CIS5200 Term Project Tutorial

Authors: Taya Stewart, Raymond Delgado, Navaneeth Visagan, Frank Chen, Ninelia Talverdi

Instructor: Jongwook Woo

Date: 12/17/2020

Lab Tutorial

Taya Stewart (tstewar2@calstatela.edu)

Raymond Delgado (rdelga40@calstatela.edu)

Navaneet Visagan (nvisaga@calstatela.edu)

Frank Chen (fchen26@calstatela.edu)

Ninelia Talverdi (ntalver2@calstatela.edu)

12/17/2020

E-commerce Analysis of Consumer Behavior using Hive

Objectives

Business data set and its customer review data is one of the popular areas for Big Data adoption. In this hands-on tutorial, you will learn how to use Big Data Compute Edition to:

- Upload and Download data file from the local system to Hadoop HDFS and vice versa

- Create tables and views in HDFS using HiveQL
- Create Hive queries to perform the analysis
- Use IBM Cognos Analytics, SAP Predictive Analytics, Tableau, Power BI for visualization

Prerequisites

Everything you need to go through the scripts and queries is already provisioned with the cluster. To analyze the data using BI tools, you need to have access to IBM Cognos Analytics, SAP Predictive Analytics, Tableau, Power BI.

Platform Spec

- Cluster Version – Oracle Big Data Compute Edition, 20.3.3-20
- Number of Nodes – 3
- Memory size – 160 GB
- # of OCPUs – 8
- CPU speed – 2.20 GHz
- HDFS capacity – 802 GB
- Local Storage – 202 GB

Step 1: Connect to Oracle Cloud: Big Data Compute

You need to remotely access your Oracle Big Data that you executed in your Oracle Cloud account putty (mintty) or terminal (Mac/Linux, Git Bash]) with ssh. For example, for the user name: **fchen26**, you need to run the following with the appropriate ip address given:

```
$ ssh fchen26@129.150.69.91
```

Note: Make sure to replace the user name with your **username** and put the appropriate ip address. When asked for a password, type in your username again and press enter.

```
$ ssh fchen26@129.150.69.91
-- WARNING -- This system is for the use of authorized users only. Individuals
using this computer system without authority or in excess of their authority
are subject to having all their activities on this system monitored and
recorded by system personnel. Anyone using this system expressly consents to
such monitoring and is advised that if such monitoring reveals possible
evidence of criminal activity system personnel may provide the evidence of such
monitoring to law enforcement officials.

fchen26@129.150.69.91's password:
Last login: Sun Nov 22 06:57:42 2020 from 147.sub-174-193-128.myvzw.com
-bash-4.1$
```

Now you are connected to Oracle cloud.

You may run the following HDFS commands to test if hdfs works well at your Oracle account.

```
$ ls -al
$ hdfs dfs –ls
```

Step 2: Download Data from Amazon S3 and Load it into Oracle Big Data

Below is the location of the e-commerce data that is used for this project. You can download the data file (2019-Oct.zip) from Amazon S3:

```
$ wget -O 2019-Oct.csv.zip https://groupthreebucket.s3-us-west-1.amazonaws.com/2019-Oct.csv.zip
```

NOTE: The dataset used for this project can be downloaded from Kaggle. However, we only used the '2019-Oct.csv' data.

It's better if you move your data file into the data folder, unzip it and then upload it into hdfs to avoid having space issues.

```
$ mv 2019-Oct.csv.zip /data/
$ cd /data/
```

```
$ unzip 2019-Oct.csv.zip
```

Now you need to upload the "2019-Oct.csv" file to a directory of HDFS. Run the following commands in order:

Create a directory named **ecommerce**.

```
$ hdfs dfs -mkdir ecommerce
$ hdfs dfs -ls
```

Put **2019-Oct.csv** file from home directory to **ecommerce** directory.

```
$ hdfs dfs -put 2019-Oct.csv /user/fchen26/ecommerce/
```

To check the file is uploaded successfully, run the below command.

```
$ hdfs dfs -ls ecommerce
```

```
-bash-4.1$ hdfs dfs -ls ecommerce
Found 1 items
-rw-r--rw-   2 fchen26 hdfs 5668612855 2020-11-20 03:53 ecommerce/2019-Oct.csv
```

Run the below commands to get the first and last 10 lines of your data file:

```
$ hdfs dfs -cat ecommerce/2019-Oct.csv | head -n 10
$ hdfs dfs -cat ecommerce/2019-Oct.csv | tail -n 10
```

```
-bash-4.1$ hdfs dfs -cat ecommerce/2019-Oct.csv | head -n 10
event_time,event_type,product_id,category_id,category_code,brand,price,user_id,user_session
2019-10-01 00:00:00 UTC,view,44600062,2103807459595387724,,shiseido,35.79,541312140,72d76fde-8bb3-4e00-8c23-a032dfed738c
2019-10-01 00:00:00 UTC,view,3900821,2053013552326770905,appliances.environment.water_heater,aqua,33.20,554748717,9333dfbd-b87a-4708-9857-6336556b0fcc
2019-10-01 00:00:01 UTC,view,17200506,2053013559792632471,furniture.living_room.sofa,,543.10,519107250,566511c2-e2e3-422b-b695-cf8e6e792ca8
2019-10-01 00:00:01 UTC,view,1307067,2053013558920217191,computers.notebook,lenovo,251.74,550050854,7c90fc70-0e80-4590-96f3-13c02c18c713
2019-10-01 00:00:04 UTC,view,1004237,2053013555631882655,electronics.smartphone,apple,1081.98,535871217,c6bd7419-2748-4c56-95b4-8cec9ff8b80d
2019-10-01 00:00:05 UTC,view,1480613,2053013561092866779,computers.desktop,pulser,908.62,512742880,0d0d91c2-c9c2-4e81-90a5-86594dec0db9
2019-10-01 00:00:08 UTC,view,17300353,2053013553853497655,,creed,380.96,555447699,4fe811e9-91de-46da-90c3-bbd87ed3a65d
2019-10-01 00:00:08 UTC,view,31500053,2053013558031024687,,luminarc,41.16,550978835,6280d577-25c8-4147-99a7-abc6048498d6
2019-10-01 00:00:10 UTC,view,28719074,2053013565480109009,apparel.shoes.keds,baden,102.71,520571932,ac1cd4e5-a3ce-4224-a2d7-ff660a105880
cat: Unable to write to output stream.
-bash-4.1$ hdfs dfs -cat ecommerce/2019-Oct.csv | tail -n 10
2019-10-31 23:59:57 UTC,view,44300011,2100825583029060150,apparel.jeans,,50.45,545220871,f278cca0-e0f6-49a3-819a-d961998282d5
2019-10-31 23:59:58 UTC,view,12800151,2053013552788144369,,sony,8.49,544578298,fb46b2fb-493b-477c-8d18-ea1c24c04020
2019-10-31 23:59:58 UTC,view,5100816,2053013553375346967,,xiaomi,29.58,543653226,ab310b47-1eb2-45f8-8e5b-21ab2010925a
2019-10-31 23:59:58 UTC,view,1004870,2053013555631882655,electronics.smartphone,samsung,275.25,518956209,6764041a-9285-4869-8a32-a79adf31d212
2019-10-31 23:59:58 UTC,view,2702331,2053013563911439225,appliances.kitchen.refrigerators,lg,527.43,524356542,153f9818-4d32-4e8b-ba9f-f355094e8ae4
2019-10-31 23:59:58 UTC,view,2300275,2053013560530830019,electronics.camera.video,gopro,527.40,537931532,22c57267-da98-4f28-9a9c-18bb5b385193
2019-10-31 23:59:58 UTC,view,10800172,2053013554994348409,,redmond,61.75,527322328,5054190a-46cb-4211-a8f1-16fc1a060ed8
2019-10-31 23:59:58 UTC,view,5701038,2053013553970938175,auto.accessories.player,kenwood,128.70,566280422,05b6c62b-992f-4e8e-91f7-961bcb4719cd
2019-10-31 23:59:59 UTC,view,21407424,2053013561579406073,electronics.clocks,tissot,689.85,513118352,4c14bf2a-2820-4504-929d-046356a5a204
2019-10-31 23:59:59 UTC,view,13300120,2053013557166998015,,swisshome,155.73,525266378,6e57d2d7-6022-46e6-81d6-fa77f14cefd8
```

Run the following HDFS command to make your beeline command works.

NOTE: There is a period at the end of the command in the below:

```
$ hdfs dfs -chmod -R o+w .
```

Step 3: Creating Hive Tables to Query Data

The following Hive statement creates an external table that allows Hive to query data stored in HDFS. External tables preserve the data in the original file format, while allowing Hive to perform queries against the data within the file.

Open another terminal and login into your account using *ssh* as in Step 1.

Open **beeline** CLI (Command Line Shell Interface) that is equivalent to **hive** CLI environment as follows. **Beeline** is for multiple users' access to Hive Server 2 of a Hadoop cluster. Press enter without putting any password when it asks for a password.

```
beeline
```

```
WARNING: Use "yarn jar" to launch YARN applications.
Beeline version 1.2.1000.2.4.2.0-258 by Apache Hive

beeline> !connect jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigdai-nov-bdcsce-2:2181,bigdai-nov-bdcsce-
3:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2?tez.queue.name=int
eractive  bdcsce_admin
Connecting to jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigdai-nov-bdcsce-2:2181,bigdai-nov-bdcsce-
3:2181/;serviceDiscoveryerver2?tez.queue.name=interactive
Enter password for jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigdai-nov-bdcsce-2:2181,bigdai-nov-
bdcsce-3:2181/;serviceDischiveserver2?tez.queue.name=interactive:
Connected to: Apache Hive (version 1.2.1000.2.4.2.0-258)
Driver: Hive JDBC (version 1.2.1000.2.4.2.0-258)
Transaction isolation: TRANSACTION_REPEATABLE_READ
0: jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigd>
```

Note: If you see "**CLOSED**" in the above beeline shell prompt, it is **not** connected to Hive Server2.

```
-bash-4.1$ beeline
WARNING: Use "yarn jar" to launch YARN applications.
Beeline version 1.2.1000.2.4.2.0-258 by Apache Hive
beeline> !connect jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigdai-nov-bdcsce-2:2181,bigdai-nov-bdcsce-3:2181/;serviceDisco
iveserver2?tez.queue.name=interactive  bdcsce_admin
Connecting to jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigdai-nov-bdcsce-2:2181,bigdai-nov-bdcsce-3:2181/;serviceDiscovery
erver2?tez.queue.name=interactive
Enter password for jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigdai-nov-bdcsce-2:2181,bigdai-nov-bdcsce-3:2181/;serviceDisc
hiveserver2?tez.queue.name=interactive:
Connected to: Apache Hive (version 1.2.1000.2.4.2.0-258)
Driver: Hive JDBC (version 1.2.1000.2.4.2.0-258)
Transaction isolation: TRANSACTION_REPEATABLE_READ
0: jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigd>
```

Now you have to create your database with your username to separate your tables with other users. For example, the user **groupthree** should run the following:

Note: Make sure to use your **username**.

```
CREATE DATABASE IF NOT EXISTS groupthree;
```

Run the below command to make sure your database is created.

```
SHOW databases;
```

Now you need to use your database to create tables.

```
USE groupthree;
```

Note: Make sure to replace the database name with your **username**.

In the beeline shell CLI, you need to copy and paste the following HiveQL code to create an external table “ecommerce”.

Note: Don’t forget to replace **fchen26** to your account name in the following HiveQL code.

```
DROP TABLE IF EXISTS ecommerce;

--create the ecommerce table on comma-separated data

CREATE EXTERNAL TABLE IF NOT EXISTS ecommerce (
event_time STRING,
event_type STRING,
product_id INT,
category_id BIGINT,
category_code STRING,
brand STRING,
price FLOAT,
user_id INT,
user_session STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LOCATION '/user/fchen26/ecommerce'
TBLPROPERTIES ('skip.header.line.count'='1');
```

Then, in the beeline shell, you need to check if the table “**ecommerce**”is shown:

```
SHOW tables;
```

Note: If you can’t see the table name, then the table is not created and you have to follow the same step again.

Now you can query the content of the **ecommerce** table to see if it has the correct data and values:

```
SELECT * FROM ecommerce LIMIT 10;
```

You will see a result similar to below:

```
0: jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigd> select * from ecommerce limit 10;
+-------------------------+---------------------+---------------------+-----------------------+-------------------------------------+-----------------+-------------
------+-------------------+-----------------------------------------+--+
|   ecommerce.event_time  | ecommerce.event_type | ecommerce.product_id | ecommerce.category_id |       ecommerce.category_code       | ecommerce.brand |   ecommerce.pr
ice   | ecommerce.user_id |        ecommerce.user_session         |
+-------------------------+---------------------+---------------------+-----------------------+-------------------------------------+-----------------+-------------
------+-------------------+-----------------------------------------+--+
| 2019-10-01 00:00:00 UTC | view                | 44600062            | 2103807459595387724   |                                     | shiseido        | 35.79000091552
7344  | 541312140         | 72d76fde-8bb3-4e00-8c23-a032dfed738c    |
| 2019-10-01 00:00:00 UTC | view                | 3900821             | 2053013552326770905   | appliances.environment.water_heater | aqua            | 33.20000076293
945   | 554748717         | 9333dfbd-b87a-4708-9857-6336556b0fcc    |
| 2019-10-01 00:00:01 UTC | view                | 17200506            | 2053013559792632471   | furniture.living_room.sofa          |                 | 543.0999755859
375   | 519107250         | 566511c2-e2e3-422b-b695-cf8e6e792ca8    |
| 2019-10-01 00:00:01 UTC | view                | 1307067             | 2053013558920217191   | computers.notebook                  | lenovo          | 251.7400054931
6406  | 550050854         | 7c90fc70-0e80-4590-96f3-13c02c18c713    |
| 2019-10-01 00:00:04 UTC | view                | 1004237             | 2053013555631882655   | electronics.smartphone              | apple           | 1081.979980468
75    | 535871217         | c6bd7419-2748-4c56-95b4-8cec9ff8b80d    |
| 2019-10-01 00:00:05 UTC | view                | 1480613             | 2053013561092866779   | computers.desktop                   | pulser          | 908.6199951171
875   | 512742880         | 0d0d91c2-c9c2-4e81-90a5-86594dec0db9    |
| 2019-10-01 00:00:08 UTC | view                | 17300353            | 2053013553853497655   |                                     | creed           | 380.9599914550
781   | 555447699         | 4fe811e9-91de-46da-90c3-bbd87ed3a65d    |
| 2019-10-01 00:00:08 UTC | view                | 31500053            | 2053013558031024687   |                                     | luminarc        | 41.15999984741
211   | 550978835         | 6280d577-25c8-4147-99a7-abc6048498d6    |
| 2019-10-01 00:00:10 UTC | view                | 28719074            | 2053013565480109009   | apparel.shoes.keds                  | baden           | 102.7099990844
7266  | 520571932         | ac1cd4e5-a3ce-4224-a2d7-ff660a105880    |
| 2019-10-01 00:00:11 UTC | view                | 1004545             | 2053013555631882655   | electronics.smartphone              | huawei          | 566.0100097656
25    | 537918940         | 406c46ed-90a4-4787-a43b-59a410c1a5fb    |
+-------------------------+---------------------+---------------------+-----------------------+-------------------------------------+-----------------+-------------
------+-------------------+-----------------------------------------+--+
10 rows selected (0.243 seconds)
```

You can see the structure of the table as well:

```
DESCRIBE ecommerce;
```

```
0: jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigd> describe ecommerce;
+----------------+------------+----------+--+
|    col_name    | data_type  | comment  |
+----------------+------------+----------+--+
| event_time     | string     |          |
| event_type     | string     |          |
| product_id     | int        |          |
| category_id    | bigint     |          |
| category_code  | string     |          |
| brand          | string     |          |
| price          | float      |          |
| user_id        | int        |          |
| user_session   | string     |          |
+----------------+------------+----------+--+
9 rows selected (0.337 seconds)
```

Next, in the beeline shell CLI, you need to copy and paste the following HiveQL code to create a table "week_days" and insert data into it.

