



دانشگاه فنی مهندسی دکتر شریعتی

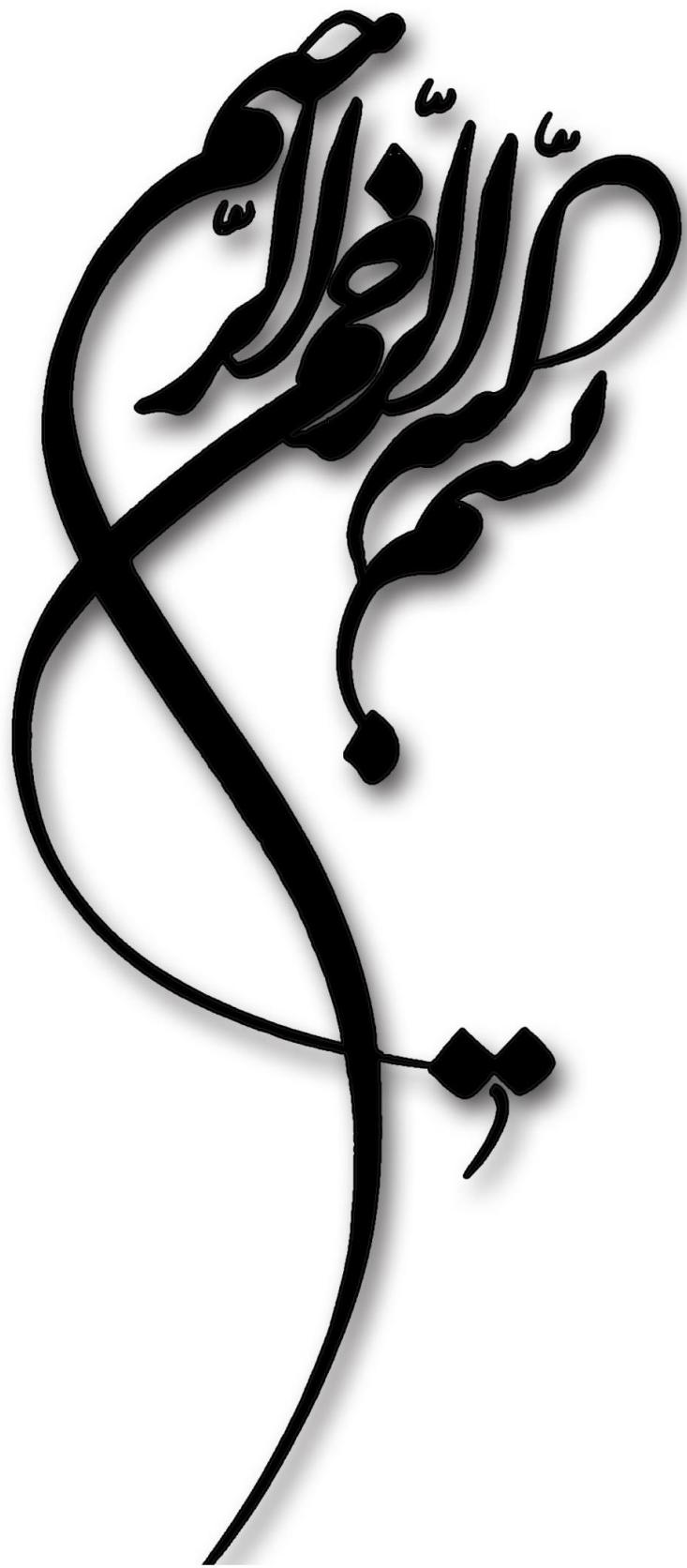
## تحلیل و مصور سازی داده های سایت airbnb وارائه مدل پیش بینی کننده

پایاننامه برای دریافت درجه کارشناسی  
در رشته مهندسی کامپیوتر گرایش نرم افزار

دانشجو:  
مهناز دیوارگر

استاد راهنما:  
سرکار خانم مهندس زمانزاده

زمان:  
نیمسال اول 1398-1399





## تقدیم به :

خدای را بسی شاکرم که از روی کرم پدر و مادری فداکار نصیبم ساخته ، تا در سایه درخت پر بار وجودشان بیاسایم و از ریشه آنها شاخ و برگ گیرم و در سایه وجودشان در راه کسب علم و دانش تلاش نمایم .

