

05b_retail_forecast_gcloud_execution.md

🔗 Retail Forecast using sessions in Serverless Spark through Vertex AI

Following are the lab modules:

1. [Understanding Data](#)
2. [Solution Architecture](#)
3. [Declaring Variables](#)
4. [Execution](#)
5. [Logging](#)

1. Understanding Data

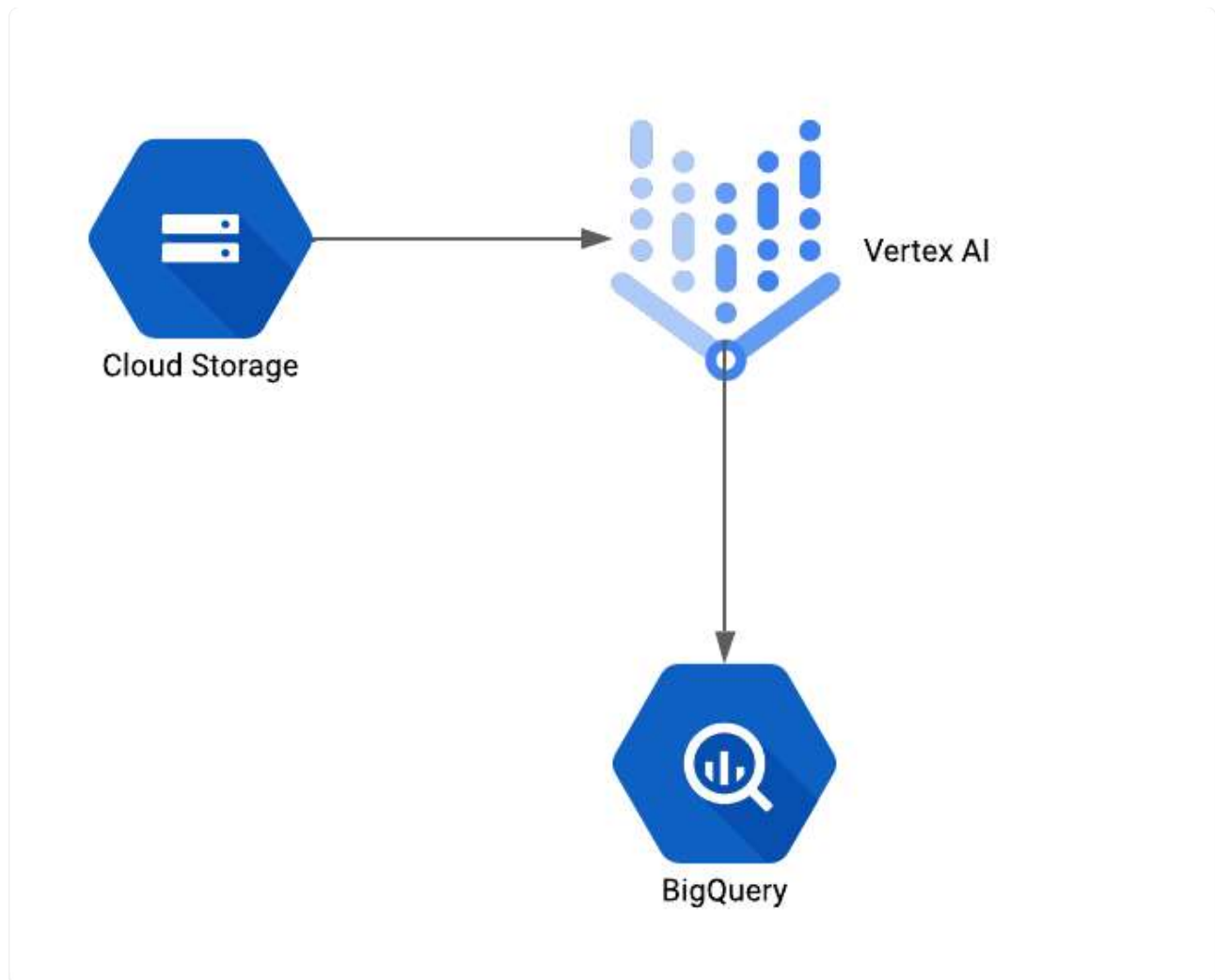
The datasets used for this project are

1. [Aisles data](#).
2. [Departments data](#) .
3. [Orders data](#).
4. [Products data](#).
5. [Order_products__prior](#).
6. [Order_products__train](#).

