Spark Course | Francisco Juan Vigo Rabolini

USED CARS - MARKET ANALYSIS

Introduction

Our company has to buy five new cars for the executive directors. It is known that cars are an asset that depreciates at an accelerated rate. Therefore, the CFO of the company requested the Data Science Department (DSD) a full report of the used car market, in order to find the model that has better resale value, in order to minimize the impact of the depreciation.

To accomplish that task, the DSD programmed a web scraper to extract all the information of the used cars published in Ebay. In this report we analyse the results, we provide some insights about the used car market in the US and finally we provide our final choice to the CFO.

Data Exploration

After loading our dataset to Spark, we performed some basic data exploration. In this case our dataset has 200.000 rows, one for each scrapped car, and 12 features.

```
root
 |-- region: string (nullable = true)
 |-- price: integer (nullable = true)
 |-- year: integer (nullable = true)
 |-- manufacturer: string (nullable = true)
 |-- model: string (nullable = true)
 |-- condition: string (nullable = true)
 |-- cylinders: string (nullable = true)
 |-- fuel: string (nullable = true)
 |-- odometer: integer (nullable = true)
 |-- transmission: string (nullable = true)
 |-- size: string (nullable = true)
 |-- paint_color: string (nullable = true)
```

Our next step is to check if we have nulls in our dataset. We have used the in-built spark function <is.Null> to perform this task.

```
Checking for nulls:
+------+-----+----+------------+-----+---------+---------+----+--------+------------+------+-----------+
|region|price|year|manufacturer|model|condition|cylinders|fuel|odometer|transmission|  size|paint_color|
+------+-----+----+------------+-----+---------+---------+----+--------+------------+------+-----------+
|     0|    0|   0|        6746| 1620|    55853|        0|1833|       0|          57|101491|      34298|
+------+-----+----+------------+-----+---------+---------+----+--------+------------+------+-----------+
```

As suspected, we have a lot of nulls in many columns. This is very common in scraped dataset. Fortunately we have enough data points to drop the nulls and continue with our analysis. We have used <na.drop(how='any')> and ended up with a cleaned dataset of 81.760 rows, less than half of the original dataset, but still representative to continue with our analysis.

After the cleaning, we compute some basic statistics of our numerical columns to better understand our data, using <.summary()>.

summary	year	price	odometer
count	81760	81760	81760
mean	2008.861350293542	24330.1167074364	117585.3887353229
stddev	7.595095067547672	2946779.863707078	140058.86616449116
min	1915	0	0
25%	2006	3995	69078
50%	2010	7995	111521
75%	2014	14995	153534
max	2020	831365932	10000000

Some insights about our numerical variables:

- We have very old cars (min year = 1915). We are going to recode the variable year because they are antique cars and therefore their price is not related to our business question.
- The mean “age” of our cars is around 2008, but the median (50% percentile) is 2010. This means that our distribution is negatively skewed because our mean is smaller than the median.
- On average, the used cars for sale from our dataset cost 24K USD and they have around 117K miles.
- We have outliers in both price and odometer.

Feature Engineering

To be able to get better insights, we are going to re-code two features: odometer and year.

To recode ‘odometer’ we are going to use its statistics and create 3 groups.

[0 - 25th percentile] : Low Miles

[25th - 75th percentile] : Mid Miles

[75th - 100th percentile] : High Miles

To re-code 'year' we are going to rename cars older than 1969 as 'Antique'. Then we are going to create a category for each decade (70's, 80's, 90's, 00's). Since we are more interested in 'recent' cars, we are going to keep the numerical value for cars above 2010.

```
carsDF3 = carsDF2\
   .withColumn("odometer",     when((col("odometer")>=0) & (col("odometer")<=69078),"Low Miles")\
                              .when((col("odometer")>69078) & (col("odometer")<=153534),"Mid Miles")\
                              .otherwise("High Miles"))\
   .withColumn("year",     when((col("year")>=0) & (col("year")<=1969),"Antique")\
                              .when((col("year")> 1970) & (col("year")<=1979),"70's")\
                              .when((col("year")> 1979) & (col("year")<=1989),"80's")\
                              .when((col("year")> 1989) & (col("year")<=1999),"90's")\
                              .when((col("year")> 1999) & (col("year")<=2009),"00's")\
                              .when((col("year")== 2010),2010)\
                              .when((col("year")== 2011),2011)\
                              .when((col("year")== 2012),2012)\
                              .when((col("year")== 2013),2013)\
                              .when((col("year")== 2014),2014)\
                              .when((col("year")== 2015),2015)\
                              .when((col("year")== 2016),2016)\
                              .when((col("year")== 2017),2017)\
                              .when((col("year")== 2018),2018)\
                              .when((col("year")== 2019),2019)\
                              .when((col("year")== 2020),2020)\
                              .otherwise("Other"))
```

Simple Business Questions

After finishing with our Feature Engineering, we started asking our data some common sense questions.

