Car store analysis

Prepared by

- Mohamed abdelwahab Farid Ashmawy[Data cleaning SQL]
- Mohamed Ayman Abdelhakam [EDA python]
- Safaa Elsayed Dawood[Data visualization]
- Aya Mahmoud Fouad [Tableau Dashboard & data visualization]

About the project

- This is the Car sales data set which include information about different cars . This data set is being taken from the Analytixlabs for the purpose of prediction
  In this we have to see two things
  First, we have seen which feature has more impact on car sales and carry out result of this
  Secondly, we have to train the classifier and to predict car sales and check the accuracy of the prediction.

When analyzing data for the car business, key insights often revolve around understanding customer preferences, market trends, operational efficiency, and profitability. Here are some of the most valuable insights you can derive from automotive data:

1. *CUSTOMER PREFERENCES & BUYING BEHAVIOR*

- *VEHICLE TYPE PREFERENCE*: ANALYZE TRENDS IN PREFERENCES FOR SUVS, SEDANS, TRUCKS, ELECTRIC VEHICLES (EVS), AND HYBRIDS. UNDERSTANDING WHICH MODELS OR SEGMENTS ARE GAINING POPULARITY CAN GUIDE INVENTORY AND MARKETING STRATEGIES.

- *PRICE SENSITIVITY*: IDENTIFY HOW PRICE CHANGES IMPACT DEMAND. SEGMENT CUSTOMERS BY BUDGET TO TARGET PRICING STRATEGIES MORE EFFECTIVELY.

2. *MARKET TRENDS*

- BRAND LOYALTY & SWITCHING*: INVESTIGATING CUSTOMER RETENTION RATES, BRAND LOYALTY, AND REASONS FOR SWITCHING BRANDS (E.G., RELIABILITY, COST OF OWNERSHIP, TECHNOLOGY) CAN HELP IMPROVE LONG-TERM CUSTOMER ENGAGEMENT.

3. *Technological Advancements*
   - *Connected Car Features*: The rise of in-car technology (e.g., infotainment systems, autonomous driving, connected car services) is shaping customer preferences. Tracking which features are most desired helps refine product offerings and marketing strategies.
   - *Autonomous & Semi-Autonomous Driving*: Analyzing the readiness of the market for autonomous features, consumer trust in these technologies, and regulatory developments is crucial for future planning.

4. *Competitive Benchmarking*
   - *Competitor Analysis*: Compare performance with competitors in terms of sales, customer satisfaction, and market share. Understanding competitor pricing, product offerings, and marketing strategies helps identify opportunities and threats.
   - *New Entrants (EV-focused companies) *: Companies like Tesla have disrupted the market. Keeping track of new entrants, especially in the EV sector, and how they are shaping customer preferences is key to staying competitive.

Cleaning data in SQL

SQLQuery3.sql - D...2N45QRK\fagr (55))

```sql
--select all data
select *
from dbo.car_prices
--delete the name of the day from the saledate column
UPDATE dbo.car_prices
SET saledate = SUBSTRING(saledate, 4, LEN(saledate))
FROM dbo.car_prices;

--review all data
select *
from dbo.car_prices

--replace column name
EXEC sp_rename 'dbo.car_prices.vin', 'Vehicle Identification Number', 'COLUMN';
EXEC sp_rename 'dbo.car_prices.Vehicle Identification Number', 'Vehicle_Id_NO', 'COLUMN';
EXEC sp_rename 'dbo.car_prices.trim', 'Car trim level', 'COLUMN';

--1* check the data type
--select the needed date only
SELECT LEFT(saledate, 12) AS ModifiedCode
from dbo.car_prices
```

SQLQuery3.sql - D...2N45QRK\fagr (55))

```sql
--update the datset with the needed date only
UPDATE dbo.car_prices
SET saledate = LEFT(saledate, 12)
from dbo.car_prices

--review all data
select *
from dbo.car_prices

--convert the salesdate from varchar into datetime
UPDATE dbo.car_prices
SET saledate = TRY_PARSE(saledate AS DATETIME USING 'en-US')
from dbo.car_prices

UPDATE dbo.car_prices
SET saledate = TRY_PARSE(saledate AS DATE USING 'en-US')
from dbo.car_prices
```

SQLQuery3.sql - D...2N45QRK\fagr (55))

```sql
--2* check null values and remove the Ineffective one
select *
from dbo.car_prices
where sellingprice is null

DELETE FROM dbo.car_prices
WHERE sellingprice IS NULL;

select *
from dbo.car_prices
where saledate is null

DELETE FROM dbo.car_prices
WHERE saledate IS NULL;

select *
from dbo.car_prices
WHERE interior IS NULL;

DELETE FROM dbo.car_prices
WHERE interior IS NULL;

select *
from dbo.car_prices
WHERE odometer IS NULL;

DELETE FROM dbo.car_prices
WHERE odometer IS NULL;
```