- Aisles: This table includes all aisles. It has a single primary key (aisle_id)
- Departments: This table includes all departments. It has a single primary key (department_id)
- Products: This table includes all products. It has a single primary key (product_id)
- Orders: This table includes all orders, namely prior, train, and test. It has single primary key (order_id).
- Order_products_train: This table includes training orders. It has a composite primary key (order_id and product_id) and indicates whether a product in an order is a reorder or not (through the reordered variable).

- `Order_products_prior` : This table includes prior orders. It has a composite primary key (`order_id` and `product_id`) and indicates whether a product in an order is a reorder or not (through the `reordered` variable).

2. Solution Architecture



3. Declaring cloud shell variables

3.1 Set the `PROJECT_ID` in Cloud Shell

Open Cloud shell or navigate to shell.cloud.google.com

Run the below

```
gcloud config set project $PROJECT_ID
```

3.2 Verify the PROJECT_ID in Cloud Shell

Next, run the following command in cloud shell to ensure that the current project is set correctly:

```
gcloud config get-value project
```

3.3 Declare the variables

Based on the prereqs and checklist, declare the following variables in cloud shell by replacing with your values:

```
PROJECT_ID=$(gcloud config get-value project)    #current GCP project where we ar
REGION=                                           #GCP region where all our resour
SUBNET=                                           #subnet which has private google
BUCKET_CODE=                                     #GCP bucket where our code, data
BUCKET_PHS=                                      #bucket where our application lc
HISTORY_SERVER_NAME=                            #name of the history server whic
BQ_DATASET_NAME=                                #BigQuery dataset where all the
SESSION_NAME=                                   # Serverless Session name.
UMSA_NAME=                                       #user managed service account re
SERVICE_ACCOUNT=$UMSA_NAME@$PROJECT_ID.iam.gserviceaccount.com
NAME=                                           #Your unique identifier
```

Note: For all the variables except 'NAME', please ensure to use the values provided by the admin team.

