

Total Auto Sales 2020(faked)

Jie Guo

2/11/2021



Objective:

As the Data Analyst Team of **Karate Of Barack Epic (A.K.A KOBE) Auto**, which is the best dealer at west coastline, we have received the report of **Total Auto Sales 2020(faked)** from our statistic department. Now we would like to combine our new technique to visualize this dataset and answer 5 business questions related to the potential benefits of K.O.B.E Vehicle.

- 1. Use Neo4j to create graphic database:
 - Nodes, Attributes and Relations

Click Here Jump To Data Wrangling Part

- 1. Five questions related to improvement if service and benefits:
 - Which manufacturer has the highest quantity of models in this database?
 - Which model is the best deal one for each horsepower?
 - The average sales for each model located in German;
 - Which manufacturer have the highest sales?
 - Compare to the horsepower, which model has the best performance on MPG?

Finally, the **conclusion and recommendation** will be given.

In [1]:

```
from neo4jrestclient.client import GraphDatabase
from py2neo import Graph, Node, Relationship
db = Graph("http://localhost:7474", username="neo4j", password='523698741')
db.delete_all()
```

1.Creating Nodes, Attributes and Relations

In [2]:

```
car1 = Node("model",name='A4')
car2 = Node("model",name='A6')
car3 = Node("model",name='A8')
car4 = Node("model",name='RS')
car5 = Node("model",name='Challenger')
car6 = Node("model",name='Avenger')
car7 = Node("model",name='323i')
car8 = Node("model",name='M2')
car9 = Node("model",name='Accord')
car10 = Node("model",name='Civic')
car11 = Node("model",name='Passport')
car12 = Node("model",name='H9')
```

```

car13 = Node("model",name='I5')
db.create(car1)
db.create(car2)
db.create(car3)
db.create(car4)
db.create(car5)
db.create(car6)
db.create(car7)
db.create(car8)
db.create(car9)
db.create(car10)
db.create(car11)
db.create(car12)
db.create(car13)

```

```

Manufacturer1 = Node("Manufacturer", name = "Audi")
Manufacturer2 = Node("Manufacturer", name = "BMW")
Manufacturer3 = Node("Manufacturer", name = "Dodge")
Manufacturer4= Node("Manufacturer", name = "Honda")

```

```

Manufacturer5= Node("Manufacturer", name = "Hongqi")

```

```

Country1 = Node("Country", country = "China", city = 'Beijing')
Country2 = Node("Country", country = "U.S", city = 'CA')
Country3 = Node("Country", country = "German", city = 'Berlin')
Country4 = Node("Country", country = "Japan", city = 'Tokyo')
db.create(Manufacturer1)
db.create(Manufacturer2)
db.create(Manufacturer3)
db.create(Manufacturer4)

```

```

db.create(Manufacturer5)

```

```

db.create(Country1)
db.create(Country2)
db.create(Country3)
db.create(Country4)

```

In [3]:

```

car1_Manufacturer1 = Relationship(car1,'belong',Manufacturer1)
db.create(car1_Manufacturer1)
car2_Manufacturer1 = Relationship(car2,'belong',Manufacturer1)
db.create(car2_Manufacturer1)
car3_Manufacturer1 = Relationship(car3,'belong',Manufacturer1)
db.create(car3_Manufacturer1)
car4_Manufacturer1 = Relationship(car4,'belong',Manufacturer1)
db.create(car4_Manufacturer1)
car5_Manufacturer3 = Relationship(car5,'belong',Manufacturer3)
db.create(car5_Manufacturer3)
car6_Manufacturer3 = Relationship(car6,'belong',Manufacturer3)
db.create(car6_Manufacturer3)
car7_Manufacturer2 = Relationship(car7,'belong',Manufacturer2)
db.create(car7_Manufacturer2)
car8_Manufacturer2 = Relationship(car8,'belong',Manufacturer2)
db.create(car8_Manufacturer2)
car9_Manufacturer = Relationship(car9,'belong',Manufacturer4)
db.create(car9_Manufacturer)
car10_Manufacturer = Relationship(car10,'belong',Manufacturer4)
db.create(car10_Manufacturer)
car11_Manufacturer = Relationship(car11,'belong',Manufacturer4)
db.create(car11_Manufacturer)
car12_Manufacturer = Relationship(car12,'belong',Manufacturer5)
db.create(car12_Manufacturer)
car13_Manufacturer = Relationship(car13,'belong',Manufacturer5)
db.create(car13_Manufacturer)

