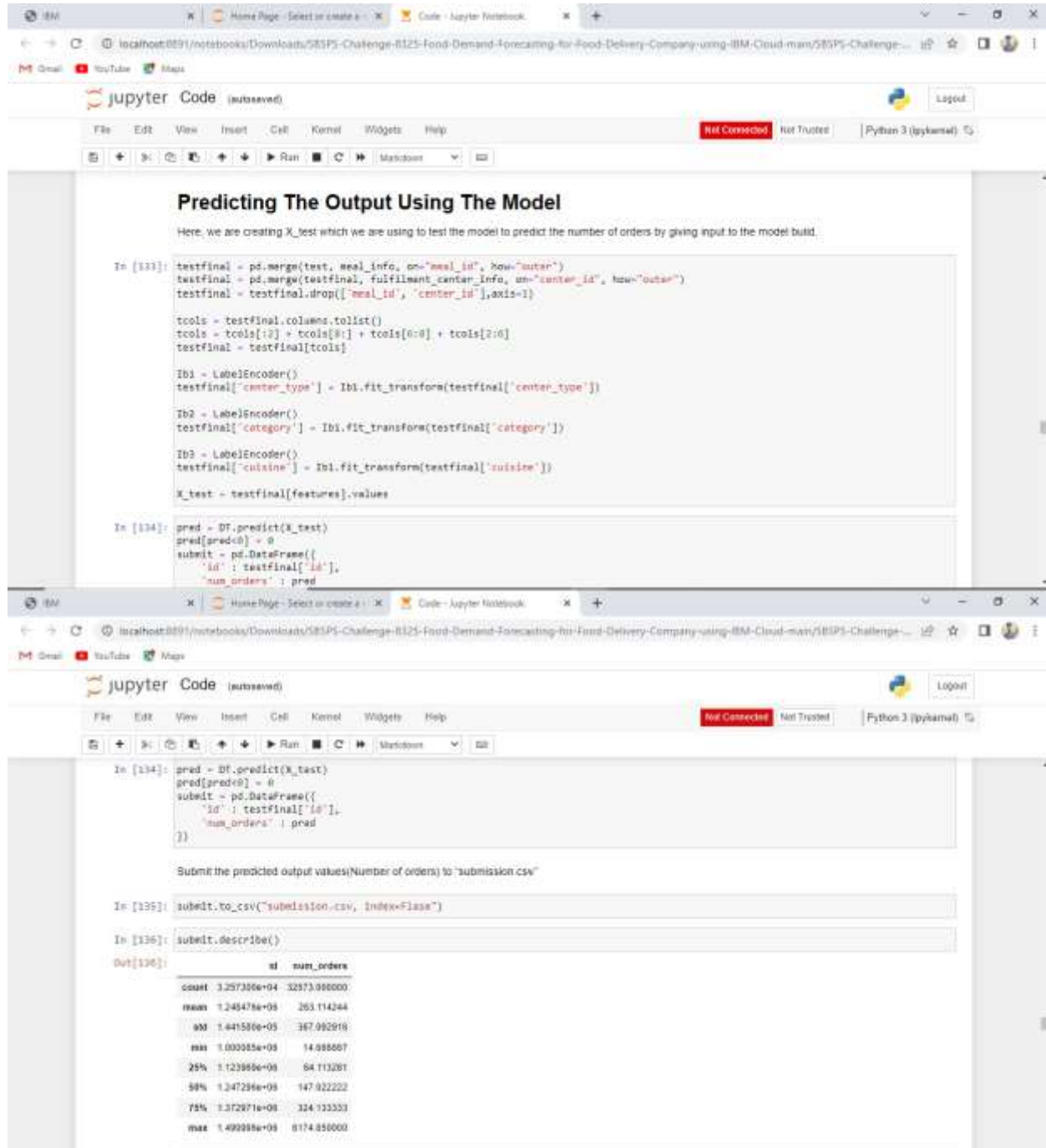


TEAM ID: PNT2022TMID32368

PROJECT NAME: DemandEst - AI powered Food Demand Forecaster

Team Leader



```
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, fulfillment_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'] = lb2.fit_transform(testfinal['category'])  
  
lb3 = LabelEncoder()  
testfinal['cuisine'] = lb3.fit_transform(testfinal['cuisine'])  
  
X_test = testfinal[features].values  
  
In [134]: pred = DF.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"

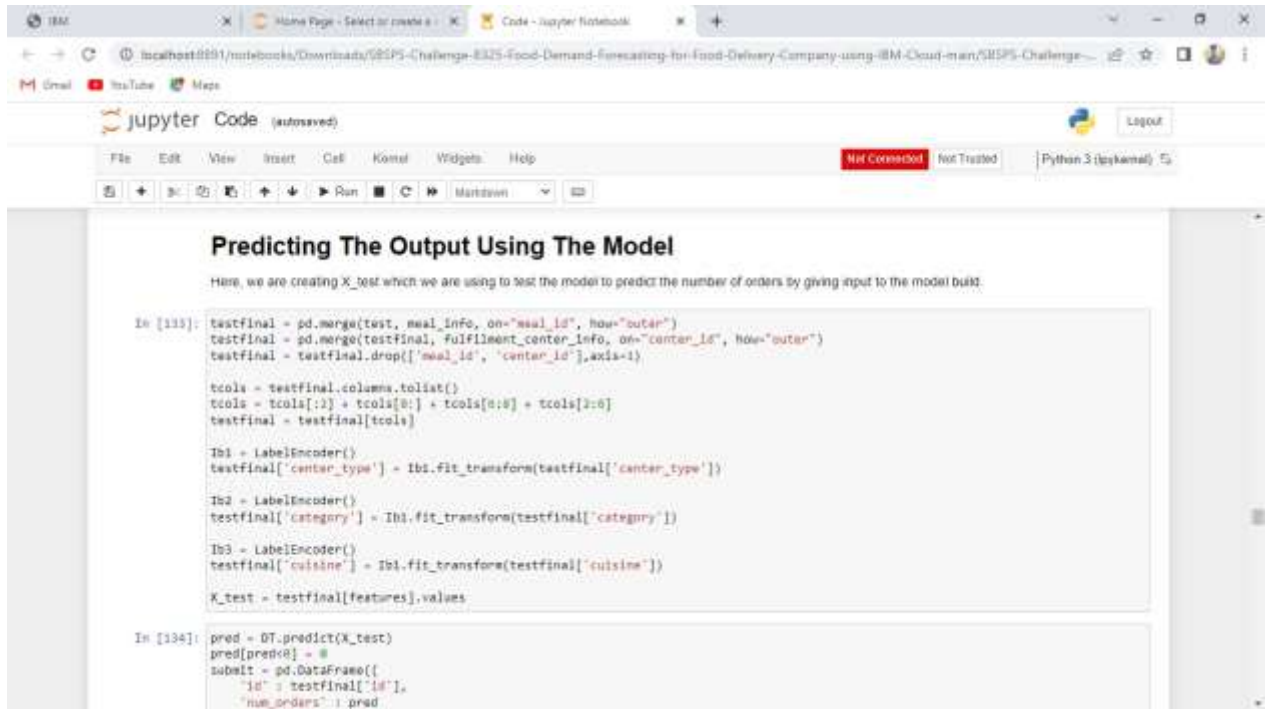
```
In [135]: submit.to_csv("submission.csv", index=False)
```

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

```
Out[136]:
```

	id	num_orders
count	3.257300e+04	32573.000000
mean	1.345475e+05	263.114244
std	1.441500e+05	367.092918
min	1.000000e+05	14.888667
25%	1.123960e+05	64.113281
50%	1.347284e+05	147.922222
75%	1.372971e+05	324.133333
max	1.499999e+05	8174.850000

Team Member 1



The screenshot shows a Jupyter Notebook interface with the title "Predicting The Output Using The Model". The text below the title states: "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 built".

```
In [133]: testfinal = pd.merge(test, meal_info, on="meal_id", how="outer")
testfinal = pd.merge(testfinal, fulfillment_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[9:] + tcols[10:]
testfinal = testfinal[tcols]

