CIS4560 Term Project Tutorial

Authors: Bryan Mendoza, Kevin De La Torre,

Jesus Cortez Bonilla, Joshua Rowill Koa,

Samuel Mendoza

Instructor: Jongwook Woo

Date: 12/20/2020

Term Project Group 4 Lab Tutorial

US Used Cars Dataset Analysis using Hadoop, Tableau, and SAP Analytic Cloud

---

Objectives

List what your objectives are. In this hands-on lab, you will learn how to:

- Get data from website and upload to Hadoop
- Create directory for file
- Use Hive to create tables
- SQL commands to perform the analysis.
- Visualization

Platform Spec

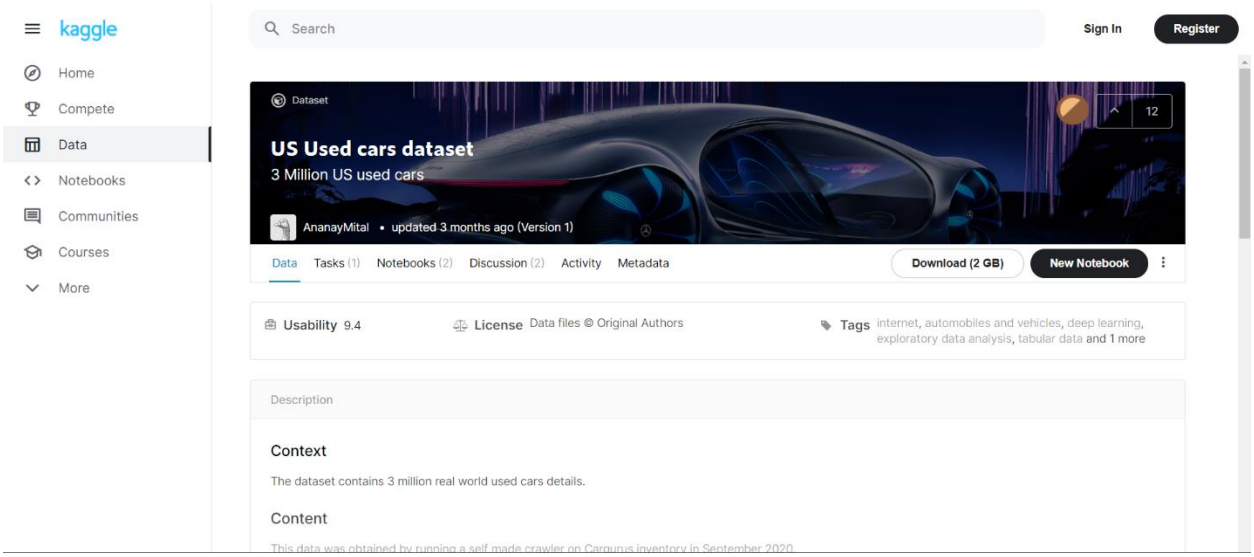
- Hadoop / Pig
- CPU Speed: 2.20GHz
- # of CPU cores: 12
- # of nodes: 3
- Total Memory Size: 180gb

Step 1: Get data manually by Downloading from website

This step is to get the necessary data file.

1. Go to https://www.kaggle.com/ananaymital/us-used-cars-dataset

2. Download (2GB) file



Step 2: Upload Car Data to Hadoop

We will begin to upload and extract the zip file in Hadoop file system.

1. Open A Shell terminal such as Git Bash, Minty, Putty, and run the ssh command to connect to the Hadoop cluster.

2. To connect to the Hadoop cluster use:

    - ssh kdelat15@129.150.64.74 (replace kdelat15 with your username)

3. Enter your password (should be the same as your username)

```
$ ssh kdelat15@129.150.64.74
-- 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.

kdelat15@129.150.64.74's password:
```

4. Once logged into your cluster we can begin to upload the dataset.

5. To upload the dataset, we will use PSCP to transfer the local file to the Hadoop Cluster. To begin, open your command prompt in Windows.

6. Once opened, you will type in the following:

   - pscp –P 22 C:\Users\Kevin\Downloads\used_cars_data.csv.zip kdelat15@129.150.64.74:/home/kdelat15 (The first highlighted portion will differ based on where you download the file, don't forget to change the username to your own username)

   - If pscp does not work for you, you will need to download it from the following: Putty

```
C:\Users\Kevin>pscp -P 22 C:\Users\Kevin\Downloads\used_cars_data.csv.zip kdelat15@129.150.64.74:/home/kdelat15
kdelat15@129.150.64.74's password:
used_cars_data.csv.zip    | 2233535 kB | 470.4 kB/s | ETA: 00:00:00 | 100%

C:\Users\Kevin>pscp -P 22 C:\Users\Kevin\Downloads\used_cars_data.csv.zip kdelat15@129.150.64.74:/home/kdelat15
```

7. Once the file completes its upload, you will want to check your Hadoop cluster to ensure it uploaded successfully. To do so, enter the following:

   - ls –al (this will show you all the files in your Hadoop cluster)

8. Once you note that the file is uploaded, you will need to unzip it since we uploaded a .zip file. To unzip it you would enter the following:

- unzip used_cars_data.csv.zip

```
-bash-4.1$ ls -al
total 2233548
drwx------.  2 kdelat15 kdelat15       4096 Nov 12 03:30 .
drwxr-xr-x. 42 root     root           4096 Nov 11 22:10 ..
-rw-r--r--.  1 kdelat15 kdelat15 2287140843 Nov 12 04:49 used_cars_data.csv.zip
-bash-4.1$ unzip used_cars_data.csv.zip
Archive:  used_cars_data.csv.zip
  inflating: used_cars_data.csv
-bash-4.1$ ls used_cars_data.csv
used_cars_data.csv
```

9. Once the file is fully uncompressed, we can now start to transfer it over to your HDFS. Before we begin transferring it over, we need to create a folder to save it in. To do so, enter the following:
   - hdfs dfs -mkdir UsedCarData
10. We then want to make sure we created the folder so we will now list the directories by using **-ls**

```
-bash-4.1$ hdfs dfs -mkdir UsedCarsData
-bash-4.1$ hdfs dfs -ls
Found 1 items
drwxr-xr-x   - kdelat15 hdfs          0 2020-11-12 04:53 UsedCarsData
```

11. Now we know that the directory was created, we can begin to move the dataset into the UsedCarsData directory by using the –put command:
   - hdfs dfs –put used_cars_data.csv UsedCarsData

12. We can then check to ensure the file transferred over to the correct directory: **hdfs dfs –ls UsedCarsData** the output will show the directory along with the fully unzipped .csv inside of it.

```
-bash-4.1$ hdfs dfs -put used_cars_data.csv UsedCarsData
-bash-4.1$ hdfs dfs -ls UsedCardsData
Found 1 items
-rw-r--r--   2 kdelat15 hdfs 9980208148 2020-11-12 04:56 UsedCarsData/used_cars_data.csv
```

13. Now that the dataset is uploaded, we can now move onto the next part, which is creating the tables and queries in hive.