This table will hold the information regarding the days of the week. We will use this table and our ecommerce_view (discussed later) to create our queries.

```
DROP TABLE IF EXISTS  week_days;

--create the week_days table on comma-separated data

CREATE TABLE IF NOT EXISTS week_days (week_day_name STRING, week_day_num STRING)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';
```

```
INSERT INTO TABLE week_days
VALUES ('Monday', '1'), ('Tuesday', '2'), ('Wednesday', '3'), ('Thursday', '4'), ('Friday', '5'),
('Saturday','6'),('Sunday','7');
```

You can run this query to make sure that your insert statement was executed successfully.

```
SELECT * FROM week_days;
```

The output should look like this:

```
0: jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigd> select * from week_days;
+-------------------------+------------------------+--+
| week_days.week_day_name | week_days.week_day_num |
+-------------------------+------------------------+--+
| Monday                  | 1                      |
| Tuesday                 | 2                      |
| Wednesday               | 3                      |
| Thursday                | 4                      |
| Friday                  | 5                      |
| Saturday                | 6                      |
| Sunday                  | 7                      |
+-------------------------+------------------------+--+
7 rows selected (0.126 seconds)
```

Run this query to ensure that your **week_days** table have the right structure

```
DESCRIBE week_days;
```

The schema of your table should look like this:

```
+----------------+------------+----------+--+
|    col_name    | data_type  | comment  |
+----------------+------------+----------+--+
| week_day_name  | string     |          |
| week_day_num   | string     |          |
+----------------+------------+----------+--+
```

Now, in the beeline shell CLI, you need to copy and paste the following HiveQL code to create a view "**ecommerce_view**".

This view will have all the data from the ecommerce table. All the subsequent queries will be based on this view and the week_days table.

```
DROP VIEW IF EXISTS ecommerce_view;

--create the ecommerce_view view on comma-separated data
```

```
CREATE VIEW IF NOT EXISTS ecommerce_view AS
SELECT
event_time,
date_format(event_time, 'u') as week_day_num,
day(event_time) as day,
hour(event_time) as hour,
unix_timestamp(event_time) as event_time_in_seconds,
event_type,
product_id,
category_id,
split(category_code, '\\.')[0] as primary_category,
brand,
price,
user_id,
user_session
FROM ecommerce
WHERE category_code!='' AND brand!='';
```

Note: We will use the value for column **week_day_num** to join this view with the week_day table.

Note: The value of the category code consists of multiple parts that are separated by period (.). You will use the "split" method to get the first category.

Note: The "Where" clause that is used in the above query is for cleaning any junk data.

Now you can query the content of the **ecommerce_view** view to see if it has the correct data and values:

```
SELECT * FROM ecommerce_view LIMIT 10;
```

You will see a result similar to below:

```
+-----------------------------+-----------------------------+-----------------+-----------------+-----------------------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+
| ecommerce_view.event_time   | ecommerce_view.week_day_num | ecommerce_view.day | ecommerce_view.hour | ecommerce_view.event_time_in_seconds | ecommerce_view.event_type | ecommerce_view.product_id | ecommerce_view.category_id | ecommerce_view.primary_category | ecommerce_view.brand | ecommerce_view.price | ecommerce_view.user_id | ecommerce_view.user_session |
+-----------------------------+-----------------------------+-----------------+-----------------+-----------------------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+
| 2019-10-01 00:00:00 UTC     | 2                           | 1               | 0               | 1569888000                        | view                  | 3900821               | 2053013552326770905   | appliances            | aqua                 | 33.20000076293945    | 554748717              | 9333dfbd-b87a-4708-9857-6336556b0fcc |
| 2019-10-01 00:00:01 UTC     | 2                           | 1               | 0               | 1569888001                        | view                  | 1307067               | 2053013558920217191   | computers             | lenovo               | 251.74000549316406   | 550050854              | 7c90fc70-0e80-4590-96f3-13c02c18c713 |
| 2019-10-01 00:00:04 UTC     | 2                           | 1               | 0               | 1569888004                        | view                  | 1004237               | 2053013555631882655   | electronics           | apple                | 1081.97998046875     | 535871217              | c6bdd419-2748-4c56-95b4-8cec9ff8b80d |
| 2019-10-01 00:00:05 UTC     | 2                           | 1               | 0               | 1569888005                        | view                  | 1480613               | 2053013561092866779   | computers             | pulser               | 908.6199951171875    | 512742880              | 0d0d91c2-c9c2-4e81-90a5-86594dec0db9 |
| 2019-10-01 00:00:10 UTC     | 2                           | 1               | 0               | 1569888010                        | view                  | 28719074              | 2053013565480109009   | apparel               | baden                | 102.70999908447266   | 520571932              | ac1cd4e5-a3ce-4224-a2d7-ff660a105880 |
| 2019-10-01 00:00:11 UTC     | 2                           | 1               | 0               | 1569888011                        | view                  | 1004545               | 2053013555631882655   | electronics           | huawei               | 566.010009765625     | 537918940              | 406c46ed-90a4-4787-a43b-59a410c1a5fb |
| 2019-10-01 00:00:11 UTC     | 2                           | 1               | 0               | 1569888011                        | view                  | 2900536               | 2053013554776244595   | appliances            | elenberg             | 51.459999084472656   | 555158050              | b5bdd0b3-4ca2-4c55-939e-9ce44bb50abd |
| 2019-10-01 00:00:11 UTC     | 2                           | 1               | 0               | 1569888011                        | view                  | 1005011               | 2053013555631882655   | electronics           | samsung              | 900.6400146484375    | 530282093              | 50a293fb-5940-41b2-baf3-17af0e812101 |
| 2019-10-01 00:00:13 UTC     | 2                           | 1               | 0               | 1569888013                        | view                  | 3900746               | 2053013552326770905   | appliances            | haier                | 102.37999725341797   | 555444559              | 98b88fa0-d8fa-4b9d-8a71-3dd403afab85 |
| 2019-10-01 00:00:16 UTC     | 2                           | 1               | 0               | 1569888016                        | view                  | 13500240              | 2053013557099889147   | furniture             | brw                  | 93.18000030517578    | 555446365              | 7f0062d8-ead0-4e0a-96f6-43a0b79a2fc4 |
+-----------------------------+-----------------------------+-----------------+-----------------+-----------------------------------+-----------------------+-----------------------+-----------------------+-----------------------+-----------------------+
```

Run the below query to see the event types:

```
SELECT event_type FROM ecommerce_view GROUP BY event_type;
```

You will see a result similar to below:

```
+--------------+--+
| event_type   |
+--------------+--+
| cart         |
| view         |
| purchase     |
+--------------+--+
3 rows selected (72.709 seconds)
```

You can also look over the structure of the view as well:

```
DESCRIBE ecommerce_view;
```

```
0: jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigd> describe ecommerce_view;
+-------------------------+------------+----------+--+
|         col_name        | data_type  | comment  |
+-------------------------+------------+----------+--+
| event_time              | string     |          |
| day                     | int        |          |
| hour                    | int        |          |
| event_time_in_seconds   | bigint     |          |
| event_type              | string     |          |
| product_id              | int        |          |
| category_id             | bigint     |          |
| primary_category        | string     |          |
| brand                   | string     |          |
| price                   | float      |          |
| user_id                 | int        |          |
| user_session            | string     |          |
+-------------------------+------------+----------+--+
12 rows selected (0.193 seconds)
```

Step 4: Creating Hive Queries to Analyze data

The following Hive queries will be used to analyze the data.

Analysis 1:

Step1: Creating a table and running a query to get the sales revenue by day.

First, run the below query:

```
SELECT day, ROUND(SUM(price), 2) AS sales_revenue FROM ecommerce_view WHERE
event_type='purchase' GROUP BY day ORDER BY day ASC;
```

You will see a result similar to below:

```
+------+----------------+--+
| day  | sales_revenue  |
+------+----------------+--+
| 1    | 5611920.0      |
| 2    | 5550036.48     |
| 3    | 5536184.77     |
| 4    | 7714341.94     |
| 5    | 6499835.96     |
| 6    | 5958345.58     |
| 7    | 5537820.9      |
| 8    | 5906031.82     |
| 9    | 5965520.86     |
| 10   | 5819145.11     |
| 11   | 6665365.3      |
| 12   | 6292749.24     |
| 13   | 7287300.74     |
| 14   | 8484248.72     |
| 15   | 7797176.28     |
| 16   | 8729024.75     |
| 17   | 8059677.38     |
| 18   | 7300257.7      |
| 19   | 6737517.05     |
| 20   | 6704844.66     |
| 21   | 7013299.0      |
| 22   | 7082430.03     |
| 23   | 6800579.05     |
| 24   | 6693980.38     |
| 25   | 6366782.23     |
| 26   | 6203638.09     |
| 27   | 6440991.65     |
| 28   | 5827527.92     |
| 29   | 5671987.66     |
| 30   | 5873179.46     |
| 31   | 5735997.89     |
+------+----------------+--+
31 rows selected (75.302 seconds)
```

The below query will create a table using the above query and store the results in hdfs for visualization:

```
DROP TABLE IF EXISTS sales_revenue_by_day;

CREATE TABLE sales_revenue_by_day
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION '/user/fchen26/ecommerce/one'
AS
select day, ROUND(SUM(price), 2) AS sales_revenue FROM ecommerce_view WHERE
event_type='purchase' GROUP BY day ORDER BY day ASC;
```

Then, you need to check if the table is created successfully or not:

```
SHOW tables;
```

Step 2: After the Hive tables are created, you can download it to your lab (or personal PC/Laptop) as follows.

Open another terminal with git bash, minty, or putty, which is to connect the Oracle cloud to download the output file 000000_0 at the HDFS path "/user/**fchen26**/ecommerce/**one** ".

First, run the following hdfs command to list what file exists at "/user/**fchen26**/ecommerce/**one**" directory that is actually the location of Hive table. It is a file named "000000_0":

Note: Do not forget to change the username.

```
$ hdfs dfs -ls ecommerce/one
```

You can view the contents of the file with the below command:

```
$ hdfs dfs -cat ecommerce/one/000000_0
```

```
-bash-4.1$ hdfs dfs -cat ecommerce/one/000000_0
1,5611920.0
2,5550036.48
3,5536184.77
4,7714341.94
5,6499835.96
6,5958345.58
7,5537820.9
8,5906031.82
9,5965520.86
10,5819145.11
11,6665365.3
12,6292749.24
13,7287300.74
14,8484248.72
15,7797176.28
16,8729024.75
17,8059677.38
18,7300257.7
19,6737517.05
20,6704844.66
21,7013299.0
22,7082430.03
23,6800579.05
24,6693980.38
25,6366782.23
26,6203638.09
27,6440991.65
28,5827527.92
29,5671987.66
30,5873179.46
31,5735997.89
```

Then, download the file to the local file systems:

```
$ hdfs dfs -get ecommerce/one/000000_0 one.csv
$ ls -al
```

```
-bash-4.1$ hdfs dfs -ls ecommerce/one
Found 1 items
-rwxr-xrwx   2 bdcsce_admin hdfs        420 2020-11-23 07:44 ecommerce/one/000000_0
-bash-4.1$ hdfs dfs -get ecommerce/one/000000_0 one.csv
-bash-4.1$ ls -al
total 28
drwx------.  4 fchen26 fchen26 4096 Nov 24 06:47 .
drwxr-xr-x. 42 root    root    4096 Nov 17 19:49 ..
-rw-r--r--.  1 fchen26 fchen26  420 Nov 23 18:36 000000_0
-rw-------.  1 fchen26 fchen26 2290 Nov 24 06:29 .bash_history
drwxrwxr-x.  2 fchen26 fchen26 4096 Nov 19 09:25 .beeline
drwxrwxr-x.  2 fchen26 fchen26 4096 Nov 23 01:55 ftest
-rw-r--r--.  1 fchen26 fchen26  420 Nov 24 06:47 one.csv
-rw-rw-r--.  1 fchen26 fchen26    0 Nov 20 13:34 .pig_history
```

Lastly, open another terminal with git bash, minty, or putty in order to read/import the output file using your lab computer (or your PC/Laptop) - you have to download the file to your lab computer (or your PC/Laptop). For example, your output file at the oracle cloud server is located at /home/**fchen26**/one.csv and remotely copied to the file "one.csv".

```
$ scp fchen26@ipaddress:/home/fchen26/one.csv one.csv
```

Note: Make sure to replace the user name with your **username** and put the appropriate ip address.

Enter your password to download the file.

Alternatively, for Windows users, you may use psftp to download the file. You need to download it at https://the.earth.li/~sgtatham/putty/latest/w64/psftp.exe.