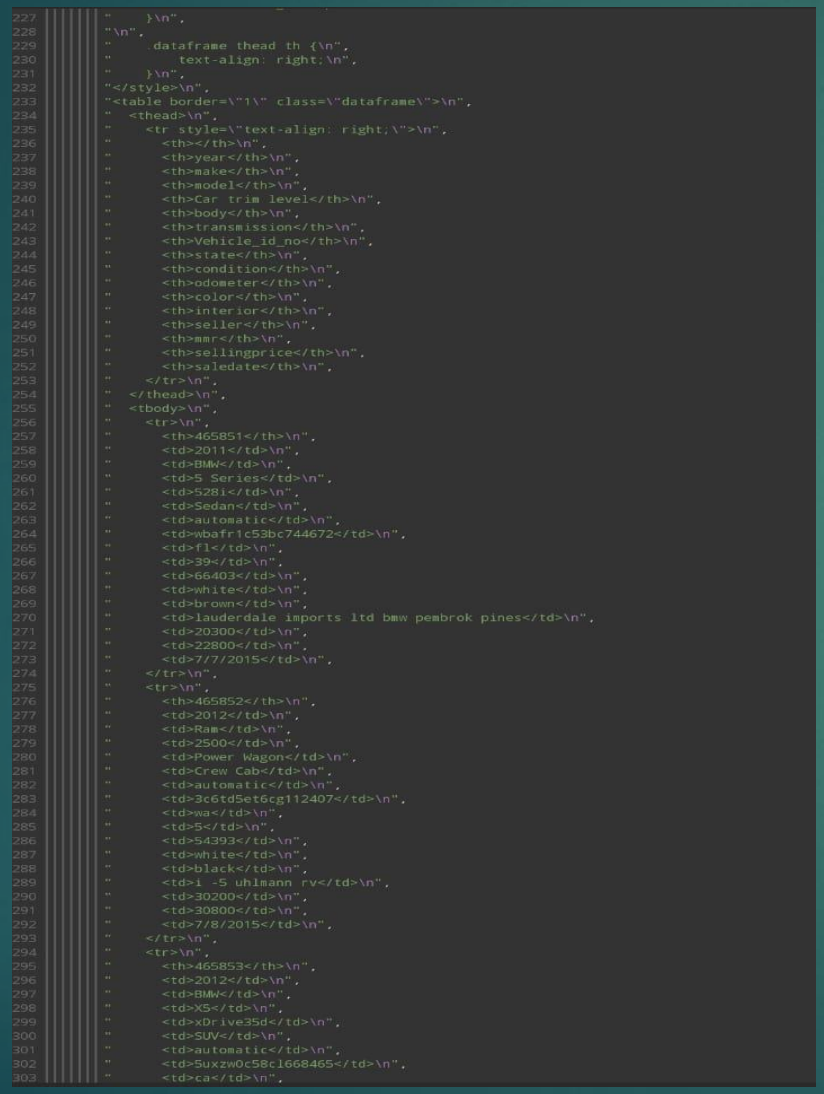
SQLQuery3.sql - D...2N45QRK\fagr (55))

```sql
DELETE FROM dbo.car_prices
WHERE odometer IS NULL;

select *
from dbo.car_prices
WHERE condition IS NULL;

DELETE FROM dbo.car_prices
WHERE condition IS NULL;

select *
from dbo.car_prices
WHERE model IS NULL and make is null and body is null

DELETE FROM dbo.car_prices
WHERE model IS NULL and make is null and body is null;

select *
from dbo.car_prices
WHERE body IS null

DELETE FROM dbo.car_prices
WHERE body IS NULL;

SELECT DISTINCT Vehicle_Id_NO
FROM dbo.car_prices;

select *
from dbo.car_prices
WHERE transmission IS null
```

SQLQuery3.sql - D...2N45QRK\fagr (55))

```sql
SELECT DISTINCT Vehicle_Id_NO
FROM dbo.car_prices;

select *
from dbo.car_prices
WHERE transmission IS null

DELETE FROM dbo.car_prices
WHERE transmission IS NULL;

select *
from dbo.car_prices

--3* check Duplicates values and remove the Ineffective one

SELECT Vehicle_Id_NO, COUNT(*) as duplicate
FROM dbo.car_prices
GROUP BY Vehicle_Id_NO
HAVING COUNT(*) > 1;
```