Step 3: Create the tables in Hive

This step will allow us to create tables from the columns

1. Run the following to run beeline and !connect
   - -bash-4.1$ hdfs dfs -chmod -R o+w .

- -bash-4.1$ beeline

2. After the beeline copy and paste the following:

   !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

3. Press enter when asked for a password

4. Call your database in Hive as follows:

```
0: jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigd> use kdelat15;
```

```
Use your own database by replacing kdelat15
```

5. Once connected, you will need to create a table using the following Hive Command.

Copy and paste the following into Hadoop

```
DROP TABLE IF EXISTS usedcartestfinal;

--create table usedcartestfinal
CREATE EXTERNAL TABLE usedcartestfinal (vin STRING, back_legroom
STRING, bed STRING, bed_height STRING, bed_length STRING, body_type
STRING, cabin STRING, city STRING, city_fuel_economy STRING,
combine_fuel_economy STRING, daysonmarket STRING, dealer_zip STRING,
description STRING, engine_cylinders STRING, engine_displacement
STRING, engine_type STRING, exterior_color STRING, fleet STRING,
frame_damaged STRING, franchise_dealer STRING, franchise_make STRING,
front_legroom STRING, fuel_tank_volume STRING, fuel_type STRING,
has_accidents STRING, height STRING, highway_fuel_economy STRING,
horsepower STRING, interior_color STRING, isCab STRING, is_certifed
STRING, is_cpo STRING, is_new STRING, is_oemcpo STRING, latitude
STRING, length STRING, listed_date STRING, listing_color STRING,
listing_id STRING, longitude STRING, main_picture_url STRING,
major_options STRING, make_name STRING, maximum_seating STRING,
mileage STRING, model_name STRING, owner_count STRING, power STRING,
price STRING, salvage STRING, savings_amount STRING, seller_rating
STRING, sp_id STRING, sp_name STRING, theft_title STRING, torque
STRING, transmission STRING, transmission_display STRING, trim_id
STRING, trim_name STRING, vehicle_damage_category STRING,
wheel_system STRING, wheel_system_display STRING, wheelbase STRING,
width STRING, year INT)

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
STORED AS TEXTFILE LOCATION '/user/kdelat15/UsedCarsData'
TBLPROPERTIES ('skip.header.line.count'='1');
```

Step 4: Creating Queries and Viewing their Output

This step we are creating queries and outputting files for visualization

1. Now that the tables have been created, we can then run our queries. This query will tell us exactly which are the top ten cars that are for sale in the United States.
2. Ensure that you are still using your table:
   - Use kdelat15;
3. Enter the following into your hive:

```
SELECT make_name, COUNT(model_name) AS total_for_sale
FROM usedcartestfinal
Group By make_name
Order By total_for_sale DESC
LIMIT 10;
```

4. When the query runs it may seem like it froze, since it is a large dataset it takes a few minutes for it to parse through all the data.
5. Once it completes it will look like this:

make_name	total_for_sale
Ford	476277
Chevrolet	376876
Toyota	239097
Nissan	217882
Honda	214447
Jeep	168427
Hyundai	136086
Kia	112322
RAM	102540
GMC	99283

6. We can now see which are the top ten cars that were for sale.
7. This next query will be for the total number of car types that are for sale.
8. Enter the following in Hive:

```
SELECT body_type, COUNT(make_name) AS total_body_type
FROM usedcartestfinal
Group By body_type
Order By total_body_type DESC
LIMIT 10;
```

9. Once it completes it should look like the following:

body_type	total_body_type
SUV / Crossover	1416312
Sedan	741973
Pickup Truck	474550
Hatchback	88366
Minivan	79800
Coupe	71594
Van	47163
Wagon	40492
Convertible	26002
	13543

10. This following query will show us what are the top car types in 25 cities throughout the United States.
11. Enter the following in Hive:

```
SELECT body_type, city, COUNT(body_type) AS total_for_body
FROM usedcartestfinal
Group By body_type, city
Order By total_for_body DESC
LIMIT 25;
```

12. Once the command is done, it should look like the following:

body_type	city	total_for_body
SUV / Crossover	Houston	20218
SUV / Crossover	San Antonio	11875
Sedan	Houston	11289
SUV / Crossover	Columbus	9219
SUV / Crossover	Miami	8090
Pickup Truck	Houston	7552
SUV / Crossover	Jacksonville	7414
SUV / Crossover	Austin	7012
SUV / Crossover	Las Vegas	6889
Sedan	San Antonio	6664
SUV / Crossover	Tampa	6604
SUV / Crossover	Cincinnati	6396
SUV / Crossover	Orlando	6391
SUV / Crossover	Dallas	6299
SUV / Crossover	Phoenix	6181
SUV / Crossover	Columbia	6151
SUV / Crossover	Indianapolis	6103
Sedan	Miami	5854
SUV / Crossover	Denver	5474
Sedan	Las Vegas	5214
SUV / Crossover	Springfield	5181
SUV / Crossover	Charlotte	4999
Sedan	Jacksonville	4881
SUV / Crossover	Madison	4820
Sedan	Phoenix	4714

13. As the idea is to identify great deals in the market, we want to evaluate dealers who would offer us the most savings.
14. Enter the following into Hive:

```
SELECT sp_id, sp_name, AVG(savings_amount) AS
Average_Savings FROM usedcartestfinal
GROUP BY sp_id, sp_name
ORDER BY Average_Savings DESC
LIMIT 1000;
```

15. Limiting the query to 1,000 rows, it will generate an output with a size of 40 kilobytes.

16. After a brief 5 minutes, the result should look something like this:

sp_id	sp_name	average_savings
285764.0	Hubbard Auto Center of Scottsdale	15780.0
433597.0	ISSIMI, Inc	12265.916666666666
400763.0	Land Rover North Dade	12071.0
404217	iLusso	12067.785714285714
438358	Precision Imports	11914.0
416003	United Sports Autotmotives LLC	11640.0
384531	Barnaba Auto Sport	11411.666666666666
303378.0	Aston Martin of Beverly Hills	11249.0
292489	Ferrari of Tampa Bay	11018.615384615385
268569	Lamborghini Houston	10397.962962962964

17. The result is the average savings of the top dealerships arranged in a series that begins with the greatest to least set.

18. The following query will show us the average mileage of cars for sale by year

```
SELECT year, ROUND(AVG(mileage),0) AS average_mileage from usedcartestfinal GROUP BY year ORDER BY average_mileage;
```

19. After a few minutes we see the result here.

2011	116770.0
1995	119004.0
2010	122246.0
2009	126963.0
1996	127718.0

20. This query will get the average price by make.

21. After a few minutes we see the result.

make_name	average_price
Pagani	2195000.0
Koenigsegg	2000000.0
Bugatti	1099998.0
Saleen	799990.0
Spyker	305500.0
McLaren	248988.0
Ferrari	222900.0
Rolls-Royce	222016.0
Lamborghini	220074.0
DeTomaso	177348.0
Aston Martin	172599.0
Jensen	159000.0
Bentley	119251.0
Karma	109569.0
Shelby	103134.0
SRT	100582.0
Humber	99895.0
Kaiser	93965.0
Lotus	91416.0
Porsche	80396.0
Sunbeam	72518.0
AM General	71702.0
Allard	65977.0
Maserati	62811.0
Maybach	57400.0
Land Rover	54648.0
Austin-Healey	52599.0
Ariel	52500.0
Tesla	50883.0
Packard	50814.0
Clenet	49900.0
DeLorean	49898.0
Mercedes-Benz	47426.0
Jaguar	46445.0
RAM	43439.0
Cadillac	42198.0
Genesis	42121.0
Audi	41995.0
BMW	41626.0
Lincoln	41370.0
Volvo	40859.0
Hudson	40465.0
Alfa Romeo	39995.0
Freightliner	39452.0
Fisker	39387.0
Plymouth	39086.0

