به نام خدا

بررسی داده های فروش ماشین ها در ایالات مختلف آمریکا

در این پروژه به بررسی مجموعه داده های (USA_cars_datasets) خواهیم پرداخت، برای شروع داده های موجود در فایل CSV مربوط به داده های خودرو را با استفاده از کتابخانه "Pandas" و به کمک تکه کد زیر در دیتا فرم ذخیره می کنیم :

```
In [1]: import numpy as np
   ... : import pandas as pd

In [2]: data = pd.read_csv('USA_cars_datasets.csv')

In [3]:
```

```
import numpy as np
import pandas as pd
data = pd.read_csv('USA_cars_datasets.csv')
```

Feature	Type	ویژگی
Price	Integer	قیمت فروش خودرو در آگهی
Years	Integer	سال ثبت نام (ساخت) خودرو
Brand	String	نام تجاری خودرو
Model	String	مدل خودرو
Color	String	رنگ خودرو
State/City	String	محلی که در آن خودروبرای خرید در دسترس است
Mileage	Float	مسافت طی شده با خودرو بر حسب مایل
Vin	String	شماره شناسایی خودرو (مجموعه ای از ارقام و حروف بزرگ)
Title Status	String	دارای دو حالت که به صورت باینری (دارای بیمه یا بدون بیمه و تمیز)
Lot	Integer	تعداد زیادی از ارقام جهت شناسایی و تشخیص اصالت خودرو
Condition	String	زمان در خواست خرید تا تحویل خودرو

بعد از ذخیره داده های بعنوان دیتا فرم می توانیم به پاکسازی داده ها و بررسی داده ها بپردازیم، برای این منظور از دستورات زیر استفاده می کنیم :

```
In [3]: pd.set_option('display.max_columns', 999)

In [4]: print(data.head())
   Unnamed: 0  price      brand    model  year   title_status   mileage  \
0           0   6300     toyota  cruiser  2008  clean vehicle  274117.0
1           1   2899       ford       se  2011  clean vehicle  190552.0
2           2   5350      dodge      mpv  2018  clean vehicle   39590.0
3           3  25000       ford     door  2014  clean vehicle   64146.0
4           4  27700  chevrolet     1500  2018  clean vehicle    6654.0

    color                 vin        lot       state country      condition
0   black    jtezu11f88k007763  159348797  new jersey     usa   10 days left
1  silver    2fmdk3gc4bbb02217  166951262   tennessee     usa    6 days left
2  silver    3c4pdcgg5jt346413  167655728     georgia     usa    2 days left
3    blue    1ftfw1et4efc23745  167753855    virginia     usa  22 hours left
4     red    3gcpcrec2jg473991  167763266     florida     usa  22 hours left

In [5]:
In [5]: print(data.columns)
Index(['Unnamed: 0', 'price', 'brand', 'model', 'year', 'title_status',
       'mileage', 'color', 'vin', 'lot', 'state', 'country', 'condition'],
      dtype='object')

In [6]: print(data.info())
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2499 entries, 0 to 2498
Data columns (total 13 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Unnamed: 0    2499 non-null   int64
 1   price         2499 non-null   int64
 2   brand         2499 non-null   object
 3   model         2499 non-null   object
 4   year          2499 non-null   int64
 5   title_status  2499 non-null   object
 6   mileage       2499 non-null   float64
 7   color         2499 non-null   object
 8   vin           2499 non-null   object
 9   lot           2499 non-null   int64
 10  state         2499 non-null   object
 11  country       2499 non-null   object
 12  condition     2499 non-null   object
dtypes: float64(1), int64(4), object(8)
memory usage: 253.9+ KB
None
```

```
In [7]: print(data.describe())
        Unnamed: 0         price          year       mileage           lot
count  2499.000000   2499.000000  2499.000000  2.499000e+03  2.499000e+03
mean   1249.000000  18767.671469  2016.714286  5.229869e+04  1.676914e+08
std     721.543484  12116.094936     3.442656  5.970552e+04  2.038772e+05
min       0.000000      0.000000  1973.000000  0.000000e+00  1.593488e+08
25%     624.500000  10200.000000  2016.000000  2.146650e+04  1.676253e+08
50%    1249.000000  16900.000000  2018.000000  3.536500e+04  1.677451e+08
75%    1873.500000  25555.500000  2019.000000  6.347250e+04  1.677798e+08
max    2498.000000  84900.000000  2020.000000  1.017936e+06  1.678055e+08
```

```
pd.set_option('display.max_columns', 999)
print(data.head())
print(data.columns)
print(data.info())
print(data.describe())
```