EDA Analysis with python



```
    }\n",
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>year</th>\n",
"      <th>make</th>\n",
"      <th>model</th>\n",
"      <th>Car trim level</th>\n",
"      <th>body</th>\n",
"      <th>transmission</th>\n",
"      <th>Vehicle_id_no</th>\n",
"      <th>state</th>\n",
"      <th>condition</th>\n",
"      <th>odometer</th>\n",
"      <th>color</th>\n",
"      <th>interior</th>\n",
"      <th>seller</th>\n",
"      <th>mmr</th>\n",
"      <th>sellingprice</th>\n",
"      <th>saledate</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>465851</th>\n",
"      <td>2011</td>\n",
"      <td>BMW</td>\n",
"      <td>5 Series</td>\n",
"      <td>528i</td>\n",
"      <td>Sedan</td>\n",
"      <td>automatic</td>\n",
"      <td>wbafr1c53bc744672</td>\n",
"      <td>fl</td>\n",
"      <td>39</td>\n",
"      <td>66403</td>\n",
"      <td>white</td>\n",
"      <td>brown</td>\n",
"      <td>lauderdale imports ltd bmw pembrok pines</td>\n",
"      <td>20300</td>\n",
"      <td>22800</td>\n",
"      <td>7/7/2015</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>465852</th>\n",
"      <td>2012</td>\n",
"      <td>Ram</td>\n",
"      <td>2500</td>\n",
"      <td>Power Wagon</td>\n",
"      <td>Crew Cab</td>\n",
"      <td>automatic</td>\n",
"      <td>3c6td5et6cg112407</td>\n",
"      <td>wa</td>\n",
"      <td>5</td>\n",
"      <td>54393</td>\n",
"      <td>white</td>\n",
"      <td>black</td>\n",
"      <td>i -5 uhlmann rv</td>\n",
"      <td>30200</td>\n",
"      <td>30800</td>\n",
"      <td>7/8/2015</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>465853</th>\n",
"      <td>2012</td>\n",
"      <td>BMW</td>\n",
"      <td>X5</td>\n",
"      <td>xDrive35d</td>\n",
"      <td>SUV</td>\n",
"      <td>automatic</td>\n",
"      <td>5uxzw0c58cl668465</td>\n",
"      <td>ca</td>\n",
"      <td>48</td>\n",
"      <td>50561</td>\n",
"      <td>black</td>\n",
"      <td>black</td>\n",
"      <td>financial services remarketing (lease)</td>\n",
"      <td>29800</td>\n",
"      <td>34000</td>\n",
"      <td>7/8/2015</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>465854</th>\n",
"      <td>2015</td>\n",
"      <td>Nissan</td>\n",
"      <td>Altima</td>\n",
"      <td>2.5 S</td>\n",
"      <td>sedan</td>\n",
"      <td>automatic</td>\n",
"      <td>1n4al3ap0fc216050</td>\n",
"      <td>ga</td>\n",
"      <td>38</td>\n",
"      <td>16658</td>\n",
"      <td>white</td>\n",
"      <td>black</td>\n",
"      <td>enterprise vehicle exchange / tra / rental / t...</td>\n",
"      <td>15100</td>\n",
"      <td>11100</td>\n",
"      <td>7/9/2015</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>465855</th>\n",
"      <td>2014</td>\n",
"      <td>Ford</td>\n",
"      <td>F-150</td>\n",
"      <td>XLT</td>\n",
"      <td>SuperCrew</td>\n",
"      <td>automatic</td>\n",
"      <td>1ftfw1et2eke87277</td>\n",
"      <td>ca</td>\n",
"      <td>34</td>\n",
"      <td>15008</td>\n",
"      <td>gray</td>\n",
"      <td>gray</td>\n",
"      <td>ford motor credit company llc pd</td>\n",
"      <td>29600</td>\n",
"      <td>26700</td>\n",
"      <td>5/28/2015</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>"
],
"text/plain": [
"        year    make     model Car trim level       body transmission  \\\n",
"465851  2011     BMW  5 Series           528i      Sedan    automatic   \n",
"465852  2012     Ram      2500    Power Wagon   Crew Cab    automatic   \n",
"465853  2012     BMW        X5      xDrive35d        SUV    automatic   \n",
"465854  2015  Nissan    Altima          2.5 S      sedan    automatic   \n",
"465855  2014    Ford     F-150            XLT  SuperCrew    automatic   \n",
"\n",
"              Vehicle_id_no state  condition  odometer  color interior  \\\n",
"465851  wbafr1c53bc744672    fl         39     66403  white    brown   \n",
"465852  3c6td5et6cg112407    wa          5     54393  white    black   \n",
"465853  5uxzw0c58cl668465    ca         48     50561  black    black   \n",
"465854  1n4al3ap0fc216050    ga         38     16658  white    black   \n",
"465855  1ftfw1et2eke87277    ca         34     15008   gray     gray   \n",
"\n",
"                                                  seller    mmr  \\\n",
"465851           lauderdale imports ltd bmw pembrok pines  20300   \n",
"465852                                i -5 uhlmann rv  30200   \n",
"465853             financial services remarketing (lease)  29800   \n",
"465854  enterprise vehicle exchange / tra / rental / t...  15100   \n",
"465855                    ford motor credit company llc pd  29600   \n",
"\n",
"        sellingprice   saledate  \n",
"465851         22800   7/7/2015  \n",
"465852         30800   7/8/2015  \n",
```

```
},
{
 "cell_type": "code",
 "execution_count": 27,
 "id": "354118d4-546c-4d40-8c13-5439389e88fb",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/html": [
     "<div>\n",
     "<style scoped>\n",
     "    .dataframe tbody tr th:only-of-type {\n",
     "        vertical-align: middle;\n",
     "    }\n",
     "\n",
     "    .dataframe tbody tr th {\n",
     "        vertical-align: top;\n",
     "    }\n",