والدینی که بودنشان تاج افتخاری است بر سرم و نامشان دلیلی است بر بودنم چرا که این دو وجود پس از پروردگار مایه هستی ام بوده اند، دستم را گرفتند و راه رفتن را در این وادی پر فراز و نشیب به من آموختند.

آموزگارانی که برایم زندگی؛ بودن و انسان بودن را معنا کردند، حال این برگ سبزی است تحفه درویش تقدیم به پدر و مادر عزیزم ...

به پاس عاطفه سرشار و گرمای امیدبخش وجودشان که در این سردرین روزگاران بهترین پشتیبان است و به پاس قلب های بزرگ شان که فریادرس است و سرگردانی و ترس در پناهشان به شجاعت می گراید و به پاس محبت های بی دریغشان که هرگز فروکش نمی کند.

## سپاسگزاری

پروردگارا به پیشگاه پاک و مقدس سر تعظیم فرود می آورم ، که بندگی فقط و فقط تورا سزد . آنچه داده ای بیش از شایستگی من است ، گرچه در خور بخشنده توست؛ پروردگارا سپاس میگوییم که بر من منت نهادی و خلعت تحصیل بر من پوشاندی؛ چه زیباست ستایش خالق ، او که زندگی می کنیم برای وصلش در حالی که تقدیر از مخلوق جذبه ای از ستایش خالق است ، برخود وظیفه میدانم تا از استاد بزرگوارم که صبورانه و دلسوزانه در راستای انجام این پروژه مرا یاری کردند؛ تشکر و قدر دانی نمایم .

چرا که اگر یاری ایشان نبود ، امروز این تلاش به پایان نمی رسید . بنابراین از استاد محترم سرکار خانم مهندس زمانزاده که در مراحل مختلف این پژوهش ، صبورانه و مشتاقانه مرا راهنمایی کردند کمال قدردانی و تشکر را دارم .

از استاد گرانقدر جناب آقای دکتر عدل دوست که زحمت داوری این پروژه را بر عهده داشتند و با دقت بسیار به مطالعه این پژوهش پرداختند تشکر و قدر دانی میکنم .

از خداوند بزرگ برای تمامی این بزرگواران اجری عظیم خواستارم .

## چکیده

مستند حاضر گزارشی است مبتنی بر تحلیل و آنالیز dataset وبسایت Airbnb؛ وبسایتی که مردم در آن مکان های اقامتی را کرایه می دهند؛ با استفاده از نمودار ها و جداول گوناگون. در این زمینه از زبان python و کتابخانه های بسیار کارآمد و پرقدرت آن؛ که برای تحلیل و مصور سازی طراحی شده اند؛ مانند Numpy , Pandas , seaborn , plotly , folium استفاده شده تا بتوانم تحلیل جامع و دقیقی ارائه کنم.

پس از بررسی دقیق اطلاعات موجود در dataset به ارائه دو مدل پیشگویی برای پیشنهاد قیمت به کاربرانی که قصد ثبت مکان اقامتی جدید را دارند میپردازیم؛ که در این زمینه از کتابخانه های sklearn و scipy استفاده شده است .

واژه های کلیدی : نمودار ، python ، تجزیه و تحلیل ، رگرسیون ، مکان های اقامتی ، dataset .