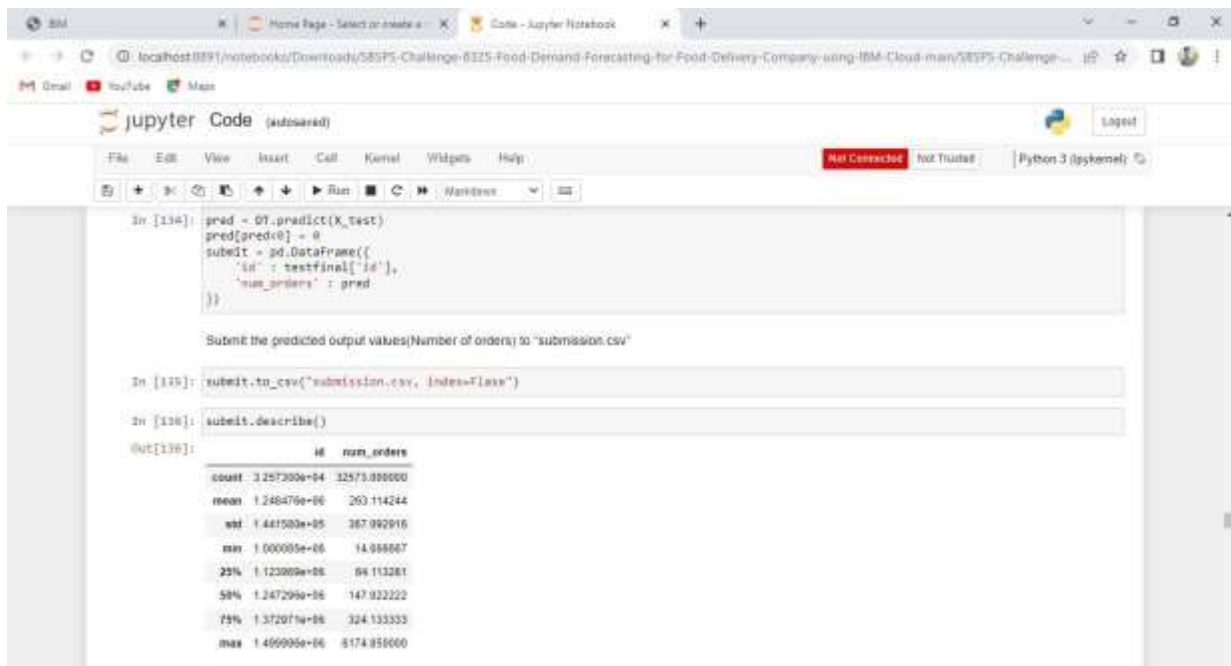
Ib1 = LabelEncoder()
testfinal['center_type'] = Ib1.fit_transform(testfinal['center_type'])

Ib2 = LabelEncoder()
testfinal['category'] = Ib2.fit_transform(testfinal['category'])

Ib3 = LabelEncoder()
testfinal['cuisine'] = Ib3.fit_transform(testfinal['cuisine'])

X_test = testfinal[features].values

In [134]: pred = DT.predict(X_test)
pred[pred<0] = 0
subdf1 = pd.DataFrame({
    'id': testfinal['id'],
    'num_orders': pred
})
```



The screenshot shows the continuation of the Jupyter Notebook. The code in cell [134] is repeated, followed by instructions to submit the predicted output values (Number of orders) to "submission.csv".

```
In [134]: pred = DT.predict(X_test)
pred[pred<0] = 0
subdf1 = pd.DataFrame({
    'id': testfinal['id'],
    'num_orders': pred
})