3.4 Update Cloud Shell SDK version

Run the below on cloud shell-

```
gcloud components update
```

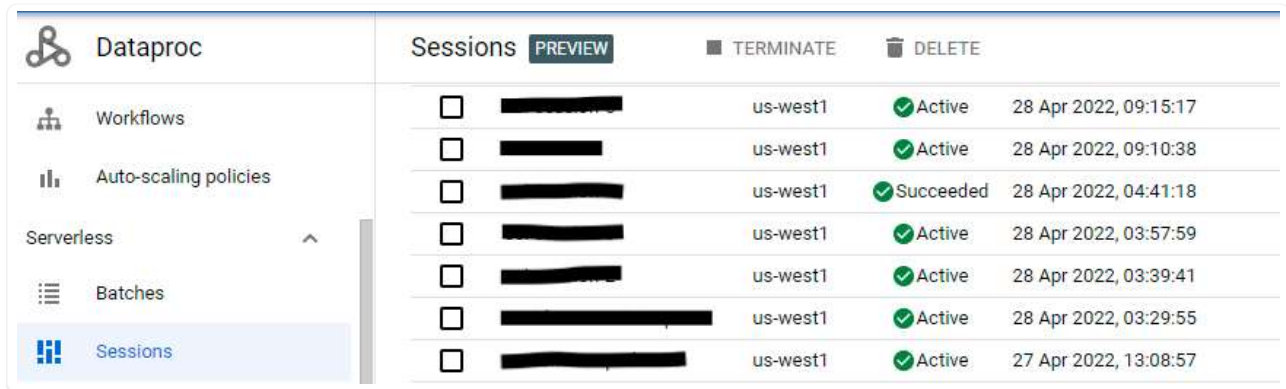
4. Execution

4.1. Run the Batch by creating sessions.

Run the below on cloud shell to create session. -

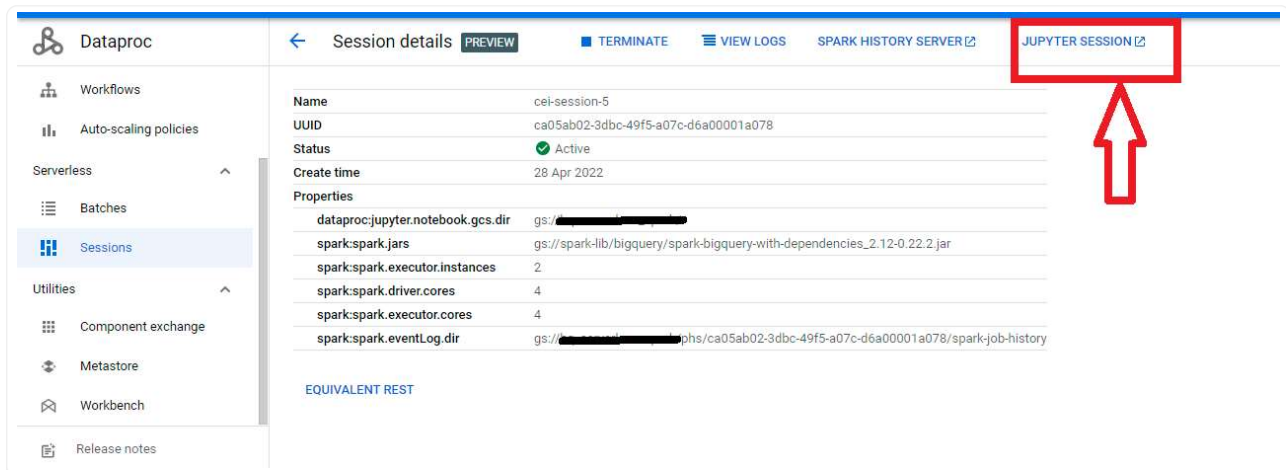
```
gcloud beta dataproc sessions create spark $SESSION_NAME \
--project=${PROJECT_ID} \
--location=${REGION} \
--property=spark.jars=gs://spark-lib/bigquery/spark-bigquery-with-dependencies_2.12-
--history-server-cluster=projects/$PROJECT_ID/regions/$REGION/clusters/$HISTORY_SERV
--subnet=$SUBNET \
--property=dataproc:jupyter.notebook.gcs.dir=$BUCKET_CODE
```

- Once the serverless spark session has been created, open the session and click on the jupyter session.



The screenshot shows the Google Cloud Dataproc console. On the left is a sidebar with navigation links: Workflows, Auto-scaling policies, Serverless, Batches, and Sessions (highlighted). The main panel is titled 'Sessions' and has tabs for 'PREVIEW', 'TERMINATE', and 'DELETE'. It displays a table of sessions with columns for checkboxes, session names (redacted), location (us-west1), status (Active or Succeeded), and create time.

	Session Name	Location	Status	Create Time
<input type="checkbox"/>	[REDACTED]	us-west1	Active	28 Apr 2022, 09:15:17
<input type="checkbox"/>	[REDACTED]	us-west1	Active	28 Apr 2022, 09:10:38
<input type="checkbox"/>	[REDACTED]	us-west1	Succeeded	28 Apr 2022, 04:41:18
<input type="checkbox"/>	[REDACTED]	us-west1	Active	28 Apr 2022, 03:57:59
<input type="checkbox"/>	[REDACTED]	us-west1	Active	28 Apr 2022, 03:39:41
<input type="checkbox"/>	[REDACTED]	us-west1	Active	28 Apr 2022, 03:29:55
<input type="checkbox"/>	[REDACTED]	us-west1	Active	27 Apr 2022, 13:08:57



The screenshot shows the 'Session details' page for a session named 'cel-session-5'. The left sidebar is the same as the previous screenshot. The main panel shows details for the session, including Name, UUID, Status (Active), and Create time. Below this is a 'Properties' section with key-value pairs for various configuration parameters. A red box highlights the 'JUPYTER SESSION' link in the top right corner, with a red arrow pointing to it.

