

Project Development Phase Model Performance Test

Date	18 November 2022
Team ID	PNT2022TMID45520
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]: RMSLE=np.sqrt(mean_squared_error(y_test,pred)) RMSLE 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)) MAE 89.10334778841495 </pre>

2.	Tune the Model	<div><div><div>Hyperparameter Tuning</div><div>RMSLE- 52.85812511759974</div><div>avg R-squared- 0.123</div><div>MSE: -64230.918</div></div></div>	<div><div><div><div><div>In [38]:</div><div>print("R-Squared:{}".format(grid_cv_dtm.best_score_)) print("Best Hyperparameters:{}".format(grid_cv_dtm.best_params_))</div></div><div><div>R-Squared:0.7601137863085042</div><div>Best Hyperparameters: {'max_leaf_nodes': None, 'min_samples_leaf': 4, 'min_samples_split': 16}</div></div></div><div><div><div>In [39]:</div><div>df = pd.DataFrame(data=grid_cv_dtm.cv_results_) df.head()</div></div><div><div><div>Out[39]:</div><div><table><tr><th></th><th>mean_fit_time</th><th>std_fit_time</th><th>mean_score_time</th><th>std_score_time</th><th>param_max_leaf_nodes</th><th>param_min_samples_leaf</th><th>param_min_samples_split</th><th>params</th></tr><tr><td>0</td><td>5.24627</td><td>1.065213</td><td>0.090588</td><td>0.028895</td><td>None</td><td>1</td><td>2</td><td>{'max_leaf_nodes': None, 'min_samples_leaf': 1}</td></tr><tr><td>1</td><td>4.93283</td><td>0.489172</td><td>0.059534</td><td>0.006248</td><td>None</td><td>1</td><td>4</td><td>{'max_leaf_nodes': None, 'min_samples_leaf': 1}</td></tr><tr><td>2</td><td>4.567815</td><td>0.329380</td><td>0.050024</td><td>0.006244</td><td>None</td><td>1</td><td>8</td><td>{'max_leaf_nodes': None, 'min_samples_leaf': 1}</td></tr><tr><td>3</td><td>4.146344</td><td>1.038443</td><td>0.043753</td><td>0.011694</td><td>None</td><td>1</td><td>16</td><td>{'max_leaf_nodes': None, 'min_samples_leaf': 1}</td></tr><tr><td>4</td><td>4.017265</td><td>0.765451</td><td>0.050551</td><td>0.006478</td><td>None</td><td>2</td><td>2</td><td>{'max_leaf_nodes': None, 'min_samples_leaf': 2}</td></tr></table></div></div></div><div><div><div>In [42]:</div><div>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:{}".format(np.mean(r2_scores))) print("MSE:{}".format(np.mean(mse_scores)))</div></div><div><div>avg R-squared:0.123</div><div>MSE:-64230.918</div></div><div><div><div>In [45]:</div><div>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)))</div></div><div><div>RMSE: 52.85812511759974</div></div><div><div><div>In []:</div><div></div></div><div><div>Tuning the model Using GridSearchCV</div><div><div><div><div>In [33]:</div><div>from sklearn import preprocessing from sklearn.model_selection import GridSearchCV, cross_val_score, cross_val_predict import seaborn as sns 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', 'tick.labelsize': 'x-large', 'tick.labelsize': 'x-large'} plt.rcParams.update(params)</div></div><div><div><div>In [37]:</div><div>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_dtm = GridSearchCV(model, param_grid, cv=5) grid_cv_dtm.fit(X_train, y_train)</div></div><div><div><div>Out[37]:</div><div>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]})</div></div></div></div></div></div></div></div></div></div></div></div></div>		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	5.24627	1.065213	0.090588	0.028895	None	1	2	{'max_leaf_nodes': None, 'min_samples_leaf': 1}	1	4.93283	0.489172	0.059534	0.006248	None	1	4	{'max_leaf_nodes': None, 'min_samples_leaf': 1}	2	4.567815	0.329380	0.050024	0.006244	None	1	8	{'max_leaf_nodes': None, 'min_samples_leaf': 1}	3	4.146344	1.038443	0.043753	0.011694	None	1	16	{'max_leaf_nodes': None, 'min_samples_leaf': 1}	4	4.017265	0.765451	0.050551	0.006478	None	2	2	{'max_leaf_nodes': None, 'min_samples_leaf': 2}
	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	5.24627	1.065213	0.090588	0.028895	None	1	2	{'max_leaf_nodes': None, 'min_samples_leaf': 1}																																																	
1	4.93283	0.489172	0.059534	0.006248	None	1	4	{'max_leaf_nodes': None, 'min_samples_leaf': 1}																																																	
2	4.567815	0.329380	0.050024	0.006244	None	1	8	{'max_leaf_nodes': None, 'min_samples_leaf': 1}																																																	
3	4.146344	1.038443	0.043753	0.011694	None	1	16	{'max_leaf_nodes': None, 'min_samples_leaf': 1}																																																	
4	4.017265	0.765451	0.050551	0.006478	None	2	2	{'max_leaf_nodes': None, 'min_samples_leaf': 2}																																																	

