TEAM ID: PNT2022TMID43603

PROJECT NAME: Demand Est - AI powered Food Demand Forecaster

© IBM X 3 Home Page - Select or create a X 2 Code - Jupyter Notebook X + v — QI X

C © localhost:8891/notebooks/Downloads/SBSPS-Challenge-8325-Food-Demand-Forecasting-for-Food-Delivery-Company-using-IBM-Cloud-main/SBSPS-Challenge-... \£ Di ^i :

M Gmail O YouTube %Maps

^ JUpyter Code (autosaved) f® Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted | Python 3 (ipykemel) ^

EI + ® CI ♦ * ► Run ■ C ►► Markdown v Q

Predicting The Output Using The Model

Here, we are creating X_test which we are using to test the model to predict the number of orders by giving input to the model build.

```
In [133]: testfinal = pd.merge(test, meal_info, on="meal_id", how="outer")
testfinal = pd.merge(testfinal, fulfilment_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]

Ibl = LabelEncoderQ
testfinal['center_type'] = Ibl.fit_transform(testfinal['center_type'])

Ib2 = LabelEncoderQ
testfinal['category'] = Ibl.fit_transform(testfinal['category'])

Ib3 = LabelEncoderQ
testfinal['cuisine'] = Ibl.fit_transform(testfinal['cuisine'])

X_test = testfinal[features].values

In [134]: pred = DT.predict(X_test) pred[pred<0] = 0 submit = pd.DataFrame({
'id' : testfinal['id'],
'num_orders' : pred_
```

© IBM X ^ Home Page - Select or create a X 2 Code - Jupyter Notebook X + — O X

C © localhost:8891/notebooks/Downloads/SBSPS-Challenge-8325-Food-Demand-Forecasting-for-Food-Delivery-Company-using-IBM-Cloud-main/SBSPS-Challenge-... \£ Q ^ :

M Gmail O YouTube Maps

^ jupyter CodG (autosaved) |® Logout

Not Trusted | Python 3 (ipykernel)

File Edit View Insert Cell Kernel Widgets Help E) + % * * ►Run BO* Markdown '

```
In [134]: |pred = DT.predict(X_test) pred[pred<0] = 0 submit =
          pd.DataFrame({
              'id' : testfinal['id'],
              'num_orders' : pred

          })

          Submit the predicted output values(Number of orders) to

"submission.csV In [135]: submit.to_csv("submission.csv, index=Flase")

In [136]: submit.describeQ
```

	id	num_orders
count	3 257300e+04	32573.000000
mean	1.248476e+06	263.114244
Std	1 441580e+05	367.092916
min	1 000085e+06	14.666667
25%	1 123969e+06	64.113281
50%	1 247296e+06	147.022222
75%	1.372971e+06	324.133333
max	1 499996e+06	6174.850000

Team Member 1

© IBM X | ^ Home Page - Select or create a X Code - Jupyter Notebook X + v 0 X

C © localhost:8891/notebooks/Downloads/SBSPS-Challenge-8325-Food-Demand-Forecasting-for-Food-Delivery-Company-using-IBM-Cloud-main/SBSPS-Challenge-... \£ ☆ □ ^ :

M Gmail O YouTube Maps

Q IBM X | ^ Home Page - Select or create a X ^ Code - Jupyter Notebook X + 0 X

C © localhost:8891/notebooks/Downloads/SBSPS-Challenge-8325-Food-Demand-Forecasting-for-Food-Delivery-Company-using-IBM-Cloud-main/SBSPS-Challenge-... \&~ff Di ^ :

M Gmail O YouTube ff Maps

^ JUpyter Code (autosaved) Logout

| Not Trusted | Python 3 (ipykernel) °o

File Edit View Insert Cell Kernel Widgets Help [~B~] [+ ft fc ♦ + ► Run ■ C » Markdown ˅

E3

In [134]:
```
pred = DT.predict(X_test)
pred[pred<0] = 0 submit =
pd.DataFrame({
'id'       :       testfinalfid'],
'num_orders' : pred
})
```

Submit the predicted output values(Number of orders) to "submission.csv"

In [135]:
```
submit.to_csv("submission.csv, index-Flase")
```

In [136]:
```
jsubmit.describe()
```

0ut[136]:

	id	num_orders	count	3 257300e+04
	32573.000000	mean	1.248476e+06	
	263 114244			
	std	1.441580e'05		3
	min	1.000085e+06	67.092916	
	25%	1.123969e+06	14.666667	
	50%	1.247296e+06	64.113281	
	75%	1.372971 e+06	147.02222	
		2	max 1 499996e+06	6174 850000
		324.13333		
		3		

Team Member 2