22. The final query will give us the average price by year

23. After a few minutes we see the result here

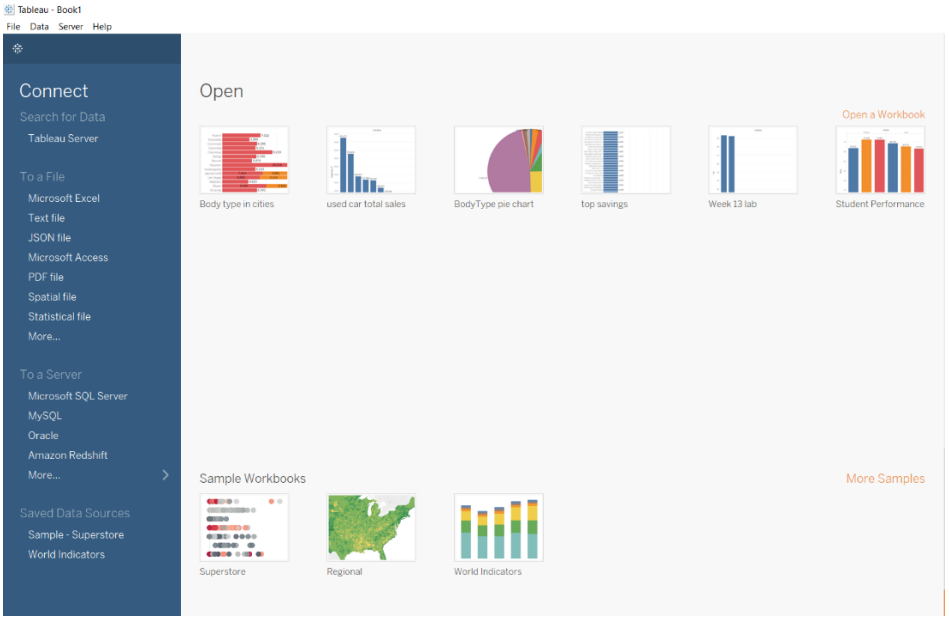
```
| 1942 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 83448.0 |
| 1941 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 37325.0 |
| 1940 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 42326.0 |
| 1939 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 50356.0 |
| 1938 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 31189.0 |
| 1937 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 56379.0 |
| 1936 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 34755.0 |
| 1935 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 686933.0 |
| 1934 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 37998.0 |
| 1933 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 30239.0 |
| 1932 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 53384.0 |
| 1931 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 32424.0 |
| 1930 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 28146.0 |
| 1929 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 24688.0 |
```

Step 5: Visualization

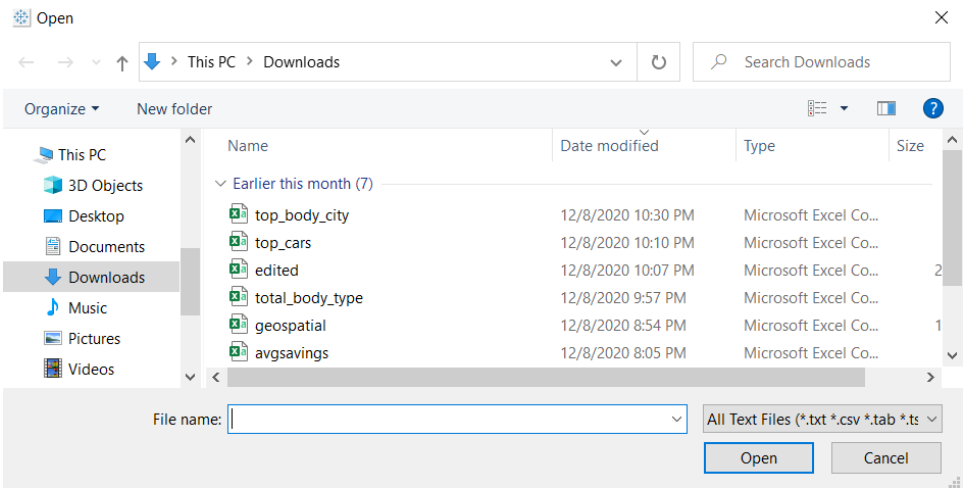
Using Tableau for Visualization of the CSV outputs

1. Open Tableau to open file download from HDFS:

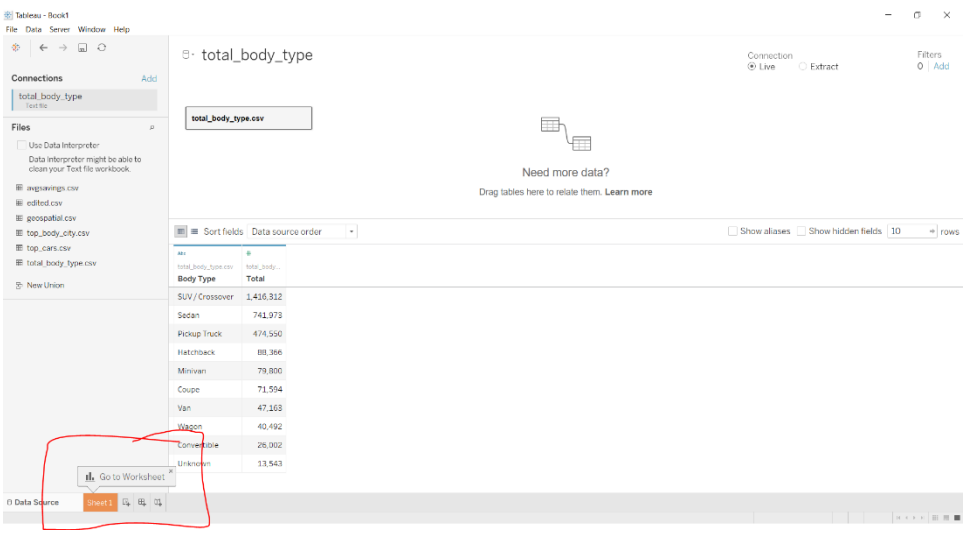
    1. Select **Text File** under **To a File** to work on a file.



2. Select file downloaded from HDFS.



3. Select **Sheet 1** on the bottom left of the screen to start working on graphs.



For Top used 10 Car Brands for Sale bar chart.