```

In [4]:

```

Manufacturer1_Country=Relationship(Manufacturer1,'located',Country3)
Manufacturer2_Country=Relationship(Manufacturer2,'located',Country3)
Manufacturer3_Country=Relationship(Manufacturer3,'located',Country2)
Manufacturer4_Country=Relationship(Manufacturer4,'located',Country4)
Manufacturer5_Country=Relationship(Manufacturer5,'located',Country1)

db.create(Manufacturer1_Country)
db.create(Manufacturer2_Country)

```

```
db.create(Manufacturer3_Country)
db.create(Manufacturer4_Country)
db.create(Manufacturer5_Country)
```

In [5]:

```
car1_country=Relationship(car1,'located',Country3)
car2_country=Relationship(car2,'located',Country3)
car3_country=Relationship(car3,'located',Country3)
car4_country=Relationship(car4,'located',Country3)
car5_country=Relationship(car5,'located',Country2)
car6_country=Relationship(car6,'located',Country2)
car7_country=Relationship(car7,'located',Country3)

car8_country=Relationship(car8,'located',Country3)
car9_country=Relationship(car9,'located',Country4)
car10_country=Relationship(car10,'located',Country4)
car11_country=Relationship(car11,'located',Country4)
car12_country=Relationship(car12,'located',Country1)
car13_country=Relationship(car13,'located',Country1)
```

```
db.create(car1_country)
db.create(car2_country)
db.create(car3_country)
db.create(car4_country)
db.create(car5_country)
db.create(car6_country)
db.create(car7_country)
db.create(car8_country)
db.create(car9_country)
db.create(car10_country)
db.create(car11_country)
db.create(car12_country)
db.create(car13_country)
```

In [6]:

```
Manufacturer1_Manufacturer2=Relationship(Manufacturer1,'coopetition',Manufacturer2)
Manufacturer1_Manufacturer3=Relationship(Manufacturer1,'coopetition',Manufacturer3)
Manufacturer1_Manufacturer4=Relationship(Manufacturer1,'coopetition',Manufacturer4)
Manufacturer1_Manufacturer5=Relationship(Manufacturer1,'coopetition',Manufacturer5)
Manufacturer2_Manufacturer3=Relationship(Manufacturer2,'coopetition',Manufacturer3)
Manufacturer2_Manufacturer4=Relationship(Manufacturer2,'coopetition',Manufacturer4)
Manufacturer2_Manufacturer5=Relationship(Manufacturer2,'coopetition',Manufacturer5)
Manufacturer3_Manufacturer4=Relationship(Manufacturer3,'coopetition',Manufacturer4)
Manufacturer3_Manufacturer5=Relationship(Manufacturer3,'coopetition',Manufacturer5)
Manufacturer4_Manufacturer5=Relationship(Manufacturer4,'coopetition',Manufacturer5)
db.create(Manufacturer1_Manufacturer2)
db.create(Manufacturer1_Manufacturer3)
db.create(Manufacturer1_Manufacturer4)
db.create(Manufacturer1_Manufacturer5)
db.create(Manufacturer2_Manufacturer3)
db.create(Manufacturer2_Manufacturer4)
db.create(Manufacturer2_Manufacturer5)
db.create(Manufacturer3_Manufacturer4)
db.create(Manufacturer3_Manufacturer5)
db.create(Manufacturer4_Manufacturer5)
```