با توجه به دستورات و نتایج به دست امده می توان دریافت که یک ستون به اسم "Unnamed: 0" در دیتا فرم وجود دارد که بدون نام بوده و حاوی اعداد از صفر تا 2498 می باشد، همچنین ستونهای " Lot " ," Vin " و "country" که حاوی اعداد بسیار بزرگ و متن usa هستند که فقط برای تشخیص کشور و اصیل بودن خودرو هست را می توان حذف کرد برای این کار یک متغیر به اسم " drop_columns " تعریف کرده و ستونهایی را که می خواهیم حذف کنیم را داخل آن ثبت می کنیم ، سپس با دستور "drop" ستونهای مورد نظر را حذف می کنیم :

```
In [8]: drop_columns = ['Unnamed: 0', 'vin', 'lot', 'country']
   ...: data.drop(drop_columns, axis = 1, inplace = True)
```

```
drop_columns = ['Unnamed: 0', 'vin', 'lot', 'country']
data.drop(drop_columns, axis = 1, inplace = True)
```

در ستون "Color" با استفاده از دستور زیر مشاهده می شود که دو مقدار " no_color " و ":color" وجود دارند :

```
data['color'].value_counts()
```

```
In [9]: data['color'].value_counts()
Out[9]:
white                                  707
black                                  516
gray                                   395
silver                                 300
red                                    192
blue                                   151
no_color                                61
green                                   24
orange                                  20
gold                                    19
charcoal                                18
brown                                   15
yellow                                   9
magnetic metallic                        6
beige                                    5
shadow black                             5
color:                                   5
ingot silver metallic                    4
oxford white                             4
triple yellow tri-coat                   3
super black                              3
billet silver metallic clearcoat         3
ruby red metallic tinted clearcoat       2
white platinum tri-coat metallic         2
```

برای رفع این مشکل از دستور "where"در کتابخانه "numpy" استفاده می کنیم و مقادیر مورد نظر را با رنگی که بیشترین تکرار را دارد جایگزین می کنیم و نتیجه را دوباره بازبینی می کنیم :

```
In [10]: data['color'] = np.where((data['color'] == 'no_color' ) |
    ... :                         (data['color'] == 'color:'), data['color'].mode(), data['color'])

In [11]: data['color'].value_counts()
Out[11]:
white                                  773
black                                  516
gray                                   395
silver                                 300
red                                    192
blue                                   151
green                                   24
orange                                  20
gold                                    19
charcoal                                18
brown                                   15
yellow                                   9
magnetic metallic                        6
shadow black                             5
beige                                    5
ingot silver metallic                    4
oxford white                             4
triple yellow tri-coat                   3
billet silver metallic clearcoat         3
super black                              3
tuxedo black metallic                    2
```

```
data['color'] = np.where((data['color'] == 'no_color' ) |
                                                (data['color'] == 'color:'),
data['color'].mode(), data['color'])
data['color'].value_counts()
```

فقط دو ستون دیگر باقیمانده یکی "price" که دارای مقادیر صفر و کمتر از 1000 می باشد و ستون "condition " که دارای مقادیر رشته ای با انواع مختلف روز و ساعت و دقیقه می باشد که باید یکپارچه سازی شوند (برای اینکه داده های حاصل اعداد بزرگی نباشند و راحتتر با انها کار کنیم بهتر است که فرمت ساعت و دقیقه را به روز تبدیل کنیم) برای این دو ستون داریم :

1- برای ستون "price" همانند ستون رنگ عمل می کنیم (برای پر کردن مقادیر مورد نظر از اعداد تصادفی مابین اعداد 1000-84000 استفاده می کنیم برای این کار ابتدا باید کتابخانه "random" را فراخوانی کنیم) :

```
In [13]: import random

In [14]: data['price'] = np.where(data['price'] < 1000, random.randrange(10000, 84000),
    ... :                         data['price'])

In [15]: data['price'].describe()
Out[15]:
count     2499.000000
mean     19736.004802
std      11644.014546
min       1000.000000
25%      11200.000000
50%      18000.000000
75%      26760.000000
max      84900.000000
Name: price, dtype: float64
```

```
import random
 data['price'] = np.where(data['price'] < 1000, random.randrange(10000,
84000),
                                                data['price'])
data['price'].describe()
```