1. In hive write the following command to save the Top Ten Cars for Sale.

```
SELECT make_name, COUNT(model_name) AS total_for_sale
FROM usedcartestfinal
Group By make_name
Order By total_for_sale DESC
LIMIT 10;
```

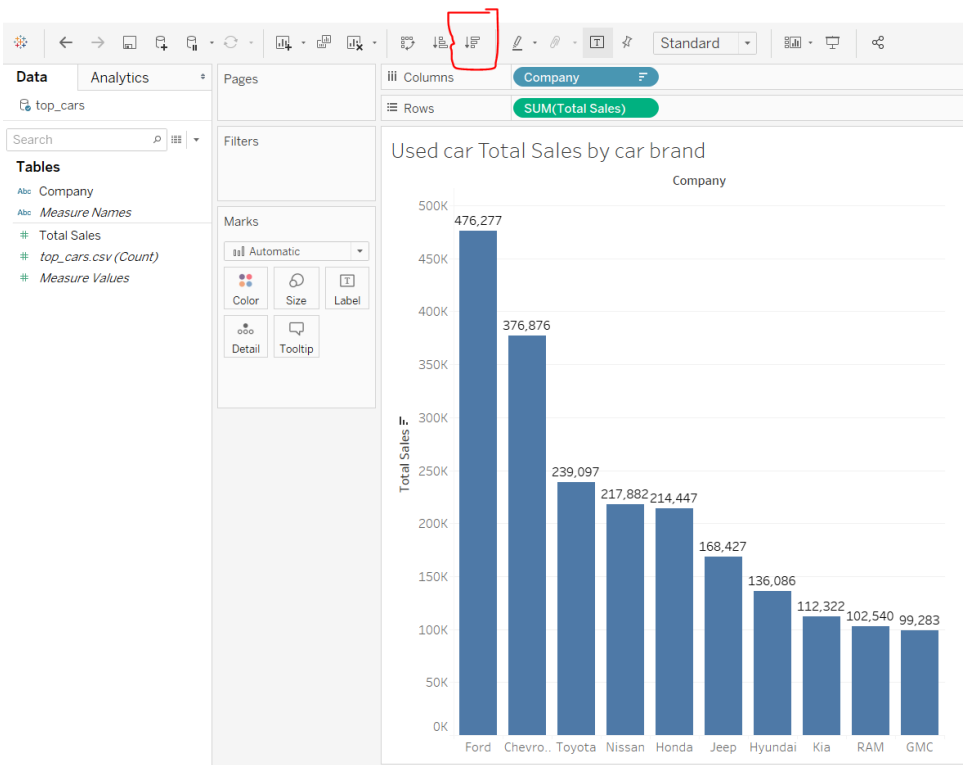
2. Download it by saving it as top_cars and enter the following:

```
hdfs dfs -get 000000_0 top_cars
```

3. Download file using pscp:

```
pscp kdelat15@129.150.64.74:/home/kdelat15/top_cars top_cars.csv
```

4. Open file on Tableau. Move **company** to columns and **total sales** to rows. Sort by descending order button shown with the red square.



For Total Body Types on Sale pie chart.

1. In hive write the following command to save Total_Body_Type:

```
 insert overwrite directory '/user/kdelat15/'
row format delimited fields terminated by ','
SELECT body_type, COUNT(make_name) AS total_body_type
FROM usedcartestfinal
Group By body_type
Order By total_body_type DESC
LIMIT 10;
```

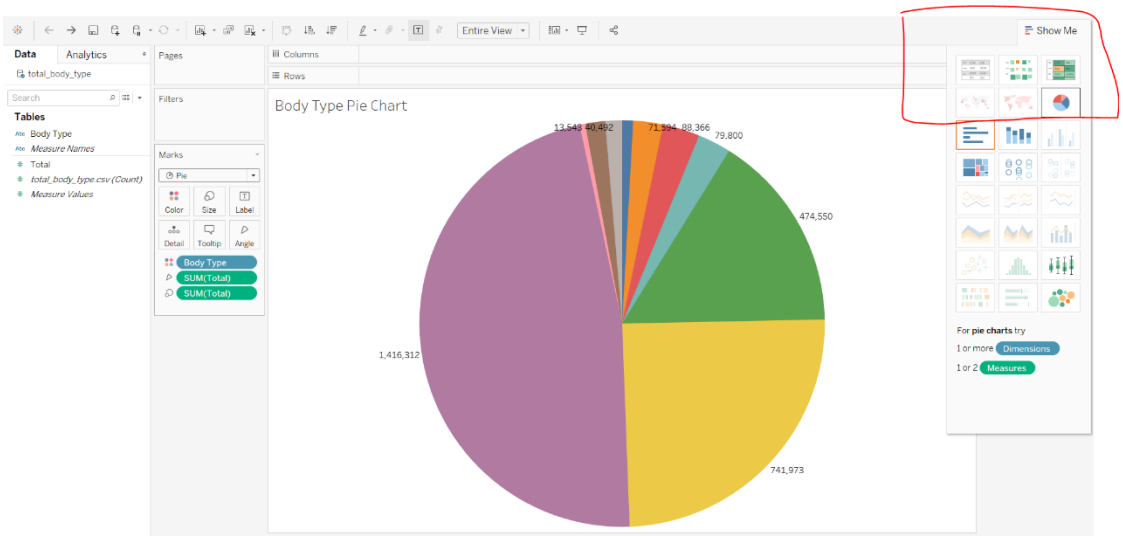
2. Download it by saving it as Total_Body_Type and enter the following:

```
hdfs dfs -get 000000_0 total_body_type
```

3. Download file using pscp:

```
pscp kdelat15@129.150.64.74:/home/kdelat15/total_body_type total_body_type.csv
```

4. Open file on Tableau. Move **Body Type** to columns and **Total** to rows. Select **Pie chart** on the top right corner of the page marked with red.



For Total Body Types on Sale pie chart.

1. In hive write the following command to save **Top_Body_City**:

```
 insert overwrite directory '/user/kdelat15/'
row format delimited fields terminated by ','
SELECT body_type, city, COUNT(body_type) AS total_for_body
FROM usedcartestfinal
Group By body_type, city
Order By total_for_body DESC
LIMIT 25;
```

2. Download it by saving it as **Total_Body_CITY** and enter the following:

```
hdfs dfs -get 000000_0 top_body_city
```

3. Download the file using pscp:

```
pscp kdelat15@129.150.64.74:/home/kdelat15/top_body_city top_body_city.csv
```

4. Open file on Tableau. Move **Total Sales** to columns and **City** to rows. Select **Stacked Bars** on the top right corner of the page marked with red.

For Average Savings bar chart.

1. In hive write the following command to save **Average_Savings**:

```
SELECT sp_id, sp_name, AVG(savings_amount) AS Average_Savings FROM
usedcartestfinal
GROUP BY sp_id, sp_name
ORDER BY Average_Savings DESC
LIMIT 1000;
```

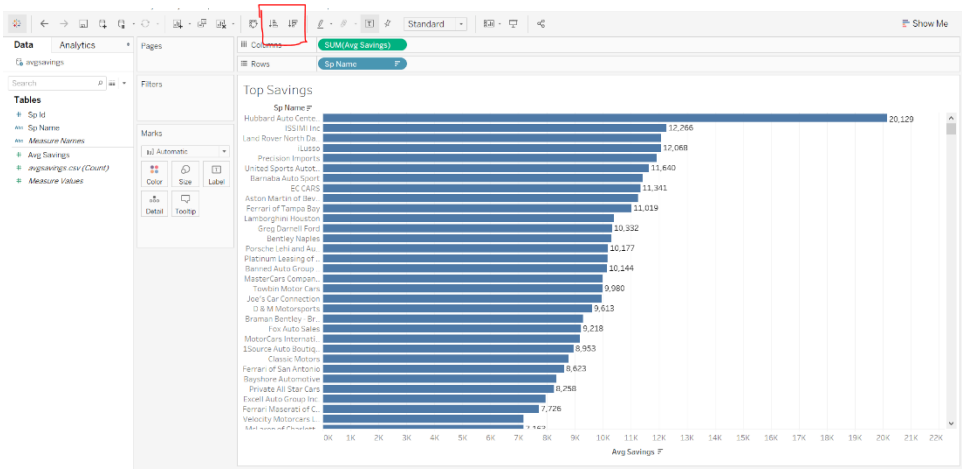
2. Download it by saving it as **average_savings** and enter the following:

```
hdfs dfs -get usedcars/000000_0 avg_savings.csv
```

3. Download the file using pscp:

```
pscp kdelat15@129.150.64.74:/home/kdelat15/avg_savings
avg_savings.csv
```

4. Open file on Tableau. Move **Average Savings** to columns and **Sp Name** to rows. Sort by descending/ascending order button shown with the red square to view highest and lowest.



