BUSINESS RECOMMENDATION REPORT

on Used Car Business Investment

Introduction

This report is requested by the Board of Directors of Georgian Investment Firm to support the decision-making process regarding the investment in a small used car business, which will sell cars to urban, middle-class citizens in Europe.

In this report, recommendations have been made about the vehicles should be invested in to maximize the profit and be able to reinvest quickly.

The data used for the analysis in this report is a 2nd party dataset collected by scrapping from car listing websites in Czech Republic and Germany by Miroslav Zoricak.

Data Source: https://www.kaggle.com/mirosval/personal-cars-classifieds

Approach

In order to maximize the profit and be able to reinvest quickly, the company should invest in these categories of cars:

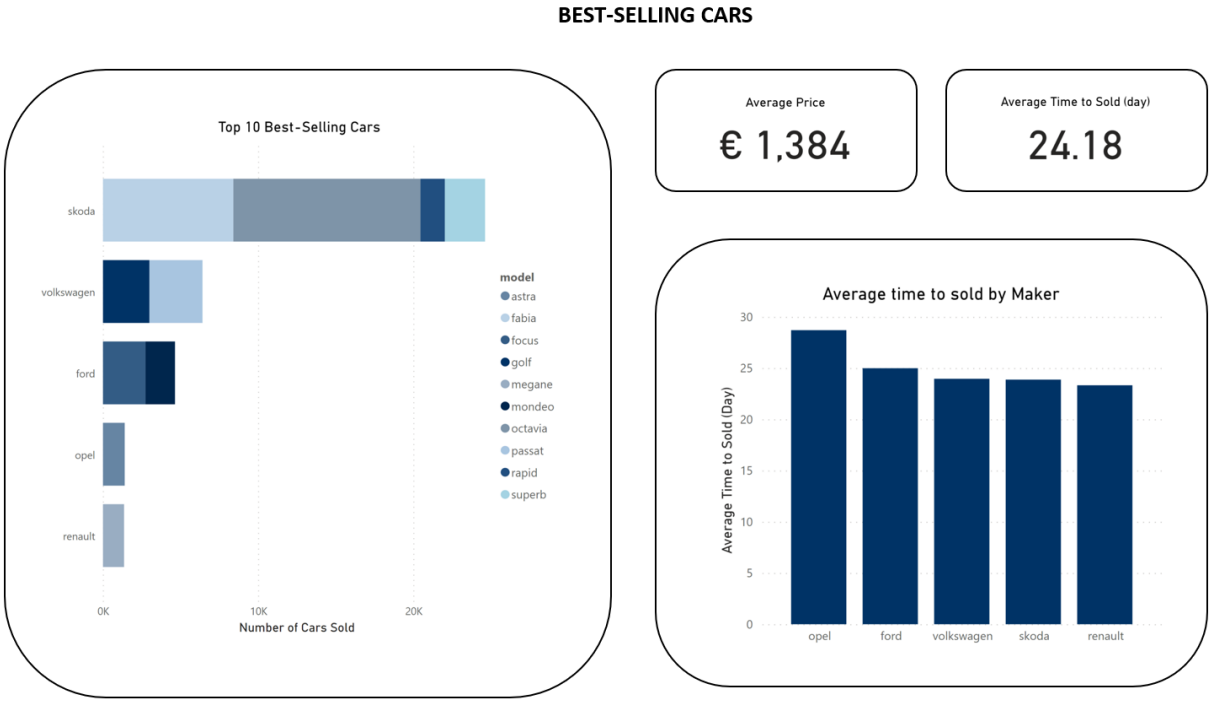
- Best-Selling Cars
- Fastest-Selling Cars
- Cars with highest Profit Margin

In this report, Best-Selling Cars and Fastest-Selling Cars is analyzed thoroughly.

Cars with highest Profit Margin cannot be analyzed due to the lack of Initial Purchasing Price of the vehicles. With the Initial Purchasing Price, the Profit Margin and Time Value of Money can be analyzed to make a more evidence-based investment decision. This is a very important aspect which should be taken into consideration when the data is collected comprehensively in the next phase of analysis.