For example, in order to download 'one.csv' from the server of Oracle Cloud, you have to run psftp as follows. You may read through the commands of *psftp* in the below:

```
psftp> open [ipaddress]

Login as: fchen26

Enter password…

psftp> ls
Listing directory /home/fchen26
psftp> get one.csv
```

Note: Do not forget to change the username and put the appropriate ip address.

Analysis 2:

Step1: Creating a table and running a query to get the total number of purchases by day and hour.

First, run the below query:

```
SELECT day, hour ,COUNT(event_type) AS total_sales, ROUND(SUM(price), 2) AS
total_sales_revenue, COUNT(DISTINCT user_id) AS total_unique_buyers FROM ecommerce_view
WHERE event_type='purchase' GROUP BY day, hour ORDER BY day ASC, hour ASC;
```

You will see a result similar to below:

day	hour	total_sales	total_sales_revenue	total_unique_buyers
1	0	9	2358.33	8
1	2	233	75459.08	196
1	3	625	219014.42	519
1	4	764	280158.54	634
1	5	946	360716.97	775
1	6	1064	375116.81	878
1	7	1087	397378.44	878
1	8	1093	438559.99	913
1	9	1136	445999.59	899
1	10	1053	431171.67	823
1	11	934	344967.48	773
1	12	870	354263.23	720
1	13	815	315840.62	677
1	14	779	288747.23	628
1	15	736	257200.85	619
1	16	721	286455.28	586
1	17	589	218902.79	468
1	18	452	193009.11	344
1	19	327	135474.25	249
1	20	186	80019.02	142
1	21	130	59534.63	96
1	22	85	31731.8	58
1	23	49	19839.87	41
2	0	59	26352.5	49
2	1	137	51090.66	111
2	2	303	97563.46	250
2	3	572	185999.09	484
2	4	820	280122.9	693
2	5	1013	377680.16	819
2	6	1129	433638.78	891
2	7	1036	373391.42	861
2	8	1017	362384.59	845
2	9	1230	472648.41	933
2	10	1144	439243.93	904
2	11	906	340110.82	745
2	12	831	331471.74	658
2	13	744	291001.66	628
2	14	756	277927.15	623
2	15	755	268719.8	611

.

.

.

day	hour	total_sales	total_sales_revenue	total_unique_buyers
30	5	923	337698.35	772
30	6	1084	429772.99	884
30	7	1071	403819.32	879
30	8	1089	397282.14	890
30	9	1077	459108.17	876
30	10	969	390219.11	815
30	11	939	376301.96	776
30	12	856	327706.37	691
30	13	858	335528.39	710
30	14	869	324623.39	706
30	15	842	302740.28	680
30	16	705	264154.11	575
30	17	601	228797.54	497
30	18	528	227398.25	417
30	19	371	155864.36	269
30	20	241	100482.62	162
30	21	117	56495.92	95
30	22	112	40502.47	76
30	23	62	28714.43	50
31	0	54	27387.96	44
31	1	95	37846.45	79
31	2	249	89443.65	210
31	3	533	195013.39	452
31	4	808	288114.44	663
31	5	983	367692.84	822
31	6	978	395150.86	788
31	7	1095	413995.29	899
31	8	1163	475376.45	911
31	9	1045	407921.2	829
31	10	1034	432819.16	854
31	11	896	369870.3	739
31	12	786	318310.76	666
31	13	709	261088.55	614
31	14	833	292626.71	680
31	15	781	287163.93	652
31	16	730	271461.74	614
31	17	616	218360.65	506
31	18	537	217722.59	420
31	19	346	147231.95	275
31	20	170	78113.6	136
31	21	144	73032.72	113
31	22	88	37412.55	72
31	23	72	32840.17	58

743 rows selected (70.51 seconds)

The below query will create a table using the above query and store the results in hdfs for visualization:

```
DROP TABLE IF EXISTS total_purchases_by_day_hour;
```

```
CREATE TABLE total_purchases_by_day_hour
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION '/user/fchen26/ecommerce/two'
AS
SELECT day, hour ,COUNT(event_type) AS total_sales, ROUND(SUM(price), 2) AS
total_sales_revenue, COUNT(DISTINCT user_id) as total_unique_buyers FROM ecommerce_view
WHERE event_type='purchase' GROUP BY day, hour ORDER BY day ASC, hour ASC;
```

Then, you need to check if the table is created successfully or not:

```
SHOW tables;
```

Step 2: After the Hive tables are created, you can download it to your lab (or personal PC/Laptop) as follows.

Open another terminal with git bash, minty, or putty, which is to connect the Oracle cloud to download the output file 000000_0 at the HDFS path "/user/**fchen26**/ecommerce/**two** ".

First, run the following hdfs command to list what file exists at "/user/**fchen26**/ecommerce/**two**" directory that is actually the location of Hive table. It is a file named "000000_0":

Note: Do not forget to change the username.

```
$ hdfs dfs -ls ecommerce/two
```

You can view the contents of the file with the below command:

```
$ hdfs dfs -cat ecommerce/two/000000_0
```

Then, download the file to the local file systems:

```
$ hdfs dfs -get ecommerce/two/000000_0 two.csv
$ ls -al
```

```
-bash-4.1$ hdfs dfs -ls ecommerce/two
Found 1 items
-rwxr-xrwx   2 bdcsce_admin hdfs      17215 2020-11-24 07:40 ecommerce/two/000000_0
-bash-4.1$ hdfs dfs -get ecommerce/two/000000_0 two.csv
-bash-4.1$ ls -al
total 44
drwx------.  4 fchen26 fchen26  4096 Nov 24 07:42 .
drwxr-xr-x. 42 root    root     4096 Nov 17 19:49 ..
-rw-------.  1 fchen26 fchen26  2290 Nov 24 06:29 .bash_history
drwxrwxr-x.  2 fchen26 fchen26  4096 Nov 19 09:25 .beeline
drwxrwxr-x.  2 fchen26 fchen26  4096 Nov 23 01:55 ftest
-rw-r--r--.  1 fchen26 fchen26   420 Nov 24 06:47 one.csv
-rw-rw-r--.  1 fchen26 fchen26     0 Nov 20 13:34 .pig_history
-rw-r--r--.  1 fchen26 fchen26 17215 Nov 24 07:42 two.csv
```

Lastly, open another terminal with git bash, minty, or putty in order to read/import the output file using your lab computer (or your PC/Laptop) - you have to download the file to your lab computer (or your PC/Laptop). For example, your output file at the oracle cloud server is located at /home/**fchen26**/two.csv and remotely copied to the file "two.csv".

```
$ scp fchen26@ipaddress:/home/fchen26/two.csv two.csv
```

Note: Make sure to replace the user name with your **username** and put the appropriate ip address.

Enter your password to download the file.

Alternatively, for Windows users, you may use psftp to download the file. You need to download it at https://the.earth.li/~sgtatham/putty/latest/w64/psftp.exe.

For example, in order to download 'two.csv' from the server of Oracle Cloud, you have to run psftp as follows. You may read through the commands of *psftp* in the below:

```
psftp> open [ipaddress]

Login as: fchen26

Enter password…

psftp> ls
Listing directory /home/fchen26
psftp> get two.csv
```

Note: Do not forget to change the **username** and put the appropriate ip address.

Analysis 3:

Step1: Creating a table and running a query to get the total number of active users by hour.

First, run the below query:

```
SELECT hour, COUNT(user_id) AS users_active FROM ecommerce_view GROUP BY hour ORDER BY hour ASC;
```

You will see a result similar to below:

hour	users_active
0	178052
1	327935
2	639179
3	954463
4	1203459
5	1346691
6	1440298
7	1484440
8	1514572
9	1493203
10	1457419
11	1395629
12	1365198
13	1496143
14	1687598
15	1863832
16	1886350
17	1675688
18	1288687
19	821616
20	461629
21	274218
22	171027
23	133296

24 rows selected (127.307 seconds)

The below query will create a table using the above query and store the results in hdfs for visualization:

```
DROP TABLE IF EXISTS active_users_by_hour;

CREATE TABLE active_users_by_hour
```

```
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION '/user/fchen26/ecommerce/three'
AS
SELECT hour , COUNT(user_id) AS users_active FROM ecommerce_view GROUP BY hour ORDER BY hour ASC;
```

Then, you need to check if the table is created successfully or not:

```
SHOW tables;
```

Step 2: After the Hive tables are created, you can download it to your lab (or personal PC/Laptop) as follows.

Open another terminal with git bash, minty, or putty, which is to connect the Oracle cloud to download the output file 000000_0 at the HDFS path "/user/**fchen26**/ecommerce/**three** ".

First, run the following hdfs command to list what file exists at "/user/**fchen26**/ecommerce/**three**" directory that is actually the location of Hive table. It is a file named "000000_0":

Note: Do not forget to change the username.

```
$ hdfs dfs -ls ecommerce/three
```

You can view the contents of the file with the below command:

```
$ hdfs dfs -cat ecommerce/three/000000_0
```

Then, download the file to the local file systems:

```
$ hdfs dfs -get ecommerce/three/000000_0 three.csv
$ ls -al
```

```
-bash-4.1$ hdfs dfs -ls ecommerce/three
Found 1 items
-rwxr-xrwx   2 bdcsce_admin hdfs        245 2020-11-24 07:48 ecommerce/three/000000_0
-bash-4.1$ hdfs dfs -get ecommerce/three/000000_0 three.csv
-bash-4.1$ ls -al
total 48
drwx------.  4 fchen26 fchen26  4096 Nov 24 07:49 .
drwxr-xr-x. 42 root    root     4096 Nov 17 19:49 ..
-rw-------.  1 fchen26 fchen26  2290 Nov 24 06:29 .bash_history
drwxrwxr-x.  2 fchen26 fchen26  4096 Nov 19 09:25 .beeline
drwxrwxr-x.  2 fchen26 fchen26  4096 Nov 23 01:55 ftest
-rw-r--r--.  1 fchen26 fchen26   420 Nov 24 06:47 one.csv
-rw-rw-r--.  1 fchen26 fchen26     0 Nov 20 13:34 .pig_history
-rw-r--r--.  1 fchen26 fchen26   245 Nov 24 07:49 three.csv
-rw-r--r--.  1 fchen26 fchen26 17215 Nov 24 07:42 two.csv
```

Lastly, open another terminal with git bash, minty, or putty in order to read/import the output file using your lab computer (or your PC/Laptop) - you have to download the file to your lab computer (or your PC/Laptop). For example, your output file at the oracle cloud server is located at /home/**fchen26**/three.csv and remotely copied to the file "three.csv".

```
$ scp fchen26@ipaddress:/home/fchen26/three.csv three.csv
```

Note: Make sure to replace the user name with your **username** and put the appropriate ip address.

Enter your password to download the file.

Alternatively, for Windows users, you may use psftp to download the file. You need to download it at https://the.earth.li/~sgtatham/putty/latest/w64/psftp.exe.

For example, in order to download 'three.csv' from the server of Oracle Cloud, you have to run psftp as follows. You may read through the commands of *psftp* in the below:

```
psftp> open [ipaddress]

Login as: fchen26

Enter password…

psftp> ls
Listing directory /home/fchen26
psftp> get three.csv
```

Note: Do not forget to change the **username** and put the appropriate ip address.

Analysis 4:

Step1: Creating a table and running a query to get the most viewed product categories.

First, run the below query:

```
SELECT primary_category, COUNT(event_type) AS total_views  FROM ecommerce_view WHERE
event_type = 'view' GROUP BY primary_category ORDER BY total_views DESC;
```

You will see a result similar to below:

primary_category	total_views
electronics	14745241
appliances	4500365
computers	2199905
apparel	1124825
auto	753578
construction	537851
furniture	534088
kids	416636
accessories	205718
sport	151166
medicine	13349
country_yard	13341
stationery	5643

13 rows selected (106.972 seconds)

The below query will create a table using the above query and store the results in hdfs for visualization:

```
DROP TABLE IF EXISTS most_viewed_categories;

CREATE TABLE most_viewed_categories
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION '/user/fchen26/ecommerce/four'
AS
SELECT primary_category, COUNT(event_type) AS total_views  FROM ecommerce_view WHERE
event_type = 'view' GROUP BY primary_category ORDER BY total_views DESC;
```

Then, you need to check if the table is created successfully or not:

```
show tables;
```

Step 2: After the Hive tables are created, you can download it to your lab (or personal PC/Laptop) as follows.

Open another terminal with git bash, minty, or putty, which is to connect the Oracle cloud to download the output file 000000_0 at the HDFS path "/user/**fchen26**/ecommerce/**four** ".

First, run the following hdfs command to list what file exists at "/user/**fchen26**/ecommerce/**four**" directory that is actually the location of Hive table. It is a file named "000000_0":

Note: Do not forget to change the username.

```
$ hdfs dfs -ls ecommerce/four
```

You can view the contents of the file with the below command:

```
$ hdfs dfs -cat ecommerce/four/000000_0
```

Then, download the file to the local file systems:

```
$ hdfs dfs -get ecommerce/four/000000_0 four.csv
$ ls -al
```

```
-bash-4.1$ hdfs dfs -ls ecommerce/four
Found 1 items
-rwxr-xrwx   2 bdcsce_admin hdfs        217 2020-11-24 07:54 ecommerce/four/000000_0
-bash-4.1$ hdfs dfs -get ecommerce/four/000000_0 four.csv
-bash-4.1$ ls -al
total 52
drwx------.  4 fchen26 fchen26  4096 Nov 24 07:56 .
drwxr-xr-x. 42 root    root     4096 Nov 17 19:49 ..
-rw-------.  1 fchen26 fchen26  2290 Nov 24 06:29 .bash_history
drwxrwxr-x.  2 fchen26 fchen26  4096 Nov 19 09:25 .beeline
-rw-r--r--.  1 fchen26 fchen26   217 Nov 24 07:56 four.csv
drwxrwxr-x.  2 fchen26 fchen26  4096 Nov 23 01:55 ftest
-rw-r--r--.  1 fchen26 fchen26   420 Nov 24 06:47 one.csv
-rw-rw-r--.  1 fchen26 fchen26     0 Nov 20 13:34 .pig_history
-rw-r--r--.  1 fchen26 fchen26   245 Nov 24 07:49 three.csv
-rw-r--r--.  1 fchen26 fchen26 17215 Nov 24 07:42 two.csv
```

Lastly, open another terminal with git bash, minty, or putty in order to read/import the output file using your lab computer (or your PC/Laptop) - you have to download the file to your lab computer (or your PC/Laptop). For example, your output file at the oracle cloud server is located at /home/**fchen26**/four.csv and remotely copied to the file "four.csv".

```
$ scp fchen26@ipaddress:/home/fchen26/four.csv four.csv
```

Note: Make sure to replace the user name with your **username** and put the appropriate ip address.