In [7]:

```
car1['HP'] = 300
car1['price'] = 17998
car1['MPG'] = 25
car1['sale'] = 21232
car1['BrandDescrpition'] = 'Make your life better!'

car2['HP'] = 310
car2['price'] = 16799
car2['MPG'] = 23
car2['sale'] = 13229

car3['HP'] = 270
car3['price'] = 17000
car3['MPG'] = 30
car3['sale'] = 10213
car3['BrandDescrpition'] = 'Try you will know'

car4['HP'] = 260
car4['price'] = 25000
car4['MPG'] = 29
car4['sale'] = 7312
```

```

car5['HP'] = 330
car5['price'] = 35000
car5['MPG'] = 26
car5['sale'] = 9895
car5['sale'] = 9899
car5['BrandDescription'] = 'Toyota the better life'
car5['rate'] = 4.5

car6['HP'] = 350
car6['price'] = 11000
car6['MPG'] = 18
car6['sale'] = 15919
car6['type'] = 'sedan'

car7['HP'] = 220
car7['price'] = 12000
car7['MPG'] = 33
car7['sale'] = 33919
car7['rate'] = 4.8

car8['HP'] = 500
car8['price'] = 13000
car8['MPG'] = 15
car8['sale'] = 12332

car9['HP'] = 230
car9['price'] = 14000
car9['MPG'] = 29
car9['sale'] = 53123
car9['BrandDescription'] = '200 years making cars perfect'

car10['HP'] = 325
car10['price'] = 17000
car10['MPG'] = 22
car10['sale'] = 62132

car11['HP'] = 525
car11['price'] = 47000
car11['MPG'] = 20
car11['sale'] = 13423
car11['type'] = 'sedan'
car11['cylinder'] = 8

car12['HP'] = 425
car12['price'] = 67000
car12['MPG'] = 19
car12['sale'] = 32142
car12['type'] = 'sedan'

car13['HP'] = 178
car13['price'] = 33333
car13['MPG'] = 35
car13['sale'] = 19823
car13['type'] = 'suv'

db.push(car1)
db.push(car2)
db.push(car3)
db.push(car4)
db.push(car5)
db.push(car6)
db.push(car7)
db.push(car8)
db.push(car9)
db.push(car10)
db.push(car11)
db.push(car12)
db.push(car13)

```

In [8]:

```

Manufacturer1['Country'] = 'Germany'
Manufacturer1['CEO'] = 'AAAA'
Manufacturer1['Year'] = 1888

Manufacturer2['Country'] = 'Germany'
Manufacturer2['CEO'] = 'BBB'

```

```

Manufacturer2['Year'] = 1900

Manufacturer3['Country'] = 'U.S'
Manufacturer3['CEO'] = 'CCC'
Manufacturer3['Year'] = 1910

Manufacturer4['Country'] = 'Japan'
Manufacturer4['CEO'] = 'DDD'
Manufacturer4['Year'] = 1915

db.push(Manufacturer1)
db.push(Manufacturer2)
db.push(Manufacturer3)
db.push(Manufacturer4)

```

1.Which manufacturer has the highest quantity of models in this database?

In [19]:

```

def most_model_manufacturer(tx):
    Audi = [i for i in tx.run('match (m:model)-[:belong]->(M:Manufacturer{name:"Audi"}) return count(m),
    Honda = [i for i in tx.run('match (m:model)-[:belong]->(M:Manufacturer{name:"Honda"}) return count(m)
    Hongqi = [i for i in tx.run('match (m:model)-[:belong]->(M:Manufacturer{name:"Hongqi"}) return count(
    Dodge = [i for i in tx.run('match (m:model)-[:belong]->(M:Manufacturer{name:"Dodge"}) return count(m)
    BMW = [i for i in tx.run('match (m:model)-[:belong]->(M:Manufacturer{name:"BMW"}) return count(m), M.
    total = [Audi, Honda, Hongqi, Dodge, BMW]
    final = Audi
    for i in total:
        if i[0].get('count(m)') > final[0].get('count(m)'):
            final = i
        else:
            pass
    print(Audi)
    print(final[0].get('M.name'), 'is the manufacturer has the most of the models as',final[0].get('count

```

In [20]:

```

most_model_manufacturer(db)

[Record({'count(m)': 4, 'M.name': 'Audi'})]
Audi is the manufacturer has the most of the models as 4 models

```

2.Which model is the best deal one for each horsepower?

