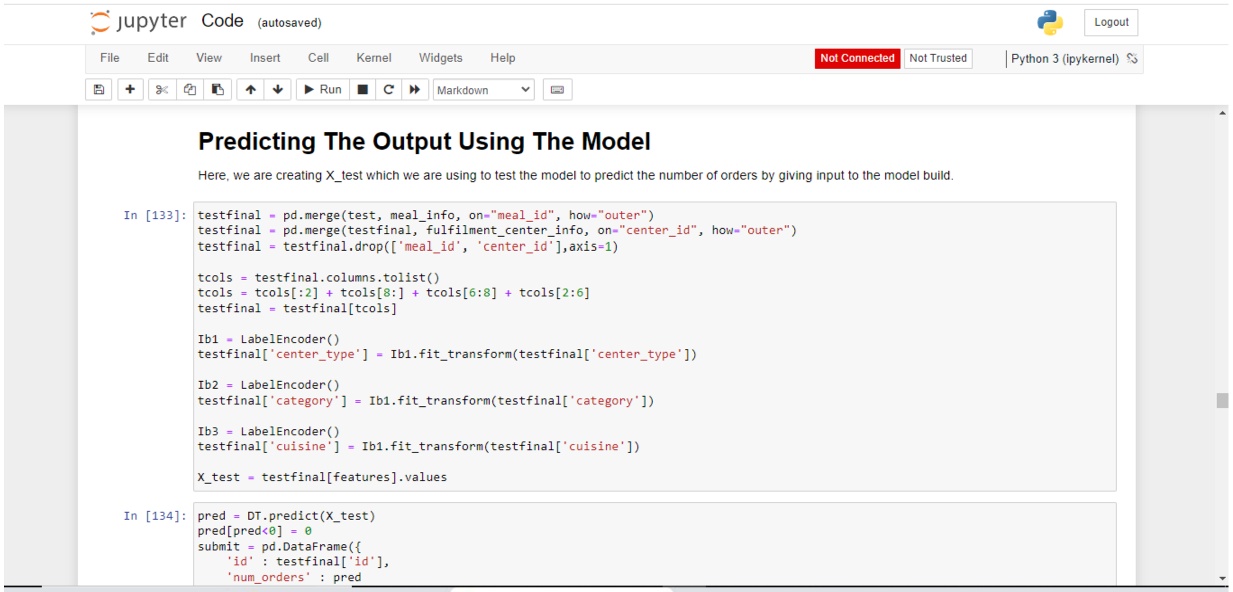
TEAM ID: PNT2022TMID51052

PROJECT NAME: DemandEst - AI powered Food DemandForecaster

Team Leader

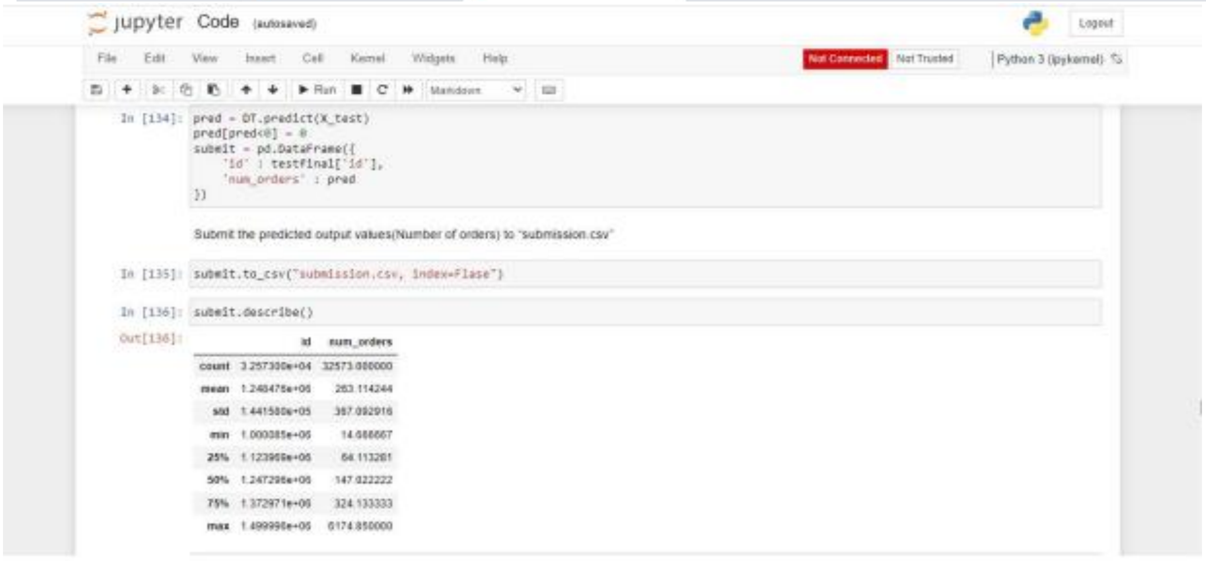


Predicting The Output Using The Model

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, 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