Analysis & Insights

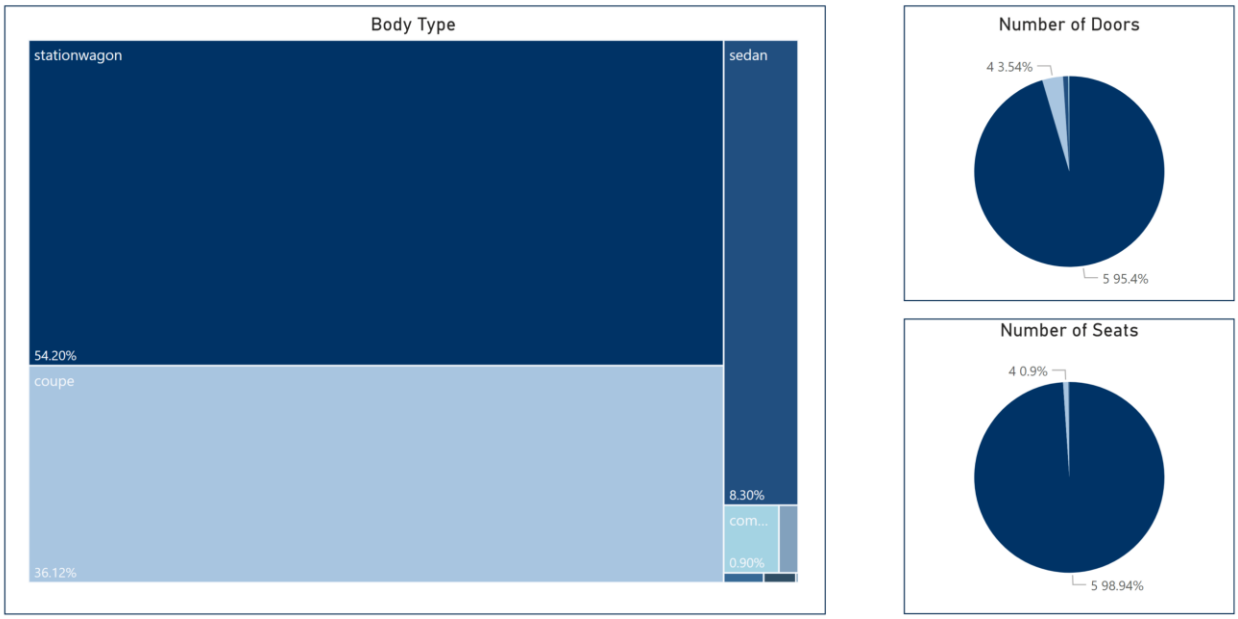
1. Best-Selling Cars



Insights

- Top 10 Best-Selling Cars consists of vehicles from:
    - Skoda
    - Volkswagen
    - Ford
    - Opel
    - Renault
- The average price of best-selling vehicles is € 1,384
- On average, it takes 24.18 days to sell a car in this category
- The average selling time is relatively similar for every maker in Top 10 Best-Selling Cars