2- برای اعمال تغییرات برای ستون "condition " از دستور "apply" از کتابخانه "Pandas" و تابع " condition" استفاده می کنیم :

```
In [16]: def condition(x):
    ... :     if x.split()[1] == 'hours':
    ... :         return int(x.split()[0]) / 24
    ... :     elif x.split()[1] == 'days':
    ... :         return int(x.split()[0])
    ... :     else :
    ... :         return 1 / 24

In [17]: data['condition'] = data['condition'].apply(condition)
```

```
def condition(x):
    if x.split()[1] == 'hours':
        return int(x.split()[0]) / 24
    elif x.split()[1] == 'days':
        return int(x.split()[0])
    else :
        return 1 / 24
data['condition'] = data['condition'].apply(condition)
```

بخش انالیز داده ها

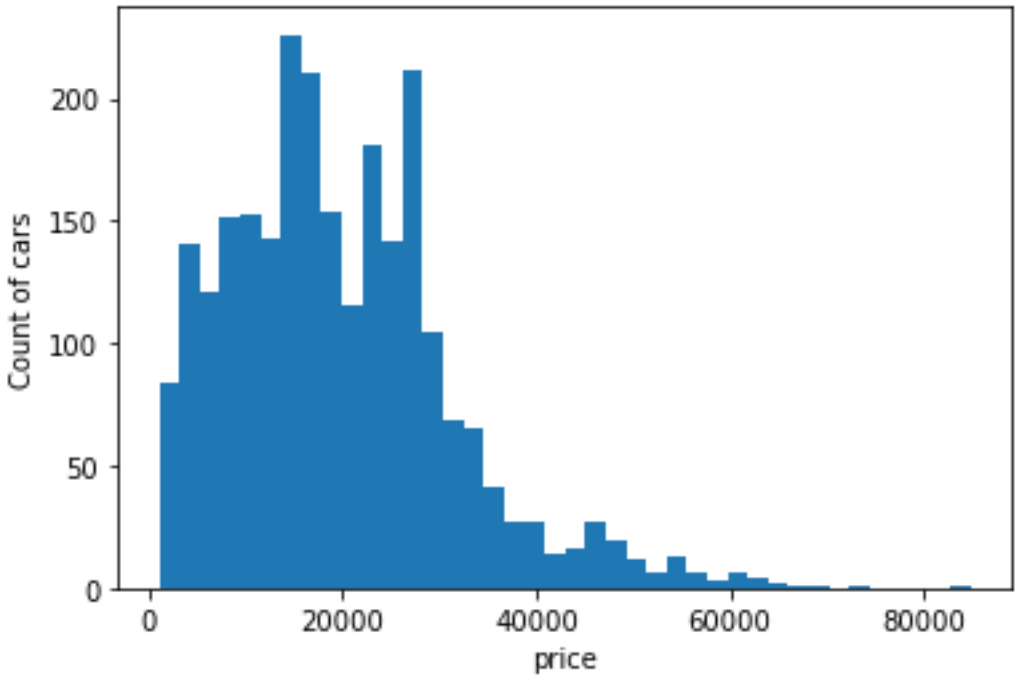
تا اینجا پاکسازی و یکپارچه سازی داده ها را انجام دادیم حال می رسیم به بخش پیش پردازش و امار توصیفی که در درک انواع داده ها بسیار موثر هستند.

حال به بررسی رنج قیمتی برای خودرو ها می پردازیم با رسم یک نمودار هیستوگرام اطلاعات مفیدی می توان به دست آورد :

```
In [20]: import matplotlib.pyplot as plt

In [21]: plt.hist(data['price'], bins = 40)
    ... : plt.xlabel('price')
    ... : plt.ylabel('Count of cars')
Out[21]: Text(0, 0.5, 'Count of cars')
```