```



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

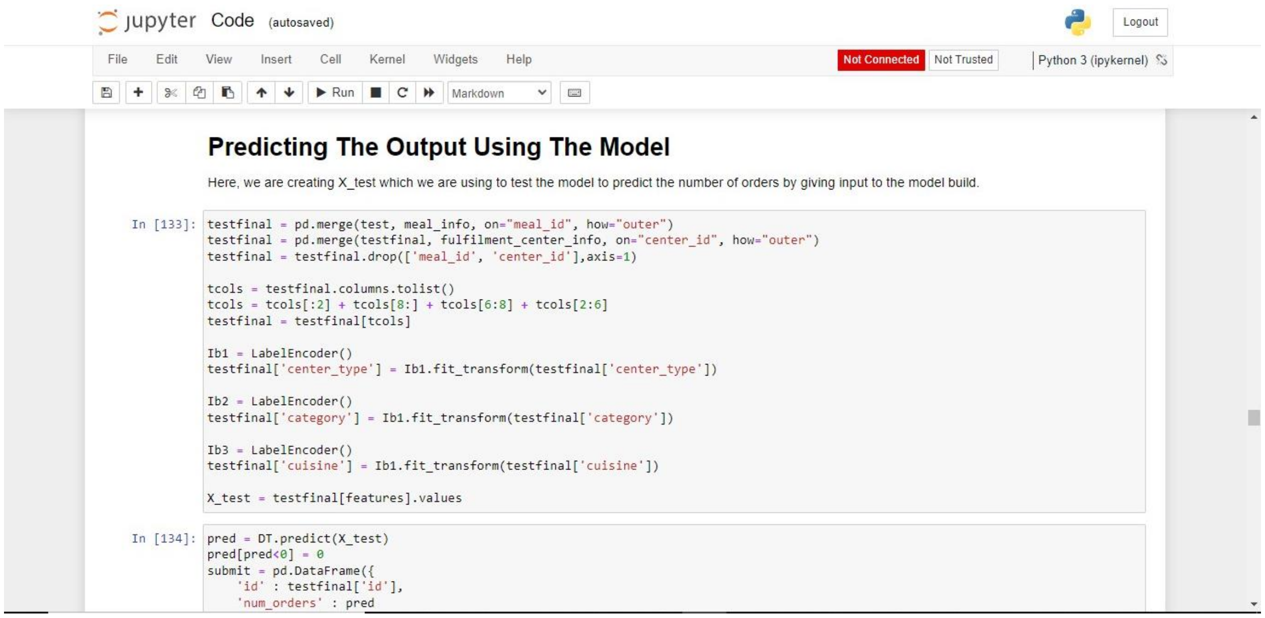
Submit the predicted output values(Number of orders) to "submission.csv"

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

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

