Here is the link to the dataset: <https://www.kaggle.com/sherrytp/airline-delay-analysis>

How to run Azure ML:

1. Log into Azure(We have a free $100 credit if we use our Cal State LA email): <https://azure.microsoft.com/en-us/account/>
2. Create a new Machine Learning resource
3. Upload the dataset to your instance of the machine learning resource
4. Click and drag the following modules: The raw dataset, The remove missing value dataset, and The split dataset module, convert to CSV module
   1. Module configuration:
      1. Removing Missing value: Max missing values 1, Min missing value 0, remove entire row
      2. Split Dataset: .1

Python Code:

Import csv

Str = “ab-cd”

Split\_string = a\_string.split(“-”,1)

Substring = split\_string[0]

print(substring)

ab

List of Datasets we can use:

<https://www.kaggle.com/mkechinov/ecommerce-behavior-data-from-multi-category-store>  
 14 gig

Help: <https://medium.com/tech4she/visualising-e-commerce-user-behaviours-c833def97cc0>

------------------------------------------------------------------------------------------------------------------------

**TO DO LIST:**

1. **Upload this to Amazon**
2. **Tables, Views, Queries**
3. **Visualization tool (Power BI, Tableau,...)**
4. **Presentation ( Dec 4)\***
5. **Lab tutorial ( Dec 9)\***
6. **Term paper**

\*So far I think

---------------------------------------------------------------------------------------------------------------

Data Set:

What you are going to do with this data set.

* Remove Nulls-Yes
  + Nulls only occurring in category\_brand and brand
  + PseudoCode: If category\_code AND brand are null REMOVE

Top Categories

-Most Expensive Categories

Least Expensive

Top Brands

-Most Expensive Brands?

What time events have the most “View events”, “Cart Events”, “Purchase Events”

---------------------------------------

ssh to Oracle BDCE: ssh [your\_username]@129.150.69.91

--Get work file from AWS:

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

2019-Nov.csv.zip file in case we need it.

<https://groupthreebucket.s3-us-west-1.amazonaws.com/2019-Nov.csv.zip>

>>>move your files into /data folder when you unzip and upload the data file into hdfs to avoid not enough space.

--Unzip and save csv files to HDFS:

mv 2019-Oct.csv.zip /data/

cd /data/

Unzip 2019-Oct.csv.zip

--Create work directory:

hdfs dfs -mkdir ecommerce

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

hdfs dfs -ls

hdfs dfs -ls ecommerce

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

hdfs dfs -chmod -R o+w .

– connect to 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=interactive bdcsce\_admin

To create database:

CREATE DATABASE IF NOT EXISTS GroupThree;

--Use GroupThree;

To drop the rows with null values:

DELETE FROM TABLE

WHERE col1 is NULL or col2 is NULL.. colN is NULL;

To drop the rows with empty strings:

DELETE FROM TABLE

WHERE col1 = ‘’ or col2 = ‘’... col3 = ‘’;

>>> To drop the NULL columns

Split category:

STRING\_SPLIT ( string , separator )

DROP TABLE IF EXISTS ecommerce;

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');

DESCRIBE ecommerce;

SELECT COUNT(\*) FROM ecommerce;

DROP VIEW IF EXISTS ecommerce\_view;

CREATE VIEW IF NOT EXISTS ecommerce\_view AS

SELECT

event\_time,

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!='';

SELECT COUNT(\*) FROM ecommerce\_view;

select size(split(category\_code, '\\.')) as s,category\_code, split(category\_code, '\\.')[4] FROM ecommerce\_view order by s DESC limit 10;

SELECT event\_time FROM ecommerce\_view limit 10 ;

SELECT event\_type FROM ecommerce\_view group by event\_type ;

SELECT category\_code FROM ecommerce\_view group by category\_code order by category\_code ;

SELECT count(event\_type) as c,primary\_category from ecommerce\_view group by primary\_category order by c DESC limit 10;

SELECT count(event\_type) as c,brand from ecommerce\_view group by brand order by c DESC limit 10;

View: Top 5 Brands

Select brand, sum(price)

from ecommerce\_view

Group by brand

Order by brand ASC LIMIT 5;

----Review disctinct titles

SELECT DISTINCT ...FROM [table name];

--Count disctinct titles

SELECT COUNT(DISTINCT ...) FROM [table name];

--Run query to select top 100

--Top 100 businesses

--Negative or Positive reviews

>>>I think they are easy to make.