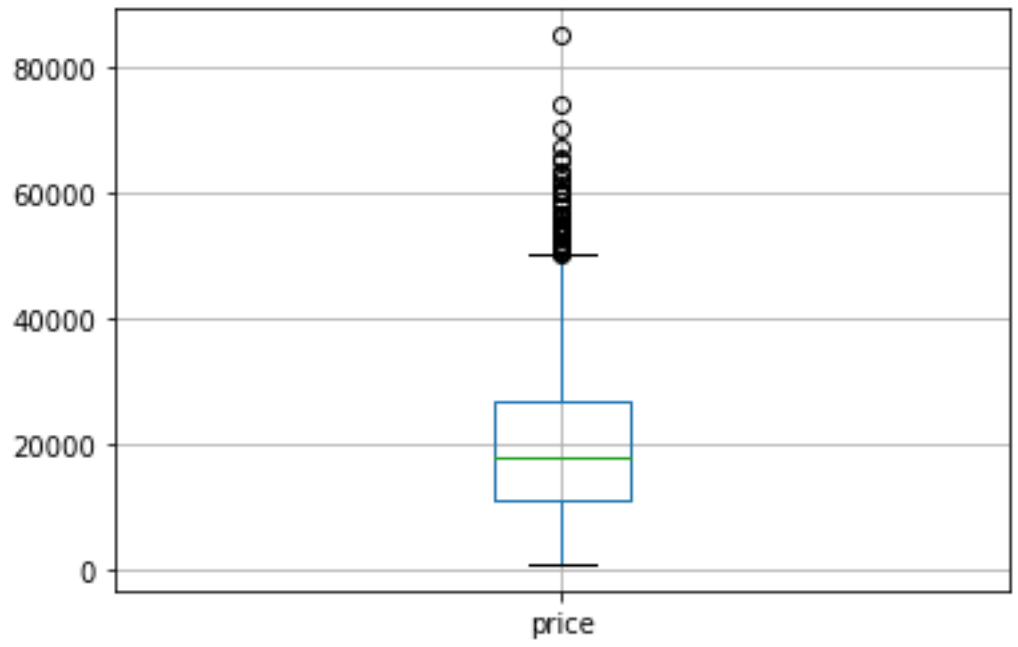
```
import matplotlib.pyplot as plt
plt.hist(data['price'], bins = 40)
plt.xlabel('price')
plt.ylabel('Count of cars')
```



برای درک بهتر از نمودار جعبه ای استفاده می کنیم :

```
In [22]: data.boxplot(column ='price')
Out[22]: <AxesSubplot:>
```

```
data.boxplot(column ='price')
```



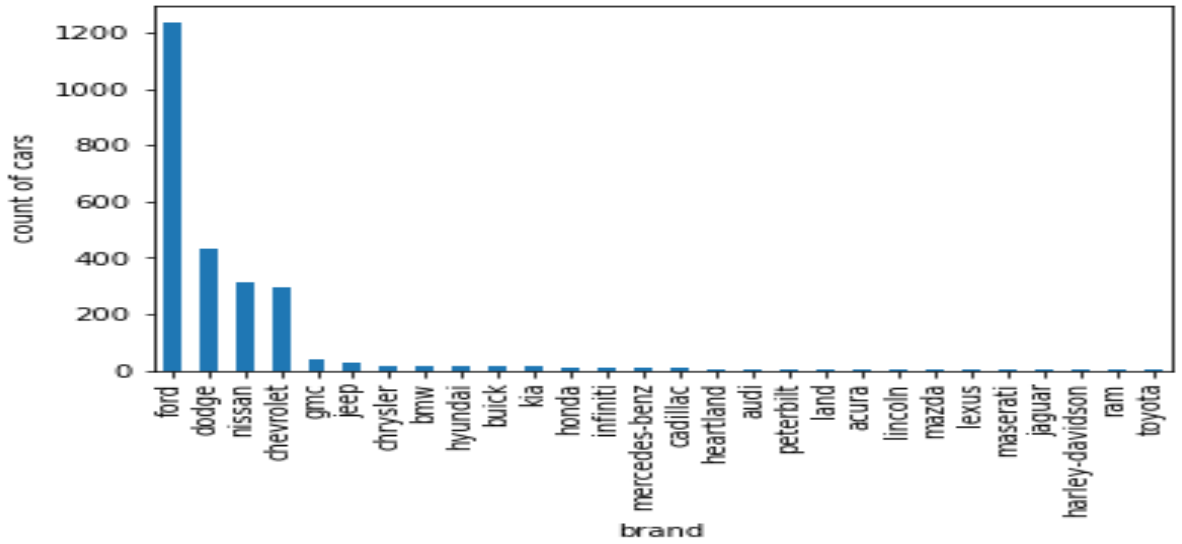
این ناهماهنگی بین داده ها به دلایل متفاوتی از جمله سال تولید، مسافت طی شده، برند، وضعیت بیمه خودرو بستگی دارد.

