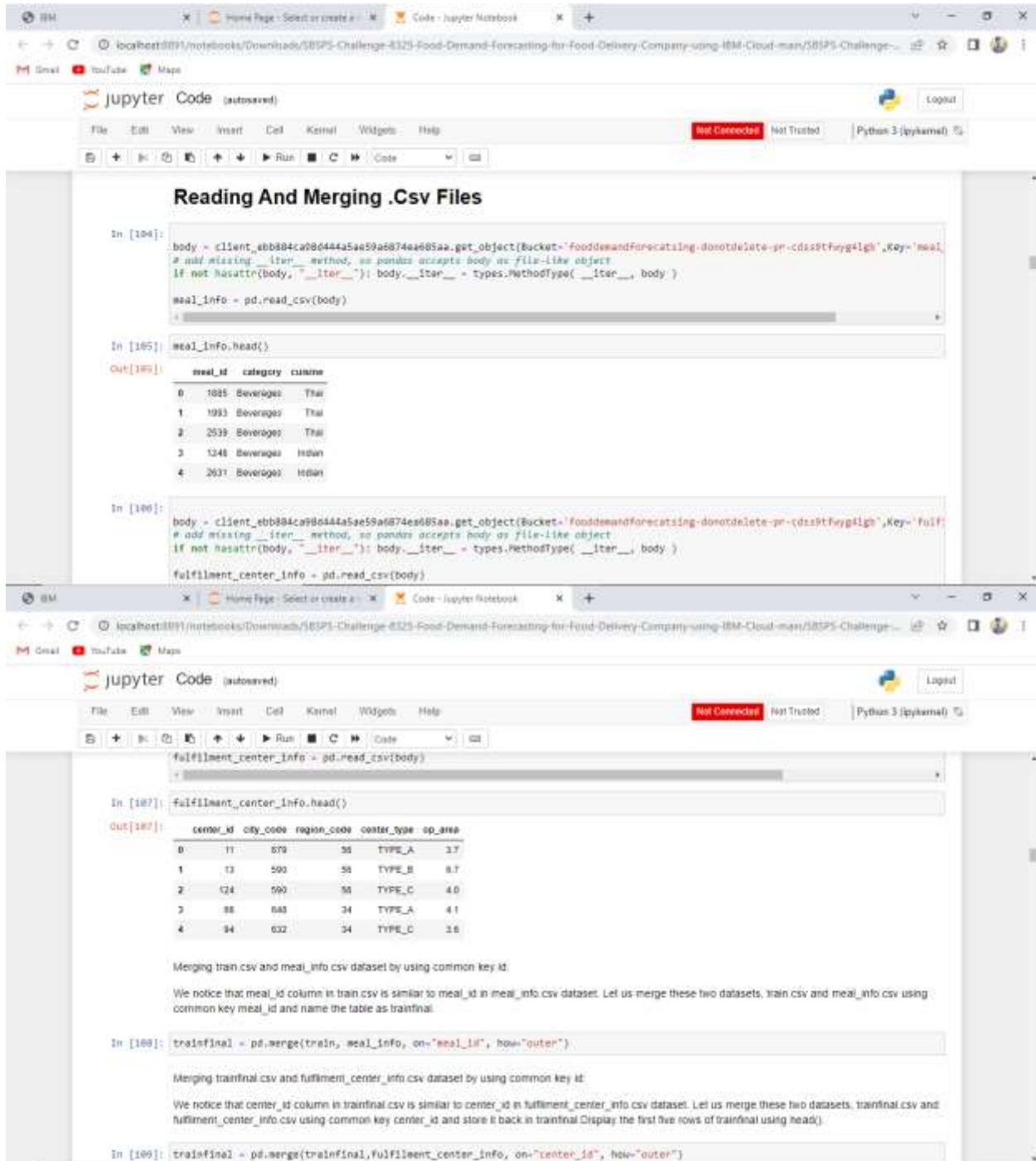


TEAM ID: PNT2022TMID32368

PROJECT NAME: DemandEst - AI powered Food Demand Forecaster

Team Leader



The screenshot displays a Jupyter Notebook interface with the title "Reading And Merging .Csv Files". The notebook is running on a local host (localhost:8891) and is using Python 3 (ipykernel). The code is divided into two main sections, each with a code cell and an output cell.