>>>For market decide what brand should take off from the website: the most viewed brand but finally not buy for the same user\_id, the most viewed brand and put into cart but finally not buy for the same user\_id.

SELECT count(ecommerce\_view.brand) as c,ecommerce\_view.brand,ecommerce\_view.category\_code FROM ecommerce\_view

LEFT JOIN (select product\_id,event\_time,user\_id from ecommerce\_view where event\_type ='view' and brand!='') AS view ON view.user\_id=ecommerce\_view.user\_id

where ecommerce\_view.event\_type='purchase' and brand!='' and ecommerce\_view.event\_time>view.event\_time group by ecommerce\_view.brand,ecommerce\_view.category\_code order by c DESC;

SELECT count(ecommerce\_view.brand) as c,ecommerce\_view.brand,ecommerce\_view.category\_code FROM ecommerce\_view

LEFT JOIN (select product\_id,event\_time,user\_id from ecommerce\_view where event\_type ='cart' and brand!='') AS cart ON cart.user\_id=ecommerce\_view.user\_id

LEFT JOIN (select product\_id,event\_time,user\_id from ecommerce\_view where event\_type ='view' and brand!='') AS view ON view.user\_id=ecommerce\_view.user\_id

where ecommerce\_view.event\_type='purchase' and brand!=''

and ecommerce\_view.event\_time>cart.event\_time and cart.event\_time>view.event\_time

group by ecommerce\_view.brand,ecommerce\_view.category\_code order by c DESC;

>>>To see what kind of product or brand are desired to buy: split the time to see the sorted gap time for users switch from view to cart, or cart to purchase.

SELECT

SUM(unix\_timestamp(ecommerce\_view.event\_time)-unix\_timestamp(view.event\_time))/COUNT(\*) as gap,

ecommerce\_view.brand,ecommerce\_view.primary\_category FROM ecommerce\_view

LEFT JOIN (select min(event\_time) as event\_time,product\_id,user\_id from ecommerce\_view where event\_type='view' group by product\_id,user\_id) AS view ON view.user\_id=ecommerce\_view.user\_id and view.product\_id=ecommerce\_view.product\_id

where ecommerce\_view.event\_type='purchase' and brand!='' group by ecommerce\_view.brand,ecommerce\_view.primary\_category order by gap DESC limit 10;

My version of the ecommerce\_view:

CREATE VIEW IF NOT EXISTS ecommerce\_view AS

SELECT

event\_time,

split(split(event\_time, '\\-')[2], '\\ ')[0] AS day,

split(split(split(event\_time, '\\-')[2], '\\ ')[1], '\\:')[0] 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,

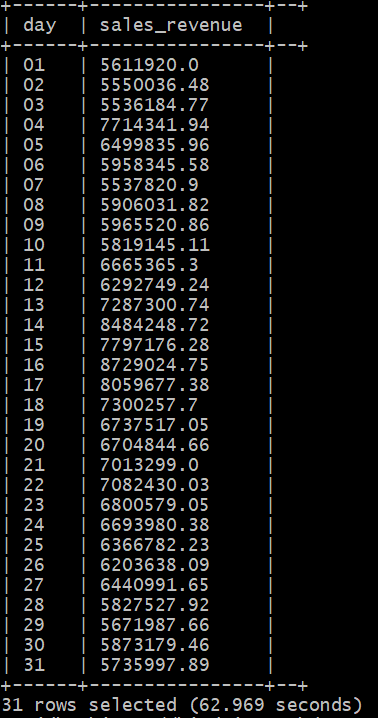
user\_session

FROM ecommerce

WHERE category\_code!='' AND brand!='';