# فهرست مطالب

## فصل 1 : مقدمه

10 .....	1-1-انگیزه
10 .....	2-1-هدف
11 .....	3-1-رئوس مطالب سایر فصل ها

## فصل 2 : ابزارها

13 .....	1-2-معرفی python
13 .....	2-2-کتابخانه Numpy
14 .....	3-2-کتابخانه Pandas
14 .....	4-2-کتابخانه matplotlib
14 .....	5-2-کتابخانه seaborn
15 .....	6-2-کتابخانه scipy
15 .....	7-2-کتابخانه sklearn

## فصل 3 : تجزیه و تحلیل

18 .....	1-3-معرفی dataset و فیلد های
18 .....	2-3-بررسی dataset در پروژه
20 .....	3-3-پیش پردازش داده ها (1)
23 .....	3-4-نقشه پراکندگی برای مکان های اقامتی و مکان های پر طرفدار
28 .....	5-3-بررسی ویژگی last_review
30 .....	6-3-بررسی ویژگی neighbourhood_group
32 .....	7-3-بررسی ویژگی neighbourhood

33	.....neighbourhood_group ارتباط بین و neighbourhood	8-3
40	.....room_type بررسی ویژگی	9-3
43	.....room_type و price بررسی ویژگیهای	10-3
44	.....availability_365 بررسی ویژگی	11-3
46	.....minimum_night بررسی ویژگی	12-3
47	..... ها ها ارتباط بین ویژگی ها	13-3
49	.....صاحبخانه های پر مشغله	14-3
51	.....لغت های پرتکرار در معرفی مکان های اقامتی	15-3
52	.....price تحلیل ویژگی	16-3

#### **فصل 4: پیشگویی قیمت بالگوریتم های یادگیری ماشین**

61	.....پیش پردازش داده ها (2)	1-4
62	.....کد گذاری برخی از ویژگی ها	1-1-4
63	.....price نرمال کردن ویژگی	2-1-4
66	.....بررسی دو الگوریتم رگرسیون	2-4
66	.....Regression بررسی الگوریتم	1-2-4
67	.....Linear Regression بررسی الگوریتم	2-2-4
71	.....Gradient Boosted Regressor بررسی الگوریتم	3-2-4
78	.....پیاده سازی الگوریتم های پیش بینی قیمت	3-4
79	.....Linear Regression پیاده سازی الگوریتم	1-3-4
82	.....Gradient Boosted Regressor پیاده سازی الگوریتم	2-3-4

## **فصل 5: جمع بندی و پیشنهاد ها**

85 .....	1-5-مقدمه
85 .....	2-پیشنهاد هایی برای کارهای آتی
86 .....	مراجع
88 .....	پیوست ها

فصل 1 :

## مقدمه

## 1-1-انگیزه

سایت Kaggle از جمله سایت های معتبر و پر کاربرد در زمینه داده کاوی و علم داده بوده و هست. کسب و کارها مشکلات داده ای خود را در این سایت مطرح کرده و تحلیل گران داده برای ارائه بهترین راه حل ها با یکدیگر رقابت می کنند. در جریان جست و جو ها با چالشی مواجه شدم که تصمیم گرفتم آن را به عنوان موضوع پژوهه کارشناسی برگزینم.

در این چالش وب سایت airbnb از تحلیل گران خواسته است با توجه به ارائه شده موضوعات زیر را بررسی کنند:

- ما چه مواردی را میتوانیم از این dataset متوجه شویم؟
- کدام میزبان ها پر مشغله تر هستند و چرا؟
- آیا تفاوت قابل ملاحظه ای در مورد ترافیک بین مناطق وجود دارد و دلیل این امر چیست؟
- یک پیش بینی در مورد قیمت که بتوان آن را به عنوان قیمت پیشنهادی به کسی که قصد افزودن مکانی را دارد ارائه داد.

## 1-2-هدف

هدف از انجام این پژوهش کشف روابط و مفاهیم مستتر در دادهها با استفاده از نمودارها و جداول مختلف و انجام چالش فوق که نمونه ای کوچک از دنیای بزرگ تحلیل داده می باشد است؛ و همچنین پیدا کردن قیمت با استفاده از الگوریتم های رگرسیون که الگوریتمی قوی برای پیش بینی داده های پیوسته میباشد؛ است.

