**CSDA1020**

**Big Data Analytics Tools**

**Project 1, Hadoop and Hive Project**

**Group 1: Ioana Moca, Emily Dang, Karly Freiburger, Kashif Mirza**

**Introduction**

In this project you will be working with the car.csv dataset. For more information about this dataset you can visit this site <https://www.kaggle.com/mirosval/personal-cars-classifieds>

**Problem Background**

You are the data analyst at a large investment firm that is **contemplating to invest in a used car business**. Your task is to **provide data driven advice** to the stakeholders, that will enable them to **make a sound investment decision**. Failure to make the best decision may result in large financial consequences and irreversible damage to the company reputation and brand.

Your manager has instructed you to use the cars.csv dataset, because the veracity of this data has been established.

**Analysis (Questions)**

1. What is the relationship between car makes, models and price?

Load cars.scv to home directory

wget http://bit.ly/kagglecars -O cars.csv

Create new directory for project1

hdfs dfs -mkdir /user/project1

Copy the file to new directory

hdfs dfs -put cars.csv /user/project1

hdfs dfs -ls /user/project1

hive

create database pj1;

use pj1;

Create a table in pj1

CREATE TABLE IF NOT EXISTS cars0   
  (   
     maker         *STRING*,   
     model         *STRING*,   
     mileage       *INT*,   
     manfacyear    *INT*,   
     enginedisp    *INT*,   
     enginepower   *INT*,   
     bodytype      *STRING*,   
     colorslug     *STRING*,   
     stkyear       *STRING*,   
     transmission *STRING*,   
     doorcount     *INT*,   
     seatcount     *INT*,   
     fueltype      *STRING*,   
     datecreated   *TIMESTAMP*,   
     datalastseen  *TIMESTAMP*,   
     price         *FLOAT*   
  ) row format delimited fields TERMINATED BY',' LINES TERMINATED BY'\n' stored   
AS textfile;

Load data from cars.csv to table cars0

load data inpath '/user/project1/cars.csv' INTO TABLE pj1.cars0;

Clean the data --- We want to analyze the data based on the reasonable input.

Create a new table for cleansed data.

CREATE TABLE IF NOT EXISTS cars1   
  (   
     maker         *STRING*,   
     model         *STRING*,   
     mileage       *INT*,   
     manfacyear    *INT*,   
     engdisp       *INT*,   
     engpow        *INT*,   
     bodytype      *STRING*,   
     transimission *STRING*,   
     fueltype      *STRING*,   
     price         *FLOAT*   
  ) row format delimited fields TERMINATED BY',' LINES TERMINATED BY'\n' stored   
AS textfile;

Insert cleansed data into new table cars1.

hive> FROM pj1.cars0 INSERT INTO pj1.cars1   
SELECT maker,   
       model,   
       mileage,   
       manfacyear,   
       enginedisp,   
       enginepower,   
       bodytype,   
       transmission,   
       fueltype,   
       price   
WHERE  length(maker)>0   
AND    length(model)>0   
AND    manfacyear BETWEEN 2000 AND    2016   
AND    enginedisp BETWEEN 800 AND    8000   
AND    enginepower BETWEEN 50 AND    1000;

Let’s take a look at the new table which includes the cleansed data.

hive> describe extended cars1;