Enter your password to download the file.

Alternatively, for Windows users, you may use psftp to download the file. You need to download it at https://the.earth.li/~sgtatham/putty/latest/w64/psftp.exe.

For example, in order to download 'four.csv' from the server of Oracle Cloud, you have to run psftp as follows. You may read through the commands of *psftp* in the below:

```
psftp> open [ipaddress]

Login as: fchen26

Enter password…

psftp> ls
Listing directory /home/fchen26
psftp> get four.csv
```

Note: Do not forget to change the **username** and put the appropriate ip address.

Analysis 5:

Step1: Creating a table and running a query to get the percentage of each event.

First, run the below query:

```
SELECT count(event_type) FROM ecommerce_view;
```

This will give you the total count of all event types.

You will see a result similar to below:

```
+-----------+--+
|    _c0    |
+-----------+--+
| 26560622  |
+-----------+--+
1 row selected (65.913 seconds)
```

Save the result and use it in the below query:

```
SELECT event_type,count(event_type) AS number_of_events, ROUND(CAST(COUNT(event_type) AS float)/26560622, 2) AS percentage FROM ecommerce_view GROUP BY event_type ORDER BY percentage DESC;
```

You will see a result similar to below:

```
+-------------+-------------------+-------------+--+
| event_type  | number_of_events  | percentage  |
+-------------+-------------------+-------------+--+
| view        | 25201706          | 0.95        |
| cart        | 809409            | 0.03        |
| purchase    | 549507            | 0.02        |
+-------------+-------------------+-------------+--+
3 rows selected (63.053 seconds)
```

The below query will create a table using the above query and store the results in hdfs for visualization:

```
DROP TABLE IF EXISTS percentage_events;

CREATE TABLE percentage_events
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION '/user/fchen26/ecommerce/five'
AS
```

```
SELECT event_type,count(event_type) AS number_of_events, ROUND(CAST(COUNT(event_type) AS float)/26560622, 2) AS percentage FROM ecommerce_view GROUP BY event_type ORDER BY percentage DESC;
```

Then, you need to check if the table is created successfully or not:

```
SHOW tables;
```

Step 2: After the Hive tables are created, you can download it to your lab (or personal PC/Laptop) as follows.

Open another terminal with git bash, minty, or putty, which is to connect the Oracle cloud to download the output file 000000_0 at the HDFS path "/user/**fchen26**/ecommerce/**five** ".

First, run the following hdfs command to list what file exists at "/user/**fchen26**/ecommerce/**five**" directory that is actually the location of Hive table. It is a file named "000000_0":

Note: Do not forget to change the username.

```
$ hdfs dfs -ls ecommerce/five
```

You can view the contents of the file with the below command:

```
$ hdfs dfs -cat ecommerce/five/000000_0
```

Then, download the file to the local file systems:

```
$ hdfs dfs -get ecommerce/five/000000_0 five.csv
$ ls -al
```

```
-bash-4.1$ hdfs dfs -ls ecommerce/five
Found 1 items
-rwxr-xrwx   2 bdcsce_admin hdfs         57 2020-11-24 07:56 ecommerce/five/000000_0
-bash-4.1$ hdfs dfs -get ecommerce/five/000000_0 five.csv
-bash-4.1$ ls -al
total 56
drwx------.  4 fchen26 fchen26  4096 Nov 24 07:57 .
drwxr-xr-x. 42 root    root     4096 Nov 17 19:49 ..
-rw-------.  1 fchen26 fchen26  2290 Nov 24 06:29 .bash_history
drwxrwxr-x.  2 fchen26 fchen26  4096 Nov 19 09:25 .beeline
-rw-r--r--.  1 fchen26 fchen26    57 Nov 24 07:57 five.csv
-rw-r--r--.  1 fchen26 fchen26   217 Nov 24 07:56 four.csv
drwxrwxr-x.  2 fchen26 fchen26  4096 Nov 23 01:55 ftest
-rw-r--r--.  1 fchen26 fchen26   420 Nov 24 06:47 one.csv
-rw-rw-r--.  1 fchen26 fchen26     0 Nov 20 13:34 .pig_history
-rw-r--r--.  1 fchen26 fchen26   245 Nov 24 07:49 three.csv
-rw-r--r--.  1 fchen26 fchen26 17215 Nov 24 07:42 two.csv
```

Lastly, open another terminal with git bash, minty, or putty in order to read/import the output file using your lab computer (or your PC/Laptop) - you have to download the file to your lab computer (or your PC/Laptop). For example, your output file at the oracle cloud server is located at /home/**fchen26**/five.csv and remotely copied to the file "five.csv".

```
$ scp fchen26@ipaddress:/home/fchen26/five.csv five.csv
```

Note: Make sure to replace the user name with your **username** and put the appropriate ip address.

Enter your password to download the file.

Alternatively, for Windows users, you may use psftp to download the file. You need to download it at https://the.earth.li/~sgtatham/putty/latest/w64/psftp.exe.

For example, in order to download 'five.csv' from the server of Oracle Cloud, you have to run psftp as follows. You may read through the commands of *psftp* in the below:

```
psftp> open [ipaddress]

Login as: fchen26

Enter password…

psftp> ls
Listing directory /home/fchen26
psftp> get five.csv
```

Note: Do not forget to change the **username** and put the appropriate ip address.

Analysis 6:

Step1: Creating a table and running a query to get the top 100 primary category, brand and total sales by product category and brand.

First, run the below query:

```
SELECT primary_category, brand, COUNT(event_type) as total_sales FROM ecommerce_view
WHERE primary_category IN('electronics', 'appliances', 'computers') AND event_type ='purchase'
GROUP BY primary_category, brand ORDER BY total_sales DESC LIMIT 100;
```

Note: That scope of the query is the top 3 most viewed product categories.

You will see a result similar to below:

primary_category	brand	total_sales
electronics	samsung	159621
electronics	apple	141394
electronics	xiaomi	44767
electronics	huawei	23220
appliances	samsung	11559
electronics	oppo	10891
computers	acer	6822
appliances	elenberg	5201
appliances	lg	5033
appliances	indesit	5023
computers	lenovo	4126
appliances	beko	3839
appliances	bosch	3407
computers	hp	3201
electronics	artel	3053
electronics	lg	2701
electronics	casio	2660
appliances	midea	2266
appliances	vitek	2252
appliances	dauscher	2219
electronics	nokia	2218
electronics	sony	2153
appliances	tefal	2126
appliances	redmond	2115
electronics	vivo	2025
electronics	haier	1923
electronics	jbl	1912
computers	asus	1804
electronics	meizu	1711
appliances	artel	1664
electronics	kivi	1584
appliances	ariston	1502
appliances	philips	1493
appliances	polaris	1410
computers	epson	1227
appliances	janome	1196
computers	apple	1173
appliances	haier	1121
electronics	prestigio	1101
appliances	braun	1069
electronics	elari	1020
appliances	xiaomi	978
appliances	atlant	931
appliances	scarlett	916
appliances	arg	815
computers	zeta	755
appliances	oasis	737
computers	kingston	720
electronics	tcl	708

appliances	asel	608
appliances	hansa	589
electronics	honor	584
electronics	yasin	578
electronics	texet	554
appliances	electrolux	526
appliances	maxwell	524
computers	samsung	522
appliances	karcher	509
electronics	oneplus	509
electronics	bq	500
electronics	panasonic	485
electronics	pioneer	477
computers	microlab	451
appliances	moulinex	439
electronics	philips	426
computers	pulser	422
appliances	thomas	407
electronics	wonlex	402
appliances	kitfort	402
computers	canon	394
appliances	nika	382
electronics	tp-link	375
appliances	chayka	373
appliances	delonghi	362
electronics	harper	362
computers	defender	347
electronics	lenovo	343
electronics	kicx	341
electronics	yamaha	337
appliances	candy	336
appliances	saturn	322
appliances	willmark	318
computers	gigabyte	314
appliances	gorenje	303
electronics	aimoto	295
appliances	galaxy	287
appliances	panasonic	283
electronics	garmin	278
appliances	thermex	276
appliances	ballu	273
appliances	arnica	271
computers	msi	256
electronics	changhong	255
electronics	orient	254
computers	pocketbook	251
computers	palit	250
electronics	cortland	249
electronics	plantronics	241
electronics	inoi	240
computers	sven	234

100 rows selected (58.709 seconds)

The below query will create a table using the above query and store the results in hdfs for visualization:

```
DROP TABLE IF EXISTS top_primary_categories;

CREATE TABLE top_primary_categories
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION '/user/fchen26/ecommerce/six'
AS
SELECT primary_category, brand, COUNT(event_type) as total_sales FROM ecommerce_view
WHERE primary_category IN('electronics', 'appliances', 'computers') AND event_type ='purchase'
GROUP BY primary_category, brand ORDER BY total_sales DESC LIMIT 100;
```

Then, you need to check if the table is created successfully or not:

```
SHOW tables;
```

Step 2: After the Hive tables are created, you can download it to your lab (or personal PC/Laptop) as follows.

Open another terminal with git bash, minty, or putty, which is to connect the Oracle cloud to download the output file 000000_0 at the HDFS path "/user/**fchen26**/ecommerce/**six** ".

First, run the following hdfs command to list what file exists at "/user/**fchen26**/ecommerce/**six**" directory that is actually the location of Hive table. It is a file named "000000_0":

Note: Do not forget to change the username.

```
$ hdfs dfs -ls ecommerce/six
```

You can view the contents of the file with the below command:

```
$ hdfs dfs -cat ecommerce/six/000000_0
```

Then, download the file to the local file systems:

```
$ hdfs dfs -get ecommerce/six/000000_0 six.csv
$ ls -al
```

```
-bash-4.1$ hdfs dfs -ls ecommerce/six
Found 1 items
-rwxr-xrwx   2 bdcsce_admin hdfs       2255 2020-11-24 08:01 ecommerce/six/000000_0
-bash-4.1$ hdfs dfs -get ecommerce/six/000000_0 six.csv
-bash-4.1$ ls -al
total 60
drwx------.  4 fchen26 fchen26  4096 Nov 24 08:02 .
drwxr-xr-x. 42 root    root     4096 Nov 17 19:49 ..
-rw-------.  1 fchen26 fchen26  2290 Nov 24 06:29 .bash_history
drwxrwxr-x.  2 fchen26 fchen26  4096 Nov 19 09:25 .beeline
-rw-r--r--.  1 fchen26 fchen26    57 Nov 24 07:57 five.csv
-rw-r--r--.  1 fchen26 fchen26   217 Nov 24 07:56 four.csv
drwxrwxr-x.  2 fchen26 fchen26  4096 Nov 23 01:55 ftest
-rw-r--r--.  1 fchen26 fchen26   420 Nov 24 06:47 one.csv
-rw-rw-r--.  1 fchen26 fchen26     0 Nov 20 13:34 .pig_history
-rw-r--r--.  1 fchen26 fchen26  2255 Nov 24 08:02 six.csv
-rw-r--r--.  1 fchen26 fchen26   245 Nov 24 07:49 three.csv
-rw-r--r--.  1 fchen26 fchen26 17215 Nov 24 07:42 two.csv
```

Lastly, open another terminal with git bash, minty, or putty in order to read/import the output file using your lab computer (or your PC/Laptop) - you have to download the file to your lab computer (or your PC/Laptop). For example, your output file at the oracle cloud server is located at /home/**fchen26**/six.csv and remotely copied to the file "six.csv".

```
$ scp fchen26@ipaddress:/home/fchen26/six.csv six.csv
```

Note: Make sure to replace the user name with your **username** and put the appropriate ip address.

Enter your password to download the file.

Alternatively, for Windows users, you may use psftp to download the file. You need to download it at https://the.earth.li/~sgtatham/putty/latest/w64/psftp.exe.

For example, in order to download 'six.csv' from the server of Oracle Cloud, you have to run psftp as follows. You may read through the commands of *psftp* in the below:

```
psftp> open [ipaddress]

Login as: fchen26

Enter password…

psftp> ls
Listing directory /home/fchen26
psftp> get six.csv
```

Note: Do not forget to change the **username** and put the appropriate ip address.

Analysis 7:

Step1: Creating a table and running a query to get the top 10 users who purchased more than once.

First, run the below query:

```
SELECT COUNT(user_id) as number_of_purchased,user_id FROM ecommerce_view WHERE
event_type ='purchase' GROUP BY user_id HAVING number_of_purchased>1 ORDER BY
number_of_purchased DESC limit 10;
```

You will see a result similar to below:

number_of_purchased	user_id
322	512386086
288	523974502
242	513320236
193	543312954
183	519267944
183	513117637
178	517728689
170	530834332
165	541510103
154	549109608

The below query will create a table using the above query and store the results in hdfs for visualization:

```
DROP TABLE IF EXISTS top_users_most_purchased;

CREATE TABLE top_users_most_purchased
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION '/user/fchen26/ecommerce/seven'
AS
SELECT COUNT(user_id) as number_of_purchased,user_id FROM ecommerce_view WHERE
event_type ='purchase' GROUP BY user_id HAVING number_of_purchased>1 ORDER BY
number_of_purchased DESC limit 10;
```

Then, you need to check if the table is created successfully or not:

```
SHOW tables;
```

Step 2: After the Hive tables are created, you can download it to your lab (or personal PC/Laptop) as follows.

Open another terminal with git bash, minty, or putty, which is to connect the Oracle cloud to download the output file 000000_0 at the HDFS path "/user/**fchen26**/ecommerce/**seven** ".