Session Details:

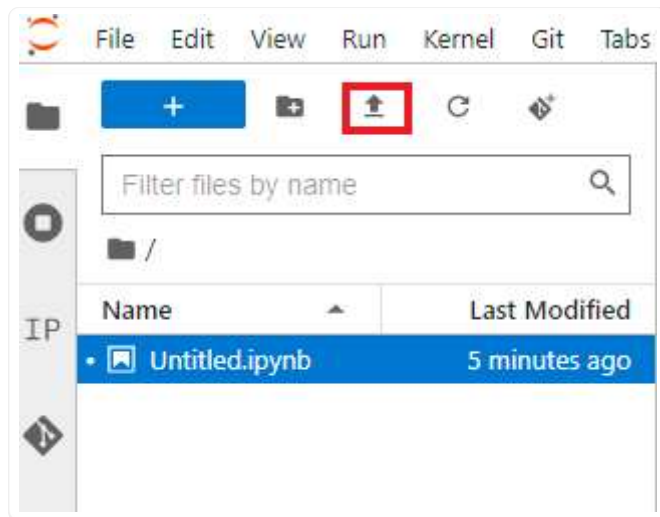
- Name: cel-session-5
- UUID: ca05ab02-3dbc-49f5-a07c-d6a00001a078
- Status: Active
- Create time: 28 Apr 2022

Properties:

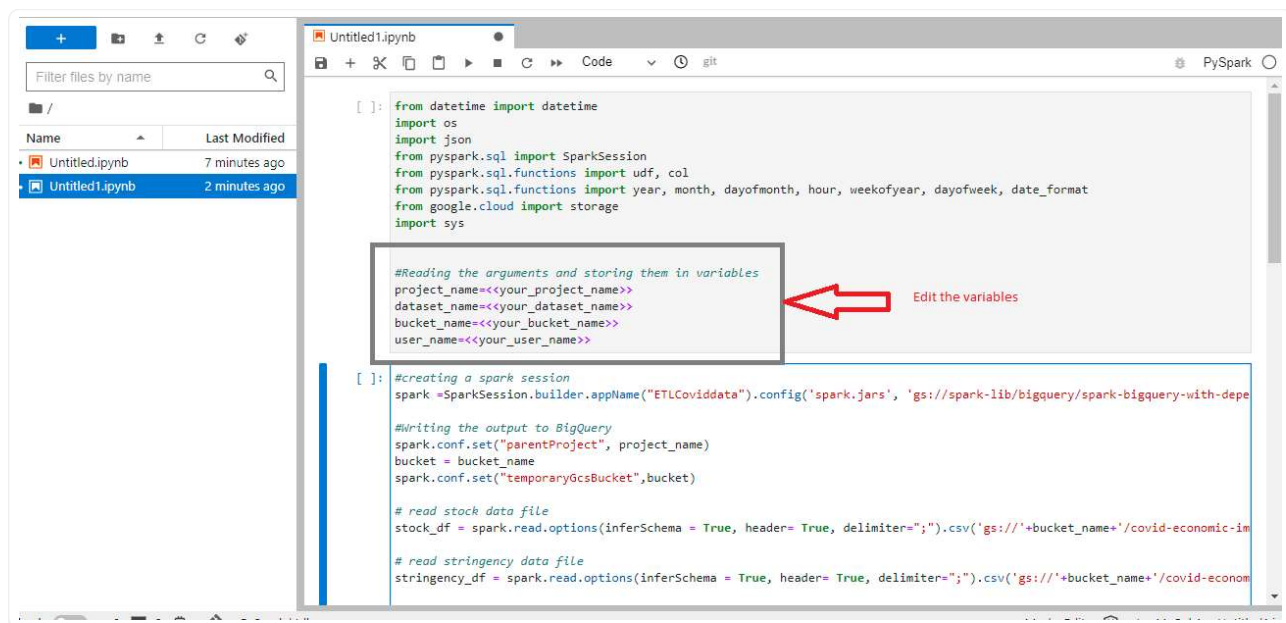
- dataproc:jupyter.notebook.gcs.dir: gs://[REDACTED]
- spark:spark.jars: gs://spark-lib/bigquery/spark-bigquery-with-dependencies_2.12-0.22.2.jar
- spark:spark.executor.instances: 2
- spark:spark.driver.cores: 4
- spark:spark.executor.cores: 4
- spark:spark.eventLog.dir: gs://[REDACTED]phs/ca05ab02-3dbc-49f5-a07c-d6a00001a078/spark-job-history