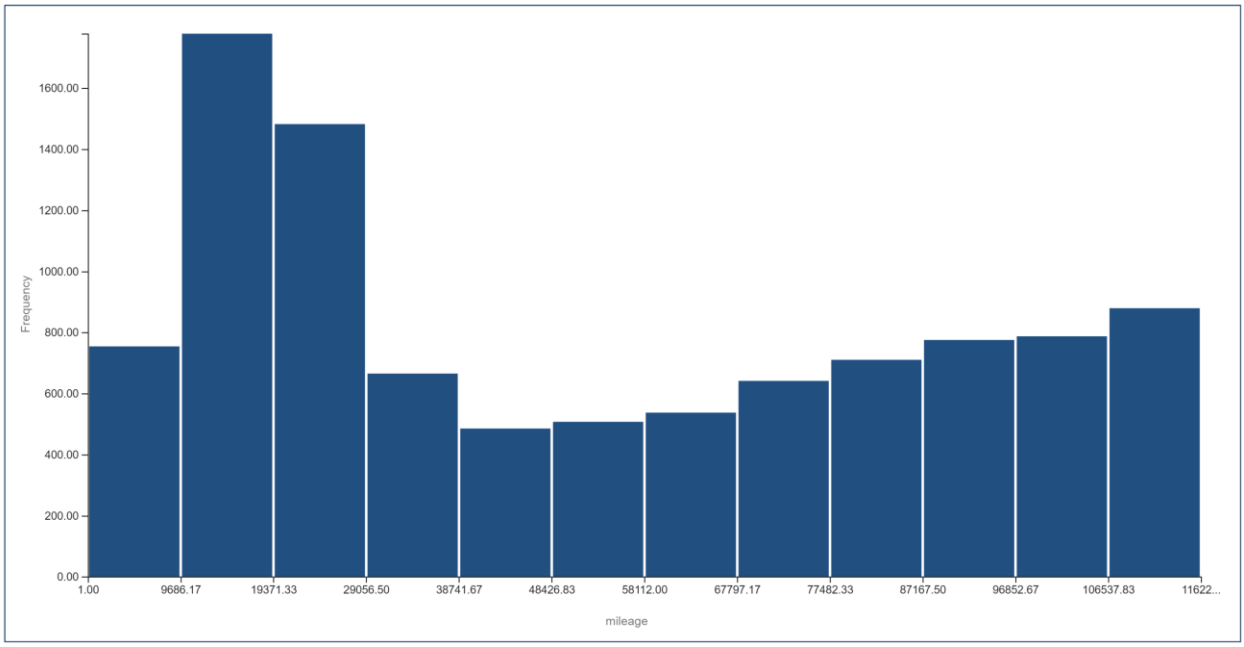
BODY TYPES



Insights

- The customers prefer Station Wagon, which also is a 5-doors, 5-seats car (54.2% total number of cars sold)
- Coupe and Sedan are second and third most popular choices (36.12% and 8.3% respectively)

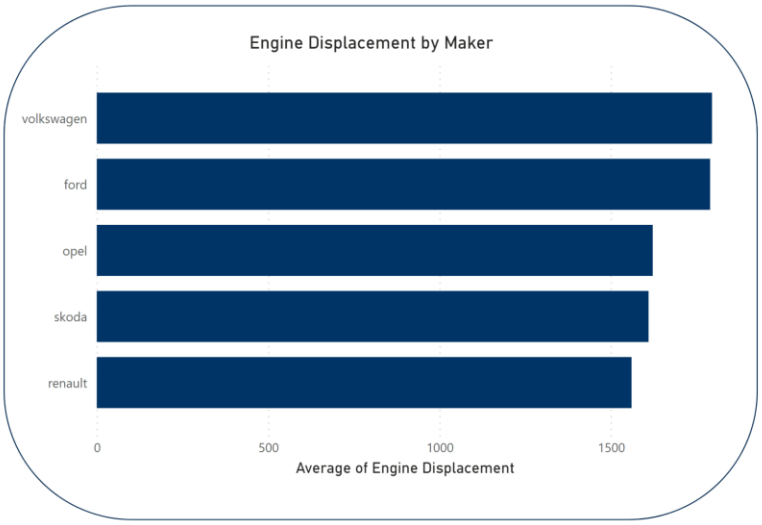
MILEAGE DISTRIBUTION

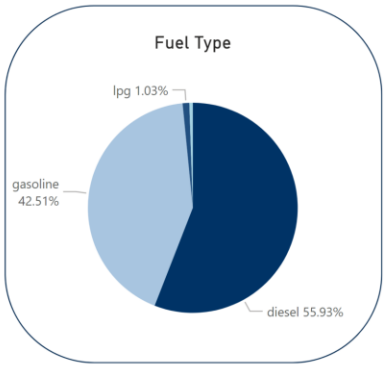


Insights

- Customers prefer cars with relatively low mileage (10,000 Km - 30,000 Km)

