Implementation of Google Cloud Dataflow Pipeline

Requirements:
Create a pipeline in Dataflow that reads data from a csv file, applies transformations and inserts resulting data into the BigQuery table.
1. Download and store NYC Airbnb dataset in a GCS bucket.
2. Create a Dataflow batch job in Python that can read and process this file.
3. In the Dataflow job, apply a "Group By" transform to get the count of listings by the "neighbourhood" field.
4. Store both the original csv data and the transformed data into their own separate BigQuery tables.

Dataset:
New York City Airbnb Open Data
https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data?select=AB_NYC_2019.csv

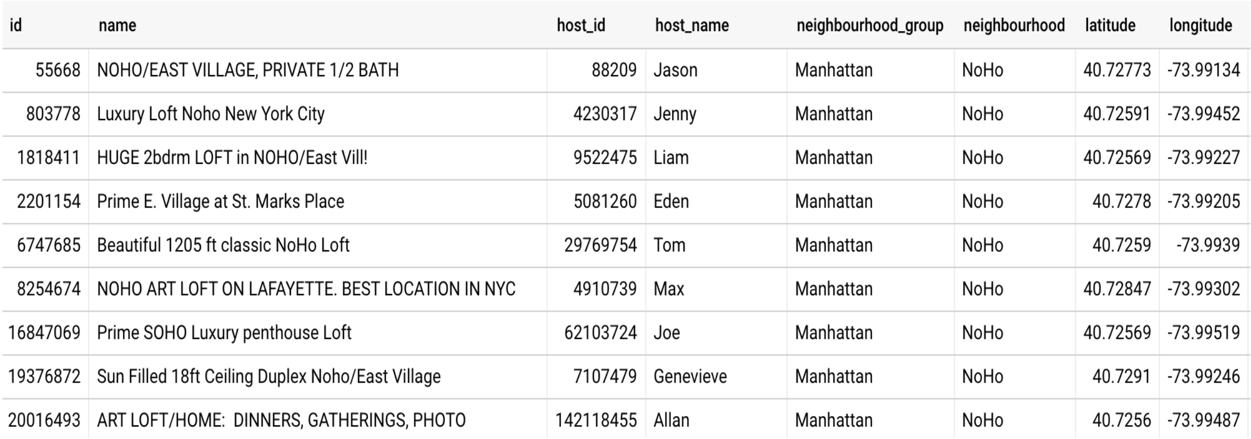
Setup:
1. Sign up on GCP.
2. On the GCP console, create a project "Project-1".
3. On the GCP console, create a bucket "input-bucket-2".
4. Install Apache Beam SDK for Python on MacOS using the link:
https://cloud.google.com/dataflow/docs/quickstarts/quickstart-python
5. Create a virtual environment using Conda and run the Python code containing the Dataflow pipelines.

Exploratory Data Analysis:
Performed the following checks for exploratory data analysis using Jupyter notebook:
1. Check the size of the dataset.
2. Check the datatype of each entity (column).
3. Check if any column/row has null values.
4. Replace the null values with relevant data.

Code: data_preprocessing.ipynb

Input:

1. Uploaded the original CSV dataset in GCS manually.
2. Uploaded the original CSV dataset in the BigQuery table with “Allow quoted newlines” option enabled.



id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude
55668	NOHO/EAST VILLAGE, PRIVATE 1/2 BATH	88209	Jason	Manhattan	NoHo	40.72773	-73.99134
803778	Luxury Loft Noho New York City	4230317	Jenny	Manhattan	NoHo	40.72591	-73.99452
1818411	HUGE 2bdrm LOFT in NOHO/East Vill!	9522475	Liam	Manhattan	NoHo	40.72569	-73.99227
2201154	Prime E. Village at St. Marks Place	5081260	Eden	Manhattan	NoHo	40.7278	-73.99205
6747685	Beautiful 1205 ft classic NoHo Loft	29769754	Tom	Manhattan	NoHo	40.7259	-73.9939
8254674	NOHO ART LOFT ON LAFAYETTE. BEST LOCATION IN NYC	4910739	Max	Manhattan	NoHo	40.72847	-73.99302
16847069	Prime SOHO Luxury penthouse Loft	62103724	Joe	Manhattan	NoHo	40.72569	-73.99519
19376872	Sun Filled 18ft Ceiling Duplex Noho/East Village	7107479	Genevieve	Manhattan	NoHo	40.7291	-73.99246
20016493	ART LOFT/HOME: DINNERS, GATHERINGS, PHOTO	142118455	Allan	Manhattan	NoHo	40.7256	-73.99487

Fig 1: Original dataset in BigQuery table

Dataflow jobs:

Dataflow job #1:
While working with the “name”, “neighbourhood_group” and “host_name” columns, some records had punctuation marks like commas, quotes and newline characters. Due to this, the comma separation code treated these entries as two and gave “Index out of range” error.
To solve this, created the first Dataflow job to:
1. Create a Pandas dataframe from the original CSV file and drop these columns.
2. Generate an intermediate output file “processed_data.csv”.
3. Store this file in GCS.

To store the pandas dataframe in GCS, use the “google-cloud-storage” module. By default, this was not installed in the Dataflow Runner environment. So, created a “requirements.txt” file and passed it as an argument while invoking Dataflow Runner.