Link: JUPYTER SESSION

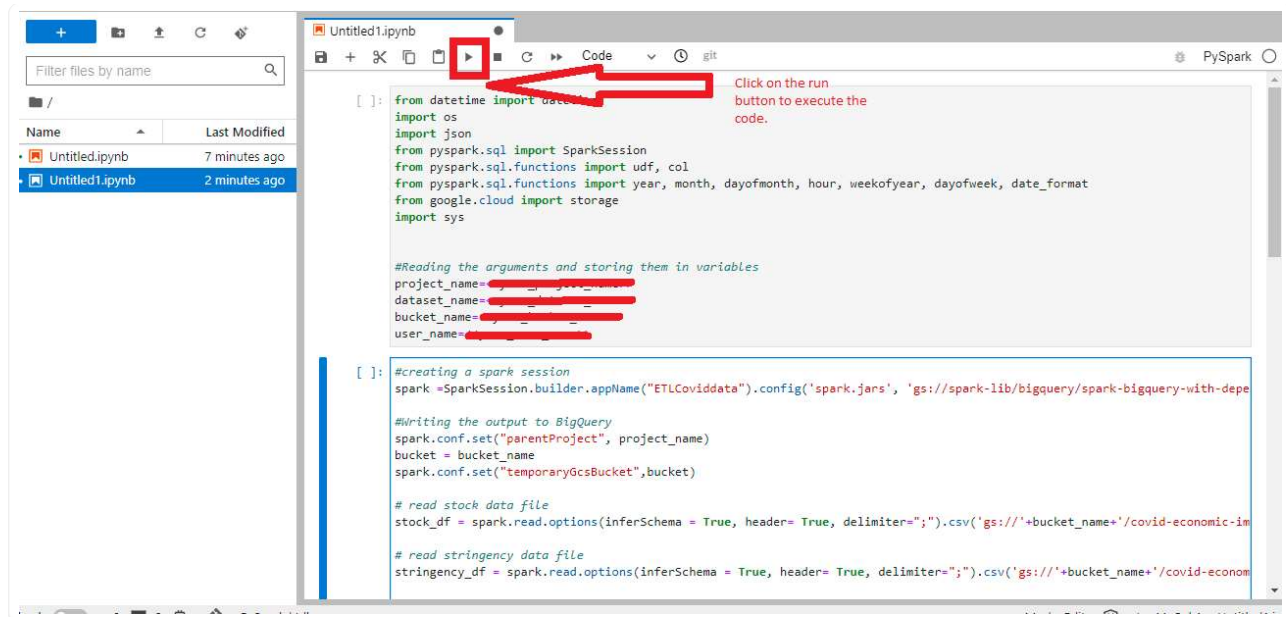
- Select Pyspark Kernel for the execution.



- Upload the notebook 00-scripts/retail-forecast.ipynb and edit the variables: project_name, dataset_name, bucket_name and name with your values.



- Hit the **Execute** button to execute the code.



4.2. Check the output table in BQ

Navigate to BigQuery Console, and check the **retail_forecast** dataset.