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

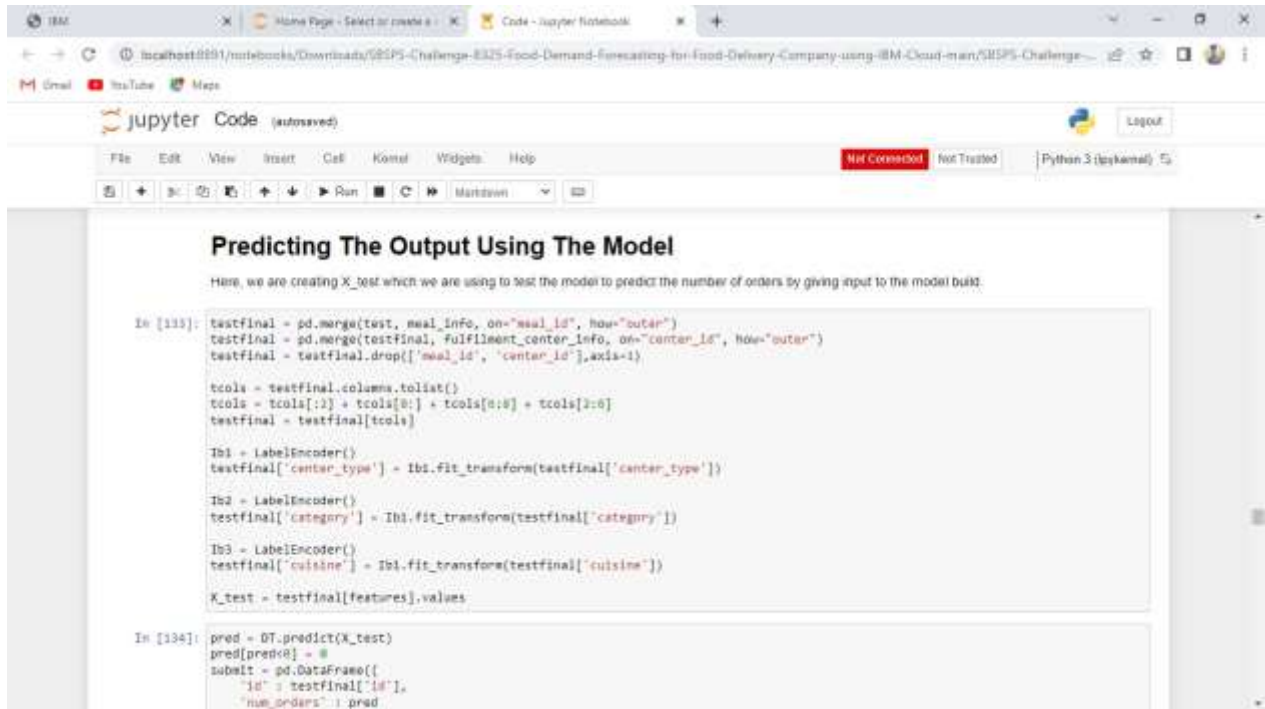
In [135]: submit.to_csv("submission.csv", index=False)

In [136]: submit.describe()

Out[136]:
```

	id	num_orders
count	3.257300e+04	32573.000000
mean	1.248476e+06	260.114244
std	1.441503e+05	267.992916
min	1.000000e+05	14.988667
25%	1.123089e+06	84.113281
50%	1.247299e+06	147.932222
75%	1.372074e+06	324.133333
max	1.466906e+06	8174.850000

Team Member 2



The screenshot shows a Jupyter Notebook interface with the title "Predicting The Output Using The Model". The text below the title states: "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 built".

```
In [133]: testfinal = pd.merge(test, meal_info, on="meal_id", how="outer")
testfinal = pd.merge(testfinal, fulfillment_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[9:] + tcols[10:]
testfinal = testfinal[tcols]

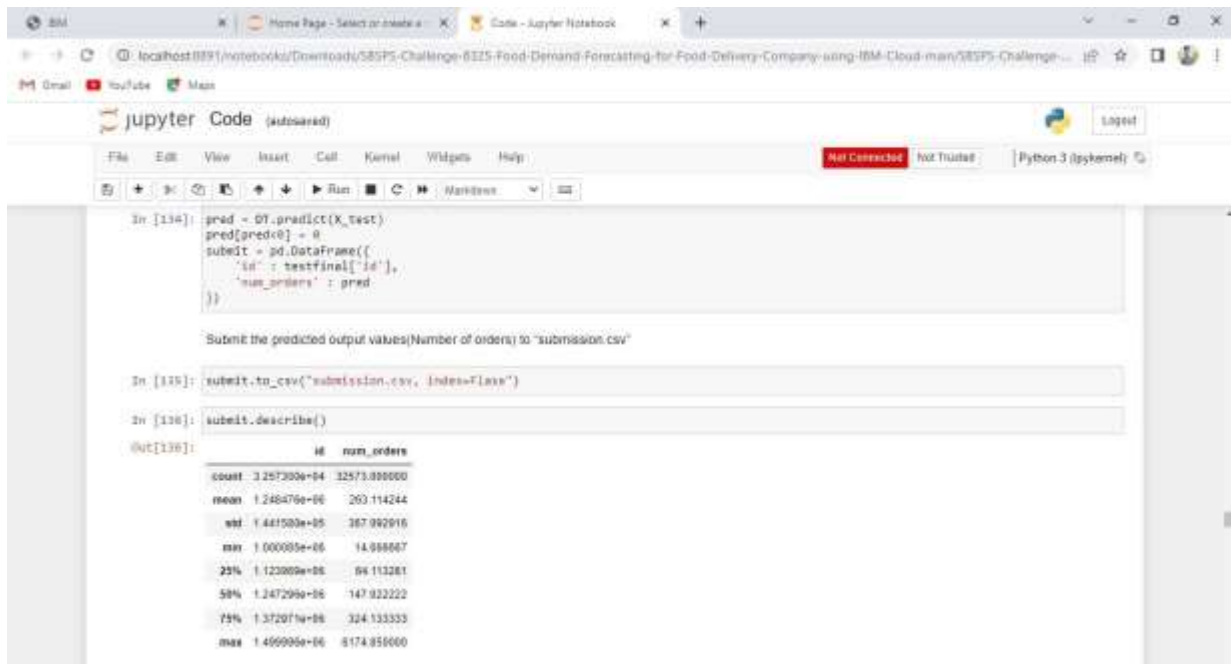
lb1 = LabelEncoder()
testfinal['center_type'] = lb1.fit_transform(testfinal['center_type'])