**Section 1: Reading meal\_info.csv**

The first code cell (In [104]) reads a CSV file from IBM Cloud Object Storage. The code uses the `client.get_object` method to retrieve the file and then `pd.read_csv` to load it into a DataFrame named `meal_info`.

```
In [104]: body = client_ebb884ca98d444a5ae59a6874ea685aa.get_object(Bucket='fooddemandforecasting-donotdelete-pr-cdss9tfuyg4lg',Key='meal_info.csv')
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, '__iter__'): body.__iter__ = types.MethodType(__iter__, body)
meal_info = pd.read_csv(body)
```

The output (Out[105]) shows the first five rows of the `meal_info` DataFrame:

	meal_id	category	cuisine
0	1085	Beverages	Thai
1	1993	Beverages	Thai
2	2539	Beverages	Thai
3	1248	Beverages	Indian
4	2631	Beverages	Indian

**Section 2: Reading fulfillment\_center\_info.csv**

The second code cell (In [106]) reads another CSV file from IBM Cloud Object Storage, loading it into a DataFrame named `fulfillment_center_info`.

```
In [106]: body = client_ebb884ca98d444a5ae59a6874ea685aa.get_object(Bucket='fooddemandforecasting-donotdelete-pr-cdss9tfuyg4lg',Key='fulfillment_center_info.csv')
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, '__iter__'): body.__iter__ = types.MethodType(__iter__, body)
fulfillment_center_info = pd.read_csv(body)
```

The output (Out[107]) shows the first five rows of the `fulfillment_center_info` DataFrame:

	center_id	city_code	region_code	center_type	op_area
0	11	679	58	TYPE_A	3.7
1	13	593	58	TYPE_B	8.7
2	124	590	58	TYPE_C	4.0
3	88	643	34	TYPE_A	4.1
4	94	632	34	TYPE_C	3.6

**Merging the datasets**

The notebook includes text explaining the merging process. It states that the `meal_id` column in the `train.csv` dataset is similar to the `meal_id` column in the `meal_info.csv` dataset. The two datasets are merged using `pd.merge` with `on="meal_id"` and `how="outer"`.

```
In [108]: trainfinal = pd.merge(train, meal_info, on="meal_id", how="outer")
```

Next, it explains that the `center_id` column in the `trainfinal.csv` dataset is similar to the `center_id` column in the `fulfillment_center_info.csv` dataset. The two datasets are merged again using `pd.merge` with `on="center_id"` and `how="outer"`.

```
In [109]: trainfinal = pd.merge(trainfinal, fulfillment_center_info, on="center_id", how="outer")
```

Merging train.csv and meal\_info.csv dataset by using common key id.

We notice that meal\_id column in train.csv is similar to meal\_id in meal\_info.csv dataset. Let us merge these two datasets, train.csv and meal\_info.csv using common key meal\_id and name the table as trainfinal.

```
In [188]: trainfinal = pd.merge(train, meal_info, on="meal_id", how="outer")
```

Merging trainfinal.csv and fulfillment\_center\_info.csv dataset by using common key id.

We notice that center\_id column in trainfinal.csv is similar to center\_id in fulfillment\_center\_info.csv dataset. Let us merge these two datasets, trainfinal.csv and fulfillment\_center\_info.csv using common key center\_id and store it back in trainfinal. Display the first five rows of trainfinal using head().

```
In [189]: trainfinal = pd.merge(trainfinal, fulfillment_center_info, on="center_id", how="outer")
trainfinal.head()
```

```
Out[189]:
```

	id	week	center_id	meal_id	checkout_price	base_price	email_for_promotion	homepage_featured	num_orders	category	cuisine	city_code	region
0	1370988	1	55	1885	136.83	152.29	0	0	177	Beverages	Thai	647	
1	1018704	2	55	1885	135.83	152.29	0	0	323	Beverages	Thai	647	
2	1196273	3	55	1885	132.92	133.92	0	0	98	Beverages	Thai	647	
3	1110527	4	55	1885	135.98	134.88	0	0	103	Beverages	Thai	647	
4	1343872	5	55	1885	145.50	147.50	0	0	215	Beverages	Thai	647	