```
Out[136]:
                    id     num_orders
        count  3.257300e+04  32573.000000
         mean  1.248475e+06    263.114244
          std  1.441580e+05    387.092916
          min  1.000085e+06     14.686667
          25%  1.123969e+06     64.113281
          50%  1.247296e+06    147.022222
          75%  1.372971e+06    324.133333
          max  1.499998e+06   6174.850000
```

Team Member 1

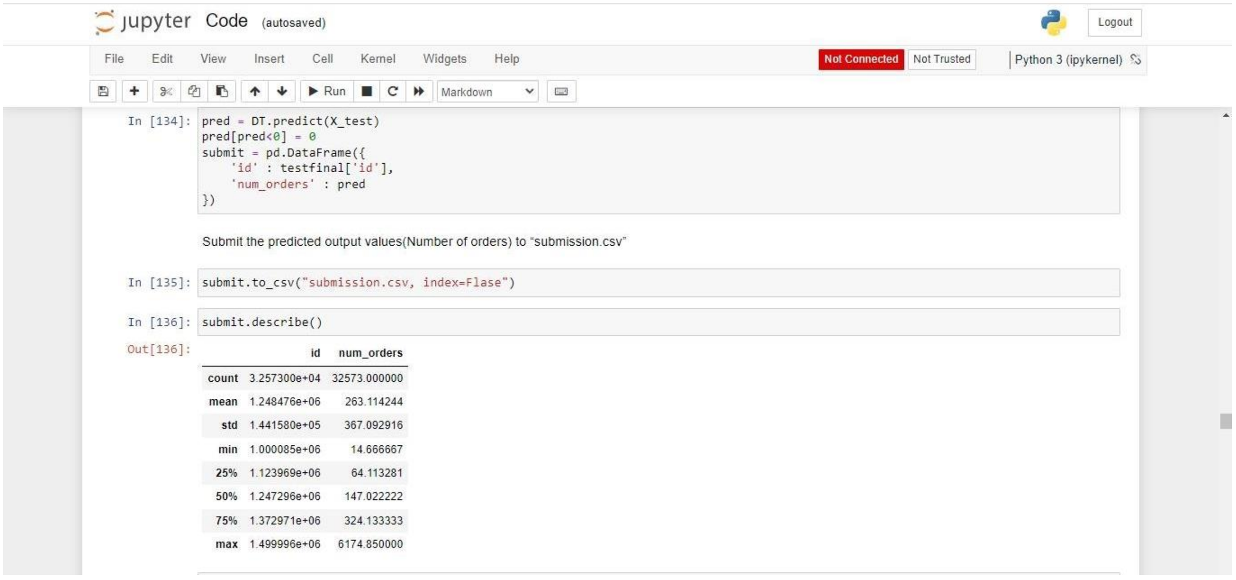


Predicting The Output Using The Model

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
```



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"

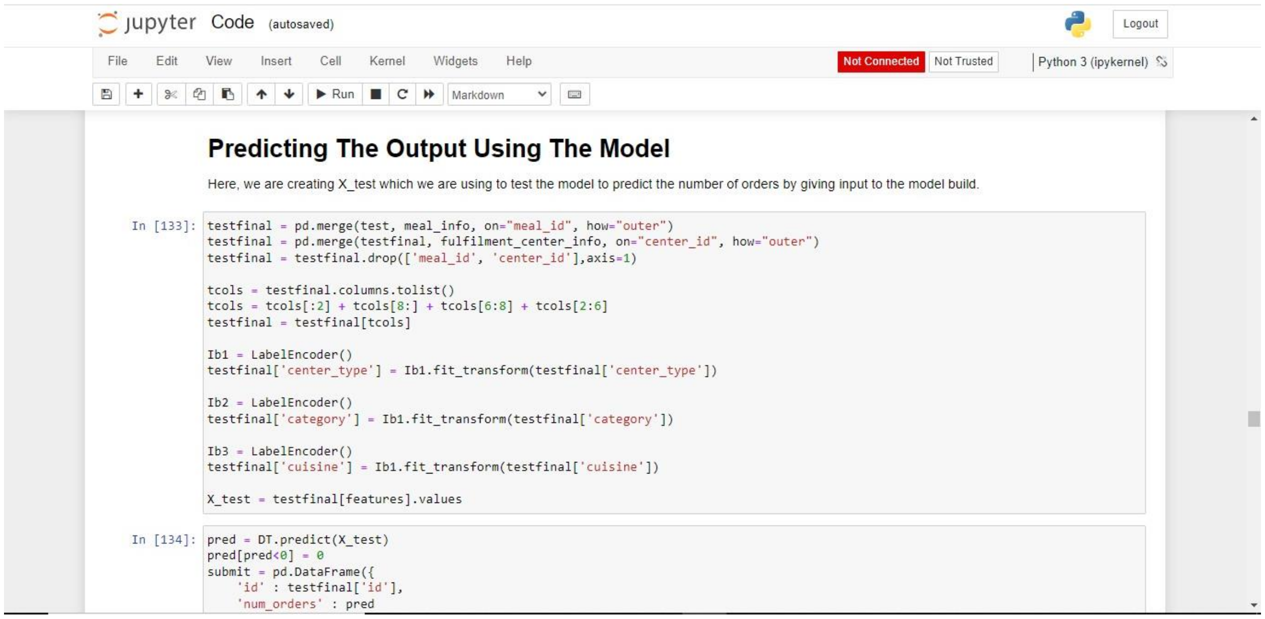
In [135]:
```
submit.to_csv("submission.csv, index=Flase")
```

