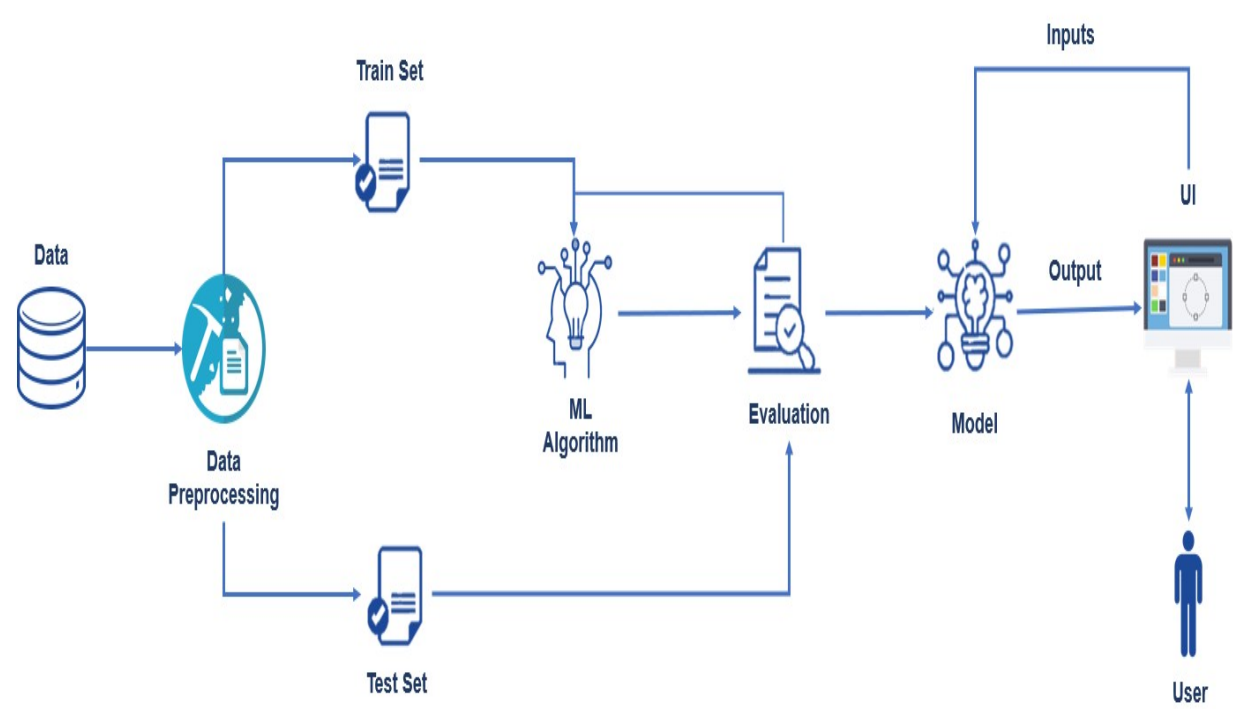




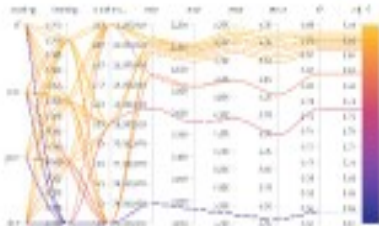
Performance Metrics



# Model Performance Testing

## Model Performance Testing:

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

S.No.	Parameter	Values	Screenshot
1.	Metrics	<b>Regression Model: LGBM Regressor</b>  MAE: 1327.55 MSE: 9492244.28 RMSE: 3080.95 RMSLE: 8.03 R2 Score: 0.8668 Adjusted R2 Score: 0.8668	
2.	Tune the Model	<b>Hyperparameter Tuning</b> 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]  <b>Validation Method:</b> Grid Search Cross Validation  <b>Best Parameters:</b> Learning Rate – 0.07 Boosting Type – 'gbdt' Number of Estimators - 300	 

## By applying train test split

### Applying Train Test Split

```
In [35]: from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)
```

```
In [74]: from sklearn.linear_model import LinearRegression
```

```
In [75]: from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import make_column_transformer
from sklearn.pipeline import make_pipeline
from sklearn.metrics import r2_score
```

### Creating an OneHotEncoder object to contain all the possible categories

```
In [39]: ohe=OneHotEncoder()
ohe.fit(X[['name','company','fuel_type']])
```

```
Out[39]: OneHotEncoder()
```

### Creating a column transformer to transform categorical columns

```
In [52]: column_trans=make_column_transformer((OneHotEncoder(categories=ohe.categories_),['name','company','fuel_type']),
remainder='passthrough')
```

### Linear Regression Model

```
In [54]: lr=LinearRegression()
```

### Making a pipeline

```
In [55]: pipe=make_pipeline(column_trans,lr)
```

### Fitting the model

```
In [59]: pipe.fit(X_train,y_train)
```

## Checking R2 Score

### Checking R2 Score

```
In [61]: r2_score(y_test,y_pred)
```

```
Out[61]: 0.76274562376113
```

### Finding the model with a random state of TrainTestSplit where the model was found to give almost 0.92 as r2\_score

```
In [62]: scores=[]
for i in range(1000):
    X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.1,random_state=i)
    lr=LinearRegression()
    pipe=make_pipeline(column_trans,lr)
    pipe.fit(X_train,y_train)
    y_pred=pipe.predict(X_test)
    scores.append(r2_score(y_test,y_pred))
```

```
In [63]: np.argmax(scores)
```

```
Out[63]: 655
```

```
In [64]: scores[np.argmax(scores)]
```

```
Out[64]: 0.920088412025344
```

```
In [65]: pipe.predict(pd.DataFrame(columns=X_test.columns,data=np.array(['Maruti Suzuki Swift','Maruti',2019,100,'Petrol']).reshape(1,5)))
```

```
Out[65]: array([400707.28215338])
```

