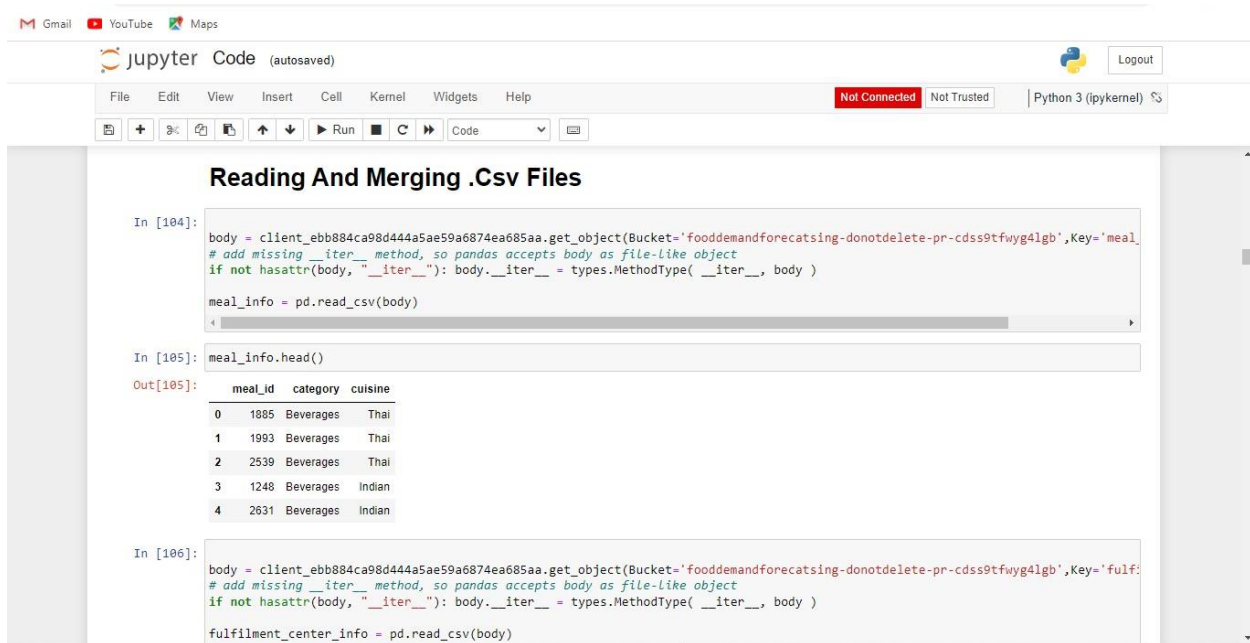


TEAM ID: PNT2022TMID32006

## PROJECT NAME: DemandEst - AI powered Food DemandForecaster

Team Leader



The screenshot shows a Jupyter Notebook interface with the title "Reading And Merging .Csv Files". The notebook is running on Python 3 (ipykernel). The code in the first cell reads a CSV file named 'meal\_info.csv' from a Google Cloud Storage bucket. The code in the second cell displays the first five rows of the 'meal\_info' DataFrame. The code in the third cell reads another CSV file named 'fulfilment\_center\_info.csv' from the same bucket.

```
In [104]: body = client_ebb884ca98d444a5ae59a6874ea685aa.get_object(Bucket='fooddemandforecatsing-donotdelete-pr-cdss9tfwyg4lgb',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)

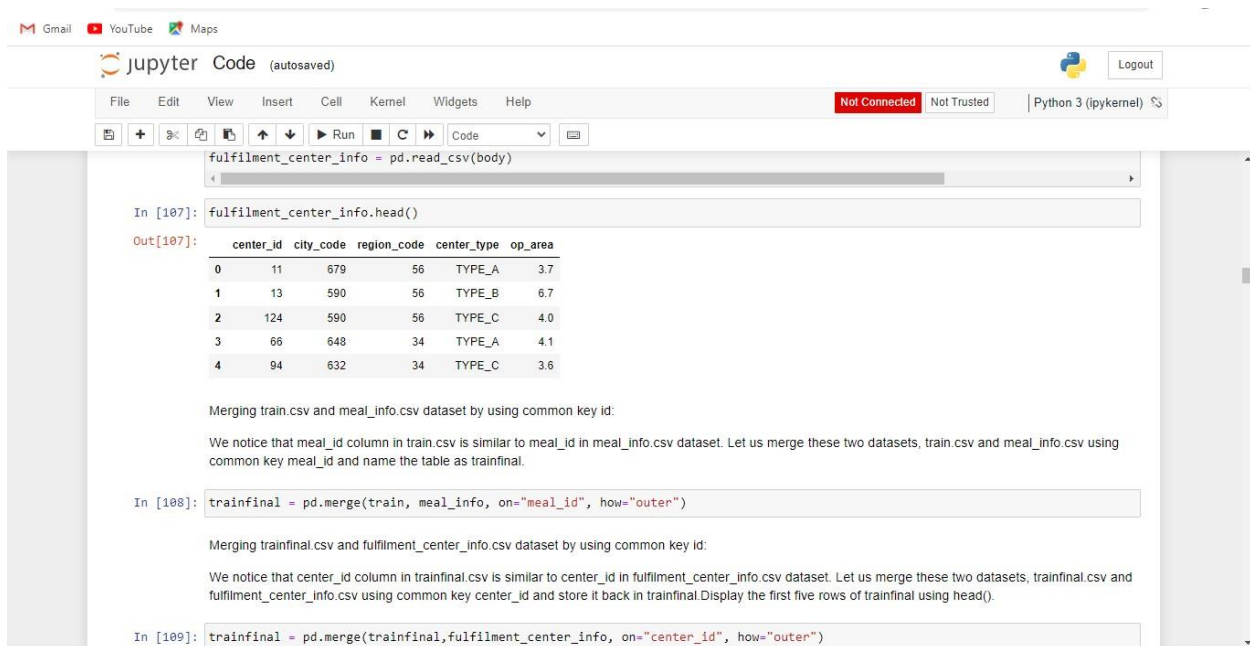
In [105]: meal_info.head()

Out[105]:
```

