Model Evaluation & Refinement Report

1. Introduction

Objective

The objective of this report is to evaluate the performance of various machine learning models for predicting traffic volumes based on historical data. The report also aims to refine the models to improve predictive accuracy using insights gained from evaluation metrics and diagnostic analyses.

Models Evaluated

- RandomForestRegressor

- GradientBoostingRegressor

- LSTM (Long Short-Term Memory)

Evaluation Metrics

- Mean Absolute Error (MAE)

- Root Mean Square Error (RMSE)

- R-squared (R2)

2. Data Preparation

Data Sources

- Traffic Data: Provided dataset containing historical traffic volumes.

- Weather Data: Retrieved from meteorological services, including temperature, precipitation, humidity, and wind speed.

Data Integration

- Merged traffic, weather, and event data into a unified dataset based on timestamps.

- Cleaned dataset to handle duplicates, missing values, and inconsistencies.

3. Model Evaluation

Evaluation Metrics Calculation

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Calculate evaluation metrics for each model

mae_rf = mean_absolute_error(y_true, y_pred_rf)
```

```
rmse_rf = mean_squared_error(y_true, y_pred_rf, squared=False)
r2_rf = r2_score(y_true, y_pred_rf)
mae_gb = mean_absolute_error(y_true, y_pred_gb)
rmse_gb = mean_squared_error(y_true, y_pred_gb, squared=False)
r2_gb = r2_score(y_true, y_pred_gb)
mae_lstm = mean_absolute_error(y_true, y_pred_lstm)
rmse_lstm = mean_squared_error(y_true, y_pred_lstm, squared=False)
r2_lstm = r2_score(y_true, y_pred_lstm)
```

Residual Analysis

Residual Plot

```
plt.figure(figsize=(10, 6))
plt.scatter(y_true, residuals_rf, label='RandomForest', alpha=0.7)
plt.scatter(y_true, residuals_gb, label='GradientBoosting', alpha=0.7)
plt.scatter(y_true, residuals_lstm, label='LSTM', alpha=0.7)
plt.axhline(0, color='red', linestyle='--')
plt.xlabel('Actual Values')
plt.ylabel('Residuals')
plt.title('Residual Plot')
plt.legend()
plt.show()
```

Prediction vs. Actual Plot

```
plt.figure(figsize=(10, 6))
plt.plot(y_true, label='Actual')
plt.plot(y_pred_rf, label='RandomForest Predicted')
plt.plot(y_pred_gb, label='GradientBoosting Predicted')
plt.plot(y_pred_lstm, label='LSTM Predicted')
plt.xlabel('Time')
plt.ylabel('Traffic Volume')
plt.title('Prediction vs. Actual')
plt.legend()
plt.show()
```

4. Model Refinement

Diagnose Model Issues

- Identified issues such as high bias in RandomForestRegressor and overfitting in LSTM based on evaluation metrics and residual analysis.

Refinement Strategies

Feature Engineering

- Introduced lag features to capture temporal dependencies.
- Included time-based features (hour, day of week, month) to improve model accuracy.

Algorithm Selection

- Experimented with ensemble methods like GradientBoostingRegressor to mitigate overfitting observed in LSTM.

Hyperparameter Tuning

- Conducted grid search to optimize hyperparameters of RandomForestRegressor for better performance.

```
from sklearn.model_selection import GridSearchCV
param_grid = {
    'n_estimators': [50, 100, 150],
    'max_depth': [None, 10, 20],
    'min_samples_split': [2, 5, 10]
}
grid_search = GridSearchCV(estimator=RandomForestRegressor(), param_grid=param_grid, cv=5,
scoring='neg_mean_squared_error')
grid_search.fit(X_train, y_train)

best_params = grid_search.best_params_
best_model = grid_search.best_estimator_
```

5. Model Performance Summary

Best Model Performance

- RandomForestRegressor:
  - MAE: 2.825
  - RMSE: 4.460
  - R2: 0.956