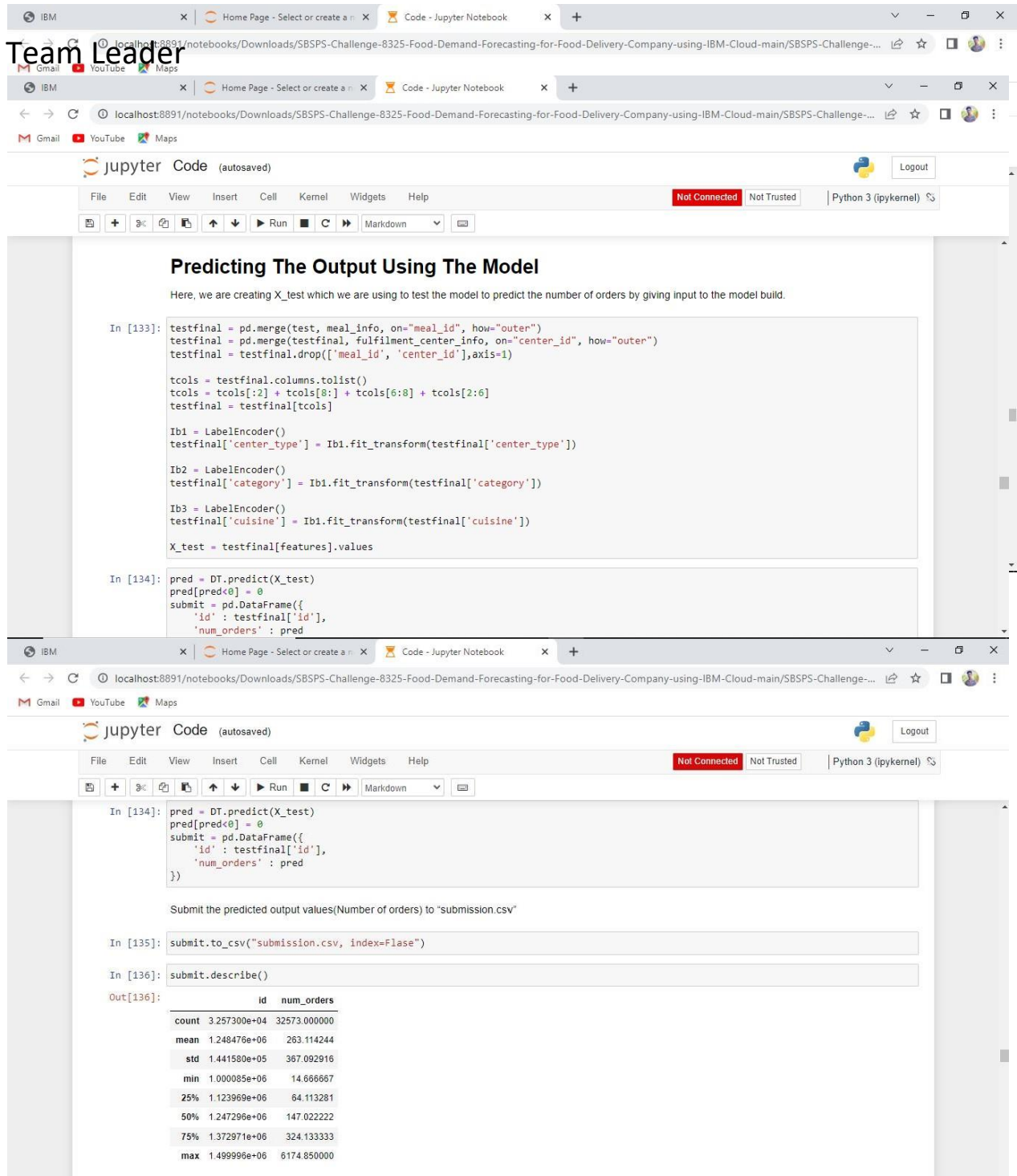


TEAM ID: PNT2022TMID48753

PROJECT NAME: DemandEst - AI powered Food Demand Forecaster

Team Member 1

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 predicting the number of orders and creating a submission DataFrame. Below the second cell, there is a text instruction and a third code cell labeled 'In [135]:' which calls the 'submit.to_csv' method. A fourth code cell labeled 'In [136]:' calls 'submit.describe()'. The output of the fourth cell, labeled 'Out[136]:', shows a summary statistics table for the 'submit' 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
})