In [136]:
```
submit.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

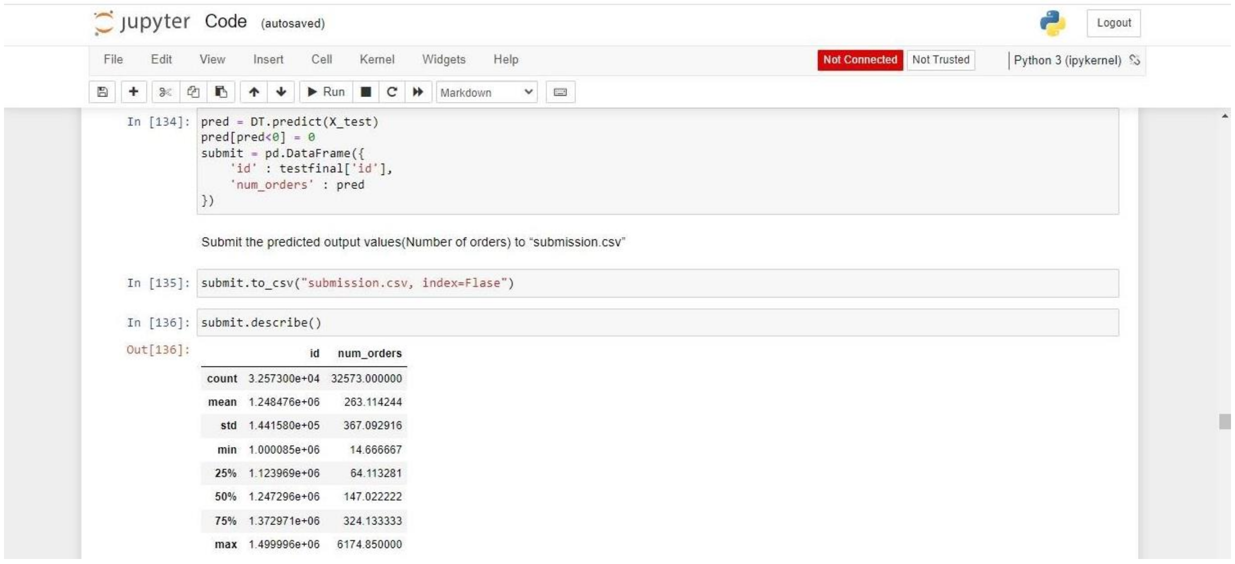


Predicting The Output Using The Model

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
```



```
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"

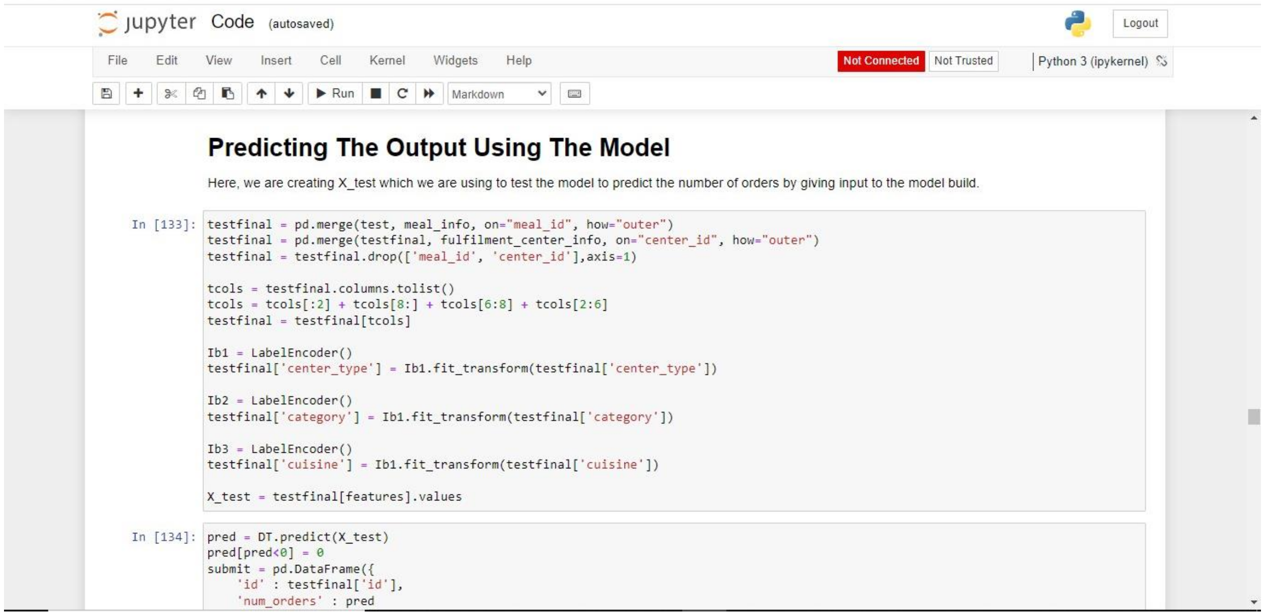
```
In [135]: submit.to_csv("submission.csv, index=Flase")
```

