Project Objective

To forecast vehicle_count at different junctions using a variety of time-series, weather, location, and event-based features. The goal is to build robust predictive models and iteratively refine them through diagnostics, validation, and tuning.

1. Dataset Summary

- Records: ~9,690 rows
- Target Variable: vehicle_count
- Temporal Features: hour, day_of_week, month, lag_1h, lag_24h, lag_168h
- Weather & Traffic: weather_condition, traffic_level, temp, humidity, precipitation, windspeed
- Event Indicators: concert, holiday, sports_event, protest
- Spatial Features: pickup_location, dropoff_location

2. Initial Model: XGBoost Regressor

Setup:

- Model: XGBoostRegressor (100 trees, default parameters)
- Validation: TimeSeriesSplit (n_splits=5)

Performance (Cross-Validation Averages):

Metric	Mean	Std Dev
MAE	~0.0219	±0.0017
RMSE	~0.0313	±0.0025
R^2	~0.8782	±0.0089

Observations:

- High R^2 values indicate strong fit.
- Residuals centered around zero but slightly skewed during event days.
- Some increase in MAE and RMSE over time indicates minor concept drift.

3. Error Analysis & Diagnostic Findings

- Underprediction during concerts, holidays, and sports events.
- MAE increased on weekends and high humidity/windspeed days.
- Feature importance indicates temporal lags and location features dominate; weather features have lower predictive power.

Symptoms Identified:

Symptom	**Diagnosis**	**Action**
R^2 drops in later folds	Concept drift	Consider time-aware retraining strategy
MAE variance across folds	High model variance	Increase regularization or reduce complexity
$R^2 < 0.9$ on average	Possible underfitting	Enhance feature representation

4. Model Comparison

We evaluated RandomForestRegressor as a baseline against XGBoost.

Model	**MAE**	**RMSE**	**R^2**
XGBoost	0.0218	0.0312	0.8792
Random Forest	0.0223	0.0324	0.8721

Conclusion: XGBoost performs better across all metrics but RF is competitive and useful for model ensembling.

5. Hyperparameter Tuning Results

XGBoost (GridSearchCV):

- Best Parameters: max_depth=5, learning_rate=0.1, n_estimators=100
- Best CV MAE: ~0.0207

Random Forest (RandomizedSearchCV):

- Best Parameters: n_estimators=150, max_depth=7, min_samples_split=4, max_features='sqrt'
- Best CV MAE: ~0.0214

6. Model Refinement Recommendations

Feature Engineering

- Add features like sin(hour*π/12) and cos(hour*π/12) for cyclical encoding.
- Construct event proximity flags (e.g., 3 hours before/after a concert).
- Create composite interaction features (e.g., is_weekend × traffic_level).

Algorithm Strategy

- Use ensemble methods: XGBoost + RF stacking.
- Try LightGBM or CatBoost for fast and effective boosting.
- Explore LSTM or GRU for sequential prediction if longer sequences are required.

Temporal Strategy

- Use expanding window or rolling window CV instead of static folds.
- Periodically retrain the model monthly to address drift.

Tuning Strategy

- Use Bayesian Optimization (e.g., Optuna) for finer hyperparameter tuning.
- Add early stopping to boosting methods to prevent overfitting.

7. Visual Diagnostic Summary

- Residual plots show small but consistent underprediction on event-heavy periods.
- Error distribution follows Gaussian-like shape with some right skew.
- Prediction vs. Actual plots show good trend tracking but dips around holidays.

8. Conclusion

- Model performance is strong (R^2 ~ 88%) but can be improved through:
    - Better temporal encoding
    - Focus on events and their lead/lag effects
    - Regular retraining or adaptive models
- Ensemble and hybrid methods recommended for deployment.