


Model Development Phase Template

Date	12 JULY 2024
Team ID	739702
Project Title	Optimising food delivery Using ML
Maximum Marks	4 Marks

Model	Classification Report	Accuracy
Random forest classifier	<pre>from sklearn.ensemble import RandomForestRegressor rfr=RandomForestRegressor()</pre> <pre>[] Predictions(rf)</pre> <pre>R2Score for Training 0.9212348902293894</pre> <pre>MSE for Training: 5.353902847665801</pre> <pre>MAE for Training: 5.353902847665801</pre>	<pre>▶ Predictions(rf)</pre> <pre>↔ R2Score for Training 0.9212348902293894</pre>

Decision Tree classifier	<pre>[] from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error def Predictions(model): model.fit(x_train,y_train) y_pred_test=model.predict(x_test) y_pred_train=model.predict(x_train) print("R2Score for Training", r2_score(y_pred_train,y_train)) print("\n") print("MSE for Training", mean_squared_error(y_pred_train,y_train)) print("\n") print("MAE for Training", mean_squared_error(y_pred_train,y_train)) print("\n") # Handle potential NaN values in predictions on test data y_pred_test_clean = np.nan_to_num(y_pred_test) # Replace NaN with 0 y_test_clean = np.nan_to_num(y_test) # Replace NaN with 0 print("R2Score for Testing", r2_score(y_pred_test_clean, y_test_clean)) # Use cleaned arrays print("\n") print("MSE for Testing", mean_squared_error(y_pred_test_clean, y_test_clean)) # Use cleaned arrays print("\n") print("MAE for Testing", mean_absolute_error(y_pred_test_clean, y_test_clean)) # Use cleaned arrays [] from sklearn.tree import DecisionTreeRegressor dt=DecisionTreeRegressor()</pre>	 R2Score for Training 0.9994962879709425 MSE for Training: 0.04437864414834311 MAE for Training: 0.04437864414834311 R2Score for Testing 0.22057379136206234 MSE for Testing: 71.564096940454 MAE for Testing: 6.4904046496326355
--------------------------	--	---

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:

```
[ ] from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42)
```

Model Validation and Evaluation Report:

Xgboost classifier	 <pre>from xgboost import XGBRegressor xgb=XGBRegressor()</pre>	 Predictions(xgb)  R2Score for Training 0.5008550984251501
--------------------	--	---