در بحث نام تجاری خودرو ها نیز می توان به اطلاعات خوبی با استفاده از نمودار میله ای دست یافت، برای سادگی ابتدا ستونی را ایجاد می کنیم که تمام رکوردهای ان 1 باشد و نام ستون مورد نظر را "count" میگذاریم :

```
In [23]: data['count'] = np.where(True, 1, 1)

In [24]: data_brand = data.groupby('brand').count()
    ...: data_brand.sort_values('count', ascending = False, inplace = True)
    ...: data_brand['count'].plot.bar()
    ...: plt.ylabel('count of cars')
Out[24]: Text(0, 0.5, 'count of cars')
```

```
data['count'] = np.where(True, 1, 1)
data_brand = data.groupby('brand').count()
data_brand.sort_values('count', ascending = False, inplace = True)
data_brand['count'].plot.bar()
plt.ylabel('count of cars')
```

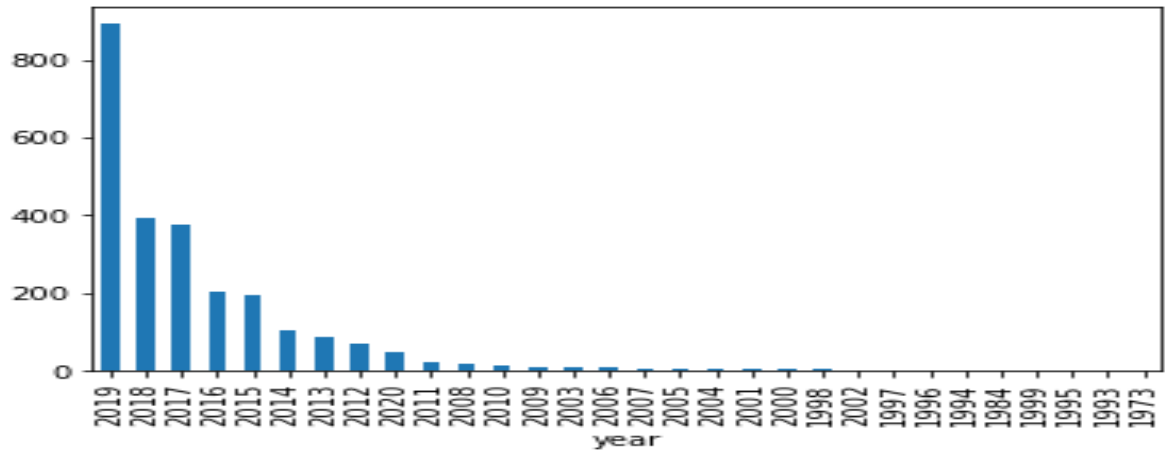


همان طور که از نمودار مشهود است خودروهای با برندهای "ford", "dodge", "nissan", "chevrolet" بیشتراز برندهای دیگر به فروش رفته اند.

همانند برندها برای سالها و ایالات مختلف نیز داریم :