ENGINE & FUEL

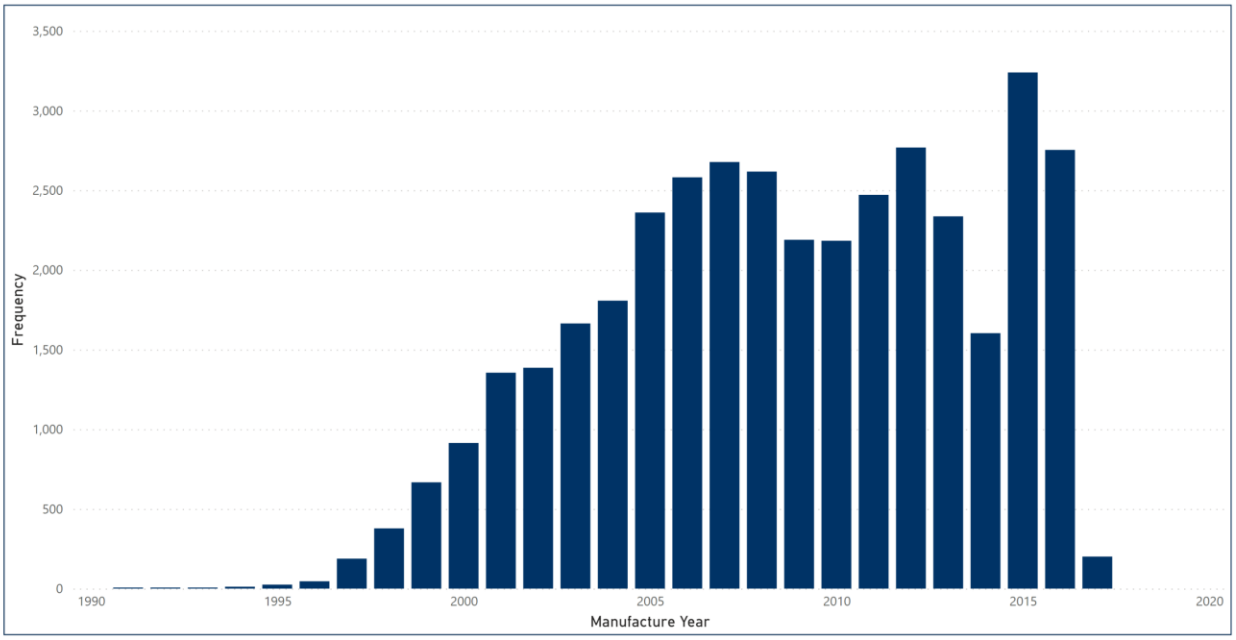




Insights

- Most of the best-selling cars have small Engine Displacement, around 1,500cc – 1,700cc, which is suitable for city roads
- The preferred fuel types are Diesel (55.93%) and Gasoline (42.51%)
- A small portion of best-selling vehicles (1.03%) runs by LPG (Liquefied Petroleum Gas)

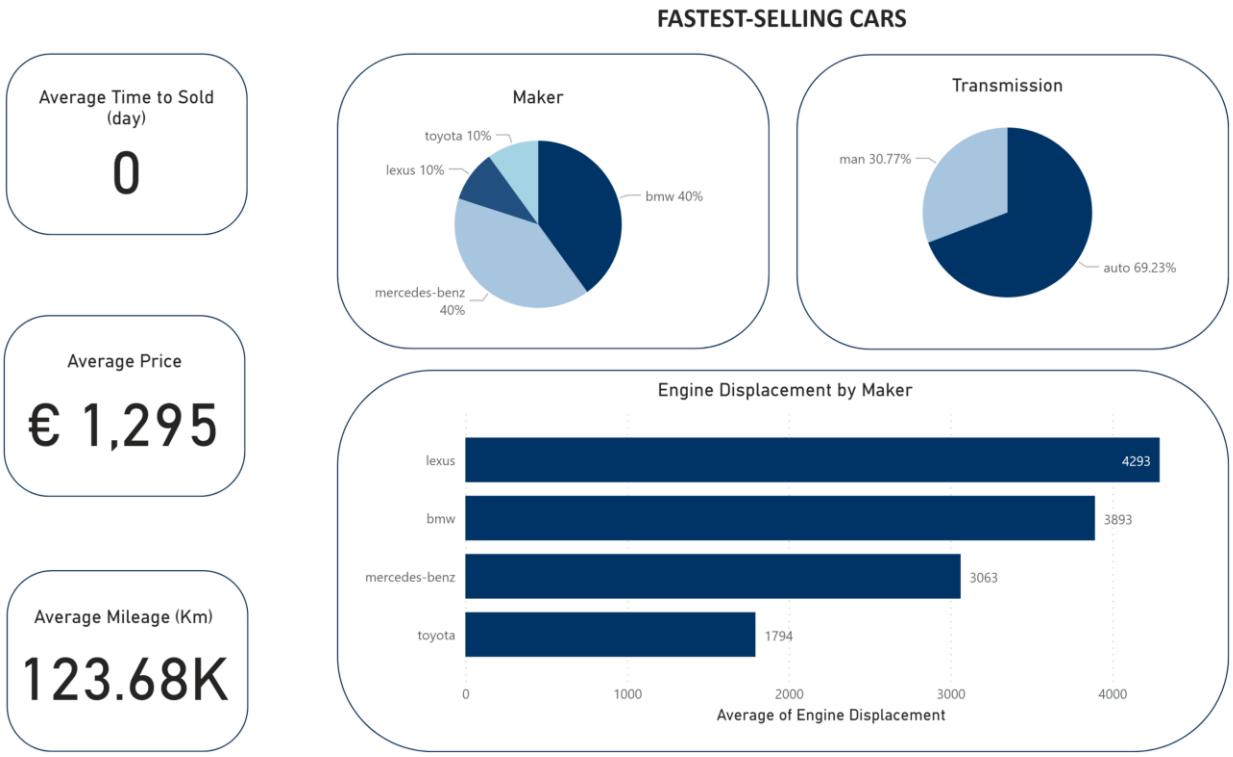
MANUFACTURE YEAR



Insights

- Most of the Best-selling cars are manufactured from 2000 to 2015

2. Fastest-Selling Cars



Insights

- Powerful cars with big engines from famous manufacturers like BMW, Mercedes-Benz, and Lexus despite having relatively high mileage still being sold very quickly, right after when the first advertisement goes online.
- Lexus is only accountable for 10% of Fastest-Selling Cars, while BMW and Mercedes-Benz are accountable for 80% in total sales. This is an indication that European citizens still prefer vehicles made by European car companies.