## 1-3-رؤوس مطالب سایر فصل ها

در فصل دوم به بررسی زبان برنامه نویسی و کتابخانه های به کار رفته در این پژوهه میپردازیم. در فصل سوم به توضیح دیتا است سایت airbnb و فیلد های آن می پردازیم و در ادامه تحلیل جامعی بر داده های آن خواهیم داشت. در فصل چهارم به بررسی مدل های Gradient و LinearRegression

Boosted Regressor از الگوریتم های یادگیری ماشین می پردازم و با استفاده از این دو الگوریتم قیمت را پیش بینی می کنیم و در نهایت نتیجه‌گیری حاصل شده ، فصل پنجم را دربر میگیرد .

## فصل 2: ابزارها

در این بخش به بررسی زبان مورد استفاده و کتابخانه های اصلی استفاده شده در این پروژه می پردازیم.

## 2-1-معرفی python

پایتون یک زبان مفسری، شی گرا و سطح بالاست که در برنامه نویسی وب، اسکریپت نویسی، هوش مصنوعی و تحلیل داده ها و داده کاوی کاربرد وسیعی دارد و به دلیل خوانایی بالا، محبوب واقع شده است. این زبان با استفاده از زبان C ایجاد شده است و یکی از مزیت های برجسته این زبان کتابخانه ها و ماثول های فراوان و قدرتمند آن است که توان برنامه نویس در حوزه مربوطه را افزایش می دهد.

این زبان در دو ورژن (ورژن 2 و ورژن 3) ارائه شده است که ما در این پروژه از ورژن 3 استفاده کرده ایم.

## 2-2-کتابخانه Numpy

Numpy یک بسته منبع باز (open source) پایتون برای محاسبات علمی است. Numpy، از آرایه ها و ماتریس های بزرگ و چند بعدی پشتیبانی می کند. این بسته به زبان C و python نوشته شده است. آرایه ها در numpy در مقایسه با لیست های پایتون سریعتر هستند. (زبان پایتون پشتیبانی داخلی برای آرایه ندارد)

## 3-2-کتابخانه pandas

یکی از کتابخانه های مهم علم داده به حساب می آید؛ این کتابخانه برای دستکاری و تجزیه و تحلیل داده ها نوشته شده است و به طور خاص ساختار داده و عملیات رابرای دستکاری جداول و سری های زمانی ارائه می دهد. یکی از کاربردهای اصلی این کتابخانه ارائه اشیا قدرتمندی مانند

است که کار با داده و تجزیه و تحلیل آن را اسان می کند. در حالی که `numpy` عمده برای کار با داده های عددی به کار می رود.

## 4-کتابخانه `matplotlib`

یک کتابخانه ترسیم برای زبان برنامه نویسی پایتون است و برای ترجمه داده های پیچیده به بینش قابل هضم برای مخاطب مورد استفاده قرار می گیرد. با استفاده از این کتابخانه شما می توانید فقط با چند خط کد انواع نمودار را رسم کنید. ما در قسمت های مختلف برنامه از این کتابخانه جهت مصور سازی و آشنا شدن با رفتار `Basic` داده ها استفاده کرده ایم.

## 5-کتابخانه `seaborn`

این کتابخانه نیز برای تجسم داده های آماری به کار می رود و رابط سطح بالایی را برای ترسیم گرافیک های آماری جذاب و آموزنده فراهم می کند. در حالی که این کتابخانه متفاوت از `matplotlib` است، می توان از آن برای توسعه جذابیت گرافیک `matplotlib` استفاده کرد.

