

Project Development Phase
Model Performance Test

Date	16 November 2022
Team ID	PNT2022TMID23326
Project Name	Project – DemandEst-AI Powered Food Demand Forecaster
Maximum Marks	10 Marks

Model Performance Testing:

S.No.	Parameter	Values	Screenshot
1.	Metrics	Regression Model: MAE 89.10334778841495, MSE - 43129.82977026746, RMSLE - 207.67722496765856, R2 score - 0.6946496854280233,	Evaluating the model <pre>In [33]: from sklearn.metrics import mean_squared_error</pre> <pre>In [34]: RMLSE=np.sqrt(mean_squared_error(y_test,pred)) RMLSE</pre> <pre>Out[34]: 209.71961740201198</pre> <pre>In [39]: from sklearn import metrics from sklearn.metrics import mean_absolute_error</pre> <pre>In [40]: MSE=print(metrics.mean_squared_error(y_test,pred)) MSE</pre> <pre>43982.31792324628</pre> <pre>In [41]: R2S=print(metrics.r2_score(y_test,pred)) R2S</pre> <pre>0.6886142448276894</pre> <pre>In [42]: MAE=print(mean_absolute_error(y_test,pred))</pre> <pre>89.10334778841495</pre>

2. Tune the Model

Hyperparameter Tuning -
 RMSLE- 52.85812511759974
 avg R-squared- 0.123
 MSE: -64230.918

```
In [36]: print("R-Squared:{}".format(grid_cv_dtm.best_score))
print("Best Hyperparameters:{}".format(grid_cv_dtm.best_params_))

R-Squared:-0.7691137863985942
Best Hyperparameters:
{'max_leaf_nodes': None, 'min_samples_leaf': 4, 'min_samples_split': 16}

In [39]: df = pd.DataFrame(data=grid_cv_dtm.cv_results_)
df.head()

Out[39]:
```

	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_max_leaf_nodes	param_min_samples_leaf	param_min_samples_split	param
0	5.324927	1.063213	0.096586	0.028955	None	1	2	{'max_leaf_nodes': None, 'min_samples_leaf': 1}
1	4.932083	0.468172	0.066534	0.006248	None	1	4	{'max_leaf_nodes': None, 'min_samples_leaf': 1}
2	4.587915	0.328380	0.050024	0.006244	None	1	8	{'max_leaf_nodes': None, 'min_samples_leaf': 1}
3	4.148344	0.108443	0.043753	0.018594	None	1	16	{'max_leaf_nodes': None, 'min_samples_leaf': 1}
4	4.017265	0.795451	0.056351	0.008479	None	2	2	{'max_leaf_nodes': None, 'min_samples_leaf': 2}

```
<
In [42]: r2_scores = cross_val_score(grid_cv_dtm.best_estimator_, X, y, cv=10)
mse_scores = cross_val_score(grid_cv_dtm.best_estimator_, X, y, cv=10, scoring='neg_mean_squared_error')

print("avg R-squared:{:.3f}".format(np.mean(r2_scores)))
print("MSE:{:.3f}".format(np.mean(mse_scores)))

avg R-squared:-0.123
MSE:-.64238.918

In [45]: grid_cv_dtm.best_estimator_.fit(X_train, y_train)
y_pred = grid_cv_dtm.best_estimator_.predict(X_test)
y_pred[y_pred<0] = 0
from sklearn import metrics
print("RMSE: ", 100*np.sqrt(metrics.mean_squared_log_error(y_test, y_pred)))

RMSE: 52.05812511759974
```

Tuning the model Using GridSearchCV

[illegible]