     "\n",
     "    .dataframe thead th {\n",
     "        text-align: right;\n",
     "    }\n",
     "</style>\n",
     "<table border=\"1\" class=\"dataframe\">\n",
     "  <thead>\n",
     "    <tr style=\"text-align: right;\">\n",
     "      <th></th>\n",
     "      <th>year</th>\n",
     "      <th>condition</th>\n",
     "      <th>odometer</th>\n",
     "      <th>mmr</th>\n",
     "      <th>sellingprice</th>\n",
     "    </tr>\n",
     "  </thead>\n",
     "  <tbody>\n",
     "    <tr>\n",
     "      <th>count</th>\n",
     "      <td>465856.000000</td>\n",
     "      <td>465856.000000</td>\n",
     "      <td>465856.000000</td>\n",
     "      <td>465856.000000</td>\n",
     "      <td>465856.000000</td>\n",
     "    </tr>\n",
     "    <tr>\n",
     "      <th>mean</th>\n",
     "      <td>2010.217565</td>\n",
     "      <td>30.767261</td>\n",
     "      <td>66643.220029</td>\n",
     "      <td>13843.730627</td>\n",
     "      <td>13694.933634</td>\n",
     "    </tr>\n",
     "    <tr>\n",
     "      <th>std</th>\n",
     "      <td>3.828143</td>\n",
     "      <td>13.304447</td>\n",
     "      <td>52060.125277</td>\n",
     "      <td>9565.598447</td>\n",
     "      <td>9643.726466</td>\n",
     "    </tr>\n",
     "    <tr>\n",
     "      <th>min</th>\n",
     "      <td>1990.000000</td>\n",
     "      <td>1.000000</td>\n",
     "      <td>1.000000</td>\n",
     "      <td>25.000000</td>\n",
     "      <td>1.000000</td>\n",
     "    </tr>\n",
     "    <tr>\n",
     "      <th>25%</th>\n",
     "      <td>2008.000000</td>\n",
     "      <td>24.000000</td>\n",
     "      <td>28021.000000</td>\n",
     "      <td>7425.000000</td>\n",
     "      <td>7200.000000</td>\n",
     "    </tr>\n",
     "    <tr>\n",
```

```
    </tr>\n",
  </thead>\n",
  <tbody>\n",
    <tr>\n",
      <th>0</th>\n",
      <td>2015</td>\n",
      <td>Kia</td>\n",
      <td>Sorento</td>\n",
      <td>LX</td>\n",
      <td>SUV</td>\n",
      <td>automatic</td>\n",
      <td>5xyktca69fg566472</td>\n",
      <td>ca</td>\n",
      <td>5</td>\n",
      <td>16639</td>\n",
      <td>white</td>\n",
      <td>black</td>\n",
      <td>kia motors america  inc</td>\n",
      <td>20500</td>\n",
      <td>21500</td>\n",
      <td>12/16/2014</td>\n",
    </tr>\n",
    <tr>\n",
      <th>1</th>\n",
      <td>2015</td>\n",
      <td>Kia</td>\n",
      <td>Sorento</td>\n",
      <td>LX</td>\n",
      <td>SUV</td>\n",
      <td>automatic</td>\n",
      <td>5xyktca69fg561319</td>\n",
      <td>ca</td>\n",
      <td>5</td>\n",
      <td>9393</td>\n",
      <td>white</td>\n",
      <td>beige</td>\n",
      <td>kia motors america  inc</td>\n",
      <td>20800</td>\n",
      <td>21500</td>\n",
      <td>12/16/2014</td>\n",
    </tr>\n",
    <tr>\n",
      <th>2</th>\n",
      <td>2014</td>\n",
      <td>BMW</td>\n",
      <td>3 Series</td>\n",
      <td>328i SULEV</td>\n",
      <td>Sedan</td>\n",
      <td>automatic</td>\n",
      <td>wba3c1c51ek116351</td>\n",
      <td>ca</td>\n",
      <td>45</td>\n",
      <td>1331</td>\n",
      <td>gray</td>\n",
      <td>black</td>\n",
      <td>financial services remarketing (lease)</td>\n",
      <td>31900</td>\n",
      <td>30000</td>\n",
      <td>1/15/2015</td>\n",
    </tr>\n",
    <tr>\n",
      <th>3</th>\n",
      <td>2015</td>\n",
      <td>Volvo</td>\n",
      <td>S60</td>\n",
      <td>T5</td>\n",
      <td>Sedan</td>\n",
      <td>automatic</td>\n",
      <td>yv1612tb4f1310987</td>\n",
      <td>ca</td>\n",
      <td>41</td>\n",
      <td>14282</td>\n",
      <td>white</td>\n",
      <td>black</td>\n",
      <td>volvo na rep/world omni</td>\n",
      <td>27500</td>\n",
      <td>27750</td>\n",
```

```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 1,
   "id": "eca6f4d1-1fb3-4b53-972e-8ad517cfd7dd",
   "metadata": {},
   "outputs": [],
   "source": [
    "import pandas as pd"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 3,
   "id": "d0d606d4-5d23-4c21-9231-e3e647e74444",
   "metadata": {},
   "outputs": [],
   "source": [
    "import numpy as np"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 9,
   "id": "f6c2ba44-6a63-4e9d-b4e1-407a28a9569b",
   "metadata": {},
   "outputs": [],
   "source": [
    "df = pd.read_csv(\"Carscleaned1.csv\", encoding='ISO-8859-1')"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 15,
   "id": "fb22284e-1ae3-4fd1-9afd-c0b6129a2116",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/html": [
       "<div>\n",
       "<style scoped>\n",
       "    .dataframe tbody tr th:only-of-type {\n",
       "        vertical-align: middle;\n",
       "    }\n",
       "\n",
       "    .dataframe tbody tr th {\n",
       "        vertical-align: top;\n",
       "    }\n",
       "\n",
       "    .dataframe thead th {\n",
       "        text-align: right;\n",
       "    }\n",
       "</style>\n",
       "<table border=\"1\" class=\"dataframe\">\n",
       "  <thead>\n",
       "    <tr style=\"text-align: right;\">\n",
       "      <th></th>\n",
       "      <th>year</th>\n",
       "      <th>make</th>\n",
       "      <th>model</th>\n",