Recommendation:

According to the plan of investing in a small used car business, which will sell cars to urban, middle-class citizens in Europe, the following recommendations have been made:

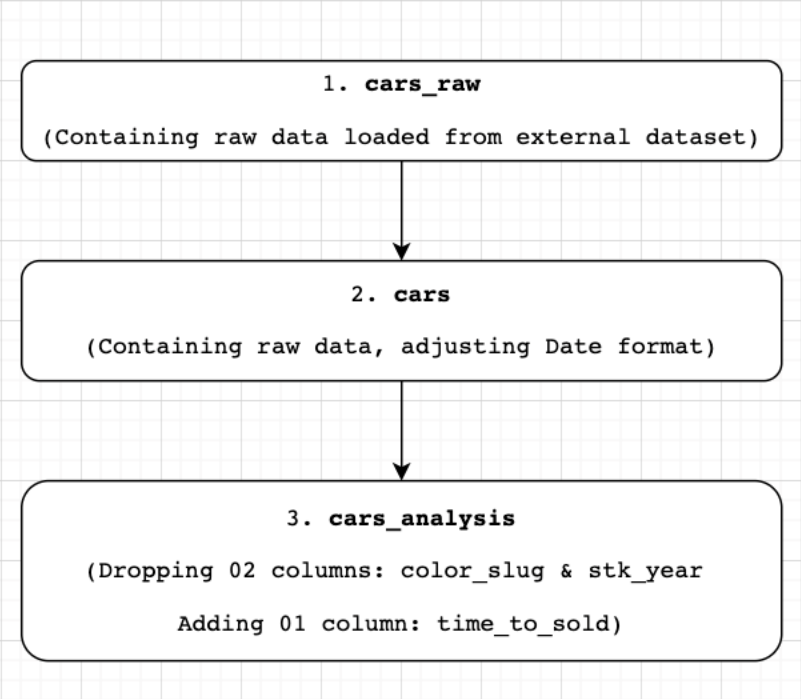
- The company should invest in Station Wagon, Coupe, and Sedan vehicles from Skoda, Volkswagen, Ford, Opel, and Renault with small engine size from 1,500cc to 1,700cc and low mileage. The selling price should be around € 1,300
- The company should also invest in luxury used cars such as BMW and Mercedes-Benz to diversify the products. These cars can be sold in a short period of time which will result in profit for the company to reinvest faster.

APPENDIX

Research Questions

1. Which are best-selling cars?
2. Which are fastest-selling cars?
3. What are the attributes of best, fastest-selling cars?

Get Data



1. Loading raw data to “cars_raw” table in Hive

```
hive (cars_db)> CREATE EXTERNAL TABLE IF NOT EXISTS cars_raw (
              > maker STRING,
              > model STRING,
              > mileage INT,
              > manufacture_year INT,
              > engine_displacement INT,
              > engine_power INT,
              > body_type STRING,
              > color_slug STRING,
              > stk_year STRING,
              > transmission STRING,
              > door_count INT,
              > seat_count INT,
              > fuel_type STRING,
              > date_created string,
              > date_last_seen string,
              > price_eur FLOAT)
              > ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
              > LOCATION '/BigData/hive'
              > TBLPROPERTIES("skip.header.line.count"="1");
OK
Time taken: 0.192 seconds
```

2. Currently, Date columns are in String data type. New table “cars” is created to adjust the Date format

```
hive (cars_db)> CREATE TABLE IF NOT EXISTS cars AS
              > SELECT
              > maker,
              > model,
              > mileage,
              > manufacture_year,
              > engine_displacement,
              > engine_power,
              > body_type,
              > color_slug,
              > stk_year,
              > transmission,
              > door_count,
              > seat_count,
              > fuel_type,
              > CAST(to_date(from_unixtime(unix_timestamp(date_created,'yyyy-MM-dd'))) AS date) as ads_created_date,
              > CAST(to_date(from_unixtime(unix_timestamp(date_last_seen,'yyyy-MM-dd'))) AS date) as ads_last_seen_date,
              > price_eur FROM cars_raw;
```

3. Dropping and Adding columns
    - Dropping the column of “color_slug”: 94% of the data values in “color_slug” column is null, so this column has been dropped to achieve a cleaner dataset
    - Dropping the column of “stk_year”: The information in this column is unrelated to the Research Questions, assuming year of the last emission control is not one of the important requirements of the buyers.
    - Adding the column of “time_to_sold”: This column has been added to calculate the average time (day) to sold of a specific vehicle.