Code: dataflow_preprocessing.py

Command used for invocation:

```
python dataflow_preprocessing.py \
   --region us-west2 \
   --runner DataflowRunner \
   --project cool-adviser-320919 \
   --temp_location gs://input-bucket-2/tmp/ \
   --input gs://input-bucket-2/processed_data.csv
```

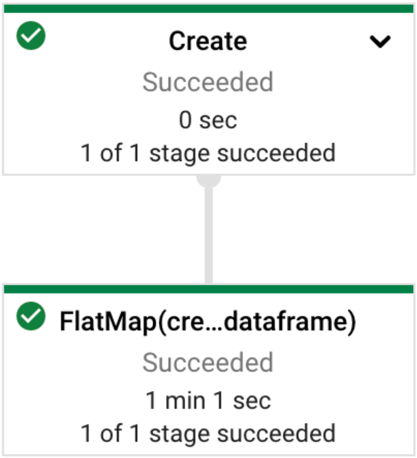


Fig 2: Dataflow Job 1 - Graph View

Dataflow job #2:

Created a second Dataflow job to:

1. Read the intermediate output file.
2. Parse the intermediate output file and separate on comma delimiter using “csv.reader”.
3. Extract the neighborhood, group according to neighborhood and count the number of occurrences using Apache beam “ParDo”, “GroupByKey” and “Map” functions.
4. Write the final output in the BigQuery dataset “result” and store the output in the table “neighborhood_count”.

Code: dataflow_pipeline.py

Command used for invocation:

```
python dataflow_pipeline.py \
    --region us-west2 \
    --runner DataflowRunner \
    --project cool-adviser-320919 \
    --temp_location gs://input-bucket-2/tmp \
    --requirements_file requirements.txt \
    --save_main_session True
```

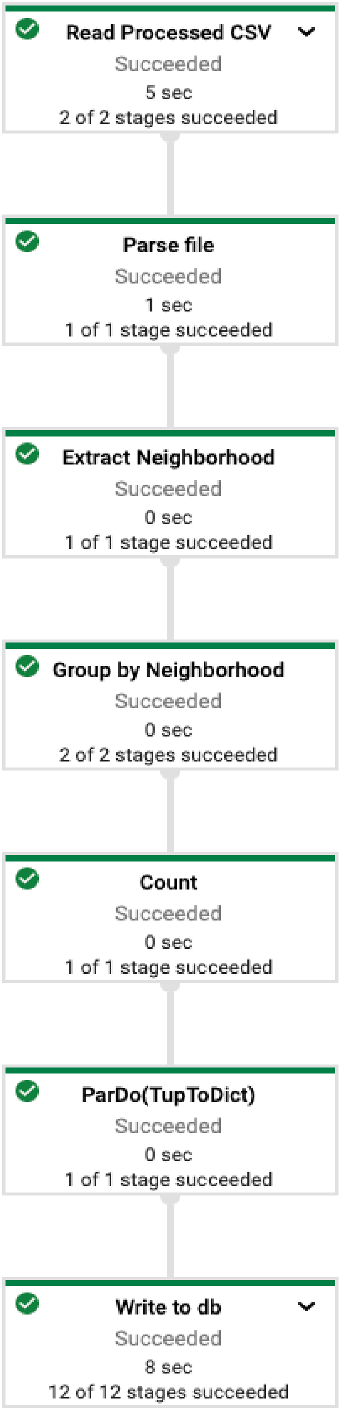
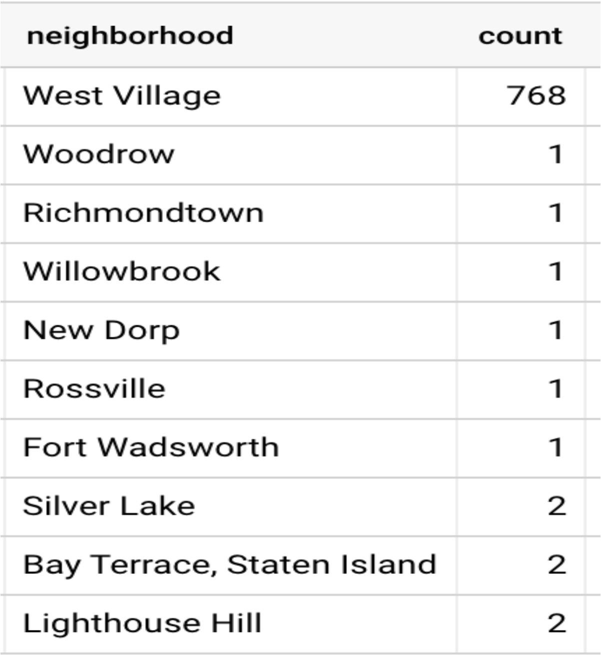


Fig 3: Dataflow Job 2 - Graph View

Output:

Total number of rows: 221



neighborhood	count
West Village	768
Woodrow	1
Richmondtown	1
Willowbrook	1
New Dorp	1
Rossville	1
Fort Wadsworth	1
Silver Lake	2
Bay Terrace, Staten Island	2
Lighthouse Hill	2

Fig 4: Result in BigQuery table

Files uploaded:

1. data_preprocessing.ipynb
2. dataflow_preprocessing.py
3. processed_data.csv
4. dataflow_pipeline.py
5. requirements.txt
6. Dataflow_Pipeline_Report.pdf

Source code:

https://github.com/PurvaDeekshit/Dataflow-AB-NYC-2019

Project Contributor:

Purva Deekshit