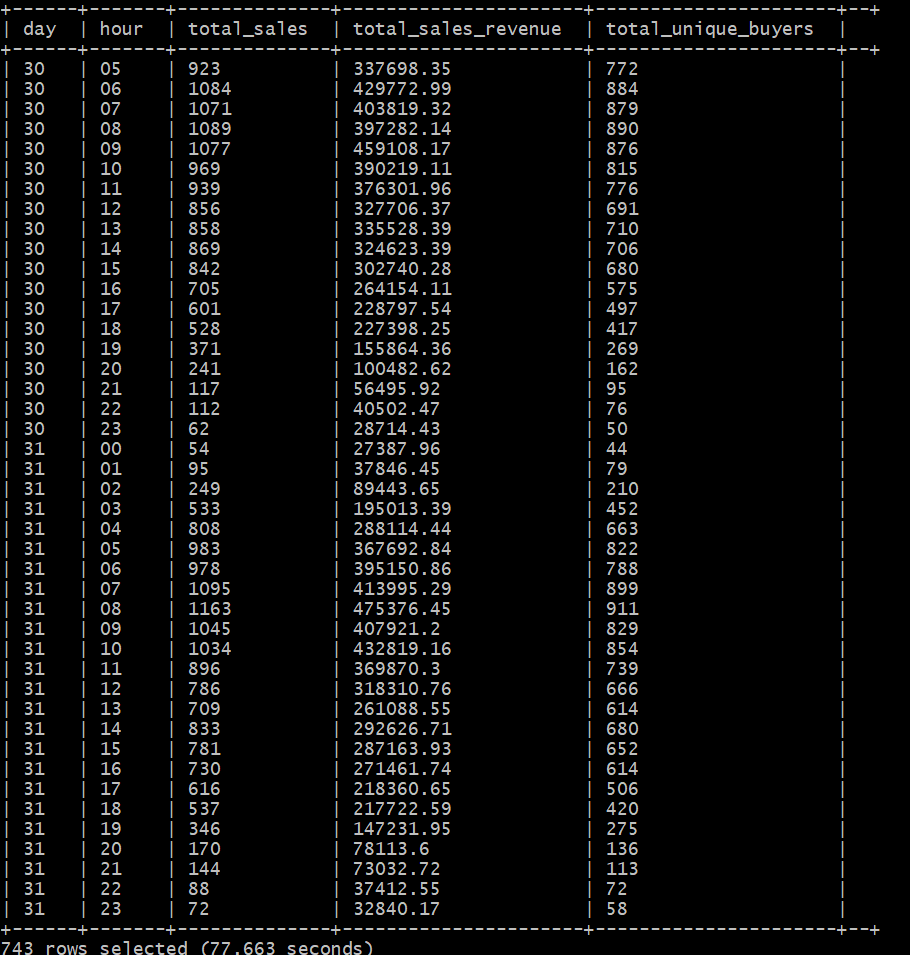
**Query to get sales revenue by day:**

SELECT day, ROUND(SUM(price), 2) AS sales\_revenue FROM ecommerce\_view WHERE event\_type='purchase' GROUP BY day ORDER BY day ASC;



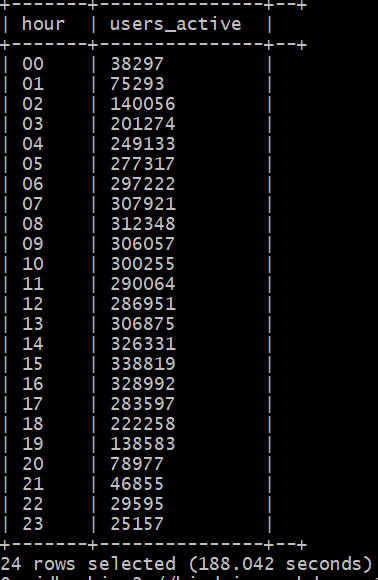
**Query to get total number of purchase by Day and hour (we can use this to make a line chart)**

