Model Building

Team ID	PNT2022TMID35762
Project Name	DEMANDEST – AI POWERED FOOD DEMAND FORECASTER

Training and testing algorithmic models:

Data can be processed using a variety of Machine Learning models, including images, sound, text and numerical values. Algorithms can be chosen according to the objective. They can be either Classification models or Regression models.

- Linear Regression Model
- Lasso Regression model
- Elastic Net Regression model
- XGB Regressor model
- Gradient Boosting Model
- Decision Tree
- K-Nearest Neighbour

```
In [85]: # Split the dataset into train and test data

         X = final_train3.values
         Y = final_train['num_orders'].values
         X_train,X_val,Y_train,Y_val = train_test_split(X,Y,test_size=0.25)
```

Model evaluation:

In Machine Learning, mathematical models are constructed to make sense of the data at hand. By fitting these models to previously observed data, new data can be predicted.

In [86]:
```
# Linear Regression Model

LR_mod = LinearRegression()
LR_mod.fit(X_train,Y_train)
Y_pred = LR_mod.predict(X_val)
Y_pred[Y_pred<0] = 0

# Root Mean Square Error

RMSE = 100*np.sqrt(metrics.mean_squared_log_error(Y_val,Y_pred))
RMSE
```

Out[86]: 129.70651046086593

In [87]:
```
# Lasso Regression model

L_mod = Lasso()
L_mod.fit(X_train,Y_train)
Y_pred = L_mod.predict(X_val)
Y_pred[Y_pred<0] = 0

# Root Mean Square Error
RMSE = 100*np.sqrt(metrics.mean_squared_log_error(Y_val,Y_pred))
RMSE
```

Out[87]: 129.20484156819074

In [90]:
```
# Decision Tree

Dec_tree = DecisionTreeRegressor()
Dec_tree.fit(X_train,Y_train)
Y_pred = Dec_tree.predict(X_val)
Y_pred[Y_pred<0] = 0

# Root Mean Square Error
RMSE = 100*np.sqrt(metrics.mean_squared_log_error(Y_val,Y_pred))
RMSE
```

Out[90]: 62.89494154472869

In [96]:
```
# K-Nearest Neighbor

KNN = KNeighborsRegressor()
KNN.fit(X_train,Y_train)
Y_pred = KNN.predict(X_val)
Y_pred[Y_pred<0] = 0

# Root Mean Square Error
RMSE = 100*np.sqrt(metrics.mean_squared_log_error(Y_val,Y_pred))
RMSE
```

Out[96]: 67.24720843560212

Save the model:

To test the machine learning model on new data, save the trained model to a file and restore them.

In [109]:
```
# To store the Decision Tree model

pickle.dump(Dec_tree,open('demand_forecast.pkl','wb'))
```

Using the model to predict the output:

In [114]:
```
# Predicting the number of orders using Decision Tree Model

final_test = pd.merge(test_data,food_data,on="meal_id",how="outer")
final_test = pd.merge(final_test,centers_data,on="center_id",how="outer")
final_test = final_test.drop(['center_id','meal_id'],axis=1)


fcolumns = final_test.columns.tolist()
fcolumns = fcolumns[:2]+fcolumns[8:]+fcolumns[6:8]+fcolumns[2:6]
final_test = final_test[fcolumns]

le = LabelEncoder()
final_test['center_type'] = le.fit_transform(final_test['center_type'])
final_test['category'] = le.fit_transform(final_test['category'])
final_test['cuisine'] = le.fit_transform(final_test['cuisine'])

X_test = final_test[features].values
obser = Dec_tree.predict(X_test)
obser[obser<0] = 0
result = pd.DataFrame({'id':final_test['id'],'num_orders':obser})

result.to_csv("C:/Users/91948/Downloads/IBM_PROJECT/Datasets/submission.csv",index=False)
result.describe()
```

Out[114]:

	id	num_orders
count	3.257300e+04	32573.000000
mean	1.248476e+06	262.516689