Project Development Phase
Model Performance Test

Date	19 November 2022
Team ID	PNT2022TMID20841
Project Name	Demand Est-AI Powered Food Demand forecaster
Maximum Marks	10 Marks

Model Performance Testing:

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

Metrics:

- **Regression Model:**

>> Decision Tree:

Values :

```
r2_score : 0.6371233948056725
MSE : 59209.67667463429
MAE : 106.73730003941613
RMSLE: 62.62626825772612
```

Screenshot :

```
In [57]: DT=DecisionTreeRegressor()
         DT.fit(x_train,y_train)
         y_pred=DT.predict(x_val)
         y_pred[y_pred<0]=0
         from sklearn import metrics
         print("r2_score :",r2_score(y_val,y_pred))
         print("MSE :",mean_squared_error(y_val,y_pred))
         print("MAE :",mean_absolute_error(y_val,y_pred))
         print('RMSLE:',100*np.sqrt(metrics.mean_squared_log_error(y_val,y_pred)))

         r2_score : 0.6371233948056725
         MSE : 59209.67667463429
         MAE : 106.73730003941613
         RMSLE: 62.62626825772612
```

>> KNN Model:

Values :

```
r2_score : 0.5943921174379555
MSE : 60606.75302417271
MAE : 116.60727721948186
RMSLE: 67.18176682966069
```

Screenshot :

In [40]:
```
KNN=KNeighborsRegressor()
KNN.fit(x_train,y_train)
y_pred=KNN.predict(x_val)
y_pred[y_pred<0]=0
from sklearn.metrics import r2_score,mean_squared_error,mean_absolute_error
print("r2_score :",r2_score(y_val,y_pred))
print("MSE :",mean_squared_error(y_val,y_pred))
print("MAE :",mean_absolute_error(y_val,y_pred))
print('RMSLE:',100*np.sqrt(metrics.mean_squared_log_error(y_val,y_pred)))
```

```
r2_score : 0.5943921174379555
MSE : 60606.75302417271
MAE : 116.60727721948186
RMSLE: 67.18176682966069
```

>> Gradient Boosting :

Values :

```
r2_score : 0.5590226511988423
MSE : 65891.7304546171
MAE : 130.064989280476
RMSLE: 98.64508400521507
```

Screenshot :

In [41]:
```
GB=GradientBoostingRegressor()
GB.fit(x_train,y_train)
y_pred=GB.predict(x_val)
y_pred[y_pred<0]=0
from sklearn.metrics import r2_score,mean_squared_error,mean_absolute_error
print("r2_score :",r2_score(y_val,y_pred))
print("MSE :",mean_squared_error(y_val,y_pred))
print("MAE :",mean_absolute_error(y_val,y_pred))
print('RMSLE:',100*np.sqrt(metrics.mean_squared_log_error(y_val,y_pred)))
```

```
r2_score : 0.5590226511988423
MSE : 65891.7304546171
MAE : 130.064989280476
RMSLE: 98.64508400521507
```

>> Lasso Model :

Values :

```
r2_score : 0.17888193284282972
MSE : 122693.12811559634
MAE : 206.15594866328124
RMSLE: 129.9773789182497
```

Screenshot :

```
In [43]: L=Lasso()
         L.fit(x_train,y_train)
         y_pred=LR.predict(x_val)
         y_pred[y_pred<0]=0
         from sklearn.metrics import r2_score,mean_squared_error,mean_absolute_error
         print("r2_score :",r2_score(y_val,y_pred))
         print("MSE :",mean_squared_error(y_val,y_pred))
         print("MAE :",mean_absolute_error(y_val,y_pred))
         print('RMSLE:',100*np.sqrt(metrics.mean_squared_log_error(y_val,y_pred)))

         r2_score : 0.17888193284282972
         MSE : 122693.12811559634
         MAE : 206.15594866328124
         RMSLE: 129.9773789182497
```

>> ElasticNet Model :

Values :

```
r2_score : 0.08920905556590097
MSE : 136092.2314361745
MAE : 213.1551819363743
RMSLE: 131.24272418094336
```

Screenshot :

In [42]:
```
EN=ElasticNet()
EN.fit(x_train,y_train)
y_pred=EN.predict(x_val)
y_pred[y_pred<0]=0
from sklearn.metrics import r2_score,mean_squared_error,mean_absolute_error
print("r2_score :",r2_score(y_val,y_pred))
print("MSE :",mean_squared_error(y_val,y_pred))
print("MAE :",mean_absolute_error(y_val,y_pred))
print('RMSLE:',100*np.sqrt(metrics.mean_squared_log_error(y_val,y_pred)))
```

```
r2_score : 0.08920905556590097
MSE : 136092.2314361745
MAE : 213.1551819363743
RMSLE: 131.24272418094336
```

>> Linear Regression :

Values :

```
r2_score : 0.17888193284282972
MSE : 122693.12811559634
MAE : 206.15594866328124
RMSLE: 129.9773789182497
```

Screenshot :

In [44]:
```
LR = LinearRegression()
LR.fit(x_train,y_train)
y_pred = LR.predict(x_val)
y_pred[y_pred<0]=0
from sklearn.metrics import r2_score,mean_squared_error,mean_absolute_error
print("r2_score :",r2_score(y_val,y_pred))
print("MSE :",mean_squared_error(y_val,y_pred))
print("MAE :",mean_absolute_error(y_val,y_pred))
print('RMSLE:',100*np.sqrt(metrics.mean_squared_log_error(y_val,y_pred)))
```

```
r2_score : 0.17888193284282972
MSE : 122693.12811559634
MAE : 206.15594866328124
RMSLE: 129.9773789182497
```

- **HyperParameter Tuning :**

In [48]:
```
dt = DecisionTreeRegressor(max_depth=3)
dt.fit(x_train, y_train)
```

Out[48]:
```
DecisionTreeRegressor(max_depth=3)
```

In [51]:
```
from sklearn import tree
fig = plt.figure(figsize=(25,20))
_ = tree.plot_tree(dt,
                   feature_names=trainfinal.columns,
                   class_names=['No Demand', "Demand"],
                   filled=True)
```