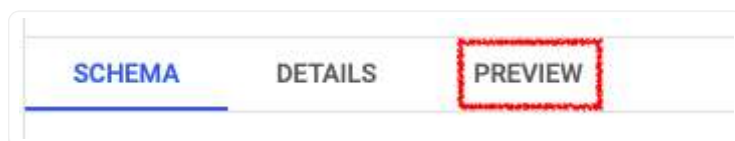
Once the data preparation batch is completed, two new tables

'<your_name_here>_train_data', '<your_name_here>_test_data',

'<your_name_here>_predictions_data' and '<your_name_here>_eval_output' will be created as shown below :

To view the data in these tables -

- Select the table from BigQuery Explorer by navigating 'project_id' > 'dataset' > 'table_name'
- Click on the **Preview** button to see the data in the table



Note: If the **Preview** button is not visible, run the below queries to view the data. However, these queries will be charged for the full table scan.

```
SELECT * FROM `<project_name>.<dataset_name>.<your_name_here>_train_data` LIMIT 10
SELECT * FROM `<project_name>.<dataset_name>.<your_name_here>_test_data` LIMIT 100
SELECT * FROM `<project_name>.<dataset_name>.<your_name_here>_predictions_data` LI
SELECT * FROM `<project_name>.<dataset_name>.<your_name_here>_eval_output` LIMIT 1
```

Note: Edit all occurrences of <project_name> and <dataset_name> to match the values of the variables PROJECT_ID, and BQ_DATASET_NAME respectively

The screenshot shows the Google Cloud Platform BigQuery console. On the left, the 'Explorer' pane displays a project structure with a folder containing 'predictions_data', 'test_data', and 'train_data'. The 'train_data' folder is highlighted with a red box. The main area shows a query editor with a query that selects from 'train_data' with a limit of 1000. Below the editor, the 'Query results' section displays a table with 13 columns: 'row', 'uwp_total_bought', 'Times_Bought_N', 'total_orders', 'first_order_number', 'Order_Range_D', 'uwp_reorder_ratio', 'u_total_orders', 'u_reordered_ratio', 'p_total_purchases', 'p_reorder_ratio', 'user_id', 'product_id', and 'reorder'. The table contains 10 rows of data. The 'Results per page' dropdown is set to 50, showing 1 - 50 of 1000 results.


5. Logging

5.1 Persistent History Server logs

To view the Persistent History server logs, click the 'View History Server' button on the Sessions monitoring page and the logs will be shown as below:

As the session is still in active state, we will be able to find the logs in show incomplete applications.

The screenshot shows the Google Cloud Platform Dataproc console. The left sidebar contains navigation links for 'Jobs on clusters', 'Clusters', 'Jobs', 'Workflows', 'Auto-scaling policies', 'Serverless', 'Batches', and 'Sessions'. The main area displays 'Session details' for a session named 'cel-session-5'. The session is in an 'Active' state and was created on '28 Apr 2022'. A red arrow points to the 'SPARK HISTORY SERVER' link in the top right corner of the session details pane. Below the session details, a table lists properties for the session, including 'dataproc:jupyter.notebook.gcs.dir', 'spark:spark.jars', 'spark:spark.executor.instances', 'spark:spark.driver.cores', 'spark:spark.executor.cores', and 'spark:spark.eventLog.dir'.

3.1.2

History Server

Event log directory: gs://[redacted]/phs/*/spark-job-history

Last updated: 2022-04-04 16:52:29


Client local time zone: Asia/Calcutta

Search:

Version	App ID	App Name	Driver Host	Started	Completed	Duration	Spark User	Last Updated	Event Log
3.2.1	[redacted]	[redacted]	10.122.15.217	2022-04-04 16:35:43	2022-04-04 16:36:44	1.0 min	spark	2022-04-04 16:36:45	Download

Showing 1 to 1 of 1 entries

[Show incomplete applications](#)

3.1.2

History Server

Event log directory: gs://[redacted]/phs/*/spark-job-history

Last updated: 2022-04-04 16:52:29

Client local time zone: Asia/Calcutta

Search:

Version	App ID	App Name	Driver Host	Started	Completed	Duration	Spark User	Last Updated	Event Log
3.2.1	app-20220404110546-0000	[redacted]	10.122.15.217	2022-04-04 16:35:43	2022-04-04 16:36:44	1.0 min	spark	2022-04-04 16:36:45	Download

Showing 1 to 1 of 1 entries

[Show incomplete applications](#)