Average Mileage by Year Visualization

1. In hive run the following command to save the average mileage of cars by year to HDFS.

```
insert overwrite directory '/user/kdelat15/'
row format delimited fields terminated by ','
SELECT year, ROUND(AVG(mileage),0) AS average_mileage
FROM usedcartestfinal
GROUP BY year
ORDER BY average_mileage;
```

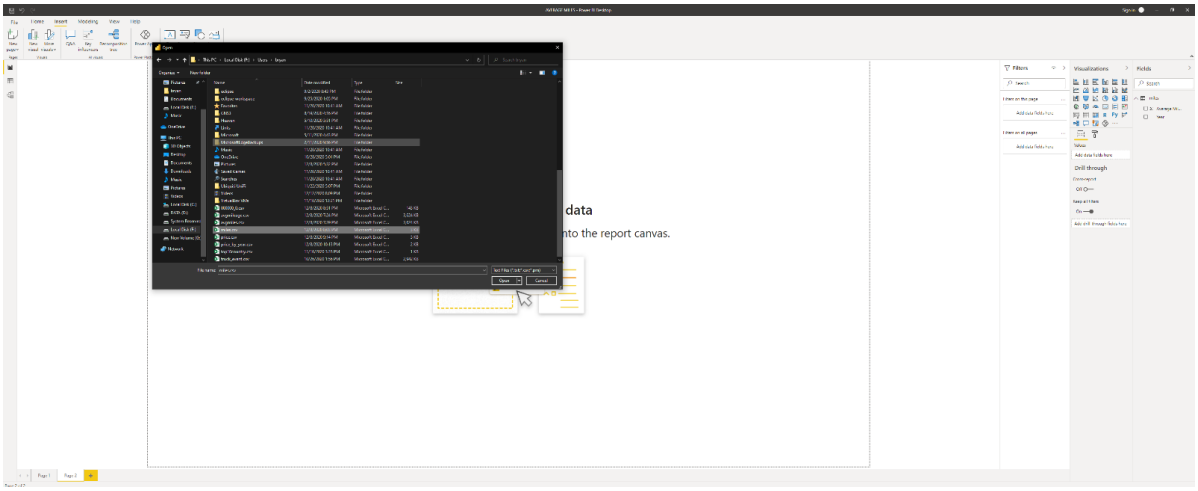
2. Exit out of hive and run the following command in HDFS to save file and rename file 000000_0 to miles

```
hdfs dfs -get 000000_0 miles
```

3. Download file using pscp/psftp

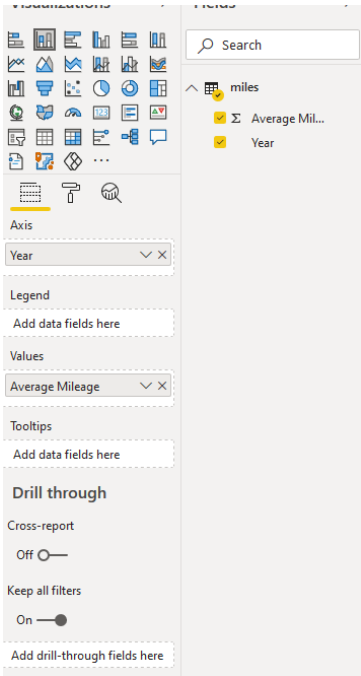
```
pscp kdelat15@129.150.64.74:/home/kdelat15/miles miles.csv
```

4. Locate miles.csv file and Upload CSV file to PowerBI

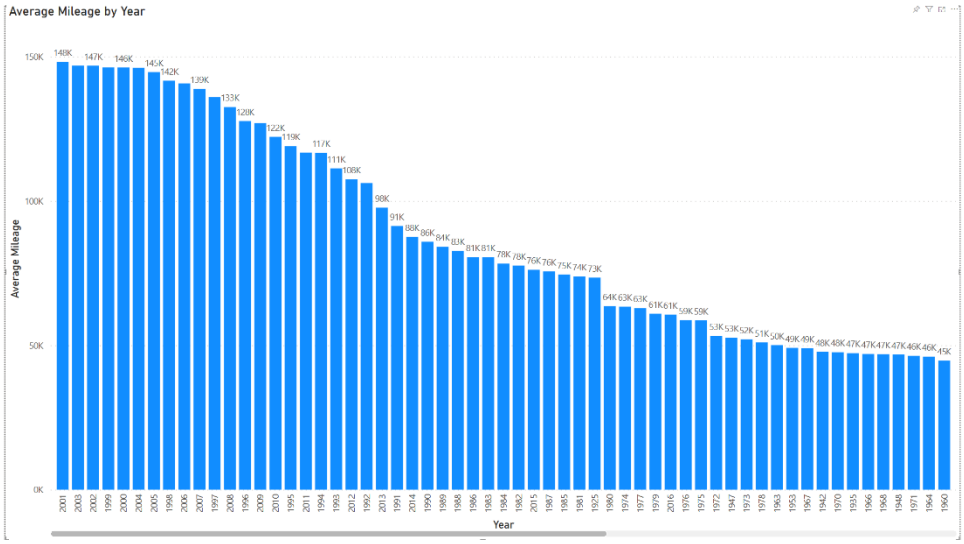


5. Make bar graph using PowerBI

   a. Select Stack Column Chart
   b. Select Year as Axis
   c. Select Average Mileage as Values



6. View graph and values



Average Price by Make Visualization

1. Using the follow command to get make name and avg price and save it to HDFS

```
insert overwrite directory '/user/kdelat15/'
row format delimited fields terminated by ','
SELECT make_name, ROUND(AVG(price),0) AS average_price
FROM usedcartestfinal
GROUP BY make_name
ORDER BY average_price
DESC LIMIT 1000;
```

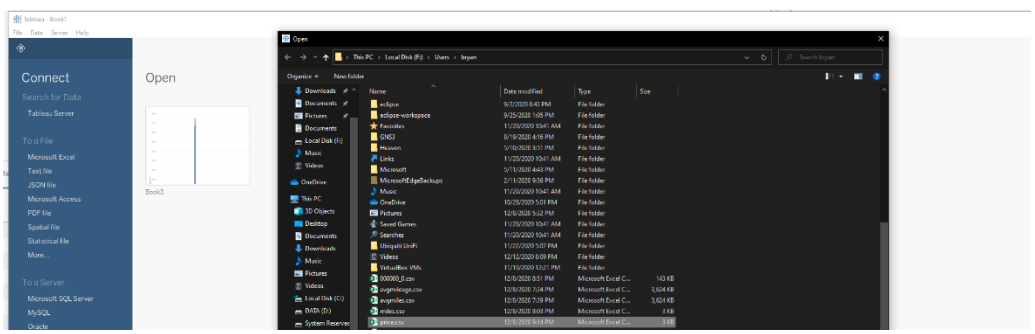
2. Go back to the shell and run the following command to get Output file and rename it to price

```
hdfs dfs -get 000000_0 price
```