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

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

X_test = testfinal[features].values

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



The screenshot shows a Jupyter Notebook interface with the title "Submitting the predicted output values (Number of orders) to 'submission.csv'".

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

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

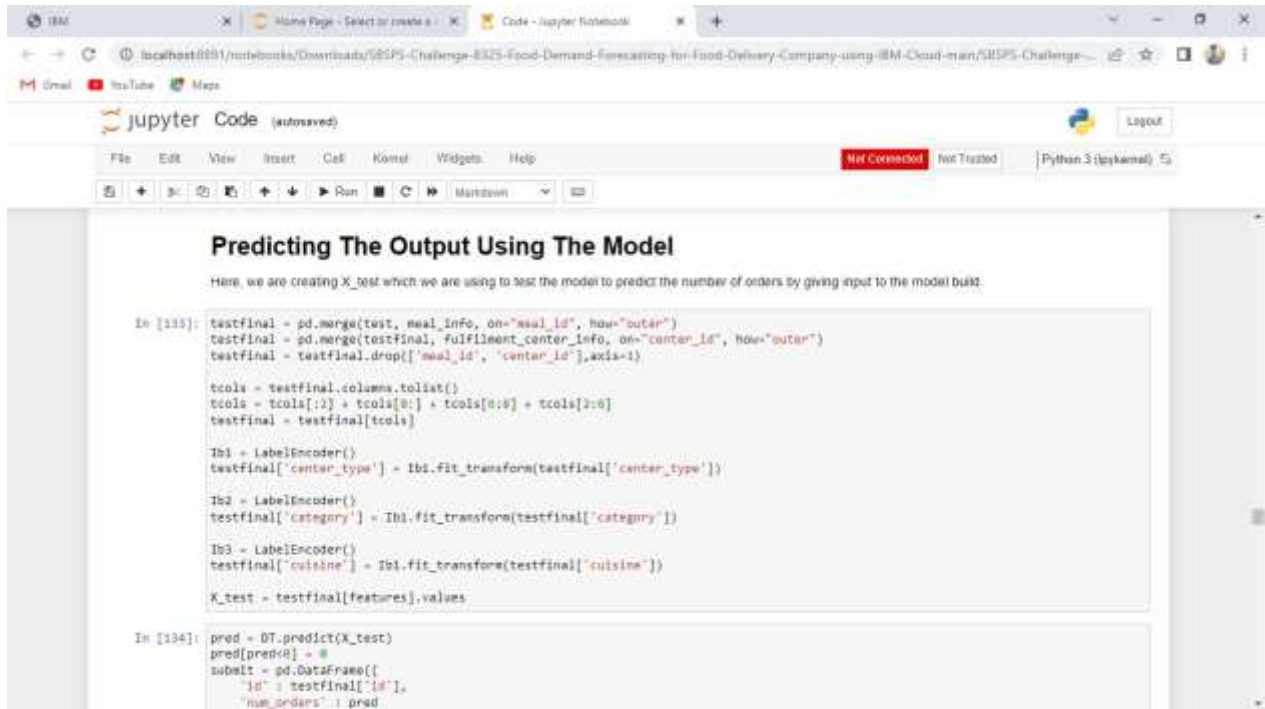
In [135]: subedit.to_csv("submission.csv", index=False)

In [136]: subedit.describe()