Team Member 1

### Reading And Merging .Csv Files

```
In [184]:
```

```
body = client_ebb884ca98d444a5ae59a6874ea085aa.get_object(Bucket='fooddemandforecasting-donotdelete-pr-cdss9tfuyg4lgh',Key='meal_
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType(__iter__, body )

meal_info = pd.read_csv(body)
```

```
In [185]: meal_info.head()
```

```
Out[185]:
```

	meal_id	category	cuisine
0	1885	Beverages	Thai
1	1983	Beverages	Thai
2	2539	Beverages	Thai
3	1248	Beverages	Indian
4	2531	Beverages	Indian

```
In [186]:
```

```
body = client_ebb884ca98d444a5ae59a6874ea085aa.get_object(Bucket='fooddemandforecasting-donotdelete-pr-cdss9tfuyg4lgh',Key='fulf
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType(__iter__, body )

fulfillment_center_info = pd.read_csv(body)
```

```
fulfilment_center_info = pd.read_csv(body)

In [187]: fulfilment_center_info.head()
Out[187]:
```

	center_id	city_code	region_code	center_type	op_area
0	11	678	56	TYPE_A	3.7
1	13	598	56	TYPE_B	8.7
2	124	598	56	TYPE_C	4.0
3	88	648	34	TYPE_A	4.1
4	94	632	34	TYPE_C	3.8

Merging train.csv and meal\_info.csv dataset by using common key id

We notice that meal\_id column in train.csv is similar to meal\_id in meal\_info.csv dataset. Let us merge these two datasets, train.csv and meal\_info.csv using common key meal\_id and name the table as trainfinal

```
In [188]: trainfinal = pd.merge(train, meal_info, on="meal_id", how="outer")
```

Merging trainfinal.csv and fulfillment\_center\_info.csv dataset by using common key id

We notice that center\_id column in trainfinal.csv is similar to center\_id in fulfillment\_center\_info.csv dataset. Let us merge these two datasets, trainfinal.csv and fulfillment\_center\_info.csv using common key center\_id and store it back in trainfinal. Display the first five rows of trainfinal using head()

```
In [189]: trainfinal = pd.merge(trainfinal, fulfillment_center_info, on="center_id", how="outer")
```

```
3  88  648  34  TYPE_A  4.1
4  94  632  34  TYPE_C  3.8
```

Merging train.csv and meal\_info.csv dataset by using common key id

We notice that meal\_id column in train.csv is similar to meal\_id in meal\_info.csv dataset. Let us merge these two datasets, train.csv and meal\_info.csv using common key meal\_id and name the table as trainfinal

```
In [188]: trainfinal = pd.merge(train, meal_info, on="meal_id", how="outer")
```

Merging trainfinal.csv and fulfillment\_center\_info.csv dataset by using common key id

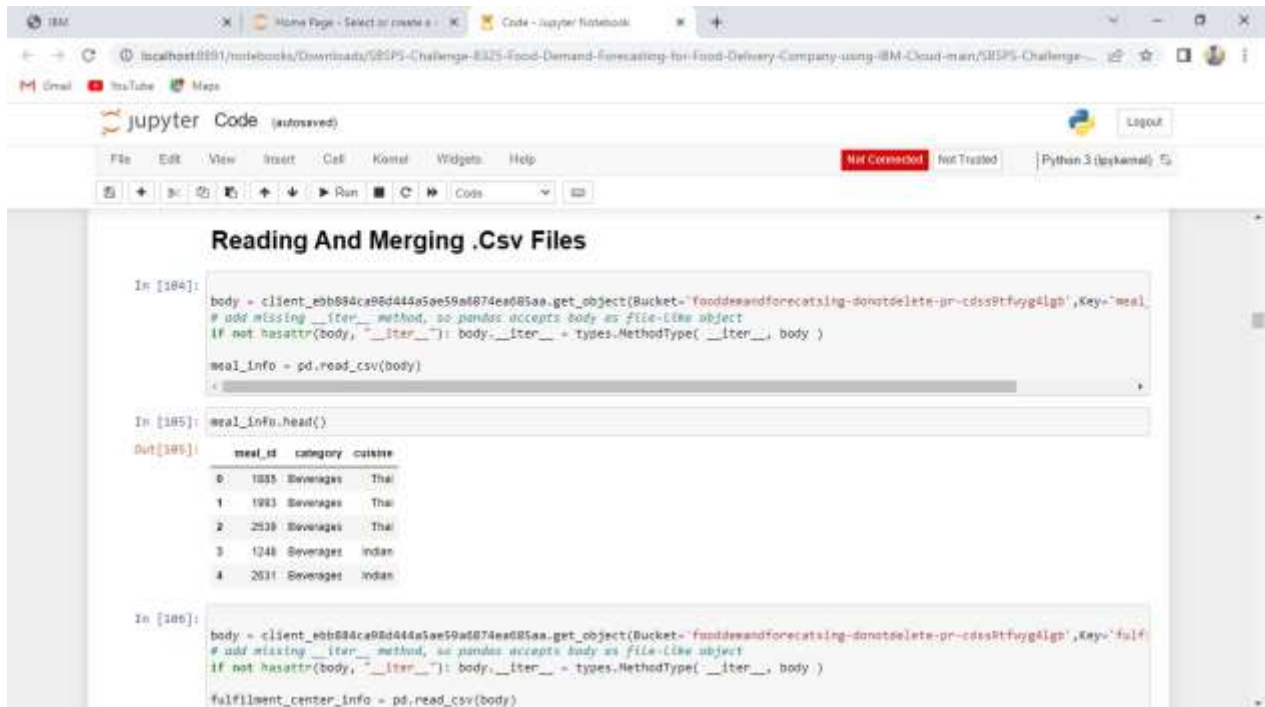
We notice that center\_id column in trainfinal.csv is similar to center\_id in fulfillment\_center\_info.csv dataset. Let us merge these two datasets, trainfinal.csv and fulfillment\_center\_info.csv using common key center\_id and store it back in trainfinal. Display the first five rows of trainfinal using head()

```
In [189]: trainfinal = pd.merge(trainfinal, fulfillment_center_info, on="center_id", how="outer")
trainfinal.head()
```

```
Out[189]:
```

