Model Development Phase Template

Date	20 JULY 2024
Team ID	SWTID1720014187
Project Title	TrafficTelligence: Advanced Traffic Volume Estimation With Machine
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training Code:

```
#splitting into independant and dependant variables
y=data['traffic_volume']
x=data.drop(columns=['traffic_volume'],axis=1)
print(x.head())

#splitting the data into train data and test data
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=42)
lin_reg = linear_model. LinearRegression()
Dtree = tree. DecisionTreeRegressor()
Rand = ensemble. RandomForestRegressor()
svr = svm. SVR( )
#XGB = xgboost . XGBRegressor ()
```

```
from sklearn import linear_model
from sklearn import tree
from sklearn import ensemble
from sklearn import svm
```

```
lin_reg.fit(x_train,y_train)
Dtree.fit(x_train,y_train)
Rand.fit(x_train,y_train)
svr.fit(x_train,y_train)
#XGB.fit(x_train,y_train)
p1 = lin_reg.predict(x_train)
p2 = Dtree.predict(x_train)
p3 = Rand.predict(x_train)
p4 = svr.predict(x_train)
#p5 = XGB.predict(x_train)
```

```
p1 = lin_reg.predict(x_test)
p2 = Dtree.predict(x_test)
p3 = Rand.predict(x_test)
p4 = svr.predict(x_test)
print(metrics. r2_score(p1,y_test))
print(metrics. r2_score(p2,y_test))
print(metrics. r2_score(p3,y_test))
print(metrics.r2_score(p4,y_test))
#print(metrics. r2_score(p5,y_test))
```

```
-5.491461561547912
0.7130190373733469
0.8117988884163669
-15966000.275938746
```

Model Validation and Evaluation Report:

Model	Regression Report	R2_score

Linear Regression	`p1 = lin_reg.predict(x_test)`
	`regression_report(y_test,p1)`
	
	{'Mean Absolute Error (MAE)': 1637.9870039113694,
	'Mean Squared Error (MSE)': 3402975.5125765526,
	'Root Mean Squared Error (RMSE)': 1844.7155641389684,
	'R-squared (R^2)': 0.1392528540190069,
	'Explained Variance Score': 0.13930894538755123}
		13%

Decision Tree Regressor	`p2 = Dtree.predict(x_test)`
	`regression_report(y_test,p2)`
	
	{'Mean Absolute Error (MAE)': 556.0734363655223,
	'Mean Squared Error (MSE)': 1118141.6407011722,
	'Root Mean Squared Error (RMSE)': 1057.4221676800482,
	'R-squared (R^2)': 0.7171777397518407,
	'Explained Variance Score': 0.7173099360906563}
		71%

Random Forest Regressor	`p3 = Rand.predict(x_test)`
	`regression_report(y_test,p3)`
	
	{'Mean Absolute Error (MAE)': 494.5744746395602,
	'Mean Squared Error (MSE)': 612380.9824446529,
	'Root Mean Squared Error (RMSE)': 782.5477509038365,
	'R-squared (R^2)': 0.8451046206638214,
	'Explained Variance Score': 0.8452203153920186}
		84%

SVR	`p4 = svr.predict(x_test)`
	`regression_report(y_test,p4)`
	
	{'Mean Absolute Error (MAE)': 1745.497301318169,
	'Mean Squared Error (MSE)': 3962326.1639990797,
	'Root Mean Squared Error (RMSE)': 1990.559259102597,
	'R-squared (R^2)': -0.002229056454693401,
	'Explained Variance Score': 0.00012691427202182748}
		0%