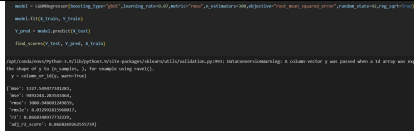



Project Development Phase Model Performance Test

Date	17 November 2022
Team ID	PNT2022TMID25623
Project Name	Project – Car Resale Value Prediction
Maximum Marks	10 Marks

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Metrics	Regression Model: LGBM Regressor MAE: 1327.56 MSE: 9492244.25 RMSE: 3080.93 RMSLE: 8.05 R2 Score: 0.8664 Adjusted R2 Score: 0.8666	 <pre> model = LGBMRegressor(n_estimators=300, learning_rate=0.07, booster='gbdt', objective='reg', num_leaves=31, min_child_samples=20, min_child_weight=0.01, min_split_gain=0.1, min_weight=1, n_jobs=-1, random_state=42, reg_alpha=0.001, reg_lambda=1.0) model.fit(X_train, y_train) # Predict on training data y_train_pred = model.predict(X_train) # Predict on test data y_test_pred = model.predict(X_test) # Calculate metrics from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score, adjusted_r2_score mae = mean_absolute_error(y_train, y_train_pred) mse = mean_squared_error(y_train, y_train_pred) rmse = np.sqrt(mse) rmsle = np.sqrt(mean_squared_error(y_train, y_train_pred) / (y_train ** 2)) r2 = r2_score(y_train, y_train_pred) adj_r2 = adjusted_r2_score(y_train, y_train_pred) print(f'MAE: {mae}, MSE: {mse}, RMSE: {rmse}, RMSLE: {rmsle}, R2 Score: {r2}, Adjusted R2 Score: {adj_r2}') </pre>
2.	Tune the Model	Hyperparameter Tuning 1) Learning Rate: [0.01, 0.03, 0.05, 0.07] 2) Boosting Type: ['gbdt', 'dart', 'goss', 'rf'] 3) Number of Estimators: [100, 200, 300] Validation Method: Grid Search Cross Validation Best Parameters: Learning Rate – 0.07 Boosting Type – 'gbdt' Number of Estimators - 300	 <pre> lgbm_config = { "name": "LGBMRegressor", "method": "grid", "metric": { "name": "adj_r2", "goal": "maximize" }, "parameters": { "learning_rate": { "values": [0.01, 0.03, 0.05, 0.07] }, "objective": { "values": ['root_mean_squared_error'] }, "boosting_type": { "values": ['gbdt', 'dart', 'goss', 'rf'] }, "reg_sqrt": { "values": [True] }, "metric": { "values": ['rmse'] }, "n_estimators": { "values": [100, 200, 300] }, "random_state": { "values": [42] } } } </pre> <p>The plot below shows the results of a grid search for the LGBMRegressor model. The x-axis represents the number of estimators (100, 200, 300), and the y-axis represents the adjusted R-squared score (adj_r2). The plot shows that the model's performance is highest when the number of estimators is 300, with an adj_r2 score of approximately 0.8664. The plot also shows that the model's performance is relatively stable across different boosting types (gbdt, dart, goss, rf) and learning rates (0.01, 0.03, 0.05, 0.07).</p>

Screenshots

1) Metrics

```
model = LGBMRegressor(boosting_type="gbdt",learning_rate=0.07,metric="rmse",n_estimators=300,objective="root_mean_squared_error",random_state=42,reg_sqrt=True)

model.fit(X_train, Y_train)

Y_pred = model.predict(X_test)

find_scores(Y_test, Y_pred, X_train)

/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/sklearn/utils/validation.py:993: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please use the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)

{'mae': 1327.549477341283,
 'mse': 9492244.283543464,
 'rmse': 3080.948601249859,
 'rmsle': 8.032992815968017,
 'r2': 0.8668348937732229,
 'adj_r2_score': 0.8668269262555739}
```

2) Tune the model

```
lgbm_configs = {
    "name": 'LGBMRegressor',
    "method": "grid",
    "metric": {
        "name": "adj_r2",
        "goal": "maximize"
    },
    "parameters": {
        "learning_rate": {
            "values": [0.01, 0.03, 0.05, 0.07]
        },
        "objective": [
            "values": ['root_mean_squared_error']
        ],
        "boosting_type": {
            "values": ['gbdt', 'dart', 'goss', 'rf']
        },
        "reg_sqrt": {
            "values": [True]
        },
        "metric": {
            "values": ['rmse']
        },
        "n_estimators": {
            "values": [100, 200, 300]
        },
        "random_state": {
            "values": [42]
        }
    }
}
```

Wandb Sweep :

