the most accurate traffic predictions.

Tree-based models (CatBoost, XGBoost) performed well.

Visualizations revealed clear traffic trends and peak times.

FUTURE SCOPE

Add live traffic data for real-time updates.

Predict congestion zones and risk areas.

Build a user-friendly mobile/web app.

Explore deep learning models for better accuracy.

THANK YOU