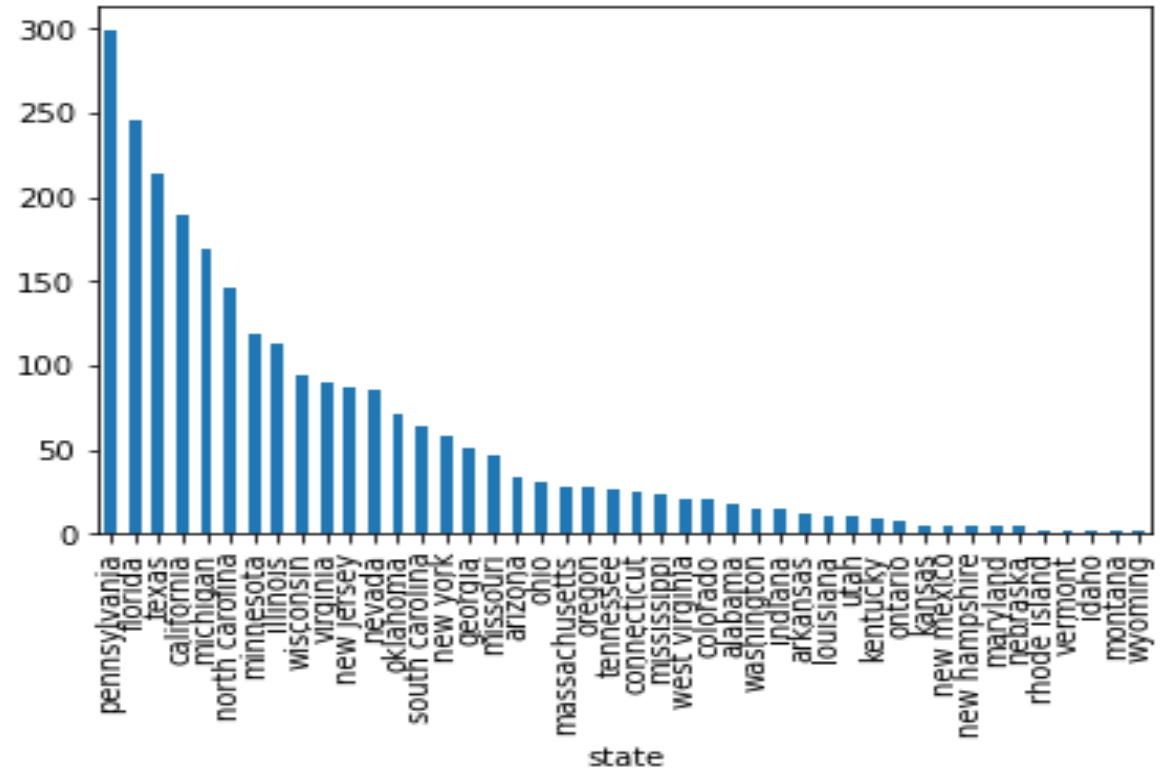
```
In [25]: data_year = data.groupby('year').count()
    ...: data_year.sort_values('count', ascending = False, inplace = True)
    ...: data_year['count'].plot.bar()
Out[25]: <AxesSubplot:xlabel='year'>
```

```
data_year = data.groupby('year').count()
data_year.sort_values('count', ascending = False, inplace = True)
data_year['count'].plot.bar()
```




```
In [26]: data_state = data.groupby('state').count()
    ... : data_state.sort_values('count', ascending = False, inplace = True)
    ... : data_state['count'].plot.bar()
Out[26]: <AxesSubplot:xlabel='state'>
```

```
data_state = data.groupby('state').count()
data_state.sort_values('count', ascending = False, inplace = True)
data_state['count'].plot.bar()
```



اکنون می خواهیم تاثیر بیمه بودن یا نبودن خودرو بر تعداد فروش و همینطور سال تولید خودرو را بررسی کنیم برای انجام این مقایسه ابتدا ستونی با نام "insured" را با شرط اینکه اگر خودرو بیمه باشد مقدار این ستون 1 و درغیر اینصورت عدد 0 قرار گیرد برای این منظور از دستور زیر استفاده می کنیم :