	meal_id	category	cuisine
0	1885	Beverages	Thai
1	1993	Beverages	Thai
2	2539	Beverages	Thai
3	1248	Beverages	Indian
4	2631	Beverages	Indian

```
In [106]: body = client_ebb884ca98d444a5ae59a6874ea685aa.get_object(Bucket='fooddemandforecatsing-donotdelete-pr-cdss9tfwyg4lgb',Key='fulfilment_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 )

fulfilment_center_info = pd.read_csv(body)
```



The screenshot shows a Jupyter Notebook interface with the title "Reading And Merging .Csv Files". The notebook is running on Python 3 (ipykernel). The code in the first cell reads a CSV file named 'fulfilment\_center\_info.csv' from a Google Cloud Storage bucket. The code in the second cell displays the first five rows of the 'fulfilment\_center\_info' DataFrame. The code in the third cell merges the 'train' DataFrame with the 'meal\_info' DataFrame using the 'meal\_id' column as the common key. The code in the fourth cell merges the 'trainfinal' DataFrame with the 'fulfilment\_center\_info' DataFrame using the 'center\_id' column as the common key. The code in the fifth cell displays the first five rows of the final merged DataFrame, 'trainfinal'.

```
fulfilment_center_info = pd.read_csv(body)

In [107]: fulfilment_center_info.head()

Out[107]:
```

	center_id	city_code	region_code	center_type	op_area
0	11	679	56	TYPE_A	3.7
1	13	590	56	TYPE_B	6.7
2	124	590	56	TYPE_C	4.0
3	66	648	34	TYPE_A	4.1
4	94	632	34	TYPE_C	3.6

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

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

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

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

Gmail YouTube Maps

jupyter Code (autosaved) Logout

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

3 66 648 34 TYPE\_A 4.1  
4 94 632 34 TYPE\_C 3.6

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

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

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

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

```
Out[109]:
```

	id	week	center_id	meal_id	checkout_price	base_price	emailer_for_promotion	homepage_featured	num_orders	category	cuisine	city_code	region
0	1379560	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	96	Beverages	Thai	647	
3	1116527	4	55	1885	135.86	134.86	0	0	163	Beverages	Thai	647	
4	1343872	5	55	1885	146.50	147.50	0	0	215	Beverages	Thai	647	

## Team Member 1

Gmail YouTube Maps

jupyter Code (autosaved) Logout

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

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

```
In [107]: fulfilment_center_info.head()
```

```
Out[107]:
```

	center_id	city_code	region_code	center_type	op_area
0	11	679	56	TYPE_A	3.7
1	13	590	56	TYPE_B	6.7
2	124	590	56	TYPE_C	4.0
3	66	648	34	TYPE_A	4.1
4	94	632	34	TYPE_C	3.6

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

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

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

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

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

```
In [107]: fulfilment_center_info.head()
```

```
Out[107]:
```

	center_id	city_code	region_code	center_type	op_area
0	11	679	56	TYPE_A	3.7
1	13	590	56	TYPE_B	6.7
2	124	590	56	TYPE_C	4.0
3	66	648	34	TYPE_A	4.1
4	94	632	34	TYPE_C	3.6