       "      <th>Car trim level</th>\n",
       "      <th>body</th>\n",
       "      <th>transmission</th>\n",
       "      <th>Vehicle_id_no</th>\n",
       "      <th>state</th>\n",
       "      <th>condition</th>\n",
       "      <th>odometer</th>\n",
       "      <th>color</th>\n",
       "      <th>interior</th>\n",
       "      <th>seller</th>\n",
       "      <th>mmr</th>\n",
       "      <th>sellingprice</th>\n",
       "      <th>saledate</th>\n",
       "    </tr>\n",
  </thead>\n",
```

```
    "state              0\n",
    "condition          0\n",
    "odometer           0\n",
    "color              0\n",
    "interior           0\n",
    "seller             0\n",
    "mmr                0\n",
    "sellingprice       0\n",
    "saledate           0\n",
    "dtype: int64"
   ]
  },
  "execution_count": 29,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "df.isnull().sum()"
]
},
{
 "cell_type": "code",
 "execution_count": 33,
 "id": "4bd38674-0db6-4b98-9cc9-88afdc0805d0",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/html": [
     "<div>\n",
     "<style scoped>\n",
     "    .dataframe tbody tr th:only-of-type {\n",
     "        vertical-align: middle;\n",
     "    }\n",
     "\n",
     "    .dataframe tbody tr th {\n",
     "        vertical-align: top;\n",
     "    }\n",
     "\n",
     "    .dataframe thead th {\n",
     "        text-align: right;\n",
     "    }\n",
     "</style>\n",
     "<table border=\"1\" class=\"dataframe\">\n",
     "  <thead>\n",
     "    <tr style=\"text-align: right;\">\n",
     "      <th></th>\n",
     "      <th>year</th>\n",
     "      <th>make</th>\n",
     "      <th>model</th>\n",
     "      <th>Car trim level</th>\n",
     "      <th>body</th>\n",
     "      <th>transmission</th>\n",
     "      <th>Vehicle_id_no</th>\n",
     "      <th>state</th>\n",
     "      <th>condition</th>\n",
     "      <th>odometer</th>\n",
     "      <th>color</th>\n",
     "      <th>interior</th>\n",
     "      <th>seller</th>\n",
     "      <th>mmr</th>\n",
     "      <th>sellingprice</th>\n",
     "      <th>saledate</th>\n",
     "    </tr>\n",
     "  </thead>\n",
     "  <tbody>\n",
     "    <tr>\n",
     "      <th>0</th>\n",
     "      <td>2015</td>\n",
     "      <td>Kia</td>\n",
     "      <td>Sorento</td>\n",
     "      <td>LX</td>\n",
     "      <td>SUV</td>\n",
     "      <td>automatic</td>\n",
     "      <td>5xyktca69fg566472</td>\n",
     "      <td>ca</td>\n",
```



```
     "    <tr>\n",
     "      <th>50%</th>\n",
     "      <td>2012.000000</td>\n",
     "      <td>35.000000</td>\n",
     "      <td>50849.500000</td>\n",
     "      <td>12300.000000</td>\n",
     "      <td>12200.000000</td>\n",
     "    </tr>\n",
     "    <tr>\n",
     "      <th>75%</th>\n",
     "      <td>2013.000000</td>\n",
     "      <td>42.000000</td>\n",
     "      <td>96604.250000</td>\n",
     "      <td>18300.000000</td>\n",
     "      <td>18200.000000</td>\n",
     "    </tr>\n",
     "    <tr>\n",
     "      <th>max</th>\n",
     "      <td>2015.000000</td>\n",
     "      <td>49.000000</td>\n",
     "      <td>999999.000000</td>\n",
     "      <td>182000.000000</td>\n",
     "      <td>230000.000000</td>\n",
     "    </tr>\n",
     "  </tbody>\n",
     "</table>\n",
     "</div>"
    ],
    "text/plain": [
     "               year      condition       odometer            mmr  \\\n",
     "count  465856.000000  465856.000000  465856.000000  465856.000000   \n",
     "mean     2010.217565      30.767261   66643.220029   13843.730627   \n",
     "std         3.828143      13.304447   52060.125277    9565.598447   \n",
     "min      1990.000000       1.000000       1.000000      25.000000   \n",
     "25%      2008.000000      24.000000   28021.000000    7425.000000   \n",
     "50%      2012.000000      35.000000   50849.500000   12300.000000   \n",
     "75%      2013.000000      42.000000   96604.250000   18300.000000   \n",
     "max      2015.000000      49.000000  999999.000000  182000.000000   \n",
     "\n",
     "        sellingprice  \n",
     "count  465856.000000  \n",
     "mean    13694.933634  \n",
     "std      9643.726466  \n",
     "min         1.000000  \n",
     "25%      7200.000000  \n",
     "50%     12200.000000  \n",
     "75%     18200.000000  \n",
     "max    230000.000000  "
    ]
   },
   "execution_count": 27,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "df.describe()"
 ]
},
{
 "cell_type": "code",
 "execution_count": 29,
 "id": "cb12f665-ae15-47e7-aa8d-ddff99ca394e",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "year               0\n",
     "make               0\n",
     "model             88\n",
     "Car trim level     0\n",
     "body               0\n",
     "transmission       0\n",
     "Vehicle_id_no      0\n",
     "state              0\n",
     "condition          0\n",
```



```
     "465851         22800   7/7/2015  \n",
     "465852         30800   7/8/2015  \n",
     "465853         34000   7/8/2015  \n",
     "465854         11100   7/9/2015  \n",
     "465855         26700  5/28/2015  "
    ]
   },