First, run the following hdfs command to list what file exists at "/user/**fchen26**/ecommerce/**seven**" directory that is actually the location of Hive table. It is a file named "000000_0":

Note: Do not forget to change the username.

```
$ hdfs dfs -ls ecommerce/seven
```

You can view the contents of the file with the below command:

```
$ hdfs dfs -cat ecommerce/seven/000000_0
```

Then, download the file to the local file systems:

```
$ hdfs dfs -get ecommerce/seven/000000_0 seven.csv
$ ls -al
```

```
-bash-4.1$ hdfs dfs -ls ecommerce/seven
Found 1 items
-rwxr-xrwx   2 bdcsce_admin hdfs        140 2020-11-25 04:00 ecommerce/seven/000000_0
-bash-4.1$ hdfs dfs -cat ecommerce/seven/000000_0
322,512386086
288,523974502
242,513320236
193,543312954
183,513117637
183,519267944
178,517728689
170,530834332
165,541510103
154,551557821
-bash-4.1$ hdfs dfs -get ecommerce/seven/000000_0 seven.csv
-bash-4.1$ ls -al
total 64
drwx------.  4 fchen26 fchen26  4096 Nov 25 06:23 .
drwxr-xr-x. 42 root    root     4096 Nov 17 19:49 ..
-rw-------.  1 fchen26 fchen26  3603 Nov 25 06:08 .bash_history
drwxrwxr-x.  2 fchen26 fchen26  4096 Nov 19 09:25 .beeline
-rw-r--r--.  1 fchen26 fchen26    57 Nov 24 07:57 five.csv
-rw-r--r--.  1 fchen26 fchen26   217 Nov 24 07:56 four.csv
drwxrwxr-x.  2 fchen26 fchen26  4096 Nov 23 01:55 ftest
-rw-r--r--.  1 fchen26 fchen26   420 Nov 24 06:47 one.csv
-rw-rw-r--.  1 fchen26 fchen26     0 Nov 20 13:34 .pig_history
-rw-r--r--.  1 fchen26 fchen26   140 Nov 25 06:23 seven.csv
-rw-r--r--.  1 fchen26 fchen26  2255 Nov 24 08:02 six.csv
-rw-r--r--.  1 fchen26 fchen26   245 Nov 24 07:49 three.csv
-rw-r--r--.  1 fchen26 fchen26 17215 Nov 24 07:42 two.csv
```

Lastly, open another terminal with git bash, minty, or putty in order to read/import the output file using your lab computer (or your PC/Laptop) - you have to download the file to your lab computer (or your PC/Laptop). For example, your output file at the oracle cloud server is located at /home/**fchen26**/seven.csv and remotely copied to the file "seven.csv".

```
$ scp fchen26@ipaddress:/home/fchen26/seven.csv seven.csv
```

Note: Make sure to replace the user name with your **username** and put the appropriate ip address.

Enter your password to download the file.

Alternatively, for Windows users, you may use psftp to download the file. You need to download it at https://the.earth.li/~sgtatham/putty/latest/w64/psftp.exe.

For example, in order to download 'seven.csv' from the server of Oracle Cloud, you have to run psftp as follows. You may read through the commands of *psftp* in the below:

```
psftp> open [ipaddress]

Login as: fchen26

Enter password…

psftp> ls
Listing directory /home/fchen26
psftp> get seven.csv
```

Note: Do not forget to change the **username** and put the appropriate ip address.

Analysis 8:

Step1: Creating a table and running a query to get the total number of active users by weekday.

First, run the below query:

```
SELECT week_days.week_day_name as week_day , COUNT(ecommerce_view.user_id) AS
users_active
FROM ecommerce_view JOIN week_days
ON (ecommerce_view.week_day_num = week_days.week_day_num)
GROUP BY week_day_name;
```

You will see a result similar to below:

week_day	users_active
Monday	3339790
Tuesday	4207913
Saturday	3523134
Sunday	3669964
Thursday	4023662
Friday	3652495
Wednesday	4143664

The below query will create a table using the above query and store the results in hdfs for visualization:

```
DROP TABLE IF EXISTS top_users_most_purchased;

CREATE TABLE active_users_by_weekday
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION '/user/fchen26/ecommerce/eight'
AS
SELECT week_days.week_day_name as week_day , COUNT(ecommerce_view.user_id) AS
users_active
FROM ecommerce_view JOIN week_days
ON (ecommerce_view.week_day_num = week_days.week_day_num)
```

```
GROUP BY week_day_name;
```

Then, you need to check if the table is created successfully or not:

```
SHOW tables;
```

Step 2: After the Hive tables are created, you can download it to your lab (or personal PC/Laptop) as follows.

Open another terminal with git bash, minty, or putty, which is to connect the Oracle cloud to download the output file 000000_0 at the HDFS path "/user/**fchen26**/ecommerce/**eight** ".

First, run the following hdfs command to list what file exists at "/user/**fchen26**/ecommerce/**eight**" directory that is actually the location of the Hive table. For some reason it is split into 47 separate files. We run a command to merge them later.

Note: Do not forget to change the username.

```
$ hdfs dfs -ls ecommerce/eight
```

This is the merge command to merge the 47 files into a single file.

```
$ hadoop fs -cat /user/fchen26/ecommerce/eight/* | hadoop fs -put -
/user/fchen26/ecommerce/eight/000046_1
```

You can view the contents of the merged file with the below command:

```
$ hdfs dfs -cat ecommerce/eight/000046_1
```

Then, download the file to the local file systems:

```
$ hdfs dfs -get ecommerce/eight/000046_1 eight.csv
$ ls -al
```

```
drwxr-xr-x. 42 root    root    4096 Nov 17 19:49 ..
-rw-------.  1 fchen26 fchen26 9106 Dec 12 16:07 .bash_history
drwxrwxr-x.  2 fchen26 fchen26 4096 Nov 19 09:25 .beeline
-rw-rw-r--.  1 fchen26 fchen26  300 Nov 29 06:33 beeline-hs2-connection.xml
-rw-r--r--.  1 fchen26 fchen26  113 Dec 16 23:35 eight.csv
drwxrwxr-x.  2 fchen26 fchen26 4096 Nov 23 01:55 ftest
-rw-rw-r--.  1 fchen26 fchen26  471 Dec  3 03:59 high_cost_sites.pig
drwxrwxr-x.  4 fchen26 fchen26 4096 Nov 15  2016 labPigETL
-rw-rw-r--.  1 fchen26 fchen26 1127 Dec  3 03:03 .pig_history
-bash-4.1$
```

Lastly, open another terminal with git bash, minty, or putty in order to read/import the output file using your lab computer (or your PC/Laptop) - you have to download the file to your lab computer (or your PC/Laptop). For example, your output file at the oracle cloud server is located at /home/**fchen26**/eight.csv and remotely copied to the file "eight.csv".

```
$ scp fchen26@ipaddress:/home/fchen26/eight.csv eight.csv
```

Note: Make sure to replace the user name with your **username** and put the appropriate ip address.

Enter your password to download the file.

Alternatively, for Windows users, you may use psftp to download the file. You need to download it at https://the.earth.li/~sgtatham/putty/latest/w64/psftp.exe.

For example, in order to download 'eight.csv' from the server of Oracle Cloud, you have to run psftp as follows. You may read through the commands of *psftp* in the below:

```
psftp> open [ipaddress]

Login as: fchen26

Enter password…

psftp> ls
Listing directory /home/fchen26
psftp> get eight.csv
```

Note: Do not forget to change the **username** and put the appropriate ip address.

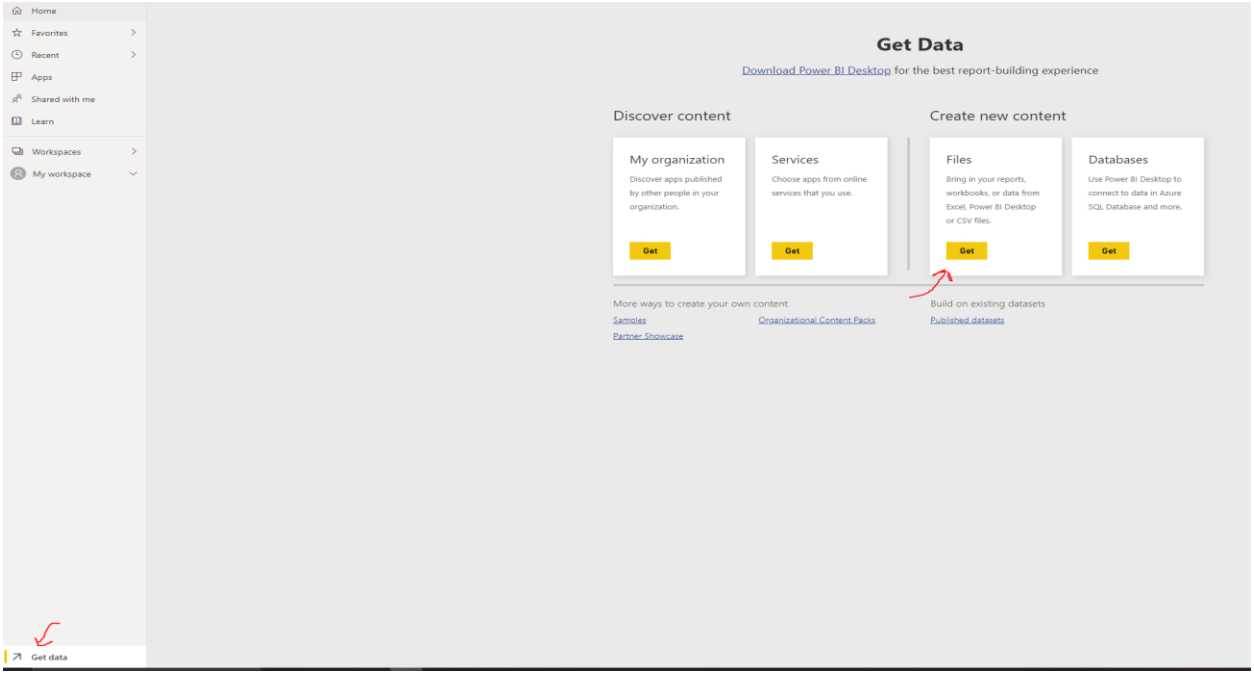
Step 5: Visualizing Data using Business Intelligence Tools

You will use the results data files (csv files) to create visualizations.

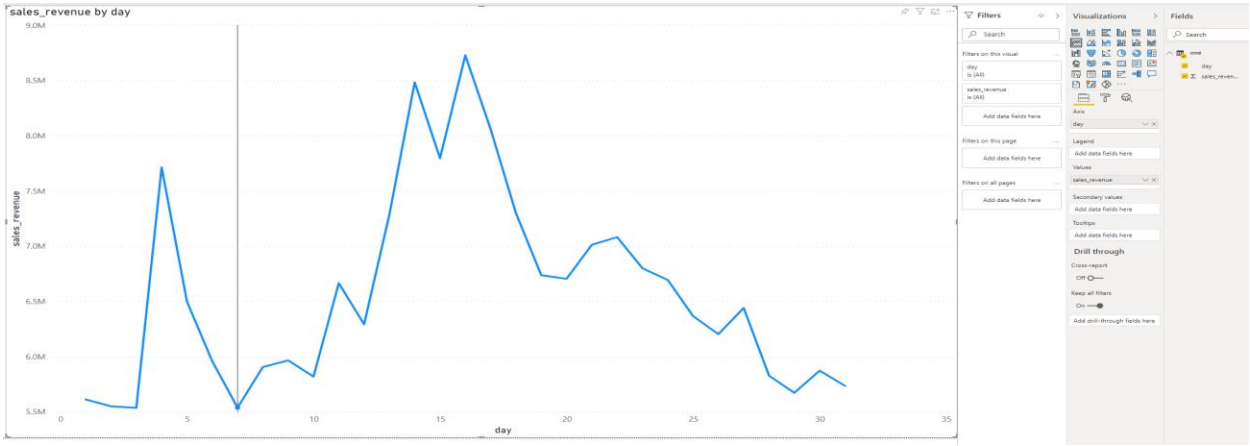
Visualization 1 and 5: Power BI Line Chart & Pie Chart

Download and install Power BI Desktop or you can use the Power BI on the web.
https://www.microsoft.com/en-us/download/details.aspx?id=58494

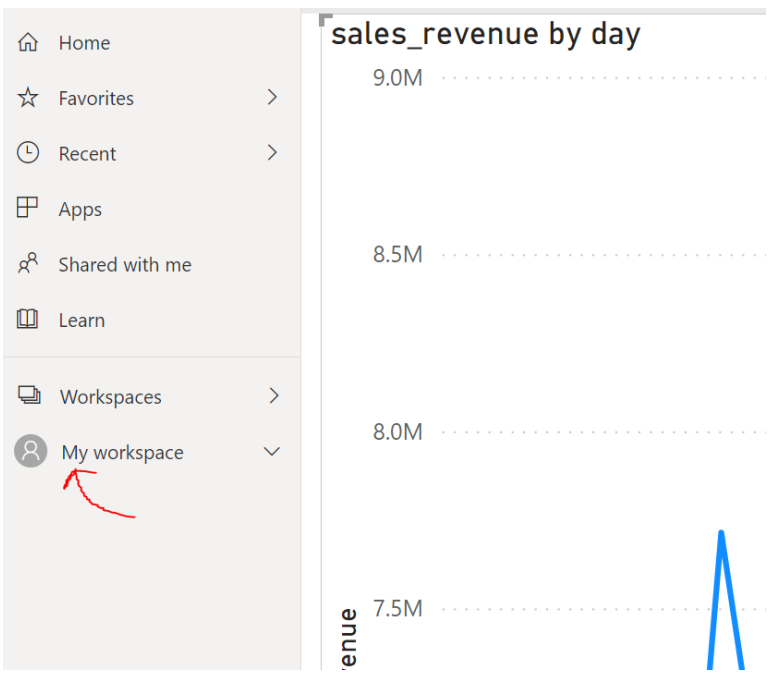
Step 1: Open PowerBI, import the data into PowerBI by clicking Get Data on the bottom left and then clicking the Get in the Files to import files **one.csv** and **five.csv**.



Step 2: PowerBI would prompt you to the dashboard. Double click **one.csv**.
Choose the line chart from the chart options and move the day field to axis and sales revenue to Values.