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

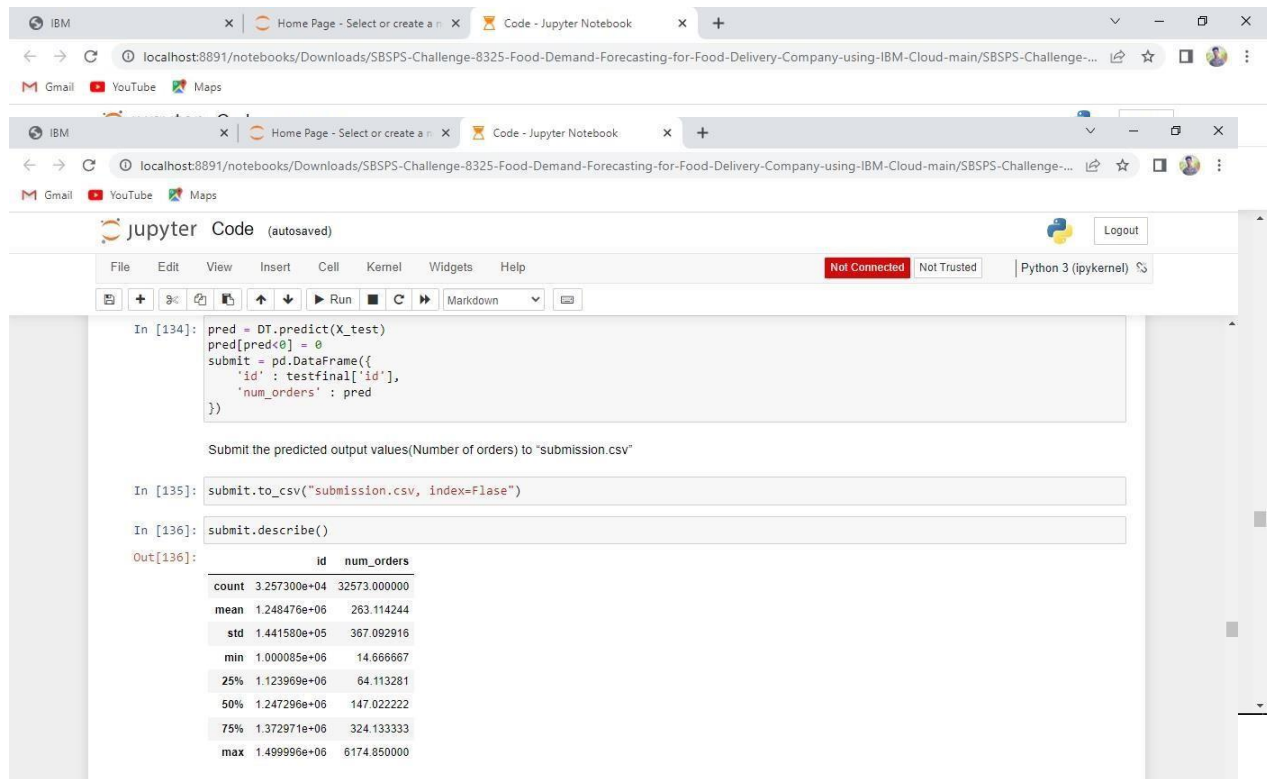
```
In [135]: submit.to_csv("submission.csv", index=False)

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 displays a Jupyter Notebook interface within a web browser. The browser's address bar shows the URL: `localhost:8891/notebooks/Downloads/SBSPS-Challenge-8325-Food-Demand-Forecasting-for-Food-Delivery-Company-using-IBM-Cloud-main/SBSPS-Challenge-...`. The notebook's title bar indicates it is an "autosaved" code file. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, cell execution, and output viewing. The status bar at the bottom shows "Not Connected", "Not Trusted", and "Python 3 (ipykernel)".

The notebook contains three input cells and one output cell. The first input cell (In [134]) contains the following Python code:

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

Below the code cell, a text prompt reads: "Submit the predicted output values(Number of orders) to 'submission.csv'".

The second input cell (In [135]) contains the code:

```
submit.to_csv("submission.csv", index=False)
```

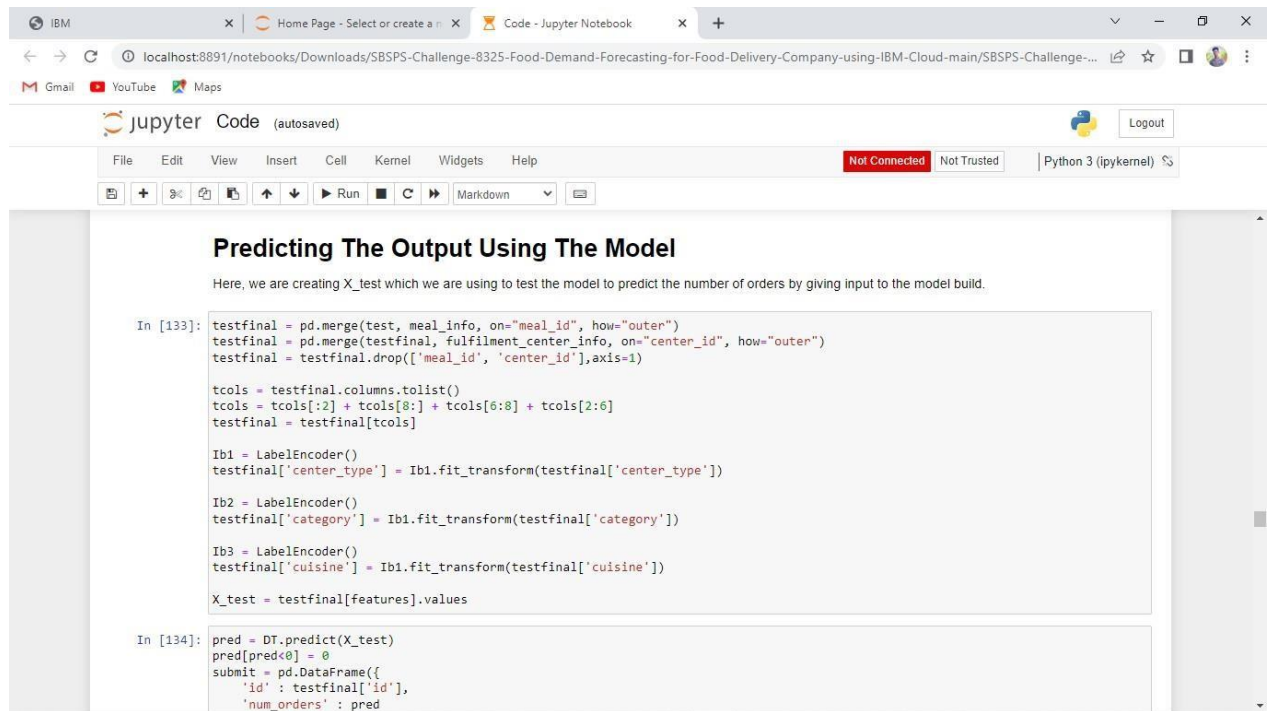
The third input cell (In [136]) contains the code:

```
submit.describe()
```

The output cell (Out[136]) displays the result of the `describe()` method, showing statistical data for the 'id' and 'num_orders' columns:

| | 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 interface with the following content:

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]

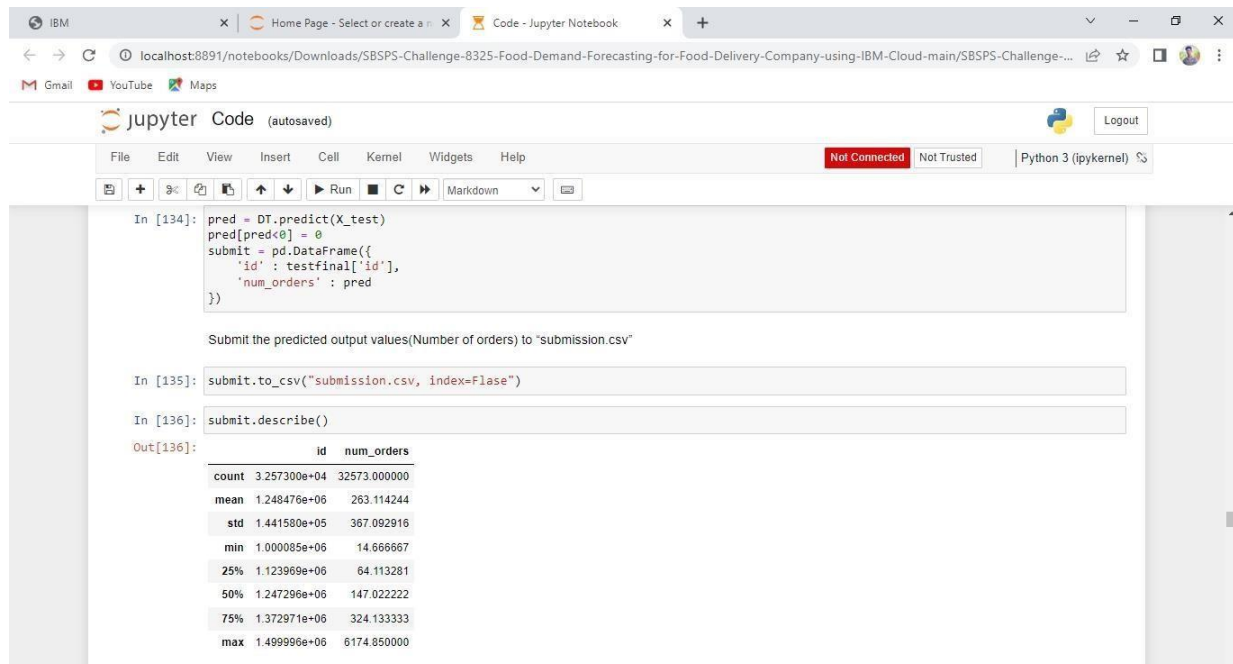
Ib1 = LabelEncoder()
testfinal['center_type'] = Ib1.fit_transform(testfinal['center_type'])

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 a Jupyter Notebook interface with the following content:

```
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"
```

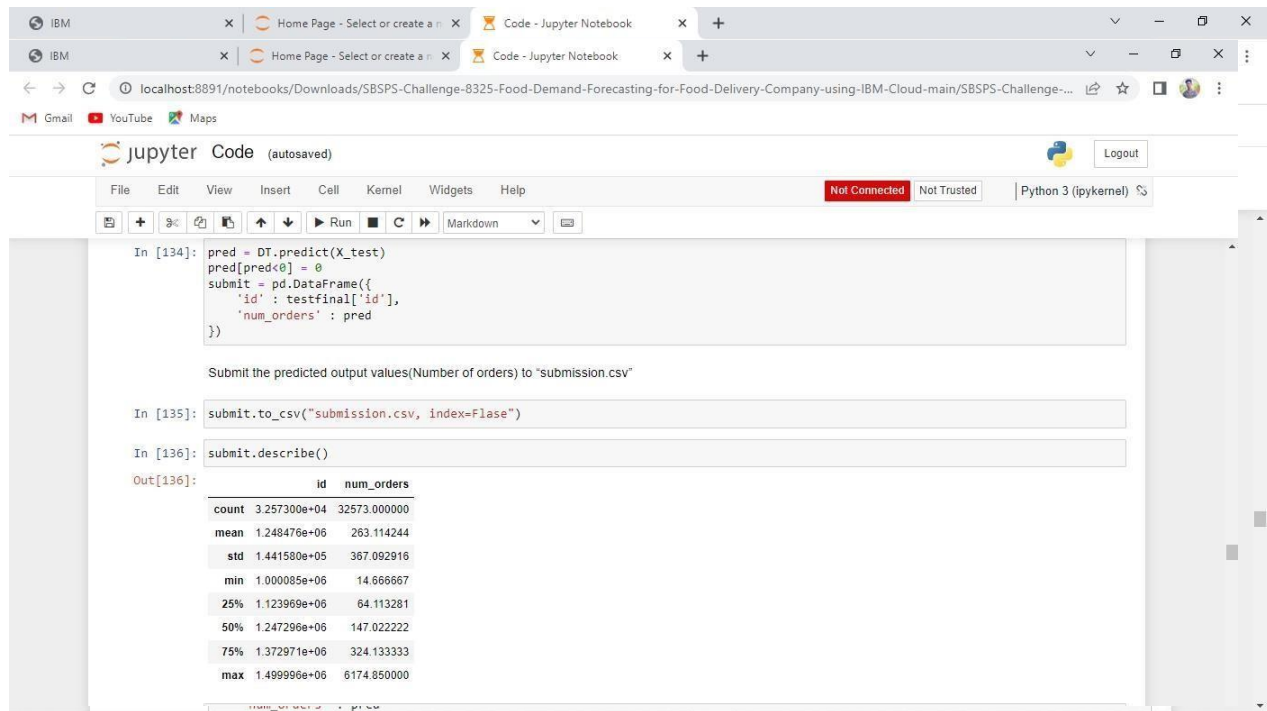
```
In [135]: submit.to_csv("submission.csv", index=False)
```

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

Out[136]:

| | id | num_orders |
|-------|--------------|--------------|
| count | 3.257300e+04 | 32573.000000 |
| mean | 1.248476e+06 | 283.114244 |
| 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 4



The screenshot shows a web browser window with multiple tabs. The active tab is titled "Code - Jupyter Notebook". The address bar shows the URL: `localhost:8891/notebooks/Downloads/SBSPS-Challenge-8325-Food-Demand-Forecasting-for-Food-Delivery-Company-using-IBM-Cloud-main/SBSPS-Challenge-...`. The Jupyter Notebook interface is displayed, showing a code cell with the following Python code:

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

Below the code cell, a text prompt reads: "Submit the predicted output values(Number of orders) to 'submission.csv'".

The next code cell is:

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

The following code cell is:

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

The output of the last cell is displayed as a table:

| | 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 |