   "execution_count": 17,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "df.tail()"
 ]
},
{
 "cell_type": "code",
 "execution_count": 19,
 "id": "a2b3bd48-818f-4dd5-a8d8-ccaecd96f4a9",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "(465856, 16)"
    ]
   },
   "execution_count": 19,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "df.shape"
 ]
},
{
 "cell_type": "code",
 "execution_count": 23,
 "id": "635a8db3-4ef5-4646-865c-709dbbccb9cf",
 "metadata": {},
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "<class 'pandas.core.frame.DataFrame'>\n",
    "RangeIndex: 465856 entries, 0 to 465855\n",
    "Data columns (total 16 columns):\n",
    " #   Column          Non-Null Count   Dtype \n",
    "---  ------          --------------   ----- \n",
    " 0   year            465856 non-null  int64 \n",
    " 1   make            465856 non-null  object\n",
    " 2   model           465768 non-null  object\n",
    " 3   Car trim level  465856 non-null  object\n",
    " 4   body            465856 non-null  object\n",
    " 5   transmission    465856 non-null  object\n",
    " 6   Vehicle_id_no   465856 non-null  object\n",
    " 7   state           465856 non-null  object\n",
    " 8   condition       465856 non-null  int64 \n",
    " 9   odometer        465856 non-null  int64 \n",
    " 10  color           465856 non-null  object\n",
    " 11  interior        465856 non-null  object\n",
    " 12  seller          465856 non-null  object\n",
    " 13  mmr             465856 non-null  int64 \n",
    " 14  sellingprice    465856 non-null  int64 \n",
    " 15  saledate        465856 non-null  object\n",
    "dtypes: int64(5), object(11)\n",
    "memory usage: 56.9+ MB\n"
   ]
  }
 ],
 "source": [
  "df.info()"
 ]
},
```

```
      <td>ca</td>\n",
      <td>43</td>\n",
      <td>2641</td>\n",
      <td>gray</td>\n",
      <td>black</td>\n",
      <td>financial services remarketing (lease)</td>\n",
      <td>66000</td>\n",
      <td>67000</td>\n",
      <td>12/18/2014</td>\n",
    </tr>\n",
    <tr>\n",
      <th>...</th>\n",
      <td>...</td>\n",
      <td>...</td>\n",
      <td>...</td>\n",
      <td>...</td>\n",
      <td>...</td>\n",
      <td>...</td>\n",
      <td>...</td>\n",
      <td>...</td>\n",
      <td>...</td>\n",
      <td>...</td>\n",
      <td>...</td>\n",
      <td>...</td>\n",
      <td>...</td>\n",
      <td>...</td>\n",
      <td>...</td>\n",
      <td>...</td>\n",
    </tr>\n",
    <tr>\n",
      <th>465851</th>\n",
      <td>2011</td>\n",
      <td>BMW</td>\n",
      <td>5 Series</td>\n",
      <td>528i</td>\n",
      <td>Sedan</td>\n",
      <td>automatic</td>\n",
      <td>wbafr1c53bc744672</td>\n",
      <td>fl</td>\n",
      <td>39</td>\n",
      <td>66403</td>\n",
      <td>white</td>\n",
      <td>brown</td>\n",
      <td>lauderdale imports ltd bmw pembrok pines</td>\n",
      <td>20300</td>\n",
      <td>22800</td>\n",
      <td>7/7/2015</td>\n",
    </tr>\n",
    <tr>\n",
      <th>465852</th>\n",
      <td>2012</td>\n",
      <td>Ram</td>\n",
      <td>2500</td>\n",
      <td>Power Wagon</td>\n",
      <td>Crew Cab</td>\n",
      <td>automatic</td>\n",
      <td>3c6td5et6cg112407</td>\n",
      <td>wa</td>\n",
      <td>5</td>\n",
      <td>54393</td>\n",
      <td>white</td>\n",
      <td>black</td>\n",
      <td>i -5 uhlmann rv</td>\n",
      <td>30200</td>\n",
      <td>30800</td>\n",
      <td>7/8/2015</td>\n",
    </tr>\n",
    <tr>\n",
      <th>465853</th>\n",
      <td>2012</td>\n",
      <td>BMW</td>\n",
      <td>X5</td>\n",
      <td>xDrive35d</td>\n",
      <td>SUV</td>\n",
      <td>automatic</td>\n",
      <td>5uxzw0c58cl668465</td>\n",
      <td>ca</td>\n",
      <td>48</td>\n",
      <td>50561</td>\n",
      <td>black</td>\n",
      <td>kia motors america  inc</td>\n",
      <td>20500</td>\n",
      <td>21500</td>\n",
      <td>12/16/2014</td>\n",
    </tr>\n",
    <tr>\n",
      <th>465854</th>\n",
      <td>2014</td>\n",
      <td>BMW</td>\n",
      <td>3 Series</td>\n",
      <td>328i SULEV</td>\n",
      <td>Sedan</td>\n",
      <td>automatic</td>\n",
      <td>wba3c1c51ek116351</td>\n",
      <td>ca</td>\n",
      <td>45</td>\n",
      <td>1331</td>\n",
      <td>gray</td>\n",
      <td>black</td>\n",
      <td>financial services remarketing (lease)</td>\n",
      <td>31900</td>\n",
      <td>30000</td>\n",
      <td>1/15/2015</td>\n",
    </tr>\n",
    <tr>\n",
      <th>465855</th>\n",
      <td>2015</td>\n",
      <td>Volvo</td>\n",
      <td>S60</td>\n",
      <td>T5</td>\n",
      <td>Sedan</td>\n",
      <td>automatic</td>\n",
      <td>yv1612tb4f1310987</td>\n",
      <td>ca</td>\n",
      <td>41</td>\n",
      <td>14282</td>\n",
      <td>white</td>\n",
      <td>black</td>\n",
      <td>volvo na rep/world omni</td>\n",
      <td>27500</td>\n",
      <td>27750</td>\n",
      <td>1/29/2015</td>\n",
    </tr>\n",
    <tr>\n",
      <th>465856</th>\n",
      <td>2014</td>\n",
      <td>BMW</td>\n",
      <td>6 Series Gran Coupe</td>\n",
      <td>650i</td>\n",
      <td>Sedan</td>\n",
      <td>automatic</td>\n",
      <td>wba6b2c57ed129731</td>\n",
      <td>ca</td>\n",
```

