**JOB-A-THON - February 2022**

**APPROACH**

1. I have targeted the problem as a regression analysis and used Xgboost regressor machine learning algorithm with RandomizedSearchCV to build my final model.
2. For data pre-processing I checked for missing values, looked for correlation in problem, Using label Encoder converted Categorical data into numeric form.
3. Before reaching my final model, I have used Random forest and Gradient boosting algorithms, however Xgboost gave a bit better results as compared.

Fine tuning and using extracted column played a crucial role in getting a descent model.

# Number of trees in random forest

n\_estimators = [int(x) for x in np.linspace(start = 200, stop = 2000, num = 10)]

# Maximum number of levels in tree

max\_depth = [int(x) for x in np.linspace(10, 110, num = 11)]

max\_depth.append(None)

# Create the params

params = { 'max\_depth': max\_depth,

'learning\_rate': [0.01, 0.1, 0.2, 0.3],

'subsample': np.arange(0.5, 1.0, 0.1),

'colsample\_bytree': np.arange(0.4, 1.0, 0.1),

'colsample\_bylevel': np.arange(0.4, 1.0, 0.1),

'n\_estimators': n\_estimators}

xgbr = xgb.XGBRegressor(seed = 20)

xgb\_rcv = RandomizedSearchCV(estimator=xgbr,

param\_distributions=params,

scoring='neg\_mean\_squared\_error',

n\_iter=25,

verbose=1)Local accuracy (using neg\_mean\_squared\_log\_error and cross validation): mean: -0.09945, std: 0.05365

**Name :** Mohammed Sohail J

**Mobile** **no. :** 9489443786

**E-mail :** mohammedsohail.j@gmail.com