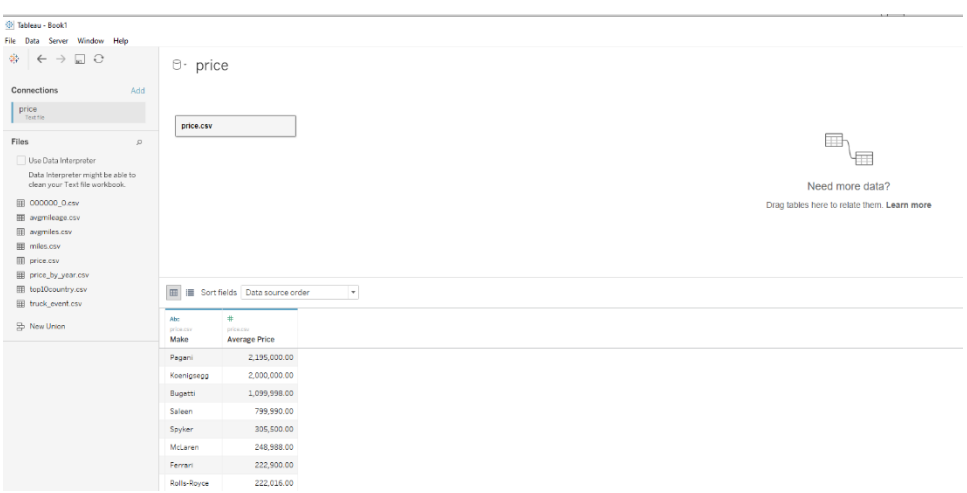
3. Use pscp to execute the following command to get file from HDFS and save to local computer

```
pscp kdelat15@129.150.64.74:/home/kdelat15/price price.csv
```

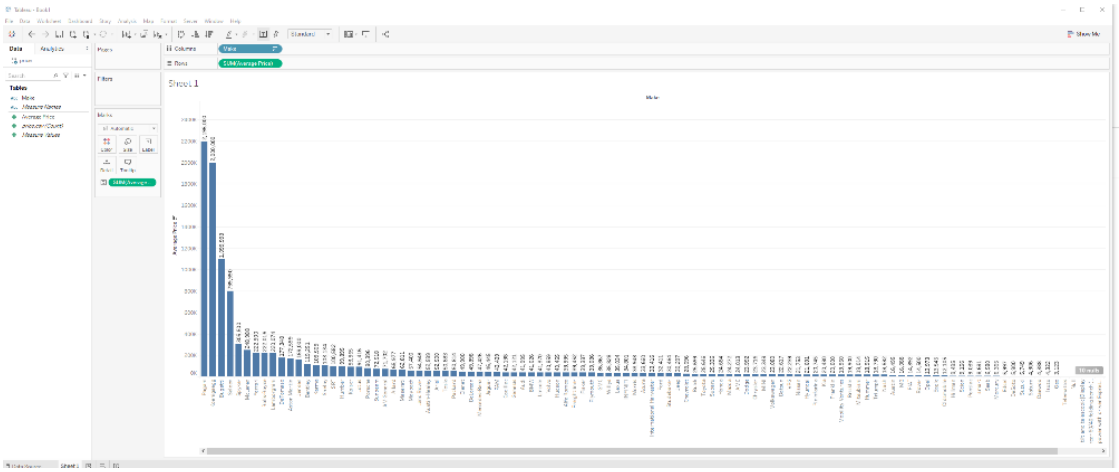
4. Open file in Tableau.



5. Rename columns to Make and Average Price



6. Go to Sheet 1 and place Make in the columns and Average Price as rows sort by Desc.



AVERAGE PRICE by YEAR Visualization

1. Run the following command to year and avg price and save to output file.

```
insert overwrite directory '/user/kdelat15/'
row format delimited fields terminated by ','
SELECT year, ROUND(AVG(price),0) AS average_price_year
FROM usedcartestfinal
GROUP BY year
ORDER BY year ASC;
```

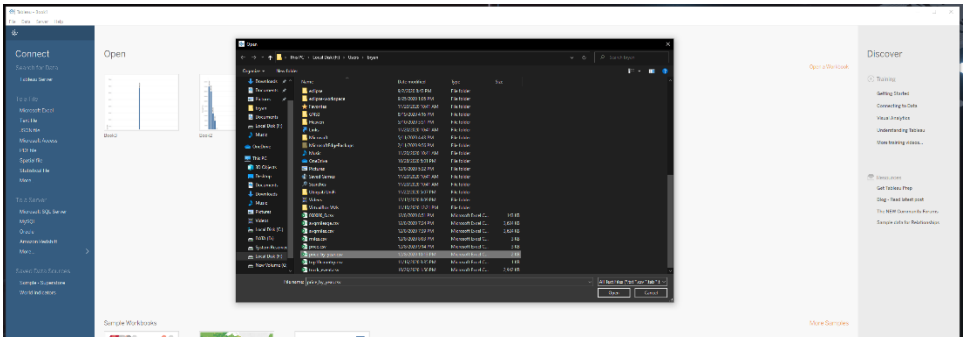
2. From the shell run following command to get output file and rename it to price_by_year

```
hdfs dfs -get 000000_0 price_by_year
```

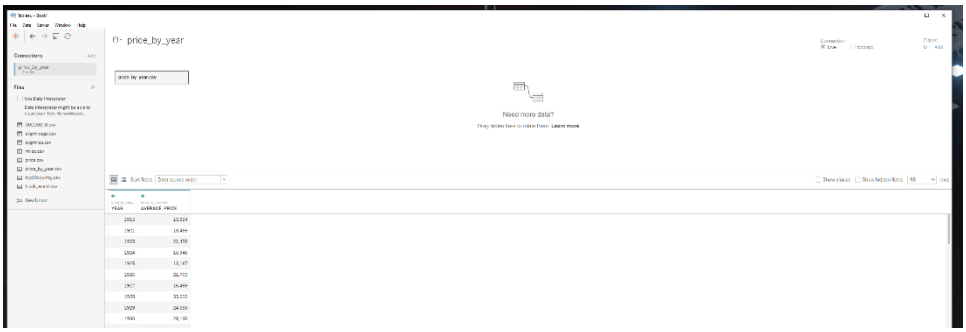
3. Use pscp to save price_by_year to local machine

```
pscp kdelat15@129.150.64.74:/home/kdelat15/price_by_year price_by_year.csv
```