Dashboard

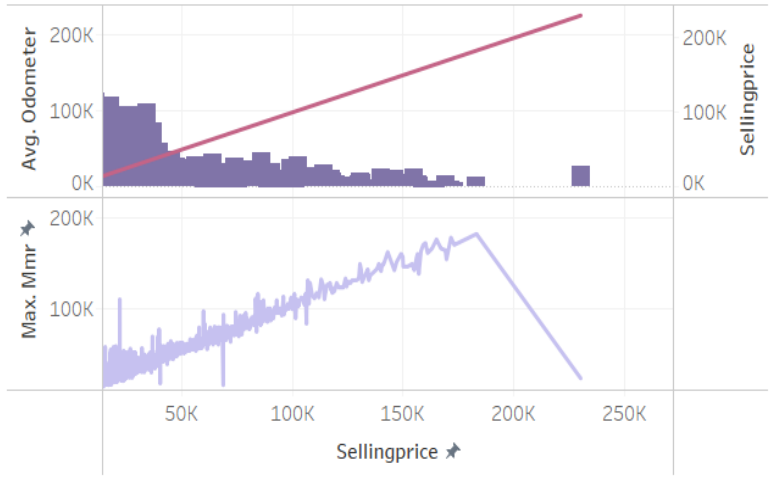
Cars Stores Sales Dashboard

Count of Seller
465.86K

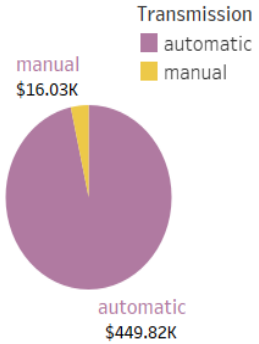
Condition
465.86K

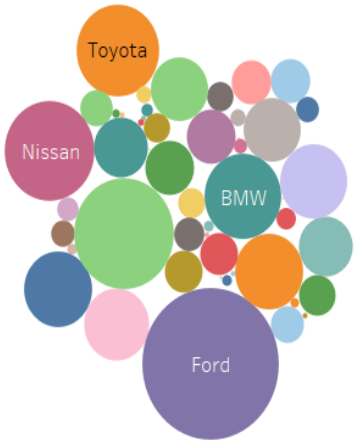
Sum of Sellingprice
$6,379.87M

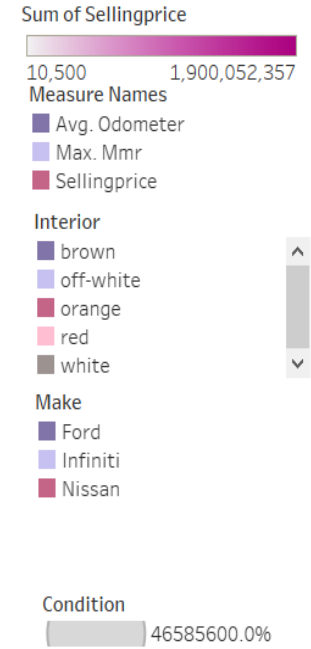
Count of Odometer
466K



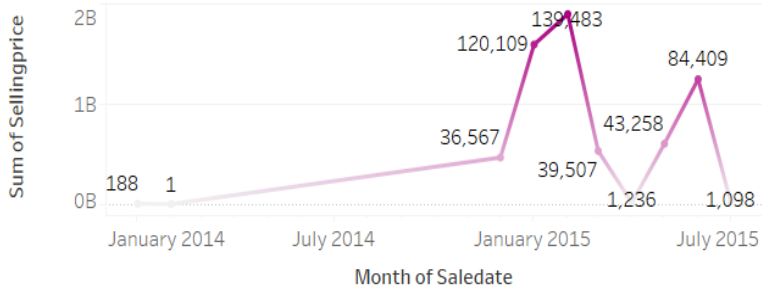
cnt of sellingprice by transmisstion



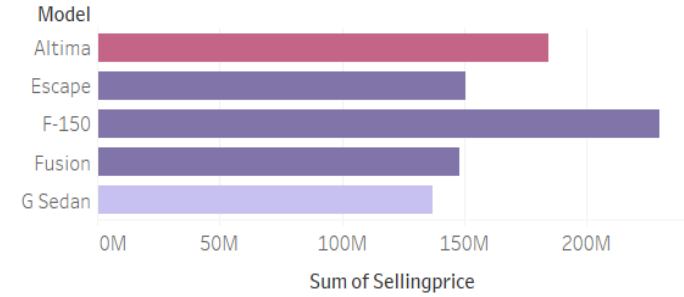




