

TEAM ID: PNT2022TMID17645

## PROJECT NAME: DemandEst - AI powered Food Demand Forecaster

### Team Leader

The screenshot displays a Jupyter Notebook interface with two visible code cells. The first cell, labeled 'In [133]:', contains Python code for data preprocessing and model prediction. The second cell, labeled 'In [134]:', contains code for submitting the predicted values to a CSV file. Below the second cell, the output of the submission is shown as a table.

**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]

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

Ib2 = LabelEncoder()
testfinal['category'] = Ib1.fit_transform(testfinal['category'])

Ib3 = LabelEncoder()
testfinal['cuisine'] = Ib1.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
})

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

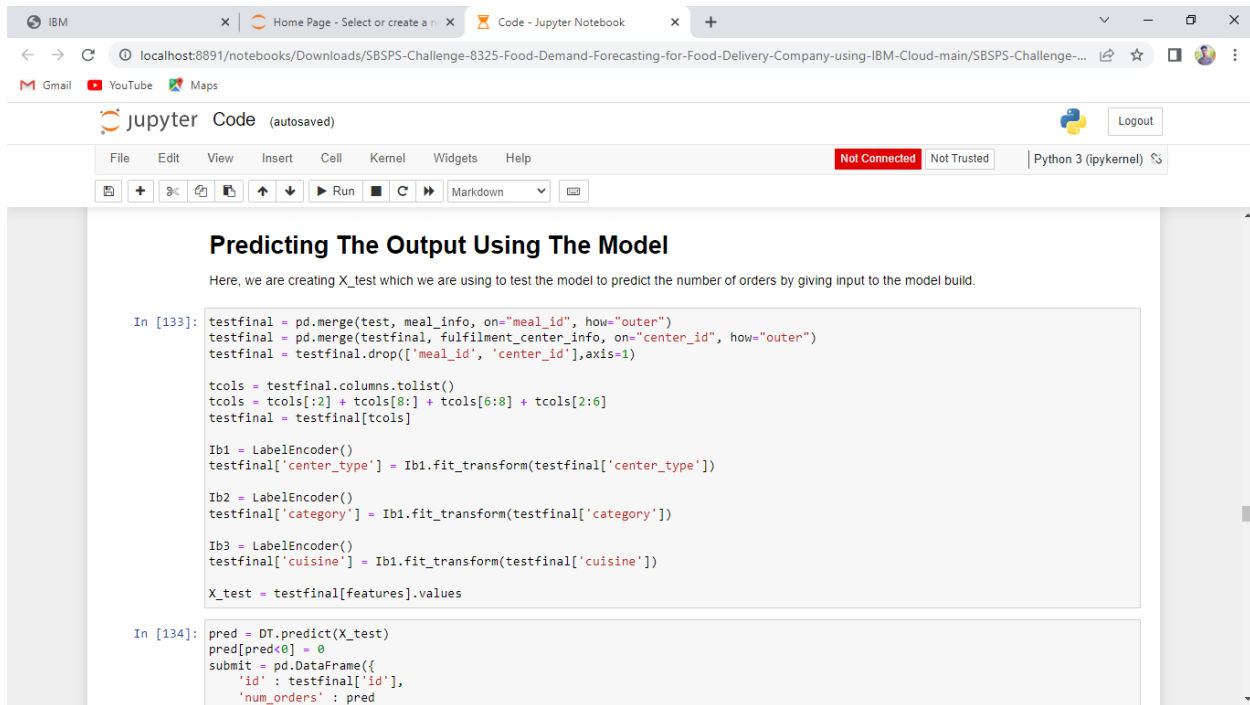
```
In [135]: submit.to_csv("submission.csv", index=False)

In [136]: submit.describe()

Out[136]:
```

	id	num_orders
count	3.257300e+04	32573.000000
mean	1.248476e+06	263.114244
std	1.441580e+05	367.092916
min	1.000085e+06	14.666667
25%	1.123969e+06	64.113281
50%	1.247299e+06	147.022222
75%	1.372971e+06	324.133333
max	1.499996e+06	6174.850000