1) *Is there a connection between price and mileage?*

odometer	Price
Low Miles	59360.0
Mid Miles	15735.0
High Miles	6503.0

Happily, our data shows normal behavior and the price decreases while the mileage increases. The *'price-step'* is bigger from low to mid mileage (-73%) in comparison with mid to high (-59%).

2) Is there a connection between price and year?

year	Price	Count	Max Prince
00's	31868.0	32835	831365932
2010	8842.0	4442	149995
2011	11547.0	5404	1950000
2012	17999.0	5661	18880499
2013	12961.0	5645	150000
2014	19452.0	5247	26880499
2015	21936.0	5224	23880499
2016	18132.0	4810	295000
2017	35665.0	3404	36980499
2018	22097.0	2176	77500
2019	20777.0	1355	85500
2020	35295.0	70	99999
70's	10714.0	403	123456
80's	6877.0	684	68500
90's	4897.0	3885	500000
Antique	278737.0	478	123456789
Other	22418.0	37	68500

- There is a strong connection between price and years, but we have some exceptions. For example, cars from 2013 are -in average- cheaper than those from 2012. Also we found a weird pattern in the case of cars from 2017, because they are more expensive than those from 2018. This might be related with the fact that it is somehow strange to sell a brand-new car (2018 or 2019). In this case, the cars that are for sale might have a mechanical problem or it is possible that they were crashed. This phenomena might also be related with the number of observations.
- Another interesting insight from this table is that cars from the 70's are more expensive than the ones from the 80's and 90's. This might be related with the revalorization of classic cars.

3) What is the most popular fuel?

fuel	Avg Price	N.	% of total
hybrid	8994.0	767	0.9
other	10418.0	96	0.1
electric	18975.0	51	0.1
diesel	22106.0	6015	7.4
gas	24688.0	74831	91.5

- Gas is the most popular fuel because it represent 91,5% of the total cars of our dataset.
- Both electric and hybrid car prices are on average cheaper than diesel and gas.
- Hybrid cars are 9 times more popular than electric cars (0.9% versus 0.1%.

4) Are larger cars more expensive?

size	Avg Price	N.	% of total
mid-size	8709.0	24190	29.6
full-size	17800.0	46339	56.7
sub-compact	28087.0	1231	1.5
compact	91914.0	10000	12.2

- Full size are more expensive than mid-size, but both sub-compact and compact are on the top of the list.
- More than half of our dataset are full-size cars (56.7%)

5) Are automatic cars more expensive on average than manual cars?

transmission	Avg Price	N.	% of total
other	7960.0	759	0.9
manual	9341.0	5427	6.6
automatic	25571.0	75574	92.4

- Manual cars are on average cheaper than automatic ones.
- 92.4% of the cars from our dataset are automatic.

6) Is there a connection between price and color?

paint_color	Avg Price	N.	% of total
green	7748.0	2602	3.2
purple	8493.0	252	0.3
brown	8659.0	2351	2.9
custom	10545.0	2156	2.6
grey	10597.0	9598	11.7
orange	12557.0	450	0.6
yellow	12737.0	559	0.7
black	13260.0	14864	18.2
red	13433.0	8185	10.0
white	17770.0	20212	24.7
blue	23800.0	8392	10.3
silver	77717.0	12139	14.8

- At a first glance, there is no connection between price and color. Silver and blue cars are the most expensive ones in our dataset.
- White, black, grey and silver are the more popular colors.

Model Hunting

After having made an exploration of our dataset, we will reduce the focus and filter the values that interest us for our business question.

We just want to keep the cars that are automatic, gas fueled, white, black, gray or silver and newer than 2014. We also excluded full-size cars, because we are not interested in trucks or 4x4.