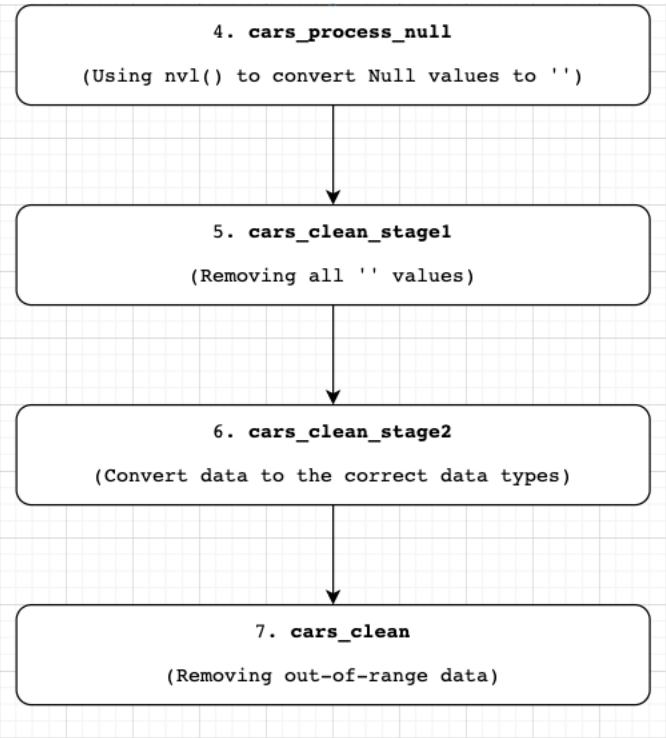
Query

```
hive (cars_db)> CREATE TABLE cars_analysis AS
              >  SELECT
              >  maker,
              >  model,
              >  mileage,
              >  manufacture_year,
              >  engine_displacement,
              >  engine_power,
              >  body_type,
              >  transmission,
              >  door_count,
              >  seat_count,
              >  fuel_type,
              >  ads_created_date,
              >  ads_last_seen_date,
              >  datediff(ads_last_seen_date, ads_created_date) as time_to_sold,
              >  price_eur
              >  FROM
              >  cars;
```

Output

```
hive (cars_db)> describe cars_analysis;
OK
maker                   string
model                   string
mileage                 int
manufacture_year        int
engine_displacement     int
engine_power            int
body_type               string
transmission            string
door_count              int
seat_count              int
fuel_type               string
ads_created_date        date
ads_last_seen_date      date
time_to_sold            int
price_eur               float
Time taken: 0.048 seconds, Fetched: 15 row(s)
```

Data Cleaning



4. Converting Null values to Empty String
   - This dataset contains two kinds of null value: Null value and '' value (Empty String).
   - To simplify the cleaning query, all Null values have been converted to Empty String

<Query>

```
hive (cars_db)> CREATE TABLE cars_process_null AS
              > SELECT
              > nvl(maker,'') AS maker,
              > nvl(model,'') AS model,
              > nvl(mileage,'') AS mileage,
              > nvl(manufacture_year,'') AS manufacture_year,
              > nvl(engine_displacement,'') AS engine_displacement,
              > nvl(engine_power,'') AS engine_power,
              > nvl(body_type,'') AS body_type,
              > nvl(transmission,'') AS transmission,
              > nvl(door_count,'') AS door_count,
              > nvl(seat_count,'') AS seat_count,
              > nvl(fuel_type,'') AS fuel_type,
              > nvl(ads_created_date,'') AS ads_created_date,
              > nvl(ads_last_seen_date,'') AS ads_last_seen_date,
              > nvl(time_to_sold,'') AS time_to_sold,
              > nvl(price_eur,'') AS price_eur
              > FROM cars_analysis;
```

5. Removing all Empty String value ‘’ to clean the dataset

<Query>

```
hive (cars_db)> create table cars_clean_stage1 as
              > select maker,
              > model,
              > mileage,
              >  manufacture_year,
              >  engine_displacement,
              >  engine_power,
              >  body_type,
              >  transmission,
              >  door_count,
              >  seat_count,
              >  fuel_type,
              >  ads_created_date,
              >  ads_last_seen_date,
              >  time_to_sold,
              >  price_eur
              >  FROM cars_process_null
              >  WHERE maker <> ''
              >  and model <> ''
              >  and mileage <> ''
              >  and manufacture_year <> ''
              >  and engine_displacement <> ''
              >  and engine_power <> ''
              >  and body_type <> ''
              >  and transmission <> ''
              >  and door_count <> ''
              >  and seat_count <> ''
              >  and fuel_type <> ''
              >  and ads_created_date <> ''
              >  and ads_last_seen_date <> ''
              >  and time_to_sold <> ''
              >  and price_eur <> '';
```