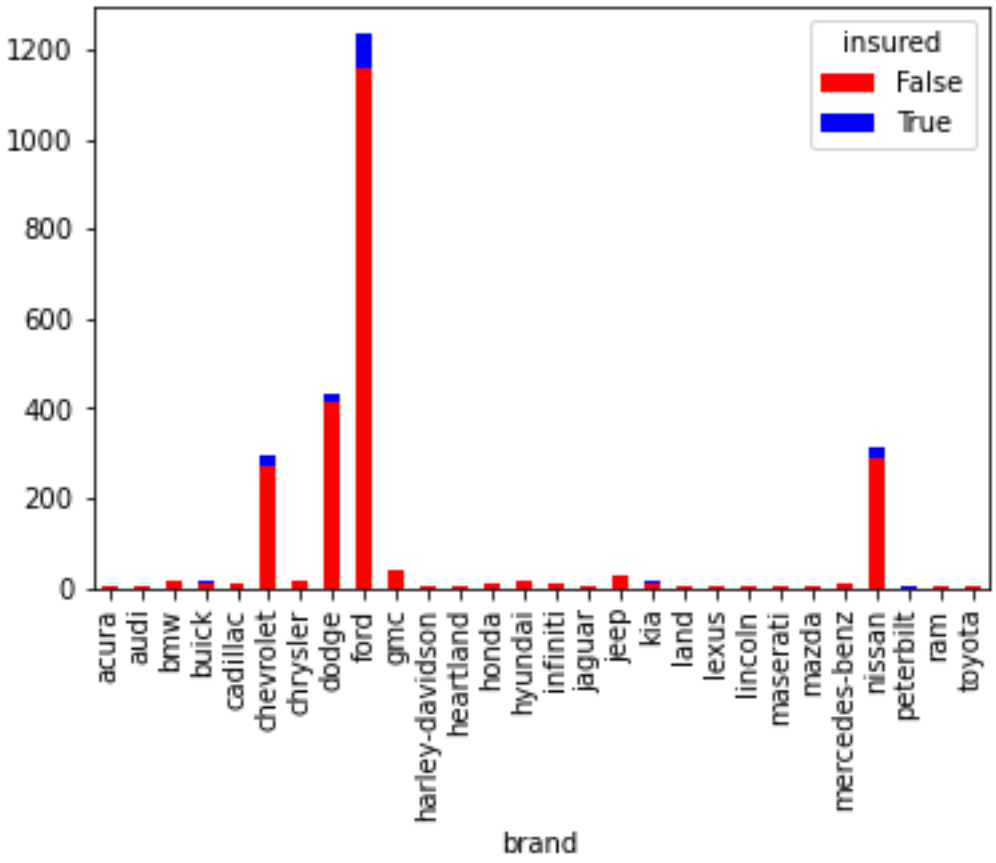
```
In [27]: data['insured']= np.where(data['title_status'] == 'clean vehicle', 0, 1)
```

```
data['insured']= np.where(data['title_status'] == 'clean vehicle', 0, 1)
```

حال با استفاده از تابع "crosstab" در کتابخانه پاندا می توانیم به اهداف خود برسیم :

```
In [28]: brand_ = pd.crosstab(data.brand, data.insured.astype(bool))
    ...: brand_.plot(kind = 'bar', stacked = True, grid = False, color = ['red', 'blue'])
Out[28]: <AxesSubplot:xlabel='brand'>
```

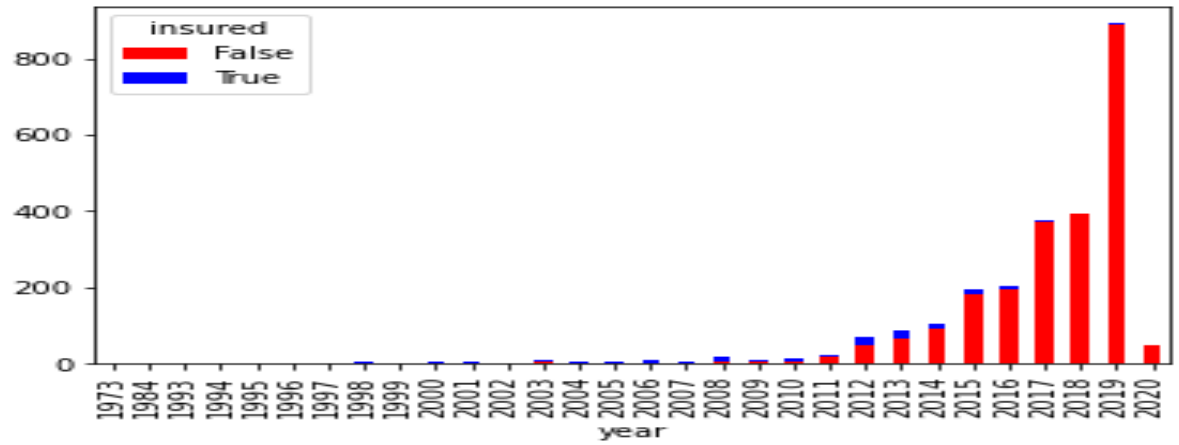
```
brand_ = pd.crosstab(data.brand, data.insured.astype(bool))
brand_.plot(kind = 'bar', stacked = True, grid = False, color = ['red',
'blue'])
```