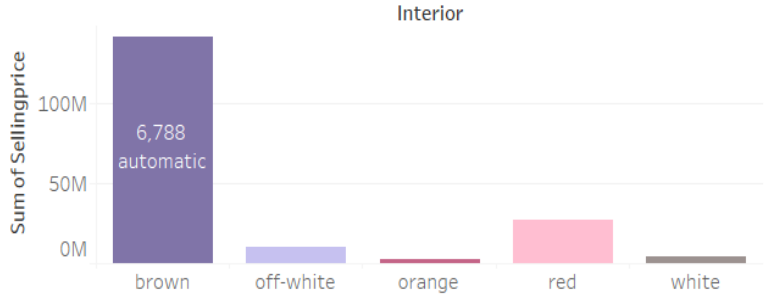
Timeline



Top 5 model/make sales

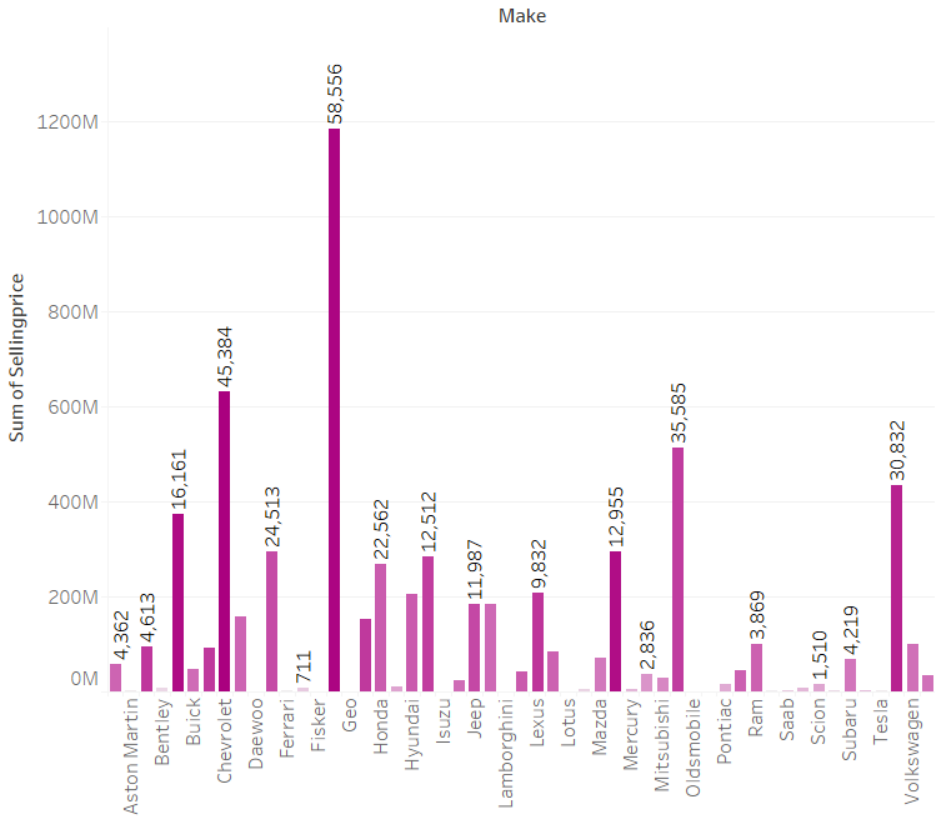


Top 5 interior color sales

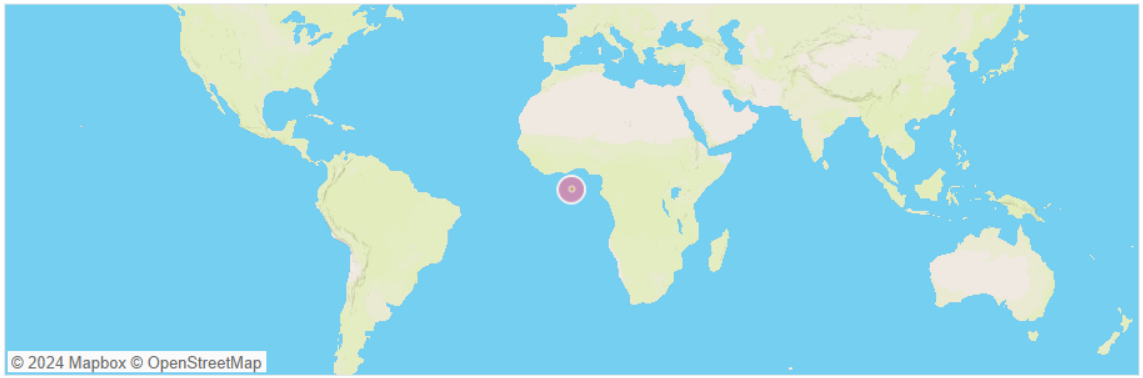


Cars Stores Sales Dashboard

Sheet 7



by state



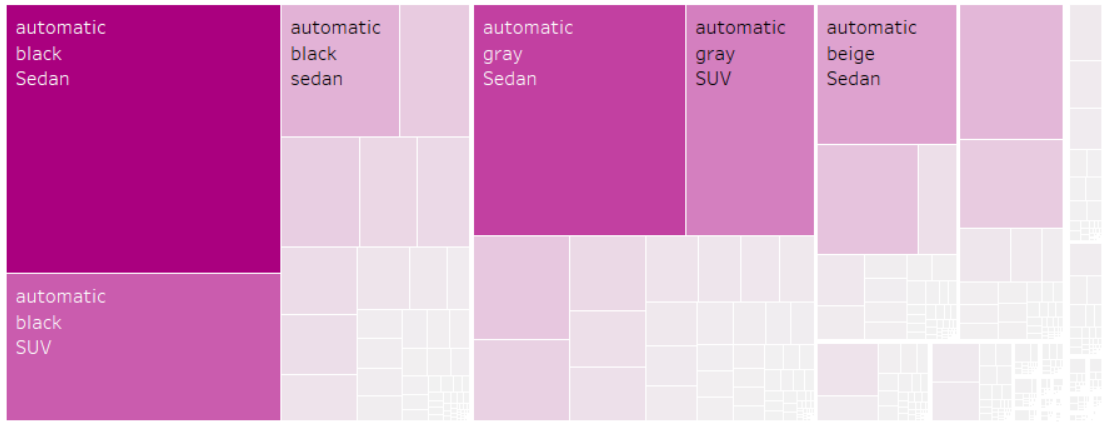
Sellingprice
6,480 17,234

Condition
21
20,000
40,000
60,000
74,378

Distinct count of Sell..
1 933

Count of Book1.csv
1 75,696

Sheet 4




Thank
you