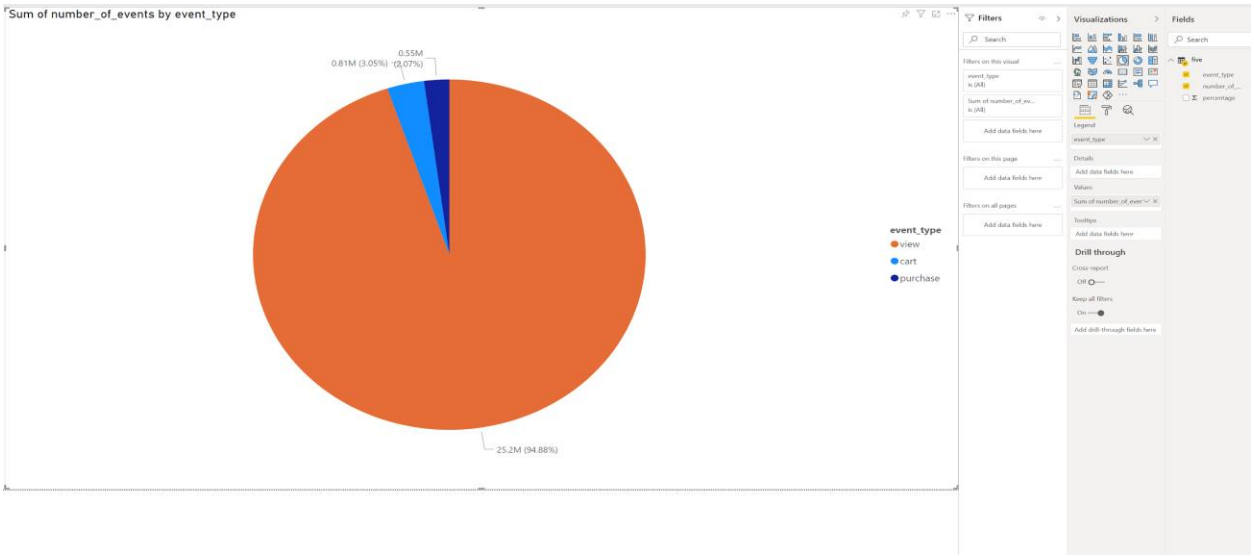
Step 3: Since we already uploaded **five.csv**, we can access it in My Workspace



Step 4: Click **five.csv** and move to the dashboard.

Step 5: Choose the Pie Chart from the options and move event type to legend and number of events into values.

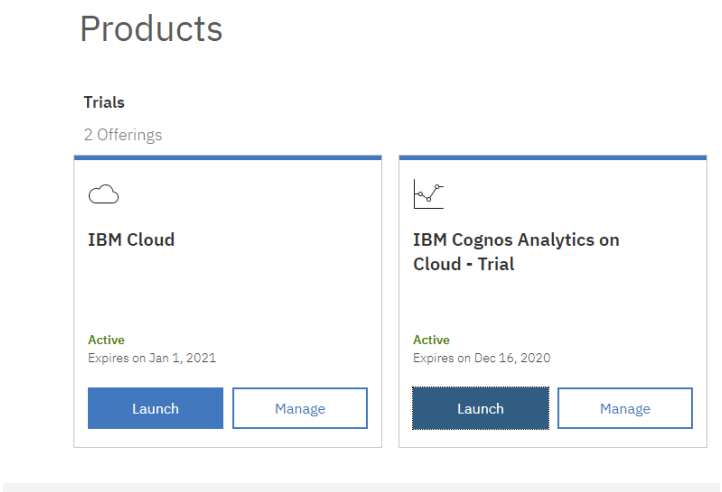
The Default for the values is *Count* so be sure to change it to *SUM*.



Visualization 2: Forecasting and Trend Lines in IBM Cognos

Step 1: Open **two.csv** in IBM Cognos.

1. Log into your IBM Cognos Account or Sign Up for a Free 30 day trial. Once you have logged in, you will see a list of your available products.



2. Launch IBM Cognos Analytics and select the My Content folder on the left panel. Select the + indicator and choose Upload Files. Search for two.csv on your desktop and open the file.

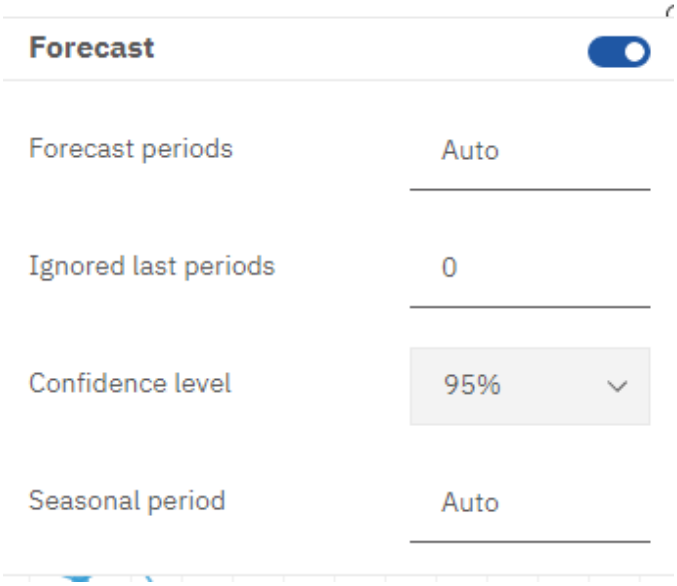
Step 2: Prepare the data for visualization.

1. Return to the Home screen and select New on the bottom of the screen under the Manage tab. Once you select New, select Exploration.
2. It will bring you to the My Content folder where you will select **two.csv** and click the Add button.

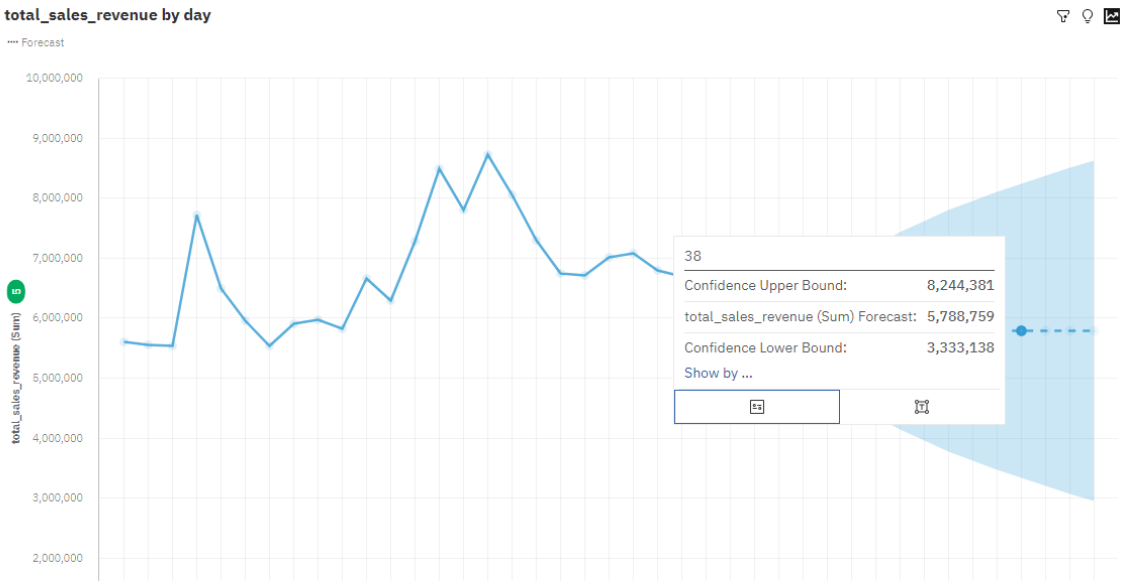
Step 3: Use the IBM Cognos Forecasting Feature.

1. Select New Card and from the drop down menu, select Single. This will bring you to the Create a visualization page.
2. Select Choose a type then choose Line under the Trend section.
3. Drag total_sales_revenue to the y-axis, and day to the x-axis. This will give you a line graph of the total_sales_revenue by day.

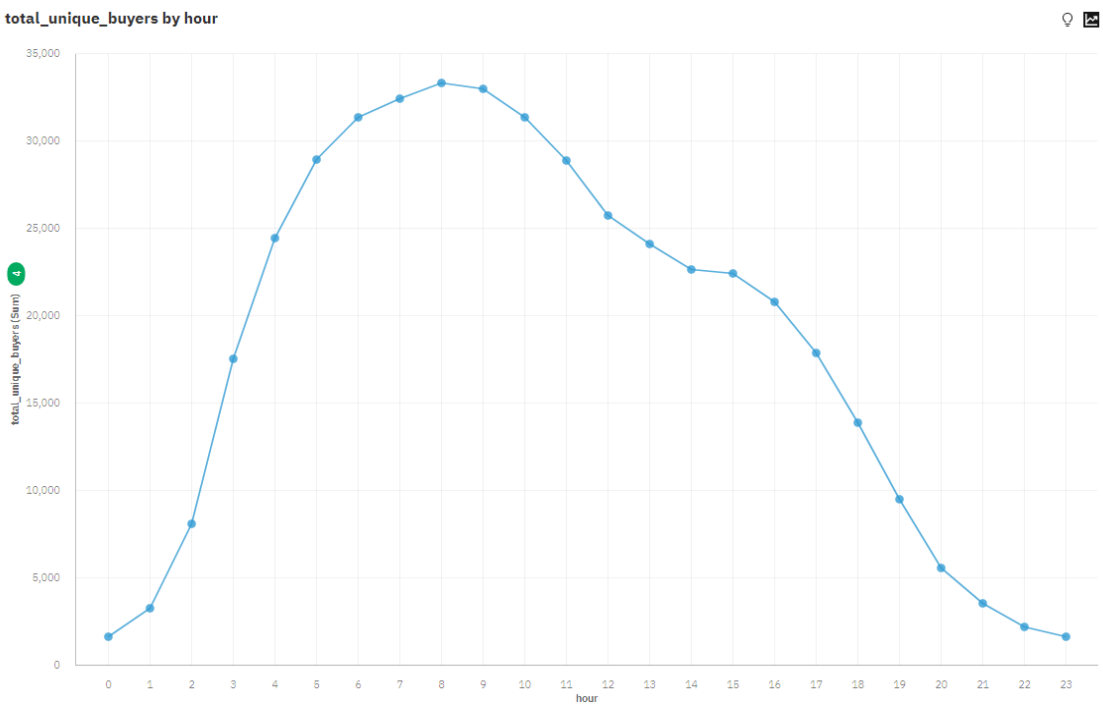
4. Select the black forecasting icon on the upper right corner of the graph and toggle the Forecast to on. Input Forecast periods as 10, a 95% confidence level, and Seasonal period to Auto.



5. Select the 38th day, you will see the Upper Bound and Lower Bound forecast for Sales Revenue on the 38th day.



6. Select New card and follow number 1 and 2 to create a new line graph. Drag total_unique_buyers to the y-axis and hour to the x-axis. You will now see an evident peak at hour 8 for unique buyers.



7. Click the blue save icon in the upper left corner to save your worksheet.

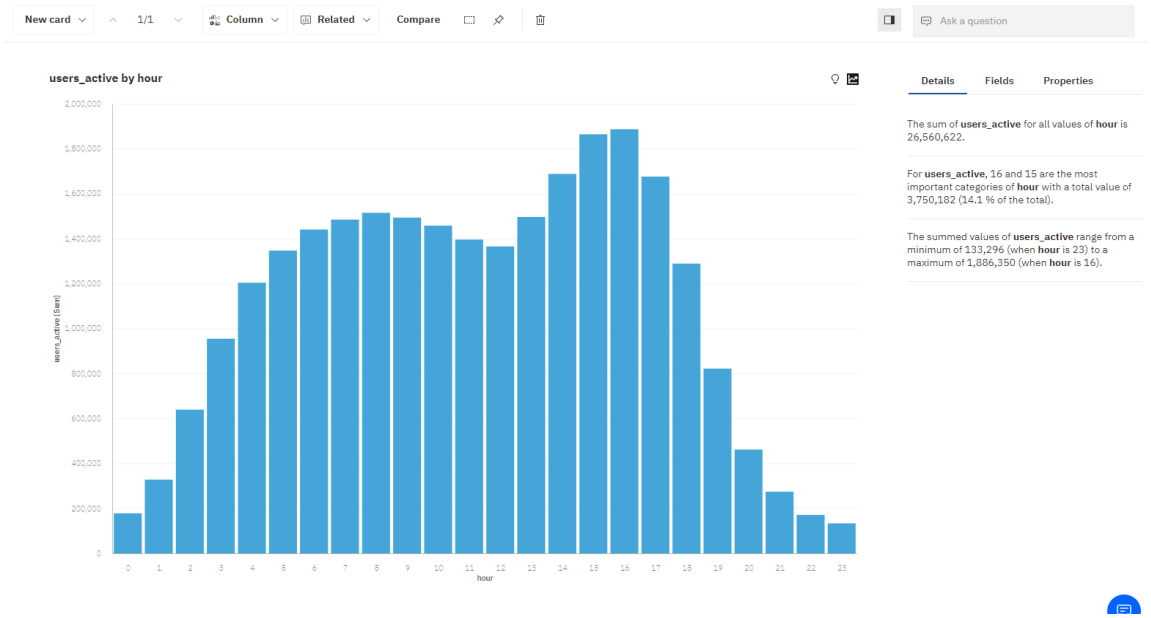
Visualization 3: Column Charts in IBM Cognos

Step 1: Open **three.csv** in IBM Cognos and prepare for visualization.

1. Return the home screen after saving Visualization 3. Select the My Content folder on the left pane and click the + indicator. Upload **three.csv** to your My Content folder.
2. Select New under the Manage and tab and choose Exploration. Add three.csv from your My Content folder.

Step 2: Create a column chart in IBM Cognos.

1. Select a New Card and choose Single. Under the create a visualization screen select Choose a Type. Select Column as the visualization type under the Comparison section.
2. Drag users_active to the Length and hour to the bars. Click on the Details tab on the left to get additional insight on the column chart created. You can see that users are most active between 14:00 and 17:00.