	id	week	center_id	meal_id	checkout_price	base_price	essaler_for_promotion	homepage_featured	num_orders	category	cuisine	city_code	region
0	1379588	1	55	1885	138.83	152.29	0	0	177	Beverages	Thai	647	
1	1018794	2	55	1925	125.83	152.29	0	0	323	Beverages	Thai	647	
2	1186273	3	55	1885	132.92	133.92	0	0	96	Beverages	Thai	647	
3	1116527	4	55	1885	135.93	134.88	0	0	163	Beverages	Thai	647	
4	1342872	5	55	1885	149.50	147.50	0	0	215	Beverages	Thai	647	

## Team Member 2



The screenshot shows a Jupyter Notebook interface with the title "Reading And Merging .Csv Files". The code in the notebook reads a CSV file from a bucket and displays its first five rows.

```
In [184]: body = client_ebb894ca9d444a5ae59a6874ea085aa.get_object(Bucket='fooddemandforecasting-donotdelete-pr-cdss9tfvvg4lgb',Key='meal_
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, '__iter__'): body.__iter__ = types.MethodType(__iter__, body)

meal_info = pd.read_csv(body)

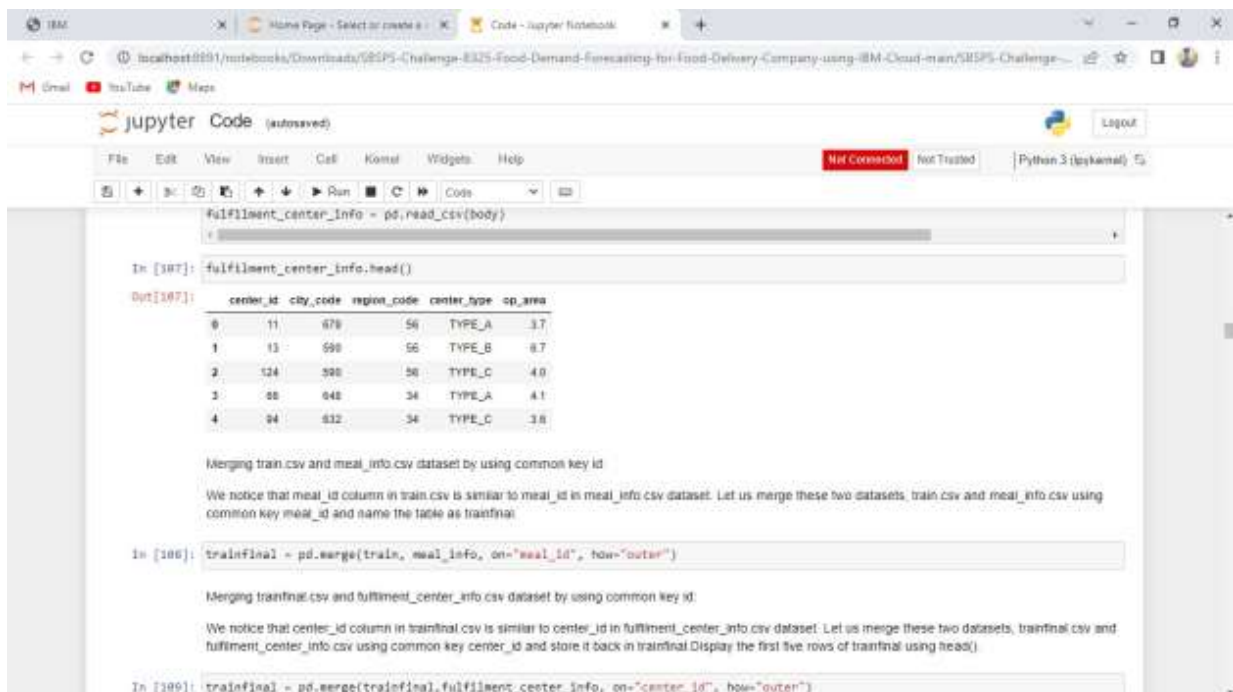
In [185]: meal_info.head()
```