```
In [136]: submit.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

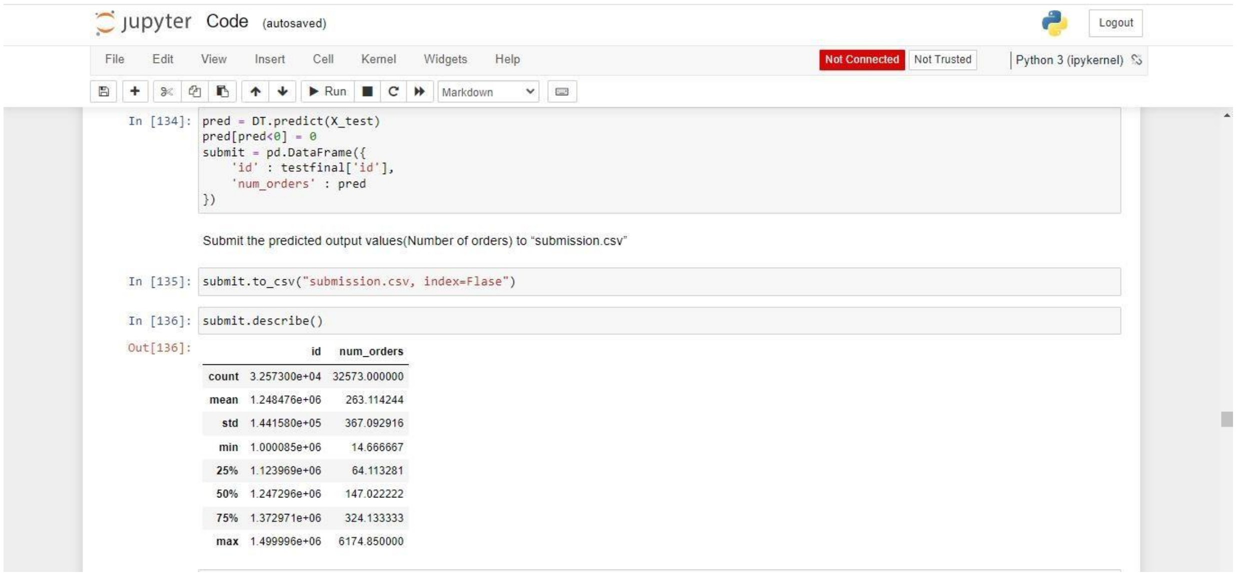


Predicting The Output Using The Model

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
```



```
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.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