# Team Member 1



The screenshot shows a Jupyter Notebook titled "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 build." The notebook contains two code cells. The first cell (In [133]) defines 'testfinal' by merging 'test', 'meal\_info', and 'fulfilment\_center\_info' on 'meal\_id' and 'center\_id' respectively, then drops these columns and selects specific columns into 'tcols'. It then uses 'LabelEncoder()' to transform 'center\_type', 'category', and 'cuisine' columns. The second cell (In [134]) uses 'DT.predict(X\_test)' to get predictions, sets the first prediction to 0, and creates a 'submit' DataFrame with 'id' and 'num\_orders' columns.

```
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[8:8] + tcols[6:8] + tcols[2:6]
testfinal = testfinal[tcols]

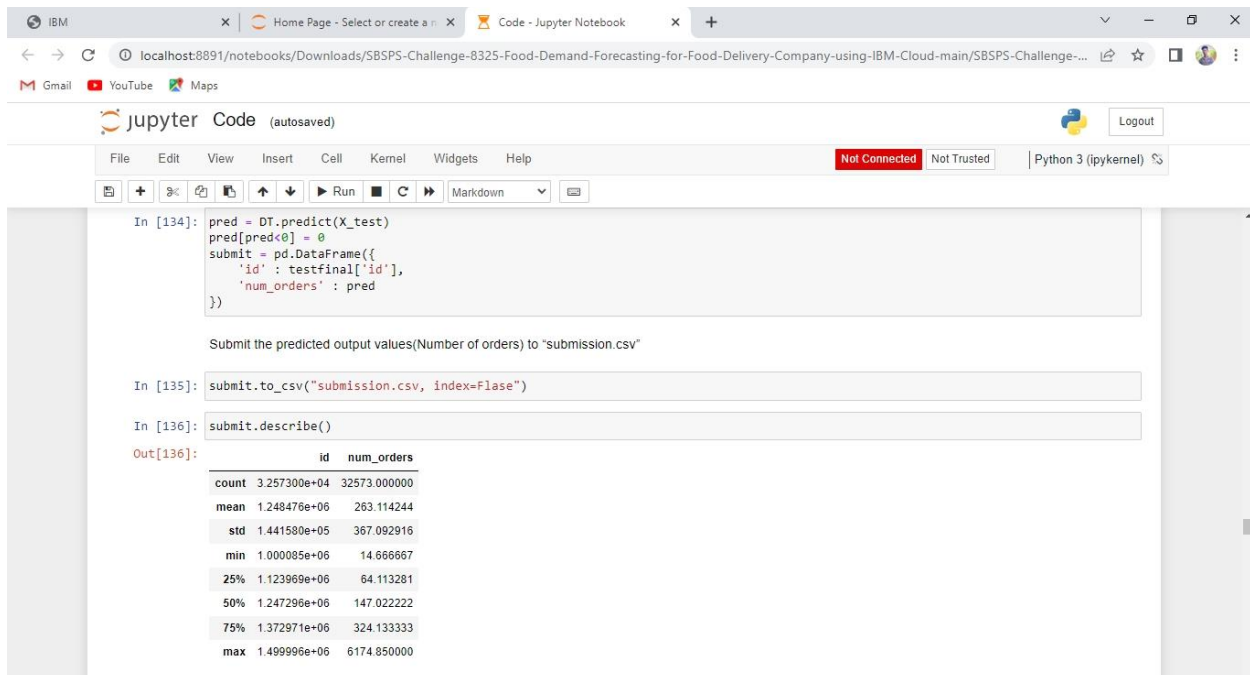
Ib1 = LabelEncoder()
testfinal['center_type'] = Ib1.fit_transform(testfinal['center_type'])

Ib2 = LabelEncoder()
testfinal['category'] = Ib1.fit_transform(testfinal['category'])

Ib3 = LabelEncoder()
testfinal['cuisine'] = Ib1.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
})
```