## 6-کتابخانه `SciPy`

این کتابخانه از آرایه های `numpy` به عنوان ساختار پایه داده استفاده می کند و ماثول هایی را برای کارهای مختلف که معمولا در برنامه نویسی علمی استفاده می شوند مثل جبر خطی، حل معادله دیفرانسیل معمولی و پردازش سیگнал طراحی می کند.

## 7-کتابخانه `sklearn`

کتابخانه `Scikit-learn` که به اختصار `sklearn` نوشته می شود یک کتابخانه رایگان زبان پایتون برای یادگیری ماشینی است. این کتابخانه دارای الگوریتم های مختلف برای پردازش داده ها، کاهش ابعاد، نرمال سازی،

طبقه بندی، رگرسیون و خوش بندی و ... است و برمبنای کتابخانه هایی مانند `matplotlib`، `pandas`، `numpy`، `SciPy` است.

فصل 3 :

## تجزیه و تحلیل

### 1-3-معرفی dataset و فیلدهای آن

این dataset شامل 16 ستون (features) و 48895 سطر (record) می باشد. هر سطر یک مشاهده تلقی می شود که حاوی اطلاعات مکانی است که یک صاحب خانه برای اجاره دادن خانه اش در New York City در سایت ثبت کرده است؛ و هر ستون یک ویژگی خاص از مکان ها را نشان میدهد که این ویژگی ها به شرح ذیل می باشند:

جدول (1-3) شرح ستون ها

ردیف	نام ستون	شرح ویژگی
1	Id	شناسه یکتای هر مکان موجود در dataset
2	name	نامی که هر مکان در dataset با آن معرفی شده است این نام به صورت توضیح خیلی مختصر از ویژگی های شاخص آن مکان است که صاحب خانه آن را تعریف میکند.
3	host_id	شناسه یکتا برای هر صاحب خانه
4	Host_name	نام هر صاحب خانه
5	neighbourhood_group	New York City شهر های موجود در
6	neighbourhood	محله های موجود در هر شهر
7	latitude	عرض جغرافیایی مکان
8	longitude	طول جغرافیایی مکان
9	room_type	نوع اتاقهایی که برای اجاره موجود اند؛ که صاحب خانه باید آن را از بین سه مقدار مشخص ( Private room , Shared room ) انتخاب کند ( Entire home/apt ,
10	price	قیمت خانه به دلار
11	minimum_nights	حداقل شبی که صاحب خانه برای اجاره مکان در نظر گرفته است.
12	number_of_reviews	تعداد بررسی هایی که گردشگران روی هر مکان انجام داده اند این بررسی ها می توانند شامل کامنت ها و امتیاز دهی

و... باشد. که پس از اقامت در مکان توسط گردشگر داده می شود.		
تاریخ آخرین بازدیدی که از این خانه شده است.	last_reviews	13
تعداد بررسی ها در هر ماه	reviews_per_month	14
تعداد خانه هایی که این صاحب خانه در کل dataset دارد	Calculated_host_listing_count	15
تعداد روزهایی از سال که آن خانه در دسترس است.	Availability_365	16

حال به بررسی های بیشتر برای تفهیم dataset می پردازیم.

## 2-3-بررسی dataset در پروژه

در کد زیر با استفاده از کتابخانه dataset، pandas به محیط برنامه وارد شده و میتوانیم 5 سطر ابتدایی این dataset را مشاهده کنیم.

### Read Dataset

```
In [3]: #using pandas library and 'read_csv' function to read csv file
airbnb=pd.read_csv('C:\\Users\\PCROOM\\Documents\\python data sets\\AB_NYC_2019.csv')
airbnb.head()
```

Out[3]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149	1	
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225	1	
2	3647	THE VILLAGE OF HARLEM...NEW YORK!	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Private room	150	3	
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89	1	
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80	10	

شکل (1-3) کد مربوط به وارد کردن dataset

## Read Dataset

id	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	last_review	reviews_per_month	calculated_host_listings_count	availability_365
ton	40.64749	-73.97237	Private room	149	1	9	2018-10-19	0.21	6	365
wn	40.75362	-73.98377	Entire home/apt	225	1	45	2019-05-21	0.38	2	355
em	40.80902	-73.94190	Private room	150	3	0	NaN	NaN	1	365
Hill	40.68514	-73.95976	Entire home/apt	89	1	270	2019-07-05	4.64	1	194
em	40.79851	-73.94399	Entire home/apt	80	10	9	2018-11-19	0.10	1	0
...										