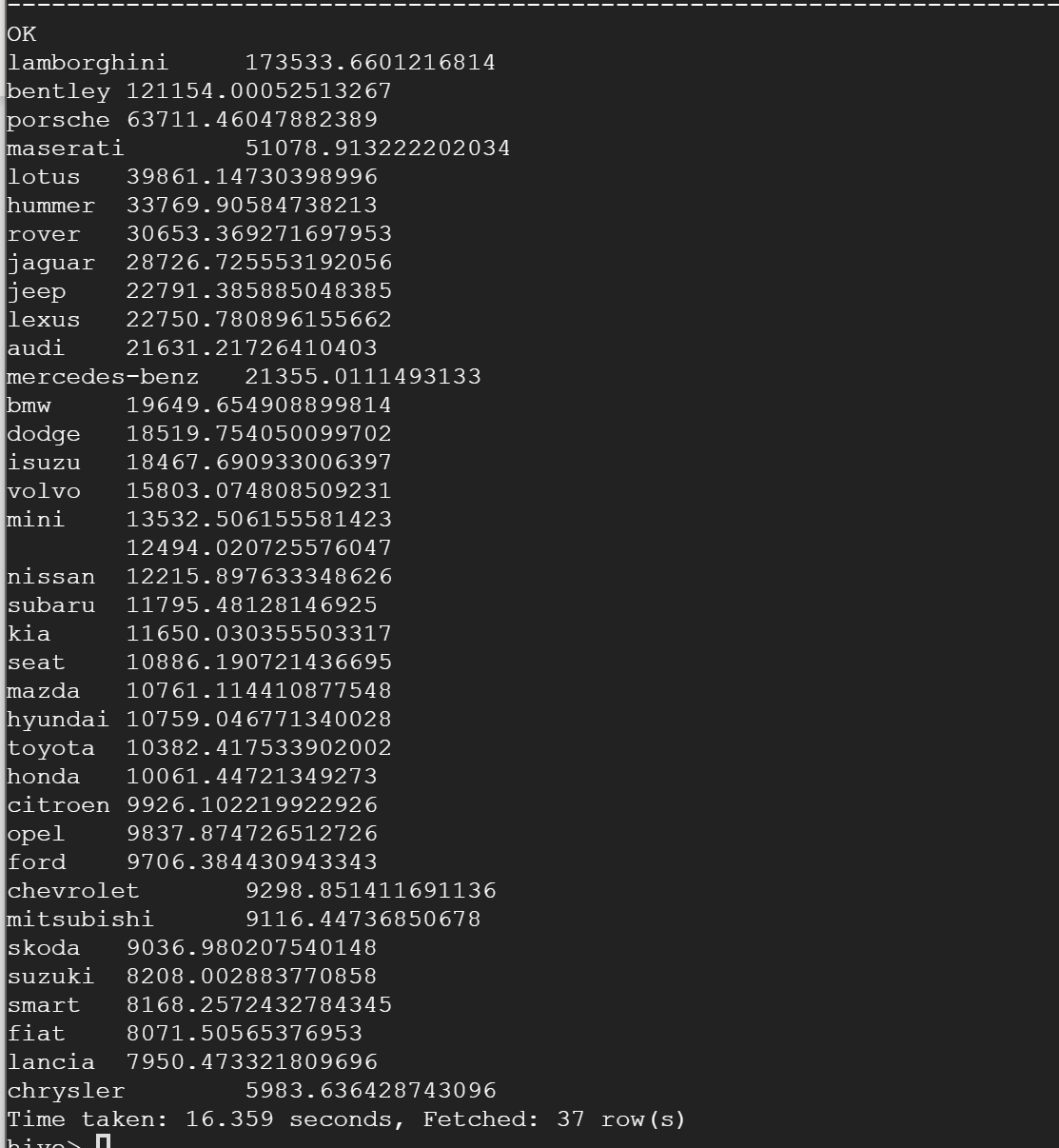


After the cleansing we have 296639 rows left. Now we can try to analyze the data in table cars1.

Get the average price by each manufacturer. We want to save the data to the home directory for further viewing.

hive>INSERT overwrite directory '/user/project1/tmp/out' > row format delimited fields terminated BY '\t' > stored AS textfile >  
SELECT   maker,   
         avg(price) AS avgprice   
FROM     cars1   
GROUP BY maker   
ORDER BY avgprice DESC;

Here is the result.



Create a new table price.

CREATE TABLE IF NOT EXISTS price   
  (   
     maker    *STRING*,   
     model    *STRING*,   
     avgprice *INT*   
  ) row format delimited fields terminated BY ',' LINES TERMINATED BY '\n'   
stored AS textfile;

Insert average price of each model into the new table.

FROM pj1.cars1INSERT INTO pj1.price   
SELECT   maker,   
         model,   
         *Avg*(price)   
GROUP BY maker,   
         model;

2. What are the top five vehicle manufacturers would you recommend? Why?

Let's recommend a car based on various use cases, assuming we want to minimize price and mileage. The variable chosen to split use cases is engine displacement, which determines both power and long-term fuel costs. Higher engine displacement will offer more power than lower ones, but will require more fuel. So there are two manufacturers we might recommend: a manufacturer offering higher power at the expense of greater fuel consumption, and manufacturer offering lower average power and better fuel economy. Both of these use cases will consider the lowest prices at reasonable mileages to minimize medium-term maintenance/replacement costs.

First let's create a new table to house our subsetted data.