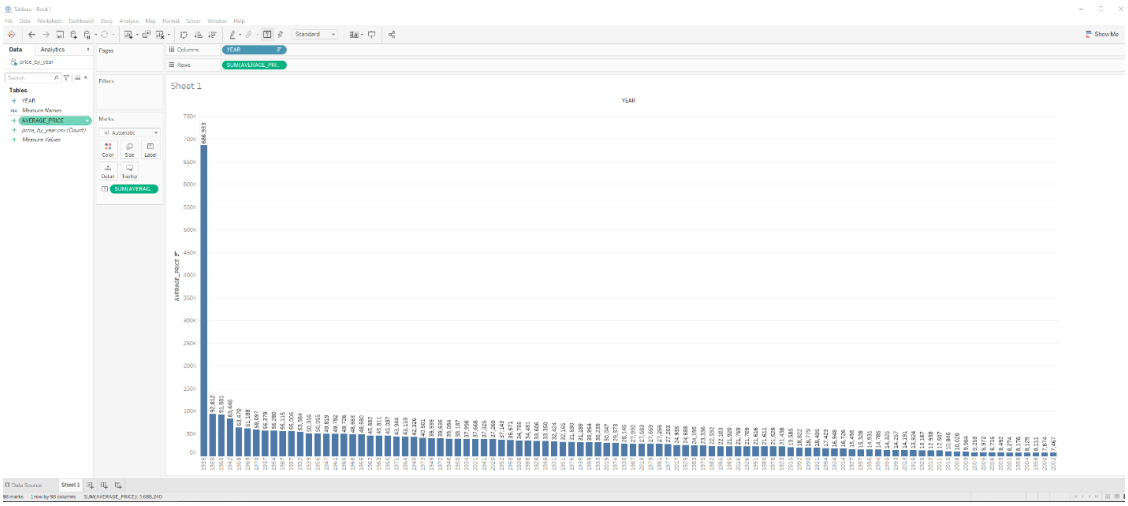
4. Locate file and Open in Tableau



5. Rename column to Year and Average Price



6. Put Year as a Column and Average Price as Row and sort by descending.



Step 6: Geo Spatial Mapping

Created a CSV file for Geo mapping Days used cars on the Market

1. Run the following code to output a CSV file which gets all data from Usedcartestfinal but only the first 1000

```
insert overwrite directory '/user/kdelat15/usedcars/'
row format delimited fields terminated by ','
SELECT *
FROM usedcartestfinal
LIMIT 1000;
```

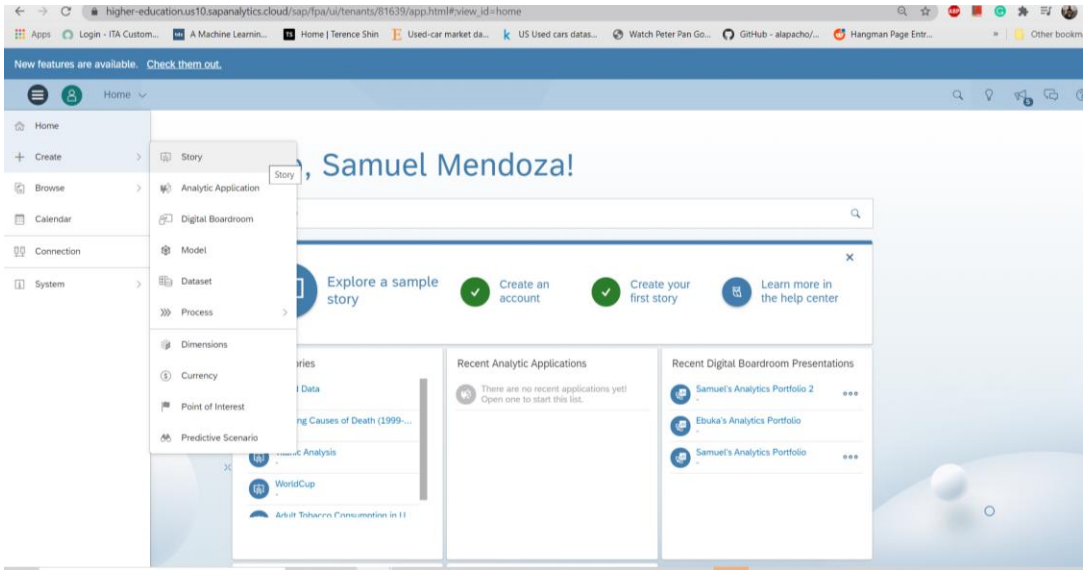
2. In Hadoop run the following command to get the output file and rename it geospatial

```
hdfs dfs -get usedcars/000000_0 geospatial
```

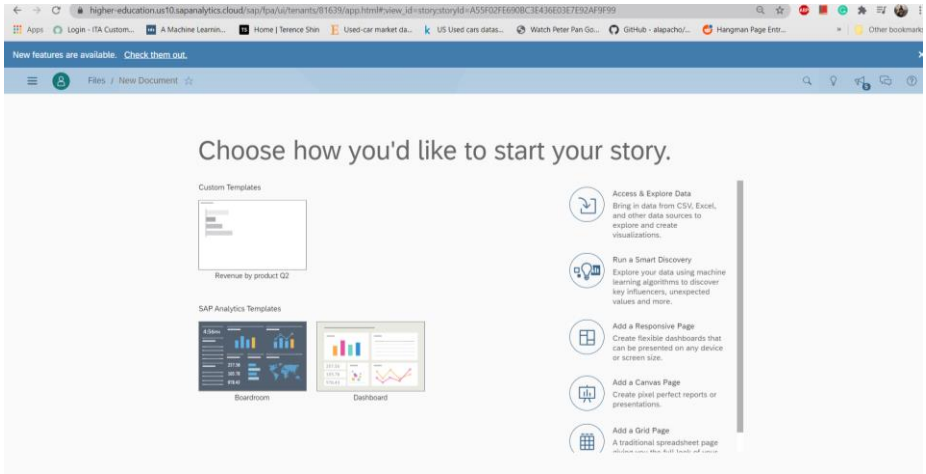
3. Using pscp connect to Hadoop and get the geospatial file and save it as a csv to your local machine

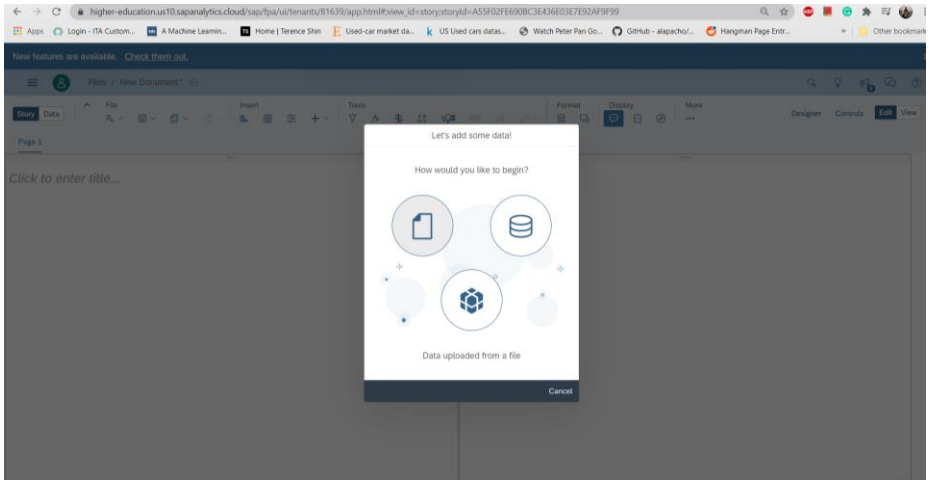
```
pscp kdelat15@129.150.64.74:/home/kdelat15/geospatial geospatial.csv
```

4. Opened SAP Analytics Cloud (Alternatively this can be done in PowerBi to get the same map)
   - Click Create -> Click on Story