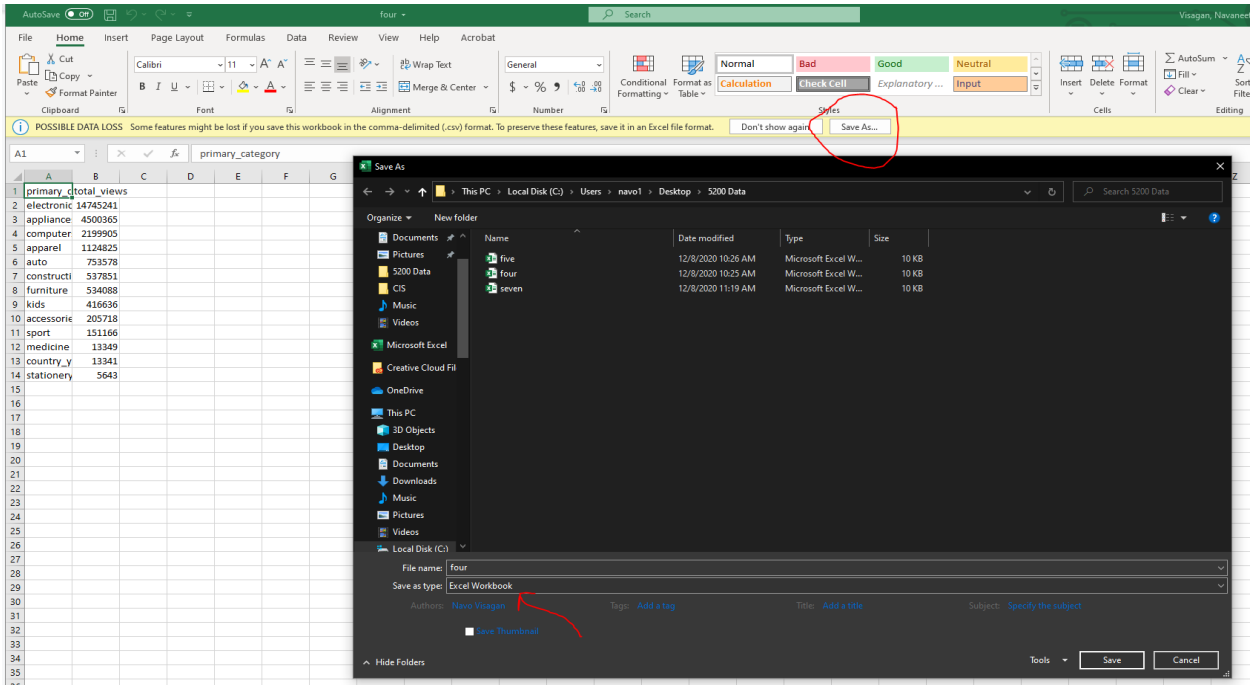


3. Save your card using the blue disk icon in the upper left corner.

Visualization 4 and 7: Bubble Map and Bar Chart in Tableau

Step 1: Convert four and seven CSV files into Excel workbook:

Open **four.csv** and **seven.csv** in Excel and save as Excel Workbook.



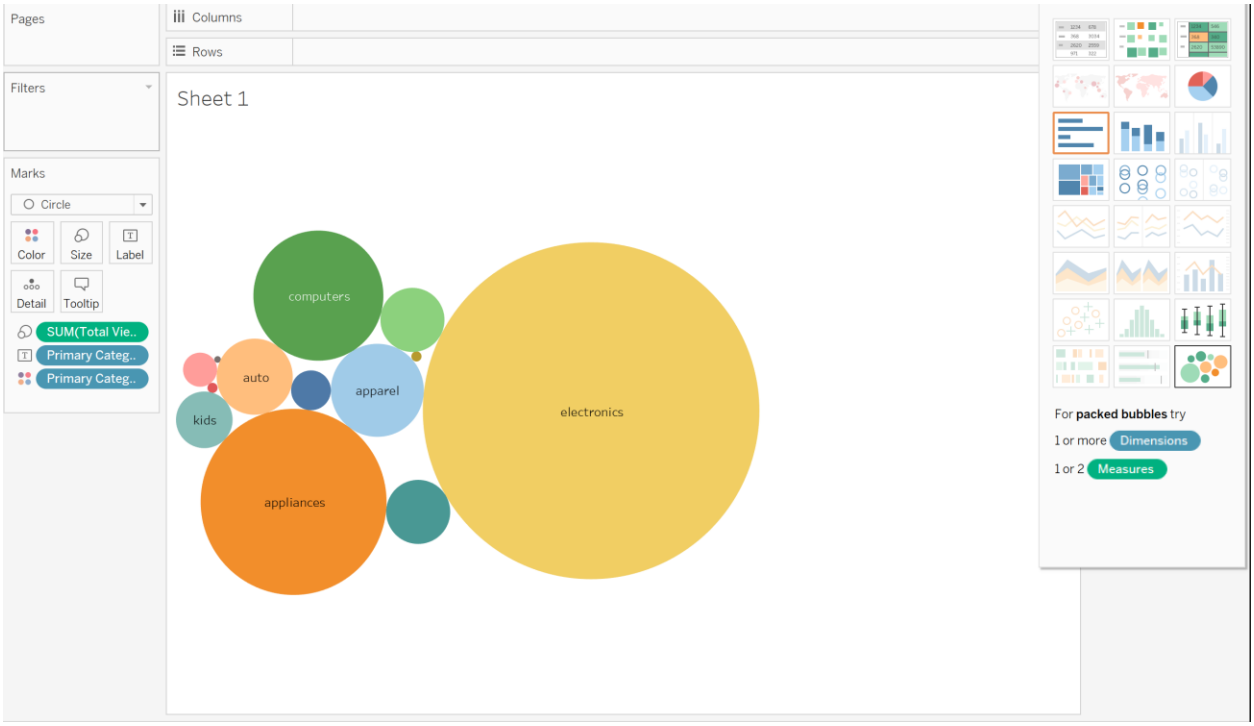
Step 2: Open Tableau and Import:

Open Tableau and choose Microsoft Excel as the data source.

When prompted choose **four.xslx** and move to Sheet 1.

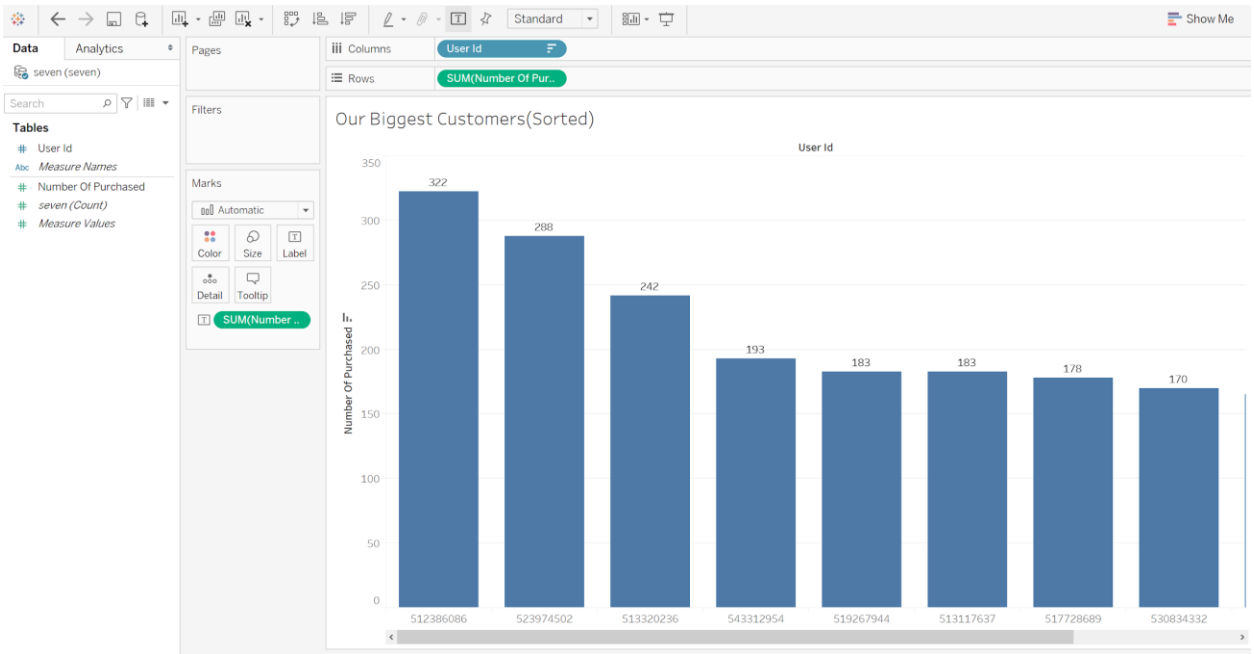
Step 3: Move the Primary Category to the Columns and the Total Views into Rows.

Step 4: Click the Show Me on the top right corner and choose the very bottom right option (Bubble Map).



Step 5: Open a new Tableau Workbook and connect the Data Source to **seven.xlsx.**

Step 6: Move User ID to the Columns Tab and Number of Purchased to Rows.



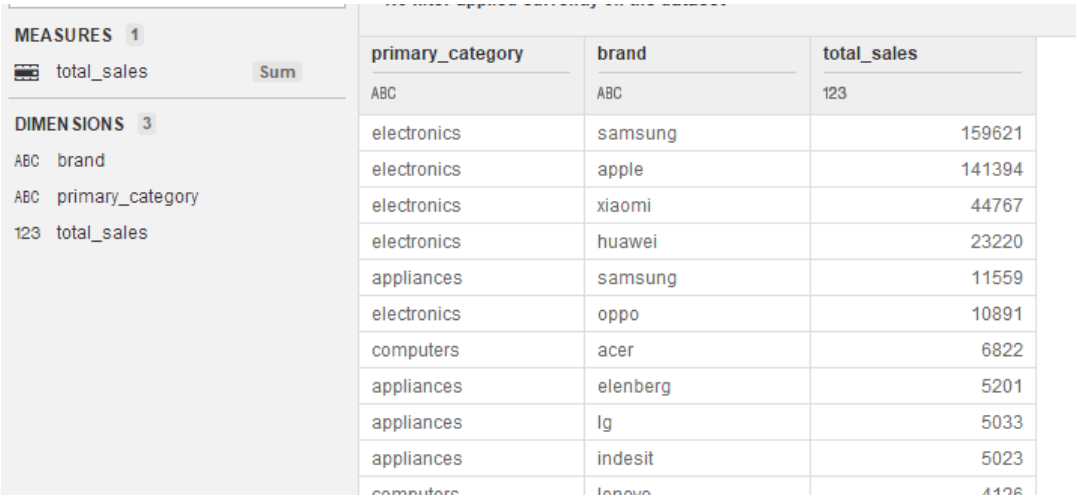
Visualization 6: Clustering in SAP Expert Analytics

Step 1: Open **six.csv** in SAP Expert Analytics.

1. You must launch SAP Predictive Analytics and select the Expert Analytics tab on the left ribbon. Once you are on the Expert Analytics screen, select Expect Analytics to launch the program.
2. In the upper left hand corner, select File and from the drop down menu click new.
3. From the Add New Dataset screen, select Text as your source type, then press the Next button. Select the dataset from your desktop and ensure that the separator is toggled on Delimited By and the drop down menu has Comma selected. Click create to create your new worksheet in SAP Expert Analytics.

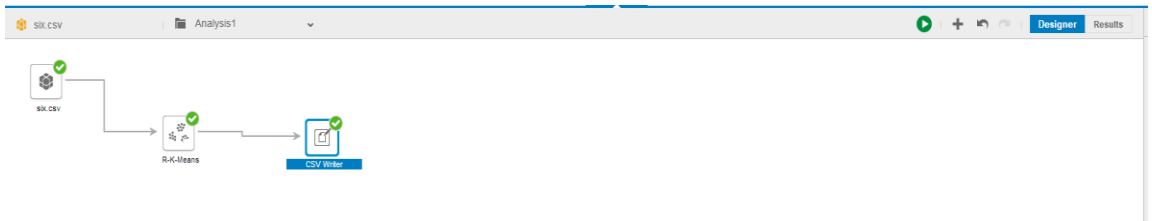
Step 2: Prepare the data for visualization.

1. In the Prepare tab, ensure that your data set has the proper measures and dimensions. Total_sales should be aggregated as a sum, while brand and primary_category should be ABC.



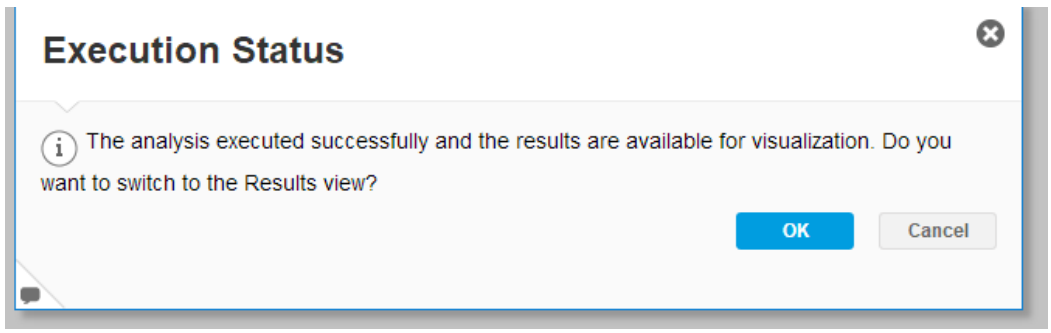
primary_category	brand	total_sales
ABC	ABC	123
electronics	samsung	159621
electronics	apple	141394
electronics	xiaomi	44767
electronics	huawei	23220
appliances	samsung	11559
electronics	oppo	10891
computers	acer	6822
appliances	elenberg	5201
appliances	lg	5033
appliances	indesit	5023

2. Select the Predict tab and you should see that **six.csv** is listed as the dataset. In the algorithms panel on the right of the screen, select R-K-Means and drag it to the center of the screen. You will see that it will link to **six.csv**, indicated by a green arrow.
3. Click the toggle icon on R-K-Means and select Configure Settings or F5. Denote the number of clusters as 3 and select total_sales before clicking Done.
4. From the right panel, select Data Writers and under File Writers, select CSV Writer and drag it to the center screen. You now have **six.csv**, R-K-Means, and CSV Writer linked.
5. Select the toggle icon on CSV Writer, then Configure Settings or F5, and give the file a generic name to be saved locally.
6. Your output should look as follows before selecting the green Run button on the upper pane.



Step3: Configure the visualization.