Out[185]:

	meal_id	category	cuisine
0	1885	Beverages	Thai
1	1983	Beverages	Thai
2	2538	Beverages	Thai
3	1248	Beverages	Indian
4	2631	Beverages	Indian

```
In [186]: body = client_ebb894ca9d444a5ae59a6874ea085aa.get_object(Bucket='fooddemandforecasting-donotdelete-pr-cdss9tfvvg4lgb',Key='fulf
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, '__iter__'): body.__iter__ = types.MethodType(__iter__, body)

fulfilment_center_info = pd.read_csv(body)
```



The screenshot shows a Jupyter Notebook interface with the title "Reading And Merging .Csv Files". The code in the notebook reads a CSV file from a bucket and displays its first five rows.

```
fulfilment_center_info = pd.read_csv(body)

In [187]: fulfilment_center_info.head()
```

Out[187]:

	center_id	city_code	region_code	center_type	op_area
0	11	678	56	TYPE_A	3.7
1	13	598	56	TYPE_B	8.7
2	124	598	56	TYPE_C	4.0
3	88	648	34	TYPE_A	4.1
4	94	632	34	TYPE_C	3.8

Merging train.csv and meal\_info.csv dataset by using common key id

We notice that meal\_id column in train.csv is similar to meal\_id in meal\_info.csv dataset. Let us merge these two datasets, train.csv and meal\_info.csv using common key meal\_id and name the table as trainfinal:

```
In [188]: trainfinal = pd.merge(train, meal_info, on="meal_id", how="outer")
```

Merging trainfinal.csv and fulfillment\_center\_info.csv dataset by using common key id

We notice that center\_id column in trainfinal.csv is similar to center\_id in fulfillment\_center\_info.csv dataset. Let us merge these two datasets, trainfinal.csv and fulfillment\_center\_info.csv using common key center\_id and store it back in trainfinal. Display the first five rows of trainfinal using head():

```
In [189]: trainfinal = pd.merge(trainfinal, fulfilment_center_info, on="center_id", how="outer")
```

Merging train.csv and meal\_info.csv dataset by using common key id.

We notice that meal\_id column in train.csv is similar to meal\_id in meal\_info.csv dataset. Let us merge these two datasets, train.csv and meal\_info.csv using common key meal\_id and name the table as trainfinal.

```
In [188]: trainfinal = pd.merge(train, meal_info, on="meal_id", how="outer")
```

Merging trainfinal.csv and fulfillment\_center\_info.csv dataset by using common key id.

We notice that center\_id column in trainfinal.csv is similar to center\_id in fulfillment\_center\_info.csv dataset. Let us merge these two datasets, trainfinal.csv and fulfillment\_center\_info.csv using common key center\_id and store it back in trainfinal. Display the first five rows of trainfinal using head().

```
In [189]: trainfinal = pd.merge(trainfinal, fulfillment_center_info, on="center_id", how="outer")
trainfinal.head()
```

```
Out[189]:
```

	id	week	center_id	meal_id	checkout_price	base_price	emailer_for_promotion	homepage_featured	num_orders	category	cuisine	city_code	region
0	1370988	1	55	1885	136.83	152.29	0	0	177	Beverages	Thai	547	
1	1018704	2	55	1885	135.83	152.29	0	0	323	Beverages	Thai	547	
2	1196273	3	55	1885	132.92	133.92	0	0	98	Beverages	Thai	547	
3	1110527	4	55	1885	135.98	134.88	0	0	103	Beverages	Thai	547	
4	1343872	5	55	1885	145.50	147.50	0	0	215	Beverages	Thai	547	