In [11]:

```

def hp_per_dollar(tx):
    print('Which model is the best deal one for each horsepower?')
    name = [i for i in tx.run('match (m:model) return m.name')]
    horsepower = [i for i in tx.run('match (m:model) return m.HP')]
    price = [i for i in tx.run('match (m:model) return m.price')]
    model = 0
    h_per_d = 10000
    for i in range(len(name)):
        h_per_d_new = price[i].get('m.price')/horsepower[i].get('m.HP')
        if h_per_d_new < h_per_d:
            h_per_d = h_per_d_new
            model = name[i].get('m.name')
        else:
            pass
    print('If you purchase', model, ", you got the best deal which is", h_per_d, "dollars per horsepower"

```

In [12]:

```

hp_per_dollar(db)

Which model is the best deal one for each horsepower?
If you purchase M2 , you got the best deal which is 26.0 dollars per horsepower

```

3.The average sales for each Germany model.

In [13]:

```

def count_german_models(tx):
    print('The average sales for each Germany model is: ')
    res_list = [res for res in tx.run('match (m:model)-[:located]->(:Country{country:"German"}) return m.
    total = 0
    for i in range(len(res_list)):
        total += res_list[i].get('m.sale')
    print(total/len(res_list))

```

In [14]:

```
count_german_models(db)
```

```
The average sales for each Germany model is:  
16372.833333333334
```

4. Which manufacturer have the highest sales?

In [15]:

```
def print_highest_Sales(tx):  
    Audi = [i for i in tx.run('match (m:model)-[:belong]->(M:Manufacturer{name:"Audi"}) return sum(m.sale  
    Honda = [i for i in tx.run('match (m:model)-[:belong]->(M:Manufacturer{name:"Honda"}) return sum(m.sa  
    Hongqi = [i for i in tx.run('match (m:model)-[:belong]->(M:Manufacturer{name:"Hongqi"}) return sum(m.  
    Dodge = [i for i in tx.run('match (m:model)-[:belong]->(M:Manufacturer{name:"Dodge"}) return sum(m.sa  
    BMW = [i for i in tx.run('match (m:model)-[:belong]->(M:Manufacturer{name:"BMW"}) return sum(m.sale),  
    total = [Audi, Honda, Hongqi, Dodge, BMW]  
    final = Audi  
    for i in total:  
        if i[0].get('sum(m.sale)') > final[0].get('sum(m.sale)'):  
            final = i  
        else:  
            pass  
    print(final[0].get('M.name'), 'is the manufacturer has the highest sales of all its models as', final[
```

In [16]:

```
print_highest_Sales(db)
```

```
Honda is the manufacturer has the highest sales of all its models as 128678 sales.
```

5. Compare to the horsepower, which model has the best performance on MPG?

In [17]:

```
def mpg_per_hp(tx):  
    print('Compare to the horsepower, which model has the best performance on MPG?')  
    name = [i for i in tx.run('match (m:model) return m.name')]  
    horsepower = [i for i in tx.run('match (m:model) return m.HP')]  
    mpg = [i for i in tx.run('match (m:model) return m.MPG')]  
    model = 0  
    m_per_h = 0  
    for i in range(len(name)):  
        m_per_h_new = horsepower[i].get('m.HP')/mpg[i].get('m.MPG')  
        if m_per_h_new > m_per_h:  
            m_per_h = m_per_h_new  
            model = name[i].get('m.name')  
        else:  
            pass  
    print('If you purchase', model, ", you got the best deal which is", m_per_h, "horsepower for Miles Pe
```

In [18]:

```
mpg_per_hp(db)
```

```
Compare to the horsepower, which model has the best performance on MPG?
```

```
If you purchase M2 , you got the best deal which is 33.33333333333336 horsepower for Miles Per Gallon
```

Conclusion and Recommendation

From our proceeding analysis, we could conclude that:

- Having 4 models, **Audi** is the manufacturer who has the most of the models;
- For the consumers who want to spend less money to get the most horsepower, we suggest them to purchase **BMW M2**;
- The average sales for the models from **German** is **17974**. Cross analyze this finding with our local brand sales could unearth valuable results;
- With disadvantage of variety of models, **Honda** still wins the competition of sales with total sales of **128678**.
- Compare to the horsepower of the vehicle, **BMW M2** is the best deal for 33 horsepower Mile Per Gallon.