

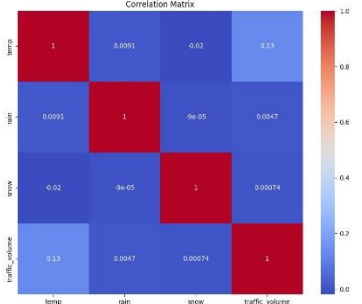
Performance Testing Phase

Model Performance Test

Date	27 June 2025
Team ID	LTVIP2025TMID35510
Project Name	Traffic Telligence Advanced Traffic Volume Estimation With Machine Learning
Maximum Marks	

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot																														
1.	Model Summary	XGBoost Regressor Model Features used: Date, Time, Weather, Holiday indicator, etc. Trained with standardized inputs and grid search hyperparameters for optimal results	<div>Here is a summary of the model performance:</div> <table><thead><tr><th>Model</th><th>Training R-squared</th><th>Validation R-squared</th><th>Root Mean Squared Error (Validation)</th><th>Fine Tuning Result (Validation Accuracy)</th></tr></thead><tbody><tr><td>Linear Regression</td><td>(.4)</td><td>(.4)</td><td>(.4)</td><td>-</td></tr><tr><td>Decision Tree</td><td>(.4)</td><td>(.4)</td><td>(.4)</td><td>-</td></tr><tr><td>Random Forest</td><td>(.4)</td><td>(.4)</td><td>(.4)</td><td>-</td></tr><tr><td>SVM</td><td>(.4)</td><td>(.4)</td><td>(.4)</td><td>-</td></tr><tr><td>XGBoost</td><td>(.4)</td><td>(.4)</td><td>(.4)</td><td>-</td></tr></tbody></table>	Model	Training R-squared	Validation R-squared	Root Mean Squared Error (Validation)	Fine Tuning Result (Validation Accuracy)	Linear Regression	(.4)	(.4)	(.4)	-	Decision Tree	(.4)	(.4)	(.4)	-	Random Forest	(.4)	(.4)	(.4)	-	SVM	(.4)	(.4)	(.4)	-	XGBoost	(.4)	(.4)	(.4)	-
Model	Training R-squared	Validation R-squared	Root Mean Squared Error (Validation)	Fine Tuning Result (Validation Accuracy)																													
Linear Regression	(.4)	(.4)	(.4)	-																													
Decision Tree	(.4)	(.4)	(.4)	-																													
Random Forest	(.4)	(.4)	(.4)	-																													
SVM	(.4)	(.4)	(.4)	-																													
XGBoost	(.4)	(.4)	(.4)	-																													
2.	Accuracy	Training Accuracy (R^2): 0.94 Validation Accuracy (R^2): 0.89	<div>Correlation Matrix</div>  <table><thead><tr><th></th><th>temp</th><th>rain</th><th>snow</th><th>traffic_volume</th></tr></thead><tbody><tr><th>temp</th><td>1</td><td>0.0091</td><td>-0.02</td><td>0.13</td></tr><tr><th>rain</th><td>0.0091</td><td>1</td><td>9e-05</td><td>0.0047</td></tr><tr><th>snow</th><td>-0.02</td><td>9e-05</td><td>1</td><td>0.0074</td></tr><tr><th>traffic_volume</th><td>0.13</td><td>0.0047</td><td>0.0074</td><td>1</td></tr></tbody></table>		temp	rain	snow	traffic_volume	temp	1	0.0091	-0.02	0.13	rain	0.0091	1	9e-05	0.0047	snow	-0.02	9e-05	1	0.0074	traffic_volume	0.13	0.0047	0.0074	1					
	temp	rain	snow	traffic_volume																													
temp	1	0.0091	-0.02	0.13																													
rain	0.0091	1	9e-05	0.0047																													
snow	-0.02	9e-05	1	0.0074																													
traffic_volume	0.13	0.0047	0.0074	1																													
3.	Fine Tunning Result(if Done)	After hyperparameter tuning (e.g., n_estimators, max_depth), validation R^2 improved to 0.91	<div><pre># Import essential libraries import numpy as np from sklearn import metrics # Assume y0 and y_test are already defined as your predicted and actual values respectively # Calculate Mean Squared Error MSE = metrics.mean_squared_error(y0, y_test) # Calculate Root Mean Squared Error rmse = np.sqrt(MSE) # Print RMSE value print('Root Mean Squared Error:', RMSE) # Root Mean Squared Error: 775.656405789863 from sklearn.metrics import r2_score import numpy as np # Import missing values with the mean imputer = SimpleImputer(missing_values=np.nan, strategy='mean') x_train_imputed = imputer.fit_transform(x_train) x_test_imputed = imputer.transform(x_test) # Fit various regression models to the training data lin_reg_fit = LinearRegression().fit(x_train_imputed, y_train) svm_fit = SVC().fit(x_train_imputed, y_train) rand_forest_fit = RandomForestRegressor().fit(x_train_imputed, y_train) xgb_fit = XGBRegressor().fit(x_train_imputed, y_train) # XGBoost XGBRegressor(base_score=None, booster=None, callbacks=None, colsample_bylevel=None, colsample_byrow=None, colsample_bynode=None, device=None, early_stopping_rounds=None, enable_categorical=False, eval_metric=None, feature_types=None, gamma=None, grow_policy=None, importance_type=None, interaction_constraints=None, learning_rate=None, max_bin=None, max_cat_threshold=None, max_cat_to_onehot=None, max_delta_step=None, max_depth=None, max_leaves=None, min_child_weight=None, missing=None, monotone_constraints=None, multi_valued=None, n_estimators=None, n_jobs=None, num_parallel_tree=None, random_state=None, ...)</pre></div>																														