CREATE TABLE IF NOT EXISTS cars2   
  (   
     maker        *STRING*,   
     mileage      *INT*,   
     manfacyear   *INT*,   
     enginedisp   *INT*,   
     enginepower  *INT*,   
     transmission *STRING*,   
     fueltype     *STRING*,   
     price        *FLOAT*   
  ) row format delimited fields TERMINATED BY ',' LINES TERMINATED BY '\n'   
stored AS textfile;

We populate our table and exclude null values for key variables.

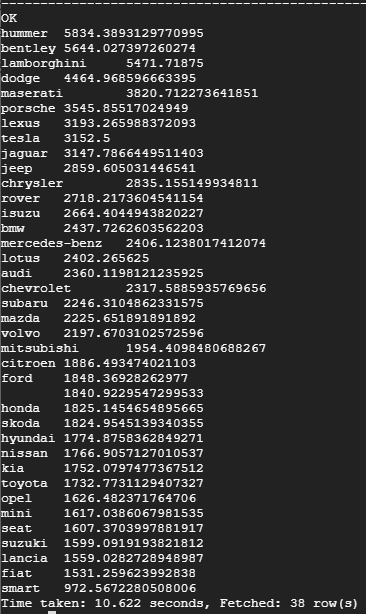
INSERT INTO cars2   
            (maker,   
             mileage,   
             manfacyear,   
             enginedisp,   
             enginepower,   
             transmission,   
             fueltype,   
             price)   
SELECT maker,   
       mileage,   
       manfacyear,   
       enginedisp,   
       engingepower,   
       transmission,   
       fueltype,   
       price   
FROM   cars0   
WHERE  enginedisp IS NOT NULL AND enginedisp <> ‘ ‘   
       AND price IS NOT NULL AND price <> ‘ ‘  
       AND mileage IS NOT NULL AND mileage <> ‘ ‘

      AND maker IS NOT NULL AND length(maker)>0;

Next we query the average engine displacement among the cars offered on the market, by maker.

SELECT maker,   
       *Avg*(enginedisp) AS avgenginedisp   
FROM   cars2   
GROUP  BY maker   
ORDER  BY avgenginedisp DESC;

Below is the output. We can see that hummers tend to offer the highest average engine displacement and smart cars offer the lowest average engine displacements.



We create a table to house the output shown above:

CREATE TABLE IF NOT EXISTS enginetable   
  (   
     maker         *STRING*,   
     avgenginedisp *INT*   
  ) row format delimited fields TERMINATED BY ',' LINES TERMINATED BY '\n';

FROM pj1.cars2 INSERT INTO pj1.enginetable   
SELECT   maker,   
         *Avg*(enginedisp)   
GROUP BY maker;

We repeat the steps above for average mileage by maker.

SELECT   maker,   
         *Avg*(mileage) AS avgmileage   
FROM     cars2   
GROUP BY maker   
ORDER BY avgmileage DESC;

CREATE TABLE IF NOT EXISTS miletable (maker string, avgmileage int) row format delimited fields terminated BY ',' lines terminated BY '\n';

FROM pj1.cars2 INSERT INTO pj1.miletable  
SELECT   maker,   
         *Avg*(mileage)   
GROUP BY maker;

\*\*Creating separate table for Average Price

SELECT   maker,   
         *Avg*(price) AS avgprice   
FROM     cars2   
GROUP BY maker   
ORDER BY avgprice DESC;

CREATE TABLE IF NOT EXISTS pricetable (maker string, avgprice int)row format delimited fields terminated BY ',' lines terminated BY '\n';

FROM pj1.cars2 INSERT INTO pj1.pricetable   
SELECT   maker,   
         *Avg*(price)   
GROUP BY maker;

We code average engine displacement by maker as follows: low is up to 1700, moderate is between 1700 and 2000, and high is >2000. Lower engine displacement offers low power and low fuel consumption, while the opposite is true for high engine displacement.

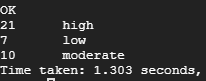
CREATE TABLE enginedisp1 AS   
  SELECT \*,   
         **IF**(avgenginedisp < 1700, 'low',

**IF**(avgenginedisp >= 1000

         AND avgenginedisp < 2000, 'moderate’, 'high')))   
         AS enginedispcategory   
  FROM   enginetable;

Below we query our new table to see how many makers offer low, moderate and high average engine displacement.