The screenshot shows the continuation of the Jupyter Notebook. The first code cell (In [134]) is identical to the one in the previous screenshot. Below the code cell, there is a text instruction: "Submit the predicted output values(Number of orders) to 'submission.csv'". The second code cell (In [135]) calls 'submit.to\_csv("submission.csv, index=False")'. The third code cell (In [136]) calls 'submit.describe()'. The output (Out[136]) is a summary statistics table for the 'submit' DataFrame.

```
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"

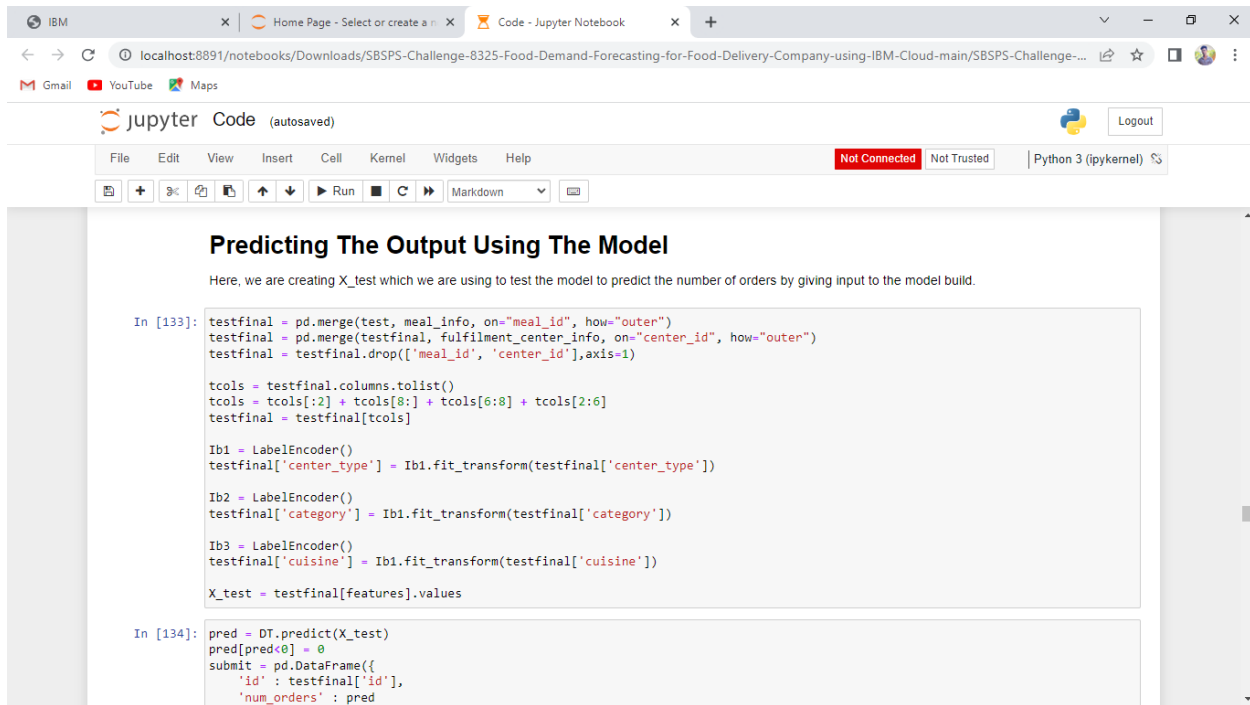
In [135]: submit.to_csv("submission.csv, index=False")

In [136]: submit.describe()

Out[136]:
```

	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 2



The screenshot shows a Jupyter Notebook interface with a browser window at the top. The notebook is titled "jupyter Code (autosaved)" and has a toolbar with various icons. The main content area displays a markdown cell with the heading "Predicting The Output Using The Model" and a text block explaining the purpose of the code. Below this, there are two code cells. The first cell (In [133]) contains Python code for merging dataframes, dropping columns, selecting specific columns, and applying label encoders to 'center\_type', 'category', and 'cuisine' columns. The second cell (In [134]) contains code for predicting the number of orders using the trained model and creating a submission dataframe.