6. Converting all the data into correct data types in order to check out-of-range values

<Query>

```
hive (cars_db)> create table cars_clean_stage2 as
              > SELECT
              >  maker,
              >  model,
              >  cast(mileage as float),
              >  cast(manufacture_year as int),
              >  cast(engine_displacement as int),
              >  cast(engine_power as int),
              >  body_type,
              >  transmission,
              >  cast(door_count as int),
              >  cast(seat_count as int),
              >  fuel_type,
              >  CAST(to_date(from_unixtime(unix_timestamp(ads_created_date,'yyyy-MM-dd'))) AS date) as ads_created_date,
              >  CAST(to_date(from_unixtime(unix_timestamp(ads_last_seen_date,'yyyy-MM-dd'))) AS date) as ads_last_seen_date,
              >  cast(time_to_sold as int),
              >  cast(price_eur as float)
              > FROM
              > cars_clean_stage1;
```

<Output>

```
col_name         data_type       comment
maker                    string
model                    string
mileage                  float
manufacture_year         int
engine_displacement      int
engine_power             int
body_type                string
transmission             string
door_count               int
seat_count               int
fuel_type                string
ads_created_date         date
ads_last_seen_date       date
time_to_sold             int
price_eur                float
Time taken: 0.042 seconds, Fetched: 15 row(s)
```

7. Removing out-of-range values
    - Quantitative variables have been checked for out-of-range values
    - Remove records with mileage = 0 because the company is investing in used cars, which are cars have already been driven.
    - engine_displacement ranges from 10 - 32,767. The target customers of this used car business are urban citizens, which only need small engine displacement, so the value has been limited to the range from 1,000cc to 7,000cc

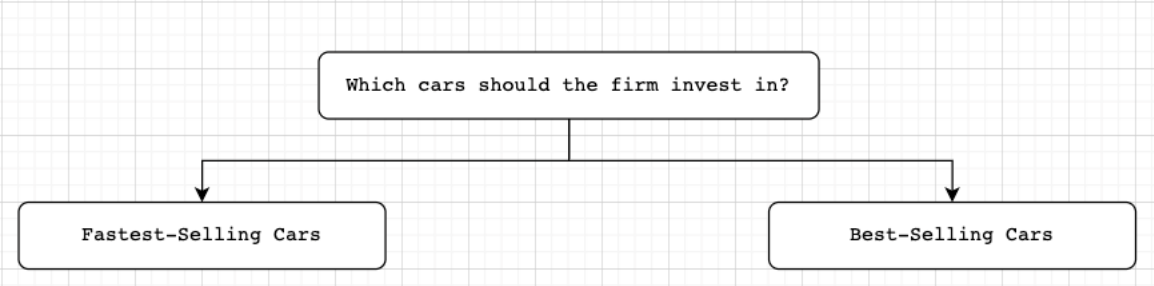
<Query>

```
hive (cars_db)> create table cars_clean as
              > select
              > maker,
              > model,
              > mileage,
              > manufacture_year,
              > engine_displacement,
              > engine_power,
              > body_type,
              > transmission,
              > door_count,
              > seat_count,
              > fuel_type,
              > ads_created_date,
              > ads_last_seen_date,
              > time_to_sold,
              > price_eur
              > FROM cars_clean_stage2
              > WHERE
              > mileage <> 0
              > AND engine_displacement between 1000 and 7000;
```

<Output>

```
hive (cars_db)> describe cars_clean;
OK
col_name        data_type       comment
maker                   string
model                   string
mileage                 float
manufacture_year        int
engine_displacement     int
engine_power            int
body_type               string
transmission            string
door_count              int
seat_count              int
fuel_type               string
ads_created_date        date
ads_last_seen_date      date
time_to_sold            int
price_eur               float
Time taken: 0.041 seconds, Fetched: 15 row(s)
```

Data Analysis



1. Top 10 Fastest-Selling Cars

<Query>

```
hive (cars_db)> CREATE TABLE fastest_selling_cars AS
              > SELECT
              >    maker,
              >    model,
              >    AVG(time_to_sold) as average_time_to_sold
              > FROM cars_clean
              > GROUP BY
              >    maker,
              >    model
              > ORDER BY
              >    average_time_to_sold
              > LIMIT 10;
```

<Output>

```
fastest_selling_cars.maker      fastest_selling_cars.model      fastest_selling_cars.average_time_to_sold
bmw     420i    0.0
toyota  mr2     0.0
mercedes-benz   ml230   0.0
bmw     850csi  0.0
bmw     760i    0.0
mercedes-benz   e500    0.0
mercedes-benz   gle350-cdi      0.0
mercedes-benz   r350-cdi        0.0
lexus   sc      0.0
bmw     x2      0.0
Time taken: 5.316 seconds, Fetched: 10 row(s)
```

