HDFS –

It is used to store the store the dataset . Data has been transferred from SMU dev server to Hadoop distributed file system using following commands.

Commands

df\_order = spark.read.option("delimiter", "\\t").option("header", "true").csv("/user/root/OS\_ORDER.txt")

df\_item = spark.read.option("delimiter", "\\t").option("header", "true").csv("/user/root/OS\_ORDER\_ITEMS.txt")

APACHE HIVE-

We imported the datasets from Pyspark datafram to apache Hive. We queried the tables to get the initial overview of the data.

df\_order.write.mode("overwrite").saveAsTable("df\_order")

df\_item.write.mode("overwrite").saveAsTable("df\_item")

TABLEAU-

Tableau is used for exploratory data analysis. We setup a server connection between tableau and Apache hive using Hortonworks Hadoop Hive server and then visualize the data to answer below questions:-

* Most frequent buyer
* Products generating most revenue
* Buyers generating the most revenue
* Trend of orders over month

Screenshot

APACHE PySPARK

We have used Apache PySpark library for data cleaning and data preparation. Following task has been done using PySpark:-

* Dta has been imported from HDFS to Spark dataframe

df\_order = spark.read.option("delimiter", "\\t").option("header", "true").csv("/user/root/OS\_ORDER.txt")

df\_item = spark.read.option("delimiter", "\\t").option("header", "true").csv("/user/root/OS\_ORDER\_ITEMS.txt")

* Data has been imported to Apache Hive for visualization

df\_order.write.mode("overwrite").saveAsTable("df\_order")

#df\_item.write.mode("overwrite").saveAsTable("df\_item")

* Extra columns have been dropped and rows containing NULL values are removed

df\_order = df\_order.drop(\*['CREATE\_DT', 'EXCEPTION\_STATUS'])

df\_order\_noNA = df\_order.where(df\_order.PAY\_DT != 'NULL')

* Both the datafames are merged based on ORDER\_ID

df\_combined = df\_order.join(df\_item,[df\_order\_noNA.ORDER\_ID == df\_item.ORDER\_ID])

FEATURE ENGINEERING

* New column has been derived by ectracting “mm-yyyy” from PAY\_DT
* Profit column has been created based on shop\_price, goods\_price and goods\_number

df\_profit = df\_combined.withColumn('profit', (df\_combined.SHOP\_PRICE - df\_combined.GOODS\_PRICE)\*df\_combined.GOODS\_NUMBER)

* Recency of the orders are created by substracting the order date from most latest order date
* RFM has been calculated for each buyer
* Joining all the dataframe together to perform Unsupervised machine learning

Apache MLlib

We have used the Apache MLlib library to perform Unsupervised machine learning i.e K-Means clustering on our dataset.

* We started with finding the optimal number of clusters based sum of square of distance and plotted elbow curve and found 5 clusters as……
* Then we performed k-means clustering with k= 5 clusters based on Profit, recency and order\_count
* After that we appended clusters numbers generated by k-means clustering to our existing dataframe