### 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]

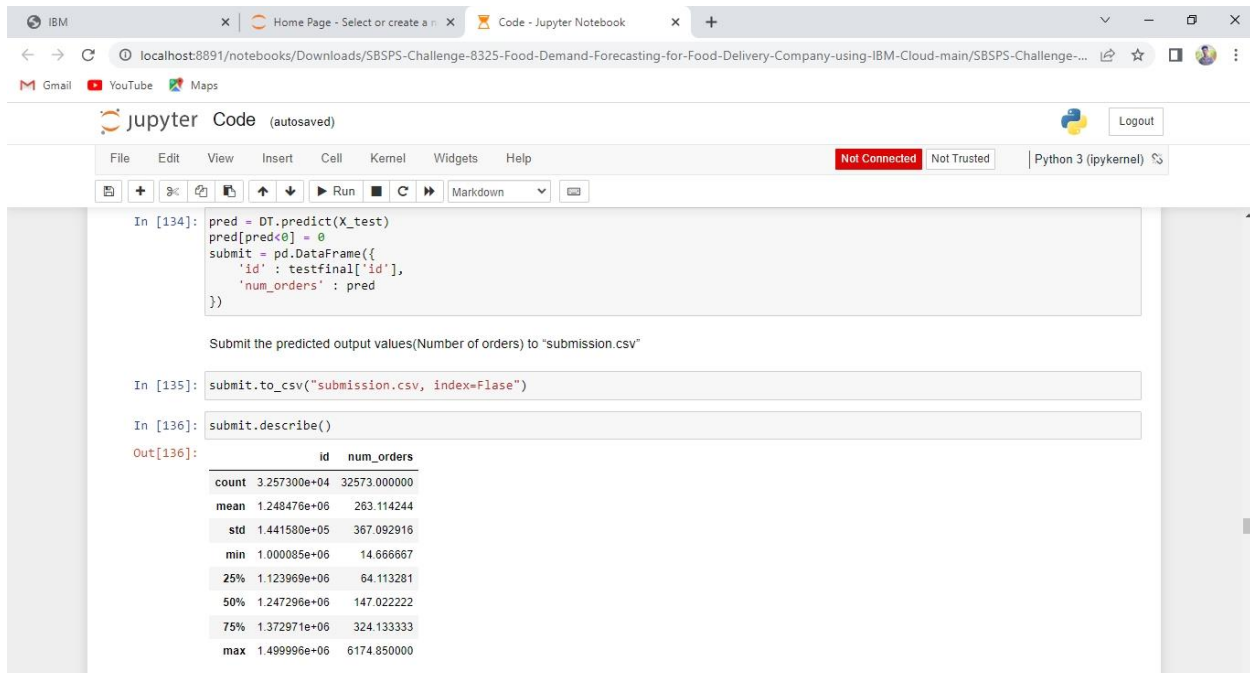
Ib1 = LabelEncoder()
testfinal['center_type'] = Ib1.fit_transform(testfinal['center_type'])

Ib2 = LabelEncoder()
testfinal['category'] = Ib1.fit_transform(testfinal['category'])

Ib3 = LabelEncoder()
testfinal['cuisine'] = Ib1.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
})
```



The screenshot shows the continuation of the Jupyter Notebook. It displays the final code cell (In [134]) which creates the submission dataframe. Below this, there is a text instruction to submit the predicted output values. The next code cell (In [135]) shows the submission being saved to a CSV file. The final code cell (In [136]) shows the submission being described. The output (Out[136]) displays a summary statistics table for the 'num\_orders' column.

```
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"

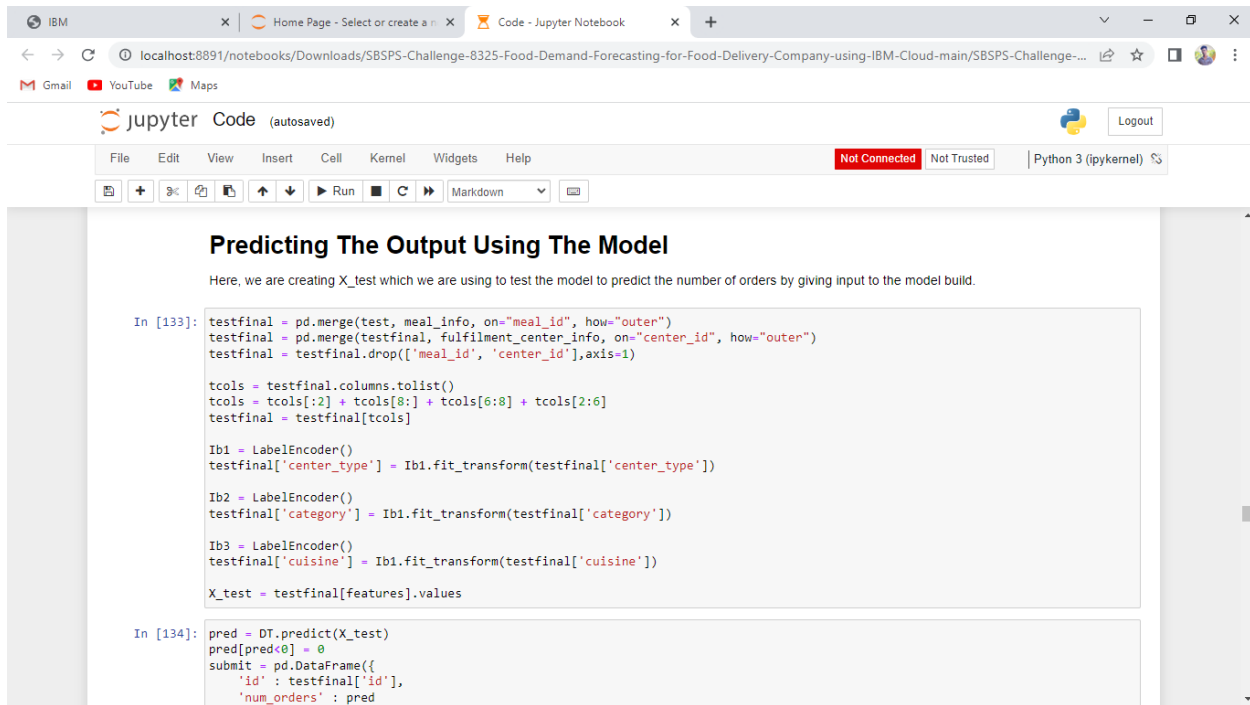
In [135]: submit.to_csv("submission.csv", index=False)