شکل (2-3) کد مربوط به وارد کردن dataset

و همین طور ابعاد dataset با قطعه کد زیر قابل مشاهده است:

## shape of dataset

```
In [3]: print('Number of features: {}'.format(airbnb.shape[1]))
print('Number of Records: {}'.format(airbnb.shape[0]))
```

Number of features: 16  
Number of Records: 48895

شکل (3-3) کد مربوط به ابعاد dataset

هر یک از این ستون‌ها دارای یک type هستند که با قطعه کد زیر میتوان هر ستون را دریافت البته باید توجه داشت که ستون type در قطعه کد زیر، type هر ستون را نشان می‌دهد ولی ستون type\_of\_each\_item هر عنصر object در ستون مربوطه را نشان می‌دهد؛ مثلاً ستون name دارای type string بوده ولی هر عنصر از این ستون دارای type هستند.

```
#checking type of every coloms in the dataset
type_of_each_item=[type(airbnb.id[0]),type(airbnb.name[0]),type(airbnb.host_id[0]),type(airbnb.host_name[0]),type(airbnb.neighbourhood_group[0]),type(airbnb.neighbourhood[0]),type(airbnb.latitude[0]),type(airbnb.longitude[0]),type(airbnb.room_type[0]),type(airbnb.price[0]),type(airbnb.minimum_nights[0]),type(airbnb.number_of_reviews[0]),type(airbnb.last_review[0]),type(airbnb.reviews_per_month[0]),type(airbnb.calculated_host_listings_count[0]),type(airbnb.availability_365[0])]
type_of_each_item=pd.DataFrame(type_of_each_item ,columns=['type_of_each_item'],index=airbnb.columns)
pd.DataFrame(airbnb.dtypes ,columns=['type']).join(type_of_each_item)
```

	type	type_of_each_item
id	int64	<class 'numpy.int64'>
name	object	<class 'str'>
host_id	int64	<class 'numpy.int64'>
host_name	object	<class 'str'>
neighbourhood_group	object	<class 'str'>
neighbourhood	object	<class 'str'>
latitude	float64	<class 'numpy.float64'>
longitude	float64	<class 'numpy.float64'>
room_type	object	<class 'str'>
price	int64	<class 'numpy.int64'>
minimum_nights	int64	<class 'numpy.int64'>
number_of_reviews	int64	<class 'numpy.int64'>
last_review	object	<class 'str'>
reviews_per_month	float64	<class 'numpy.float64'>
calculated_host_listings_count	int64	<class 'numpy.int64'>
availability_365	int64	<class 'numpy.int64'>

شکل (4-3) کد مربوط به type در dataset

### 3-3-پیش پردازش داده ها (1)

یکی از موار مهم و قابل توجه در داده کاوی و علم تحلیل داده مقادیر گمشده (missing values) هستند؛ یعنی زمانی که در یک سطر مقداری برای یک یا چند ویژگی وجود ندارد؛ مثلا در شکل (2-3) مقدار ویژگی last\_review در سطر سوم کم شده و برابر None است؛ و این مقادیر در نهایت تاثیر ناخواهاندی بر مدل پیش بینی دارند پس باید این مقادیر را در dataset پیدا کرده با الگوریتم های مختلف اثر سوء آنها را از بین برد:

