

## Model Building

Team id	PNT2022TMID45520
Project name	DemandEst - AI powered Food Demand Forecaster

### Train And Test Model Algorithms:

There are several Machine learning algorithms to be used depending on the data you are going to process such as images, sound, text, and numerical values. The algorithms that you can choose according to the objective that you might have it may be Classification algorithms or Regression algorithms.

1. Linear Regression.
2. Lasso Regression.
3. ELasticNet Regression / Classification.
4. Decision Tree Regression / Classification.

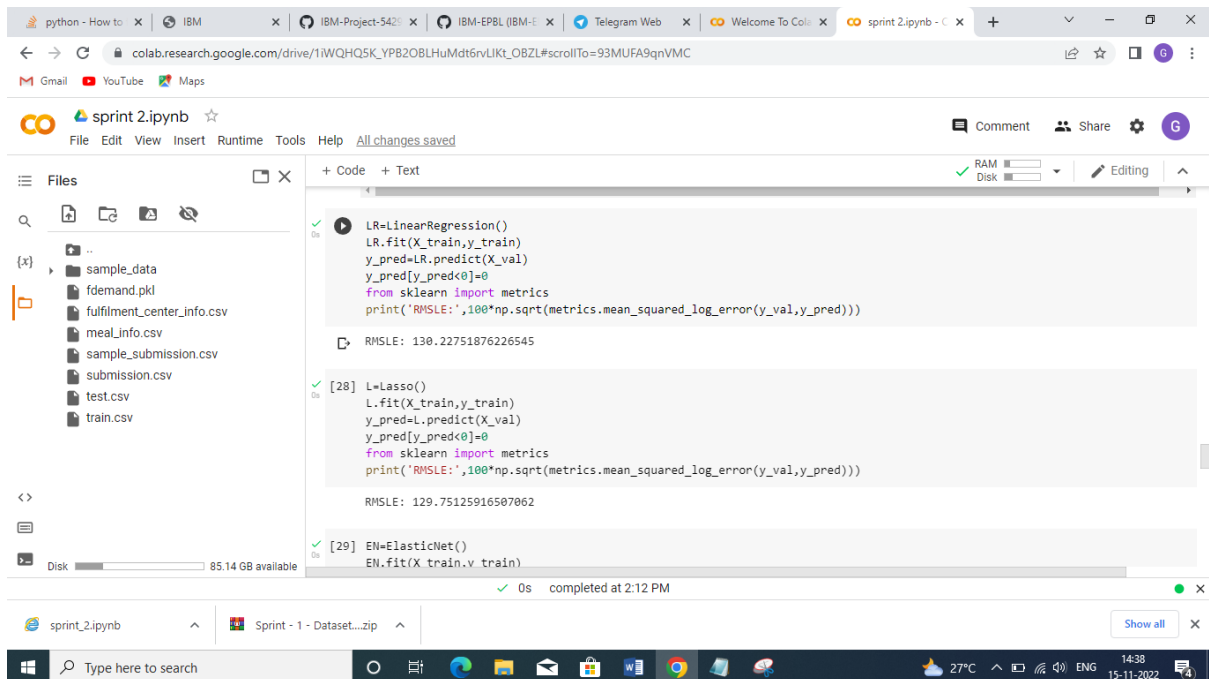
The screenshot shows a Jupyter Notebook interface. On the left is a file explorer with a tree view containing the following files and folders: `..`, `sample_data`, `fdemand.pkl`, `fulfilment_center_info.csv`, `meal_info.csv`, `sample_submission.csv`, `submission.csv`, `test.csv`, and `train.csv`. The main area is divided into two sections: a table and code cells. The table has 9 columns and 3 rows of data. The code cells contain Python code for importing libraries and splitting data.

2	0	0	2.0	3	647	56	0
3	0	0	2.0	3	647	56	0
4	0	0	2.0	3	647	56	0

```
[24] from sklearn.model_selection import train_test_split
      X_train,X_val,y_train,y_val=train_test_split(X,y,test_size=0.25)

[25] from sklearn.linear_model import LinearRegression
      from sklearn.linear_model import Lasso
      from sklearn.linear_model import ElasticNet
      from sklearn.tree import DecisionTreeRegressor
      from sklearn.neighbors import KNeighborsRegressor
      from sklearn.ensemble import GradientBoostingRegressor
      from xgboost import XGBRegressor
```

## Model Evaluation :



## Save The Model:

```
[ ] import pickle
    pickle.dump(DT,open('fdemand.pkl','wb'))
```

```
[ ] testfinal=pd.merge(test,meal_info,on="meal_id",how="outer")
    testfinal=pd.merge(testfinal,center_info,on="center_id",how="outer")
    testfinal=testfinal.drop(['meal_id','center_id'],axis=1)

    tcols=testfinal.columns.tolist()
    tcols=tcols[:2]+tcols[8:]+tcols[6:8]+tcols[2:6]
    testfinal=testfinal[tcols]
```

## Predicting The Output Using The Model:

```
[ ] testfinal=pd.merge(test,meal_info,on="meal_id",how="outer")
    testfinal=pd.merge(testfinal,center_info,on="center_id",how="outer")
    testfinal=testfinal.drop(['meal_id','center_id'],axis=1)

    tcols=testfinal.columns.tolist()
    tcols=tcols[:2]+tcols[8:]+tcols[6:8]+tcols[2:6]
    testfinal=testfinal[tcols]

    lb1=LabelEncoder()
    testfinal['center_type']=lb1.fit_transform(testfinal['center_type'])

    lb2=LabelEncoder()
    testfinal['category']=lb1.fit_transform(testfinal['category'])

    lb3=LabelEncoder()
    testfinal['cuisine']=lb1.fit_transform(testfinal['cuisine'])

    X_test=testfinal[features].values
```