In [136]: submit.describe()

Out[136]:
```

	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 3



The screenshot shows a Jupyter Notebook titled "Predicting The Output Using The Model". The notebook is running on a local host (localhost:8891) and is connected to a Python 3 (ipykernel) environment. The code in the notebook is as follows:

```
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[8:8] + tcols[6:8] + tcols[2:6]
testfinal = testfinal[tcols]

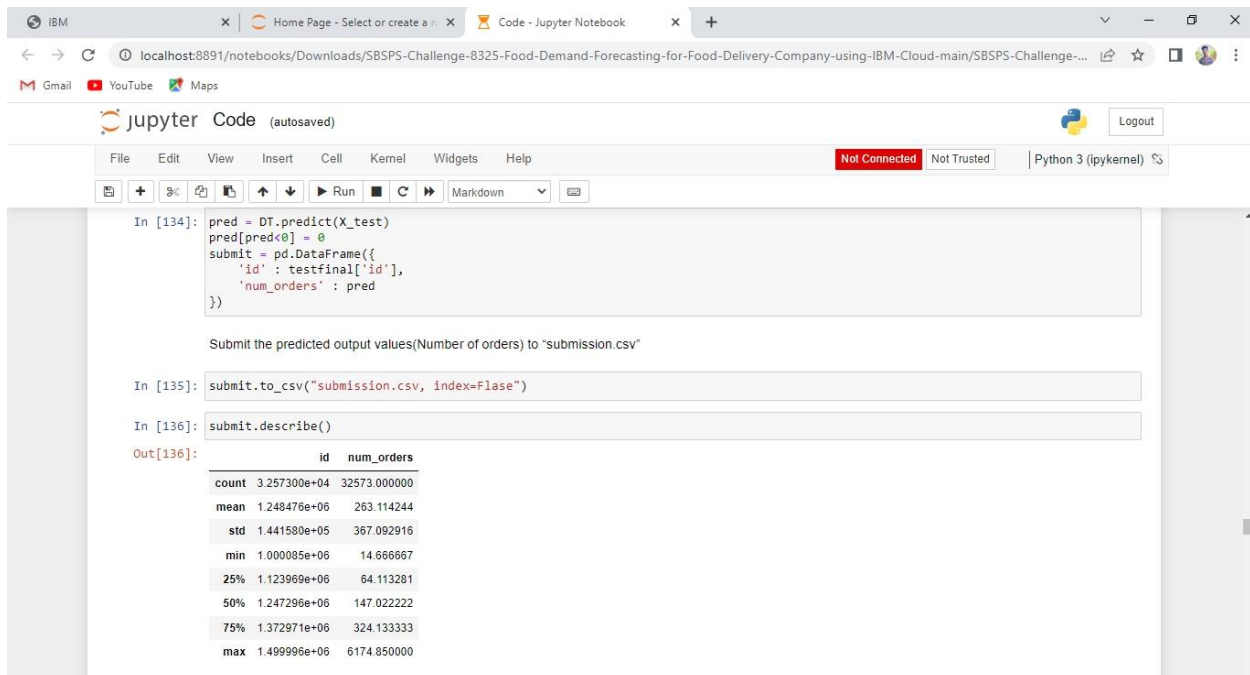
Ib1 = LabelEncoder()
testfinal['center_type'] = Ib1.fit_transform(testfinal['center_type'])

Ib2 = LabelEncoder()
testfinal['category'] = Ib1.fit_transform(testfinal['category'])

Ib3 = LabelEncoder()
testfinal['cuisine'] = Ib1.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
})
```



The screenshot shows a Jupyter Notebook titled "Submit the predicted output values (Number of orders) to 'submission.csv'". The notebook is running on a local host (localhost:8891) and is connected to a Python 3 (ipykernel) environment. The code in the notebook is as follows:

```
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=False)

In [136]: submit.describe()

Out[136]:
```

	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