

Project Development Phase Model Performance Test

Date	15 November 2022
Team ID	PNT2022TMID09878
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,	<p style="text-align: center;">Evaluating the model</p> <pre> In [33]: from sklearn.metrics import mean_squared_error In [34]: RMLSE=np.sqrt(mean_squared_error(y_test,pred)) RMLSE Out[34]: 209.71961740201198 In [39]: from sklearn import metrics from sklearn.metrics import mean_absolute_error In [40]: MSE=print(metrics.mean_squared_error(y_test,pred)) MSE 43982.31792324628 In [41]: R2S=print(metrics.r2_score(y_test,pred)) R2S 0.6886142448276894 In [42]: MAE=print(mean_absolute_error(y_test,pred)) 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_dtm.best_score_))
print("Best Hyperparameters:\n{}".format(grid_cv_dtm.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	params
0	5.324927	1.095213	0.095589	0.028995	None	1	2	{'max_leaf_nodes': None, 'min_samples_leaf': 1, 'min_samples_split': 2}
1	4.932083	0.489172	0.066534	0.006240	None	1	4	{'max_leaf_nodes': None, 'min_samples_leaf': 1, 'min_samples_split': 4}
2	4.597915	0.326580	0.050324	0.006244	None	1	8	{'max_leaf_nodes': None, 'min_samples_leaf': 1, 'min_samples_split': 8}
3	4.148344	1.038043	0.043753	0.011894	None	1	16	{'max_leaf_nodes': None, 'min_samples_leaf': 1, 'min_samples_split': 16}
4	4.017295	0.795451	0.056361	0.006479	None	2	2	{'max_leaf_nodes': None, 'min_samples_leaf': 2, 'min_samples_split': 2}

```
[4]: 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_dt.bst_estimator.fit(x_train, y_train)
         y_pred = grid_cv_dt.bst_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: 52.85812511759974
```

In []:

Tuning the model Using GridSearchCV

```
In [39]: 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'}
```

[illegible]