

Project Development Phase
Model Performance Test

Date	15 November 2022
Team ID	PNT2022TMID19852
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]: RMSLE=np.sqrt(mean_squared_error(y_test,pred)) RMSLE</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>

Tune the Model

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

```
In [38]: print("R-Squared:{}".format(grid_cv_dtn.best_score_))
          print("Best Hyperparameters:{}".format(grid_cv_dtn.best_params_))

R-Squared:0.7601137863085042
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()
```

	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_max_leaf_nodes	param_min_samples_leaf	param_min_samples_split
0	5.324927	1.065213	0.090598	0.028985	None	1	2	(max_leaf_nodes: None, min_samples_leaf: 1,		
1	4.932083	0.489172	0.065634	0.009248	None	1	4	(max_leaf_nodes: None, min_samples_leaf: 1,		
2	4.587815	0.326380	0.050024	0.009244	None	1	8	(max_leaf_nodes: None, min_samples_leaf: 1,		
3	4.148344	1.038443	0.043753	0.011854	None	1	16	(max_leaf_nodes: None, min_samples_leaf: 1,		
4	4.017285	0.795451	0.058351	0.009479	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: -64230.918
```

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

RMSE: 52.85812511759074
```

In []:

Tuning the model Using GridSearchCV

```
In [31]: from sklearn import preprocessing
from sklearn.model_selection import GridSearchCV, cross_val_score, cross_val_predict
from sklearn import svm
import matplotlib.pyplot as plt
sns.set_style('whitegrid')
sns.set_context('talk')
params = {'legend.fontsize': 'x-large',
          'figure.figsize': (30, 10),
          'axes.labelsize': 'x-large',
          'axes.titlesize': 'x-large',
          'xtick.labelsize': 'x-large',
          'ytick.labelsize': 'x-large'}

plt.rcParams.update(params)
```

```
In [37]: param_grid = {'min_samples_split': [2, 4, 8, 16], 'min_samples_leaf': [1, 2, 3, 4], 'max_leaf_nodes': [None, 10, 20, 100]}

grid_cv_dt = GridSearchCV(model, param_grid, cv=5)

grid_cv_dt.fit(X_train, y_train)

Out[37]: GridSearchCV(cv=5, estimator=DecisionTreeRegressor(),
    param_grid={'max_leaf_nodes': [None, 10, 20, 100],
    'min_samples_leaf': [1, 2, 3, 4],
    'min_samples_split': [2, 4, 8, 16]})
```