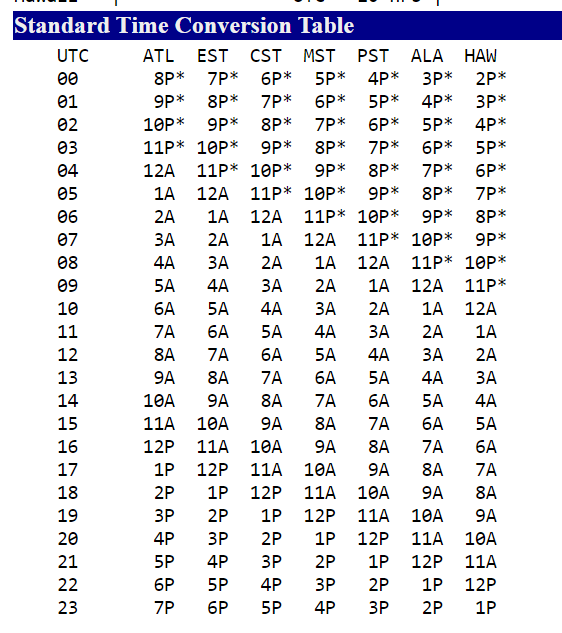
SELECT day, hour , COUNT(event\_type) AS total\_sales, ROUND(SUM(price), 2) AS total\_sales\_revenue, COUNT(DISTINCT user\_id) as total\_unique\_buyers FROM ecommerce\_view WHERE event\_type='purchase' GROUP BY day, hour ORDER BY day ASC, hour ASC;



**Query to display how many users are active by hour**

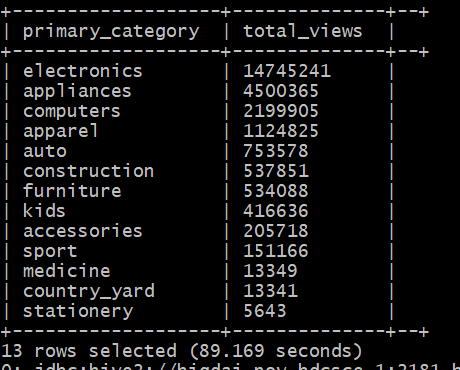
SELECT hour , COUNT(user\_id) AS users\_active FROM ecommerce\_view GROUP BY hour ORDER BY hour ASC;





**Query to display the most viewed product categories**

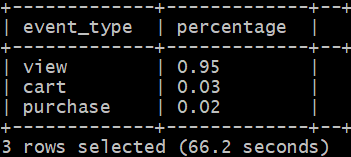
SELECT primary\_category, COUNT(event\_type) AS total\_views FROM ecommerce\_view WHERE event\_type = 'view' GROUP BY primary\_category ORDER BY total\_views DESC;



Query to find the percentage of each event:

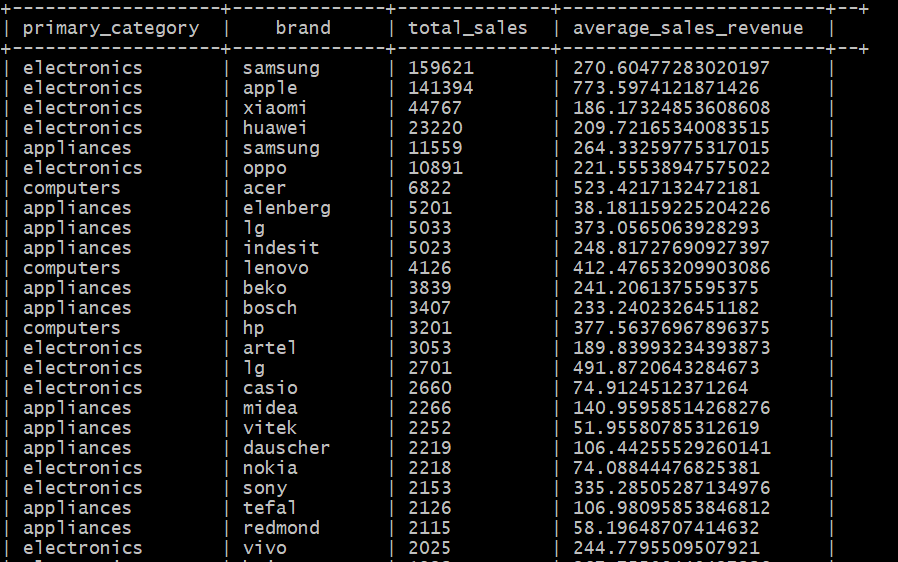
SELECT event\_type,count(event\_type) AS number\_of\_events, ROUND(CAST(COUNT(event\_type) AS float)/26560622, 2) AS percentage FROM ecommerce\_view GROUP BY event\_type ORDER BY percentage DESC;