SELECT *Count*(maker),   
       enginedispcategory   
FROM   enginedisp1   
GROUP  BY enginedispcategory;



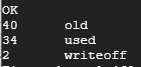
Next we bin average mileage of cars on the market by manufacturer. We do this for two reasons, first, price may be misleading if the group of cars at the lowest price points actually feature high mileages. High mileage will also introduce high long-term costs and low useful life. New is less than 1000 miles, used is between 1000 and 100,000 miles, old is between 100,000 and 200,000 miles and a writeoff is anything with more than 200,000 miles.

CREATE TABLE mileage1 AS   
  SELECT \*,   
         **IF**(avgmileage < 1000, 'new',

**IF**(avgmileage >= 1000   
         AND avgmileage < 100000, 'used',   
         **IF**(avgmileage >= 100000   
         AND avgmileage < 200000, 'old', 'writeoff'))) AS   
         mileagecategory   
  FROM   miletable;

Below we query our new table to see how many manufacturers tend to offer new, used, old cars and writeoffs. We can see that there are no new cars available, and most of the cars are old or used.

SELECT *Count*(maker),   
       mileagecategory   
FROM   mileage1   
GROUP  BY mileagecategory;

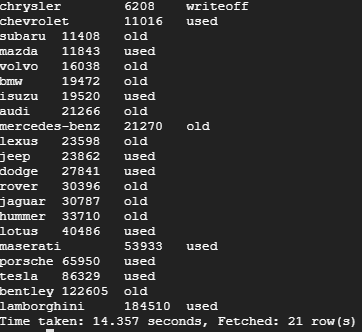


Next we query for manufacturers that offer high engine displacement, at the lowest average price. These will offer power in exchange for high fuel consumption, at a low price with long useful life.

SELECT DISTINCT enginedisp1.maker,   
                pricetable.avgprice,   
                mileage1.mileagecategory   
FROM   enginedisp1   
       JOIN pricetable   
         ON ( enginedisp1.maker = pricetable.maker   
              AND enginedisp1.enginedispcategory = 'high' )   
       JOIN mileage1   
         ON ( mileage1.maker = pricetable.maker )   
ORDER  BY pricetable.avgprice ASC;

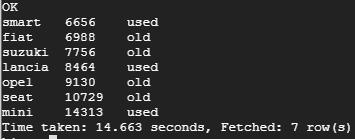
We can see that among the high engine displacement group, the least expensive maker on average is Chrysler, but we recommend ignoring this as the average mileage is so high for this group of cars that they are considered write-offs.

Chevrolets tend to be the second most affordable at an average price of approximately $11.0K for used vehicles, followed by Subarus at $11.4K. We therefore recommend chevrolets or subarus if high engine power is needed.



Next we consider the other use case, lower engine displacement. This group will offer highest fuel economy.

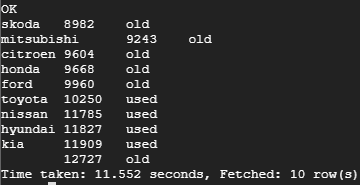
SELECT DISTINCT enginedisp1.maker,   
                pricetable.avgprice,   
                mileage1.mileagecategory   
FROM   enginedisp1   
       JOIN pricetable   
         ON ( enginedisp1.maker = pricetable.maker   
              AND enginedisp1.enginedispcategory = 'low' )   
       JOIN mileage1   
         ON ( mileage1.maker = pricetable.maker )   
ORDER  BY pricetable.avgprice ASC;



Among the low engine displacement category, the most affordable car is a smart car at approximately $6.7K, which tend to be in the used category of mileage. This is followed by fiats, at just under $7K, but these cars tend to be higher mileage than the smart cars.

Next we repeat the process for moderate engine displacement.

SELECT DISTINCT enginedisp1.maker,   
                pricetable.avgprice,   
                mileage1.mileagecategory   
FROM   enginedisp1   
       JOIN pricetable   
         ON ( enginedisp1.maker = pricetable.maker   
              AND enginedisp1.enginedispcategory = 'moderate' )   
       JOIN mileage1   
         ON ( mileage1.maker = pricetable.maker )   
ORDER  BY pricetable.avgprice ASC;



Skodas and Mitsubishis tend to offer moderate engine displacement at the lowest prices of around $9K. However, these cars are generally old. The most affordable used cars in this category are Toyotas.

3. Does fuel type have any impact on the car price? Explain

SELECT fueltype,   
       *Avg*(price) AS avgprice   
FROM   cars2   
GROUP  BY fueltype;

We can see average price is higher for cars that take diesel.