### The best model is found at a certain random state

```
In [67]: X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.1,random_state=np.argmax(scores))
lr=LinearRegression()
pipe=make_pipeline(column_trans,lr)
pipe.fit(X_train,y_train)
y_pred=pipe.predict(X_test)
r2_score(y_test,y_pred)
```

```
Out[67]: 0.920088412025344
```

## 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)

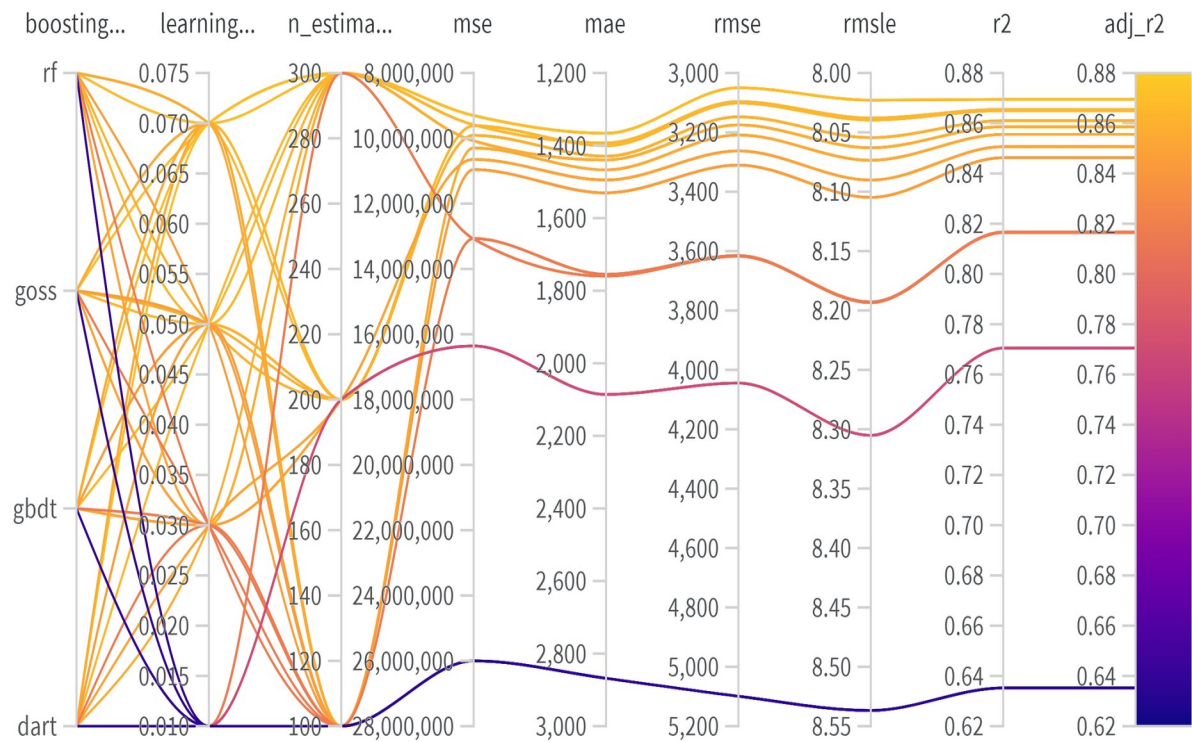
/opt/conda/envs/python-3.9/lib/python3.9/site-packages/sklearn/utils/validation.py:999: DataConversionWarning: A column-vector y was passed when a 1d array was expected. The shape of y is (n_samples, 1), for example using y.ravel().
  y = column_or_1d(y, warn=True)

{'mse': 1327.548477341383,
 'mae': 9492344.283543464,
 'rmse': 36430.486801349059,
 'rmse_l1': 8.032993815668017,
 'r2': 0.8668148917732209,
 'adj_r2_score': 0.8846262262553739}
```

## Tuning 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]
        }
    }
}
```

# LGBM Regressor



# Importing model on IBM :

IBM Cloud

Search resources and products...

Catalog Manage Mugundan P's Account

## Dashboard

For you

**Build**

Explore IBM Cloud with this selection of easy starter tutorials and services.

**Build a web app with Watson Speech to Text**

Deploy a conversational interface compatible with any application, device, or channel.

Getting started 15 min

**Get Started with Watson Studio**

Get started with using AI and Cloud Object Storage in 15 minutes.

Popular 2 hr

**Build a virtual machine**

Lift and shift your VMware workloads to the IBM Cloud.

Getting started 7 min

**Unlock the entire catalog**

Upgrade your account to access Virtual Servers, Baremetal, and other infrastructure resources.

Recommended 10 min

**Best practice**

Review for provisioning and account service.

Recommended

**User access**

Manage users

Enter email addresses below to jump directly into the invite user setup:

Enter up to 100 email addresses

**News**

View all

- IBM Cloud Satellite New Pricing
- IBM Cloud Data Shield Deprecation
- IBM Watson Orchestrate Is Integrating with ThisWay Global
- SLSA Support in IBM Cloud Continuous Delivery

**Planned maintenance**

View all