SELECT event\_type,count(event\_type) AS number\_of\_events, ROUND(CAST(COUNT(event\_type) AS float)/CAST(COUNT(event\_type) AS float), 2) AS percentage FROM ecommerce\_view GROUP BY event\_type ORDER BY percentage DESC;



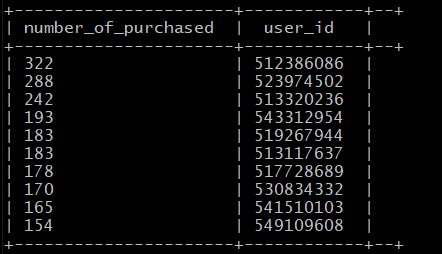
Query to list the top 100 primary category, brand and total sales by product category and brand

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;



**Query to list the top 10 users who purchased more than once:**

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;



CREATE TABLE ttt row format delimited fields terminated by '|' 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 5;

INSERT OVERWRITE LOCAL DIRECTORY '/user/fchen26/ecommerce' ROW FORMAT DELIMITED FIELDS TERMINATED BY '|' select \* from ttt;

DROP TABLE IF EXISTS

DROP TABLE IF EXISTS test;

CREATE TABLE test

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/fchen26/ecommerce'

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 5;

**Team Project Group 3:** Taya Stewart, Nel Talverdi Feridani, Raymond Delgado, Navaneeth Visagan, Frank Chen

1. **Data Set:**

This dataset consists of seven months(October 2019 to April 2020) of sales data. The sales data is categorize by four events:

1. View: The user has viewed an item
2. Cart: The user has put the item in their cart
3. Remove\_From\_Cart: The user removed the item from the cart
4. Purchased: The user has purchased the item

[https://www.kaggle.com/mkechinov/ecommerce-behavior-data-from-multi-category-sto](https://www.kaggle.com/mkechinov/ecommerce-behavior-data-from-multi-category-store)

1. **Size:** 13.67 GB
2. **Objective:**

Our team plans to serve as consultants for the major eCommerce brands included in the data set: LG, Apple, HP, and Samsung. Utilizing Hive, we will clean the dataset and develop queries that will categorize the data into necessary subtopics. Using quality management techniques, our team of consultants will analyze the results to fully address our clients concerns. Additionally, we will employ business intelligence programs such as Tableau, Power BI, and SAP Predictive Analytics to visualize the large amounts of data. Our final presentation will address key business findings, including a market analysis, profitability indexes according to product, and consumer engagement.

Nowadays online shopping is very popular. E-commerce websites such as Amazon, ebay, shopify,...have a great impact on our daily lives. The data generated from user activity on ecommerce websites is extremely valuable. It includes information about consumers and how they interact with the websites. Based on this information, these companies can try to understand their users' behaviour on their websites and improve their products and the user interaction based on it.

That is why our team decided to work on this project. Our goal is to analyze ecommerce data and provide insight that can be used to enhance the ecommerce platform.

We decided to use eCommerce Behaviour Data from Multi-category Store from Kaggle. With over 42 million rows of data, it tracks second-by-second actions taken by every user on the website for the month of October 2018. Each user makes three main actions — view a product (event\_type = “view”), add a product to cart (event\_type = “cart”) or make a purchase (event\_type = “purchase”).

Nowadays online shopping is very popular. E-commerce websites such as Amazon, ebay, shopify,...have a great impact on our daily lives. The data generated from user activity on these websites is extremely valuable. It includes information about users actions when they visit these websites. For example when they view or purchase a product and we can find out what brands do people prefer. That is why our team thought it can be interesting to work on this topic. Our goal was to analyze this data and provide insights that can be used to improve ecommerce platforms.

We used eCommerce Behaviour Data from Multi-category Store from Kaggle. we decided to focus on one of the largest data files, 1 month. Oct 2019.

We are also able to observe product characteristics such as category, brand and price, event\_time that let us to create multiple metrics to do our analysis.

Oracle big data service.