For that analysis, we created a new dataset called “targetDF”, applying all filters mentioned previously.

```
targetDF = carsDF3\
    .where(col("transmission") == "automatic" )\
    .where((col("paint_color") == "black") |\
           (col("paint_color") == "silver") |\
           (col("paint_color") == "white") |\
           (col("paint_color") == "grey"))\
     .where(col("fuel") == "gas")\
     .where(col("year") > 2014)\
     .where(col("size") != "full-size")
```

Our new dataset has only 5122 rows. Finally, we also filtered our new dataset in order to keep brands that have more than 5% of market share.

manufacturer	Avg Price	N.	% of total
chevrolet	14343.0	754	14.7
ford	15230.0	672	13.1
nissan	11381.0	513	10.0
toyota	13765.0	485	9.5
honda	63834.0	306	6.0
jeep	17487.0	262	5.1
hyundai	11268.0	258	5.0

During this exploration, we discovered that Honda cars are 5 times more expensive than the rest of the brands in our dataset. This looks weird and it's highly likely due to outliers.

model	Avg Price	AVG Year	N.	% of total
civic sedan	831221.0	2016.0	19	0.4
pilot touring	26963.0	2016.0	1	0.0
pilot ex-l	26880.0	2017.0	1	0.0
ridgeline sport	26500.0	2017.0	1	0.0
cr-v ex-l	25650.0	2017.0	1	0.0
odyssey	25000.0	2016.0	1	0.0
clarity plug-in h...	24999.0	2018.0	1	0.0
civic sport hatch...	23995.0	2019.0	1	0.0
crv ex	21498.0	2018.0	3	0.1
cr-v ex-l grey	21495.0	2016.0	1	0.0

Doing some exploration, we found that the average price for the Honda Civic Sedan is around 831K USD. It is definitely an outlier. We proceed by configuring a function to remove outliers in price.

```
# Calculate my lower bound

targetDF2.approxQuantile("price",[0.25], 0)
```

```
[7970.0]
```

```
# Calculate my upper bound

targetDF2.approxQuantile("price",[0.75], 0)
```

```
[18900.0]
```

```
q1 = 7970
q3 = 18900

iqr = q3-q1 #Interquartile range
fence_low  = q1-1.5*iqr
fence_high = q3+1.5*iqr

targetDF3 = targetDF2.where((col("price") > fence_low) & (col("price") < fence_high))

print("I have removed", targetDF2.count() - targetDF3.count(), "outliers.-")
```

```
I have removed 82 outliers
```

After running our function, we check how our new data is distributed using the function <summary()>.

```
Before...
+-------+-----------------+
|summary|            price|
+-------+-----------------+
|  count|             3250|
|   mean|18641.657846153845|
| stddev|272722.38852896006|
|    min|                0|
|    25%|             7970|
|    50%|            13600|
|    75%|            18900|
|    max|         15550499|
+-------+-----------------+

After removing outliers...
+-------+-----------------+
|summary|            price|
+-------+-----------------+
|  count|             3168|
|   mean|13053.252525252525|
| stddev| 8742.648262886763|
|    min|                0|
|    25%|             7700|
|    50%|            13450|
|    75%|            17997|
|    max|            35000|
+-------+-----------------+
```

Our function worked and now we can proceed with our analysis: grouping by manufacturer to check if there are some average price differences.

manufacturer	Avg Price	AVG Year	N.	% of total
chevrolet	12754.0	2017.0	719	14.0
ford	14416.0	2016.0	652	12.7
nissan	11157.0	2017.0	511	10.0
toyota	13172.0	2017.0	474	9.3
honda	13058.0	2016.0	305	6.0
hyundai	10787.0	2017.0	257	5.0
jeep	16336.0	2016.0	250	4.9

Jeep and Ford look like the most expensive cars,but we do not observe significant price differences. We continue our analysis grouping by brand and model.

manufacturer	model	Avg Price	N.	% of total
toyota	tacoma	21649.0	64	1.2
chevrolet	tahoe	14139.0	60	1.2
ford	escape	12896.0	65	1.3
chevrolet	silverado 1500	11147.0	90	1.8
ford	fusion	10621.0	59	1.2
nissan	altima	9870.0	67	1.3
toyota	camry	8985.0	68	1.3
ford	focus	8780.0	55	1.1
nissan	sentra	7796.0	76	1.5
toyota	corolla	6942.0	69	1.3

Finally, our recommendation for the CFO is the Ford Fusion, since it is the sedan that best maintains the resale value, above the Nissan Altima and the Honda Civic.

We have skipped the first options since they are pick-ups and they are not useful for our business question.