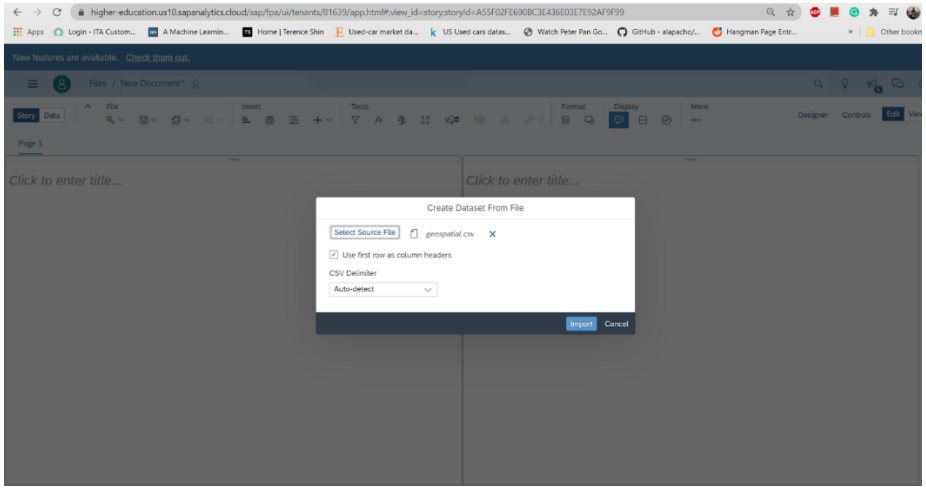


5. Next, click on Access and Explore Data. After you will be given three choices where you choose data upload from a file

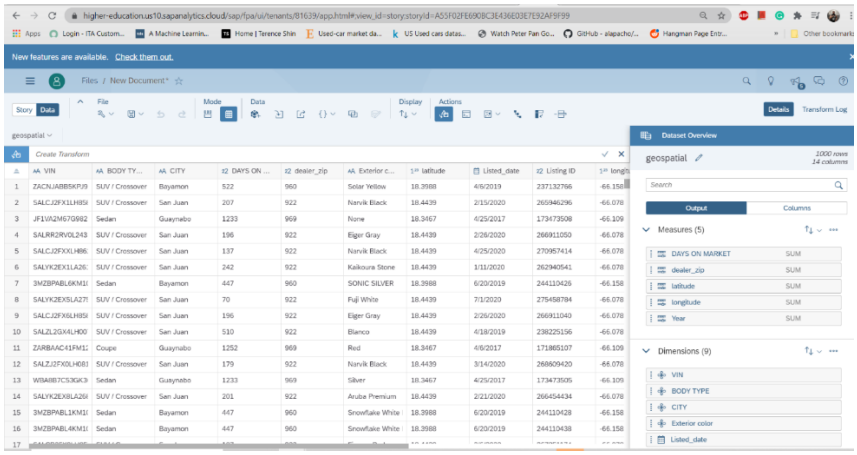
- 

- 

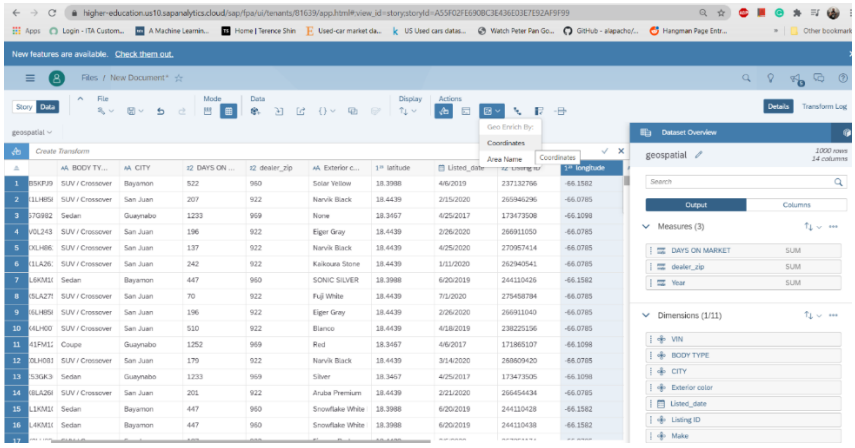
6. Select Source file and import the geospatial csv file that was created in hive.

- 

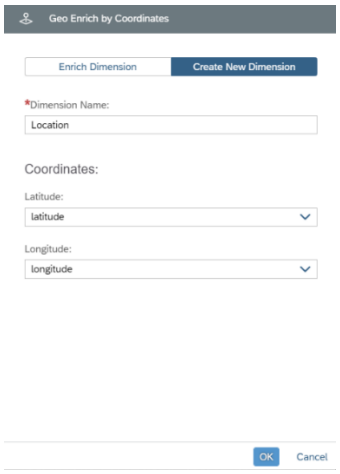
7. After the file has uploaded, In the Data section, click and drag longitude and latitude from measures to dimensions



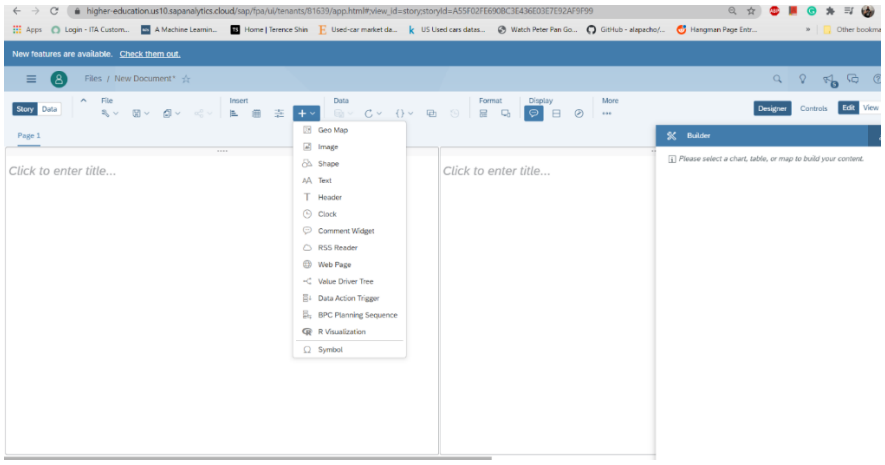
8. Then Select Geo Enrichment and select Coordinates



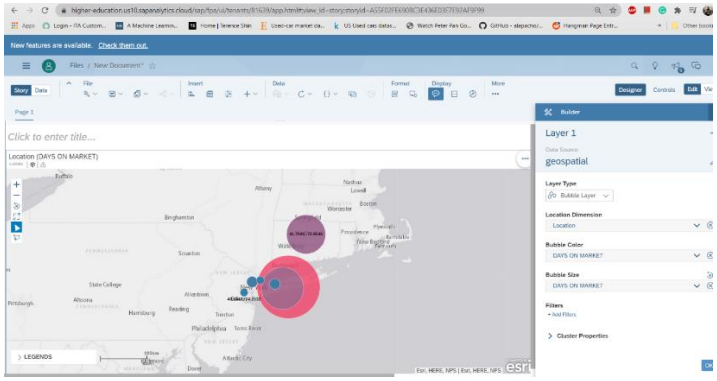
9. Here you want to create a new dimension by putting longitude and latitude together into A New Location dimension



10. Here you want to switch from Data View to Story View by clicking on story on the left upper side. From there you want to click on the plus on the tool bar and select Geo Map



11. Finally, create a layer where you will enter the Newly created Location in Dimensions, then Bubble color -> Days on the Market and Bubble size -> Days on the Market. This will create the Geo Mapping from the created CSV file created in Hive.



References

1. https://www.kaggle.com/ananaymital/us-used-cars-dataset
2. https://github.com/Smendo105/CIS4560
3. https://powerbi.microsoft.com/en-us/
4. https://hadoop.apache.org/
5. https://saphanajourney.com/sap-analytics-cloud/