Team Member 3

### Reading And Merging .Csv Files

```
In [184]: body = client_ebb884ca98d444a5ae59a6874ea085aa.get_object(Bucket='fooddemandforecasting-donotdelete-pr-cdss9tfuyg4lgh', Key='meal_
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType(__iter__, body)

meal_info = pd.read_csv(body)
```

```
In [185]: meal_info.head()
```

```
Out[185]:
```

	meal_id	category	cuisine
0	1885	Beverages	Thai
1	1983	Beverages	Thai
2	2539	Beverages	Thai
3	1248	Beverages	Indian
4	2531	Beverages	Indian

```
In [186]: body = client_ebb884ca98d444a5ae59a6874ea085aa.get_object(Bucket='fooddemandforecasting-donotdelete-pr-cdss9tfuyg4lgh', Key='fulf
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType(__iter__, body)

fulfillment_center_info = pd.read_csv(body)
```



IBM Home Page - Select or create a ... Code - Jupyter Notebook

localhost8891/notebooks/Downloads/SESPS-Challenge-8325-Food-Demand-Forecasting-for-Food-Delivery-Company-using-IBM-Cloud-man/SESPS-Challenge...

jupyter Code (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Connected Not Trusted Python 3 (ipykernel)

```
fulfillment_center_info = pd.read_csv(body)
```

In [187]: fulfillment\_center\_info.head()

Out[187]:

	center_id	city_code	region_code	center_type	op_area
0	11	678	56	TYPE_A	3.7
1	13	598	56	TYPE_B	8.7
2	124	598	56	TYPE_C	4.0
3	88	648	34	TYPE_A	4.1
4	84	632	34	TYPE_C	3.8

Merging train.csv and meal\_info.csv dataset by using common key id

We notice that meal\_id column in train.csv is similar to meal\_id in meal\_info.csv dataset. Let us merge these two datasets, train.csv and meal\_info.csv using common key meal\_id and name the table as trainfinal

In [188]: trainfinal = pd.merge(train, meal\_info, on="meal\_id", how="outer")

Merging trainfinal.csv and fulfillment\_center\_info.csv dataset by using common key id

We notice that center\_id column in trainfinal.csv is similar to center\_id in fulfillment\_center\_info.csv dataset. Let us merge these two datasets, trainfinal.csv and fulfillment\_center\_info.csv using common key center\_id and store it back in trainfinal. Display the first five rows of trainfinal using head()

In [189]: trainfinal = pd.merge(trainfinal, fulfillment\_center\_info, on="center\_id", how="outer")

IBM Home Page - Select or create a ... Code - Jupyter Notebook

localhost8891/notebooks/Downloads/SESPS-Challenge-8325-Food-Demand-Forecasting-for-Food-Delivery-Company-using-IBM-Cloud-man/SESPS-Challenge...

jupyter Code (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Connected Not Trusted Python 3 (ipykernel)

3	88	648	34	TYPE_A	4.1
4	84	632	34	TYPE_C	3.8

Merging train.csv and meal\_info.csv dataset by using common key id

We notice that meal\_id column in train.csv is similar to meal\_id in meal\_info.csv dataset. Let us merge these two datasets, train.csv and meal\_info.csv using common key meal\_id and name the table as trainfinal

In [188]: trainfinal = pd.merge(train, meal\_info, on="meal\_id", how="outer")

Merging trainfinal.csv and fulfillment\_center\_info.csv dataset by using common key id

We notice that center\_id column in trainfinal.csv is similar to center\_id in fulfillment\_center\_info.csv dataset. Let us merge these two datasets, trainfinal.csv and fulfillment\_center\_info.csv using common key center\_id and store it back in trainfinal. Display the first five rows of trainfinal using head()

In [189]: trainfinal = pd.merge(trainfinal, fulfillment\_center\_info, on="center\_id", how="outer")

trainfinal.head()

Out[189]:

	id	week	center_id	meal_id	checkout_price	base_price	enable_for_promotion	homepage_featured	num_orders	category	country	city_code	region
0	1379588	1	55	1885	138.83	152.29	0	0	177	Beverages	Thai	647	
1	1018794	2	55	1925	135.83	152.29	0	0	323	Beverages	Thai	647	
2	1186273	3	55	1885	132.92	133.92	0	0	96	Beverages	Thai	647	
3	1116527	4	55	1885	135.93	134.88	0	0	163	Beverages	Thai	647	
4	1342872	5	55	1885	149.50	147.50	0	0	215	Beverages	Thai	647	