با مشاهده نمودار بالا به اطلاعات جالبی مبنی بر اینکه چهار برندی که دارای بیشترین فروش بودند بیشترین تعداد خودرو های بیمه شده نیز مربوط به همین چهار برند است .

```
In [29]: year_ = pd.crosstab(data.year, data.insured.astype(bool))
    ...: year_.plot(kind = 'bar', stacked = True, grid = False, color = ['red', 'blue'])
Out[29]: <AxesSubplot:xlabel='year'>
```

```
year_ = pd.crosstab(data.year, data.insured.astype(bool))
year_.plot(kind = 'bar', stacked = True, grid = False, color = ['red',
'blue'])
```

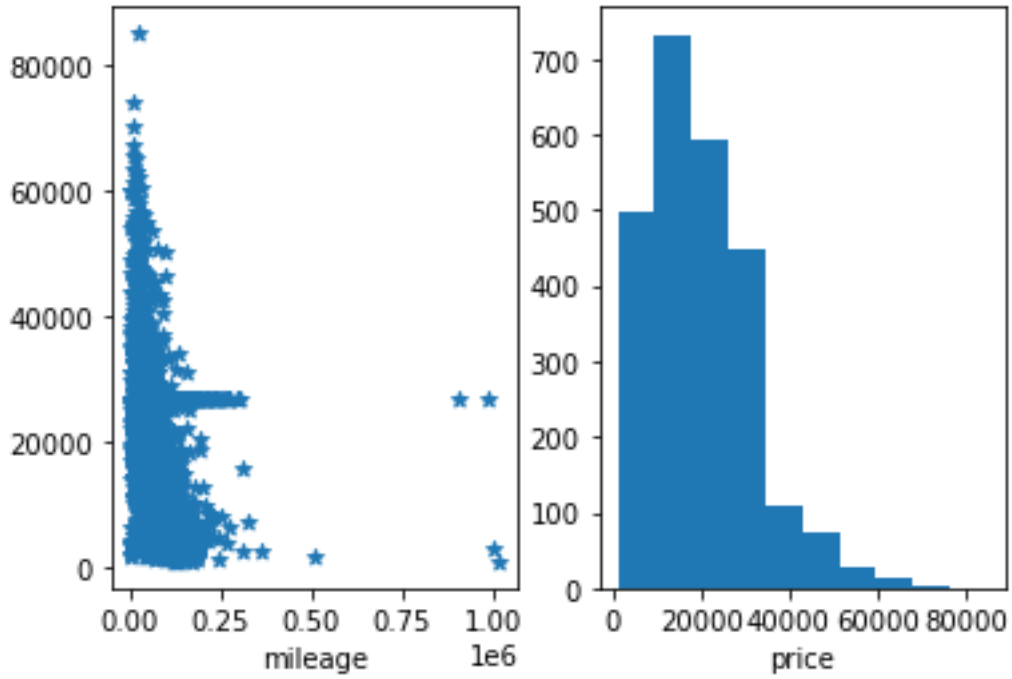


با توجه به نمودار بالا مشخص میشود که اکثرا خودرو های با سال تولید بین 2015 تا 2000 دارای بیمه هستند.

موضوع دیگر تاثیر میزان مسافت طی شده در قیمت خودرو هست :

```
In [30]: fig = plt.figure()
    ... : g1 = fig.add_subplot(121)
    ... : g1.scatter(data.mileage, data.price, marker = '*')
    ... : g1.set_xlabel('mileage')
    ... : g2 = fig.add_subplot(122)
    ... : g2.hist(data.price)
    ... : g2.set_xlabel('price')
Out[30]: Text(0.5, 0, 'price')
```

```
fig = plt.figure()
g1 = fig.add_subplot(121)
g1.scatter(data.mileage, data.price, marker = '*')
g1.set_xlabel('mileage')
g2 = fig.add_subplot(122)
g2.hist(data.price)
g2.set_xlabel('price')
```



بنابراین هر چقدر میزان مسافت طی شده توسط خودرو بیشتر با شد آن خودرو دارای قیمت کمتری نسبت به خودرو های دیگر می باشد. (چند تا داده پرت که در نمودار پراکندگی مشاهده می شود به احتمال زیاد به دلیل جایگذاری اعداد تصادفی بین 1000 و 84000 به جای قیمت های کمتر از 1000 می باشد)