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

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

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

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

```
3 66 648 34 TYPE_A 4.1
```

4	94	632	34	TYPE_C	3.6
---	----	-----	----	--------	-----

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

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

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

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

```
Out[109]:
```

	id	week	center_id	meal_id	checkout_price	base_price	emailer_for_promotion	homepage_featured	num_orders	category	cuisine	city_code	region_code
0	1379560	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	96	Beverages	Thai	647	
3	1116527	4	55	1885	135.86	134.86	0	0	163	Beverages	Thai	647	
4	1343872	5	55	1885	146.50	147.50	0	0	215	Beverages	Thai	647	

## Team Member 2

Gmail YouTube Maps

jupyter Code (autosaved) Logout

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

Run Code

### Reading And Merging .Csv Files

```
In [104]: body = client_ebb884ca98d444a5ae59a6874ea685aa.get_object(Bucket='fooddemandforecatsing-donotdelete-pr-cdss9tfwyg4lgb',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)
```

```
In [105]: meal_info.head()
```

```
Out[105]:
```

	meal_id	category	cuisine
0	1885	Beverages	Thai
1	1993	Beverages	Thai
2	2539	Beverages	Thai
3	1248	Beverages	Indian
4	2631	Beverages	Indian

```
In [106]: body = client_ebb884ca98d444a5ae59a6874ea685aa.get_object(Bucket='fooddemandforecatsing-donotdelete-pr-cdss9tfwyg4lgb',Key='fulfilment_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 )

fulfilment_center_info = pd.read_csv(body)
```

Gmail YouTube Maps

jupyter Code (autosaved) Logout

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

Run Code

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

```
In [107]: fulfilment_center_info.head()
```

```
Out[107]:
```

	center_id	city_code	region_code	center_type	op_area
0	11	679	56	TYPE_A	3.7
1	13	590	56	TYPE_B	6.7
2	124	590	56	TYPE_C	4.0
3	66	648	34	TYPE_A	4.1
4	94	632	34	TYPE_C	3.6

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

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

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

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

Gmail YouTube Maps

jupyter Code (autosaved) Python 3 (ipykernel) Logout

File Edit View Insert Cell Kernel Widgets Help

Not Connected Not Trusted

Code

```

3      66      648      34      TYPE_A      4.1
4      94      632      34      TYPE_C      3.6

```

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

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

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

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

```
Out[109]:
```

	id	week	center_id	meal_id	checkout_price	base_price	emailer_for_promotion	homepage_featured	num_orders	category	cuisine	city_code	regio
0	1379560	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	96	Beverages	Thai	647	
3	1116527	4	55	1885	135.86	134.86	0	0	163	Beverages	Thai	647	
4	1343872	5	55	1885	146.50	147.50	0	0	215	Beverages	Thai	647	

## Team Member 3

Gmail YouTube Maps

jupyter Code (autosaved) Python 3 (ipykernel) Logout

File Edit View Insert Cell Kernel Widgets Help

Not Connected Not Trusted

Code

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

```
In [107]: fulfilment_center_info.head()
```

```
Out[107]:
```

	center_id	city_code	region_code	center_type	op_area
0	11	679	56	TYPE_A	3.7
1	13	590	56	TYPE_B	6.7
2	124	590	56	TYPE_C	4.0
3	66	648	34	TYPE_A	4.1
4	94	632	34	TYPE_C	3.6

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

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

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

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



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

In [107]: fulfilment\_center\_info.head()

```
Out[107]:
```

	center_id	city_code	region_code	center_type	op_area
0	11	679	56	TYPE_A	3.7
1	13	590	56	TYPE_B	6.7
2	124	590	56	TYPE_C	4.0
3	66	648	34	TYPE_A	4.1
4	94	632	34	TYPE_C	3.6

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

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

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

In [109]: trainfinal = pd.merge(trainfinal, fulfilment\_center\_info, on="center\_id", how="outer")

```
3 66 648 34 TYPE_A 4.1
```

4	94	632	34	TYPE_C	3.6
---	----	-----	----	--------	-----

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

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

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

In [109]: trainfinal = pd.merge(trainfinal, fulfilment\_center\_info, on="center\_id", how="outer")  
trainfinal.head()

```
Out[109]:
```

	id	week	center_id	meal_id	checkout_price	base_price	emailer_for_promotion	homepage_featured	num_orders	category	cuisine	city_code	regio
0	1379560	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	96	Beverages	Thai	647	
3	1116527	4	55	1885	135.86	134.86	0	0	163	Beverages	Thai	647	
4	1343872	5	55	1885	146.50	147.50	0	0	215	Beverages	Thai	647	