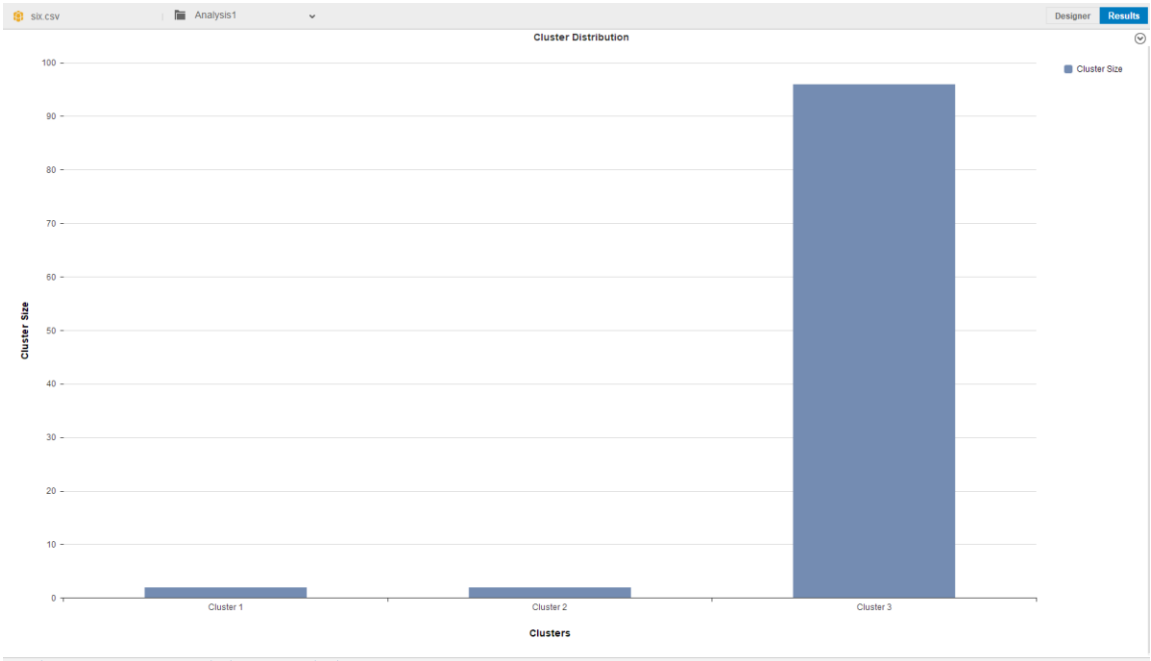
1. After selecting Run, you will get the following notification. Select OK.



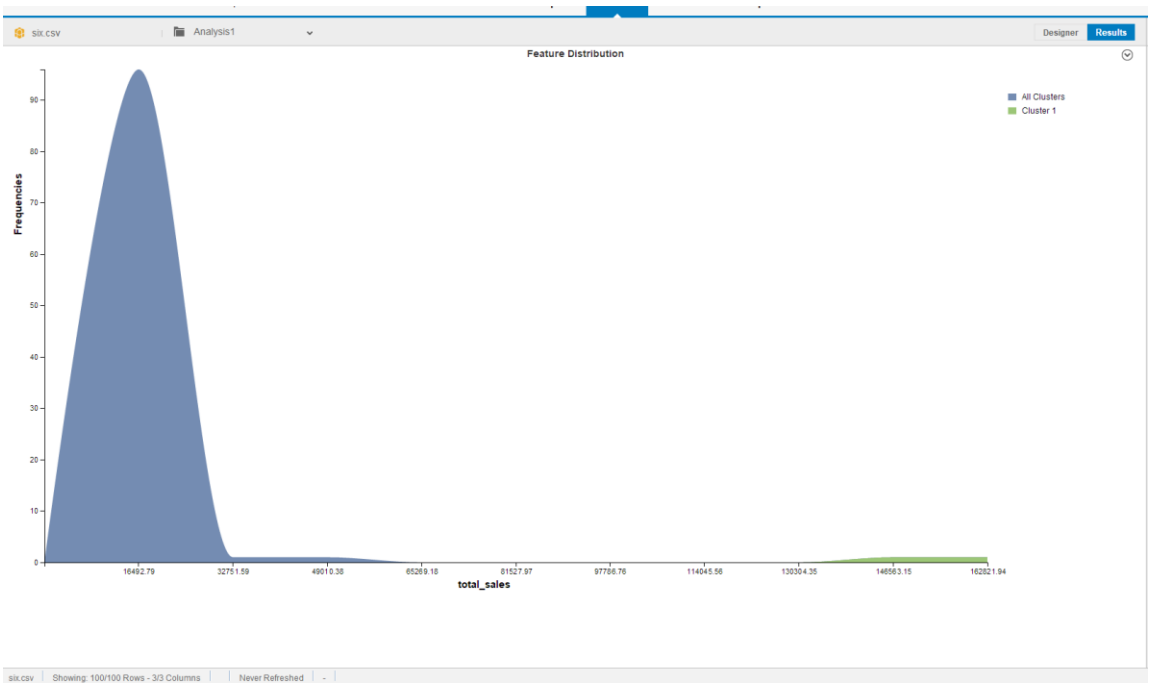
2. On the Results View, you will now see ClusterNumber added as a column to your data set. Select Summary on the right panel to see a summary of the K-Means-Analysis.

```
Summary of the model from R Scripts

Information of the columns used in the algorithm
---------------------------------------------------
Independent Column
 total_sales : Integer

Summary of the Model
             Length Class  Mode
cluster      100    -none- numeric
centers        3    -none- numeric
totss          1    -none- numeric
withinss       3    -none- numeric
tot.withinss   1    -none- numeric
betweenss      1    -none- numeric
size           3    -none- numeric
iter           1    -none- numeric
ifault         1    -none- numeric


Centers
  total_sales
1  150507.500
2   33993.500
3    1392.458


Within cluster sum of squares
[1] 166111765 232136605 352955098


The size of each cluster
[1]  2  2 96
```

3. Select the Bar Column under Cluster Representations to see the distribution of the clusters. As you can see, most of the data lies in cluster 3.



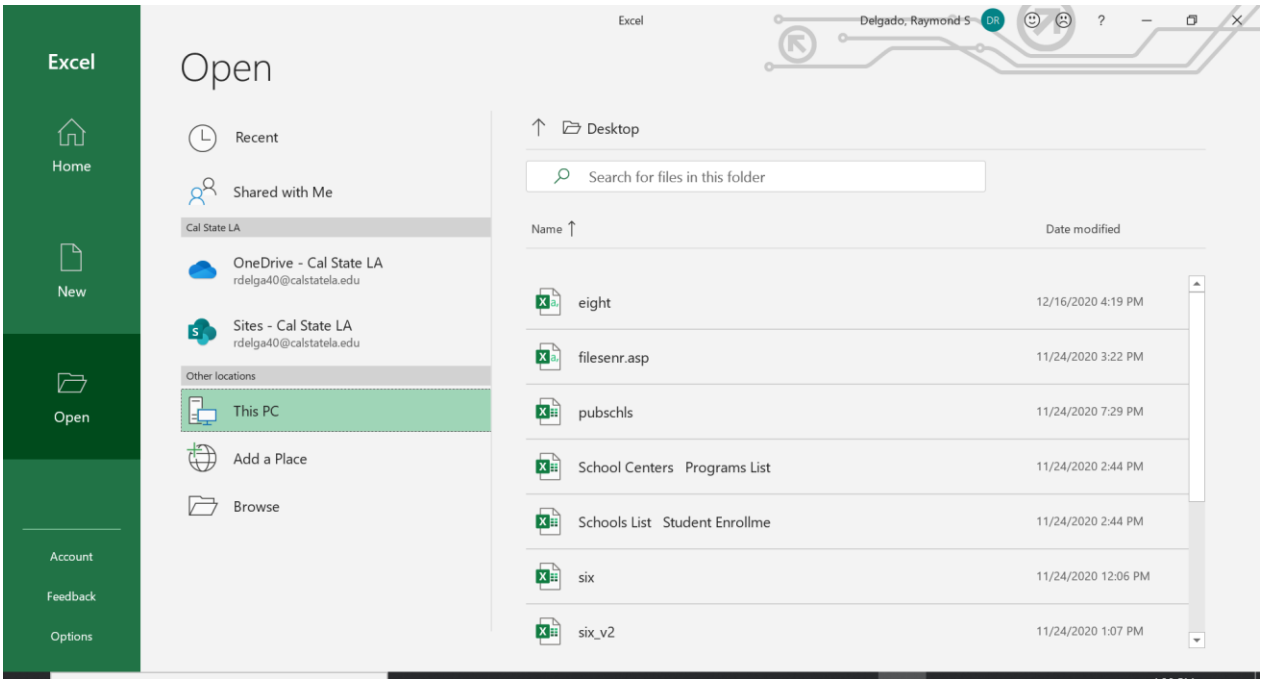
4. Select Feature Distribution under the Cluster Representations tab. You will see that the output is skewed to the left.



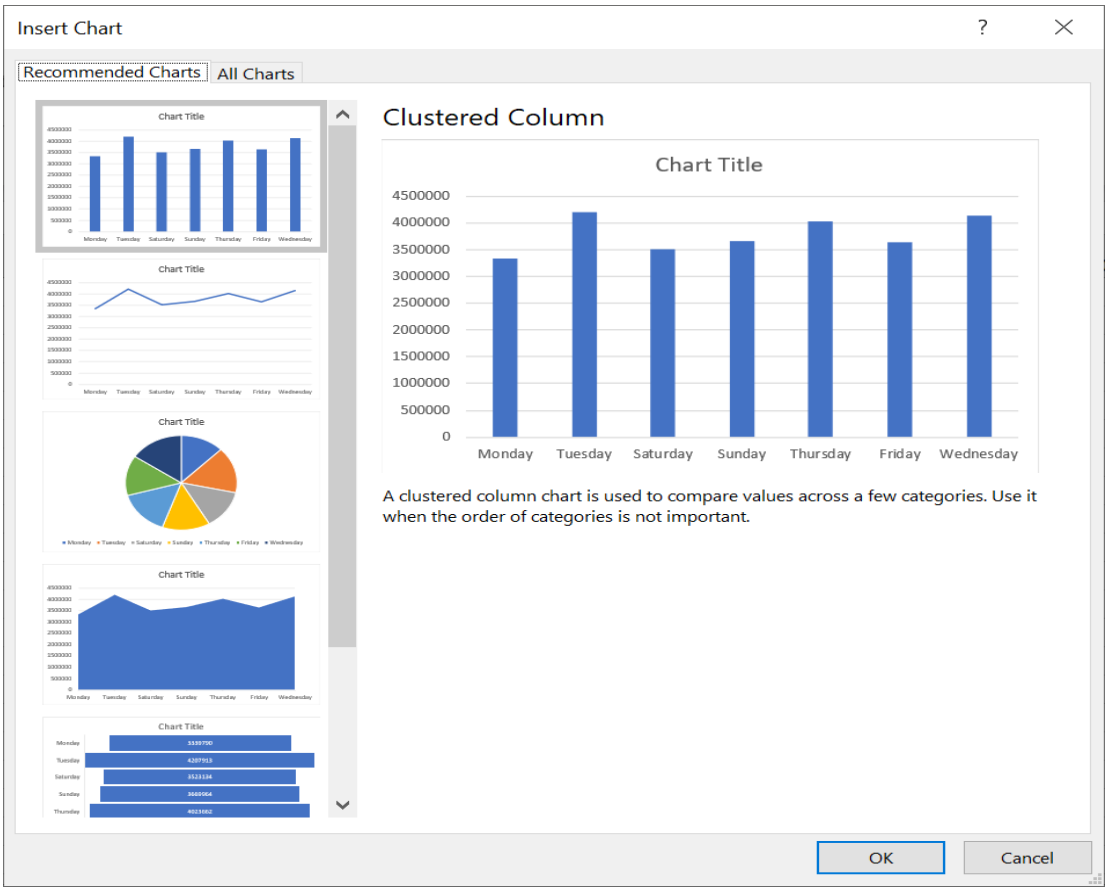
5. Select File in the upper left and Save your worksheet.

Visualization 8: Column Chart in Microsoft Excel

Step 1: Open the "**eight.csv**" in Microsoft Excel.



Step 2: Click on the Insert Tab, then select the recommended chart, and finally select the clustered column chart. Click Ok to confirm your decision.

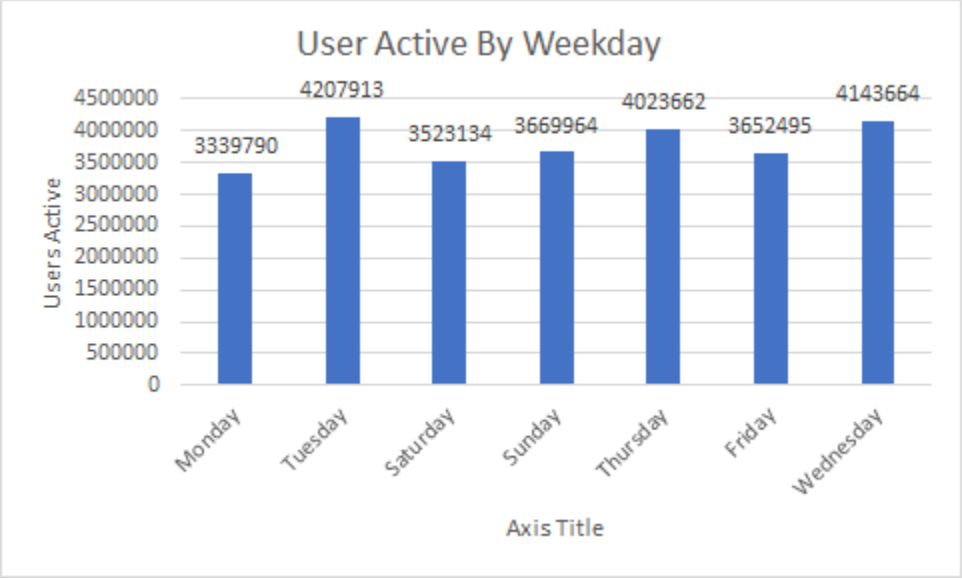


Step 3: Click on the green plus(+) icon to add chart elements. The chart elements that we want to add are:

- Axis titles: We are going to name the y-axis as "Active Users"
- Data Label: This will show us the total number of users active per weekday.

Please Note: It is recommended to rename the chart.

Your chart should look similar to the picture below:



References

1. URL of Data Source: eCommerce behavior data from multi category store
2. URL of Github: https://github.com/neltf/CSULA_CIS_5200
3. URL of References: https://medium.com/tech4she/visualising-e-commerce-user-behaviours-c833def97cc0