Out[136]:
```

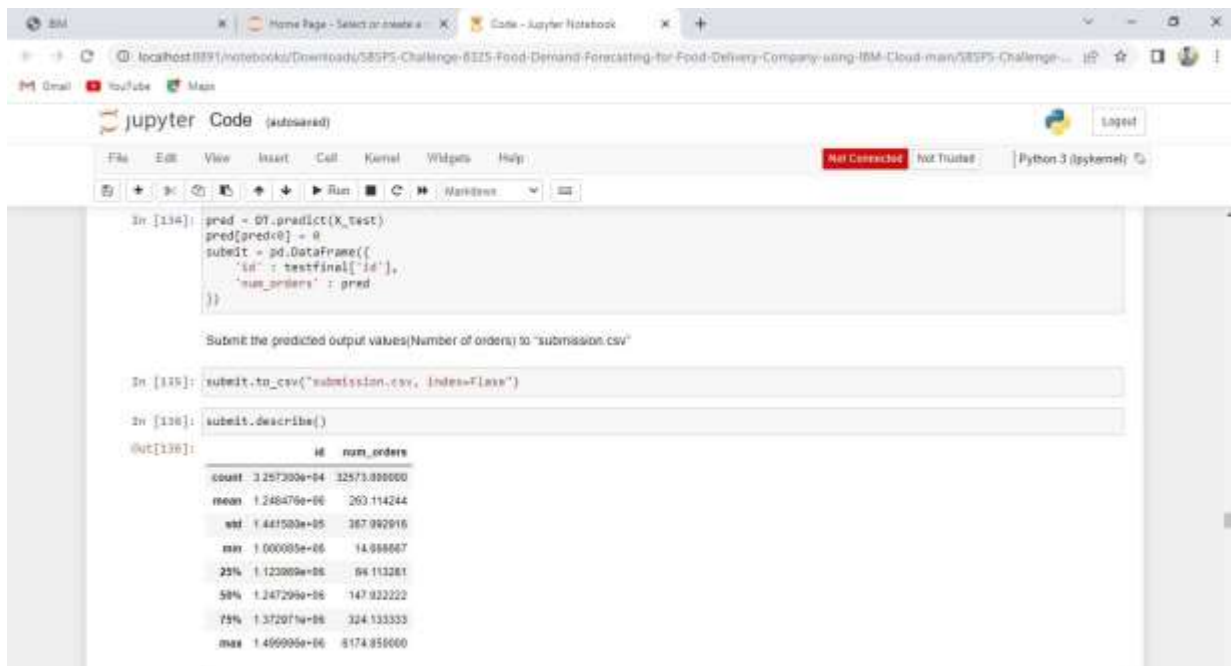
	id	num_orders
count	3.257300e+04	32573.000000
mean	1.248476e+06	260.114244
std	1.441500e+05	267.982916
min	1.000000e+06	14.988867
25%	1.123089e+06	84.113281
50%	1.247296e+06	147.922222
75%	1.372071e+06	324.133333
max	1.499999e+06	8174.856000

Team Member 3



The screenshot shows a Jupyter Notebook interface with a title bar indicating it's running on IBM Cloud. The notebook is titled "Predicting The Output Using The Model". Below the title, there is a text block explaining that the goal is to create `X_test` to test a model. The code in the notebook performs the following steps:

- Merge `test`, `meal_info`, and `fulfillment_center_info` into `testfinal` based on `meal_id` and `center_id`.
- Drop `meal_id` and `center_id` from `testfinal`.
- Concatenate columns `0:2` and `0:8` into a new column `0:8`.
- Fit and transform categorical variables: `center_type`, `category`, and `cuisine`.
- Extract features from `testfinal` into `X_test`.
- Use the trained model `DT` to predict the number of orders for `X_test`.
- Create a `DataFrame` with `id` and `num_orders` (predicted values).



The screenshot shows the continuation of the Jupyter Notebook. The code in the notebook performs the following steps:

- Use the trained model `DT` to predict the number of orders for `X_test`.
- Create a `DataFrame` with `id` and `num_orders` (predicted values).
- Submit the predicted output values (Number of orders) to "submission.csv".
- Describe the `submission.csv` file.

The output of the `submit.describe()` command is displayed as follows:

	id	num_orders
count	3.257300e+04	32573.000000
mean	1.248476e+06	260.114244
std	1.441503e+05	267.992916
min	1.000000e+05	14.988667
25%	1.123089e+06	84.113281
50%	1.247299e+06	147.932222
75%	1.372071e+06	324.133333
max	1.466906e+06	8174.850000