2. Top 10 Best-Selling Cars

<Query>

```
hive (cars_db)> create table best_selling_cars as
              > SELECT
              >    maker,
              >    model,
              >    COUNT(*) as number_of_car_sold
              > FROM cars_clean
              > GROUP BY
              >    maker,
              >    model
              > ORDER BY
              >    number_of_car_sold DESC
              > LIMIT 10;
```

<Output>

```
best_selling_cars.maker best_selling_cars.model best_selling_cars.number_of_car_sold
skoda   octavia 12053
skoda   fabia   8401
volkswagen      passat  3419
volkswagen      golf    2995
ford    focus   2738
skoda   superb  2588
ford    mondeo  1903
skoda   rapid   1573
opel    astra   1402
renault megane  1361
Time taken: 4.542 seconds, Fetched: 10 row(s)
```

3. Join best_selling_cars & cars_clean to get attributes of cars into the new table of best_selling_cars_information

```
hive (cars_db)> CREATE TABLE best_selling_cars_information AS
              > SELECT
              > c.*
              > FROM
              > best_selling_cars AS b JOIN cars_clean AS c ON b.maker = c.maker AND b.model = c.model;
```

```
OK
best_selling_cars_information.maker     best_selling_cars_information.model      best_selling_cars_information.mileage   best_selling_cars_information.manufacture_year  best_selling_cars_information.engine_displacement       best_selling_cars_information.engine_power      best_selling_cars_information.body_type be
st_selling_cars_information.transmission        best_selling_cars_information.door_count        best_selling_cars_information.seat_count        best_selling_cars_information.fuel_type best_selling_cars_information.ads_created_date  best_selling_cars_information.ads_last_seen_date        best_selling_cars_informat
ion.time_to_sold        best_selling_cars_information.price_eur
volkswagen      golf    172000.0        2005    1598    85      coupe   man     3       5       gasoline        2016-11-09      2016-11-17      8       1295.34
renault megane  80880.0 2012    1598    96      stationwagon    man     5       5       diesel  2016-11-09      2017-01-09      61      1295.34
renault megane  170105.0        2005    1461    63      stationwagon    man     5       5       diesel  2016-11-09      2016-11-14      7       1295.34
skoda   rapid   36612.0 2013    1200    63      coupe   man     5       5       gasoline        2016-11-09      2016-12-07      28      1295.34
volkswagen      golf    23878.0 2015    1197    77      coupe   man     5       5       gasoline        2016-11-09      2016-11-17      8       1295.34
renault megane  89274.0 2014    1461    81      coupe   man     5       5       diesel  2016-11-09      2016-11-20      11      1295.34
volkswagen      golf    144676.0        2005    1400    55      coupe   man     5       5       gasoline        2016-11-09      2016-12-01      22      1295.34
```

4. Join fastest_selling_cars & cars_clean to get attributes of cars into the new table of fastest_selling_cars_information

```
hive (cars_db)> CREATE TABLE fastest_selling_cars_information AS
              > SELECT
              > c.*
              > FROM
              > fastest_selling_cars AS f JOIN cars_clean AS c ON f.maker = c.maker AND f.model = c.model;
```

```
OK
fastest_selling_cars_information.maker  fastest_selling_cars_information.model     fastest_selling_cars_information.mileage        fastest_selling_cars_information.manufacture_year       fastest_selling_cars_information.engine_displacement    fastest_selling_cars_information.engine_power   fastest_selling_cars_infor
mation.body_type        fastest_selling_cars_information.transmission   fastest_selling_cars_information.door_count     fastest_selling_cars_information.seat_count     fastest_selling_cars_information.fuel_type      fastest_selling_cars_information.ads_created_date       fastest_selling_cars_information.ads_last_
seen_date       fastest_selling_cars_information.time_to_sold   fastest_selling_cars_information.price_eur
mercedes-benz   s350-cdi        79000.0 2012    2987    195     stationwagon    auto    5       5       diesel  2017-03-08      2017-03-08      0       1295.34
bmw     m2      187635.0        2014    1999    135     offroad man     5       5       diesel  2017-03-10      2017-03-10      0       1295.34
mercedes-benz   ml230   245560.0        1998    2245    110     offroad man     5       7       ipg     2017-03-10      2017-03-10      0       1295.34
mercedes-benz   ml230   245560.0        1998    2245    110     offroad man     5       7       gasoline        2017-03-11      2017-03-11      0       1295.34
```

5. Visualize the data with Microsoft Power BI, recommend the appropriate vehicles.