## what are missing\_data

```
: missing_data=airbnb.isnull().sum().sort_values(ascending=False)
missing_data = pd.DataFrame(missing_data,columns=['count of missing value'])
missing_data
```

	count of missing value
reviews_per_month	10052
last_review	10052
host_name	21
name	16
availability_365	0
calculated_host_listings_count	0
number_of_reviews	0
minimum_nights	0
price	0
room_type	0
longitude	0
latitude	0
neighbourhood	0
neighbourhood_group	0
host_id	0
id	0

شکل (5-3) کد مربوط به مقادیر گمشده در dataset

همانطور که مشاهده میشود reviews\_per\_month و last\_review و reviews\_per\_month دارای مقادیر گمشده هستند. تحلیلی داده به کشف روابط مرتبط و مستتر در داده ها می پردازد ولی برخی از ویژگی های موجود در dataset هیچ کمکی به این هدف نمی کنند از جمله id که برای هر مکان یکتا است و اطلاعات ویژه ای از آن استنباط نمی شود. همینطور ویژگی host\_name در حالی که ما شناسه هر صاحب خانه را داریم . پس می توان آنها را حذف کرد.

## Delete some not significant column

```
# proceed with removing columns that are not important and handling of missing data.
#dropping columns that are not significant or could be unethical to use for our future data exploration and predictions
airbnb.drop(['id','host_name'], axis=1, inplace=True)
#test
airbnb.columns

Index(['name', 'host_id', 'neighbourhood_group', 'neighbourhood', 'latitude',
       'longitude', 'room_type', 'price', 'minimum_nights',
       'number_of_reviews', 'last_review', 'reviews_per_month',
       'calculated_host_listings_count', 'availability_365'],
      dtype='object')
```

شکل (6-3) کد مربوط حذف ویژگی های اضافه

با حذف این دو ویژگی سه ویژگی دیگر وجود دارند که در بعضی سطر ها مقادیر NULL دارند؛ همانطور که مشاهده میشود مقادیر NULL در دو ویژگی last\_review و reviews\_per\_month برابر است و این فرضیه وجود دارد که خانه هایی که هیچ نظری راجع به آنها ثبت نشده و number\_of\_reviews در آنها صفر است و در واقع گردشگری آنها را رزرو نکرده؛ reviews\_per\_month و last\_review برای آنها وجود ندارد و برای بررسی این فرضیه قطعه کد زیر را در نظر گرفتیم:

```
test=airbnb[airbnb.number_of_reviews == 0]
print(test.shape)
print(test.last_review.unique())
print(test.reviews_per_month.unique())
test.head()

(10052, 14)
[nan]
[nan]
```

host_id	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	last_review	reviews_per_month	calculated_host_listings_count	availability_365
em	40.80902	-73.94190	Private room	150	3	0	NaN	NaN	1	365
em	40.79685	-73.94872	Entire home/apt	190	7	0	NaN	NaN	2	249
cod	40.86754	-73.92639	Private room	80	4	0	NaN	NaN	1	0
ord-ant	40.68876	-73.94312	Private room	35	60	0	NaN	NaN	1	365
ush	40.63702	-73.96327	Private room	150	1	0	NaN	NaN	1	365

**شکل (7-3)** کد مربوط بررسی دلیلی null بودن reviews\_per\_month و last\_review

در این قطعه کد نشان داده شده است که تعداد رکوردهایی که مقدار ویژگی number\_of\_reviews برای آنها صفر است؛ برابر 10052 تاست که در این 10052 رکورد مقدار reviews\_per\_month و last\_review null بوده است. پس می توان مقدار ویژگی reviews\_per\_month را در این رکورد ها مساوی صفر قرار داد:

### replacing nan value

```
#replacing all NaN values in 'reviews_per_month' with 0
airbnb.fillna({'reviews_per_month':0}, inplace=True)
#examining changes
airbnb.reviews_per_month.isnull().sum()
0
```

شکل (8-3) کد مربوط جایگزین کردن مقادیر null در reviews\_per\_month

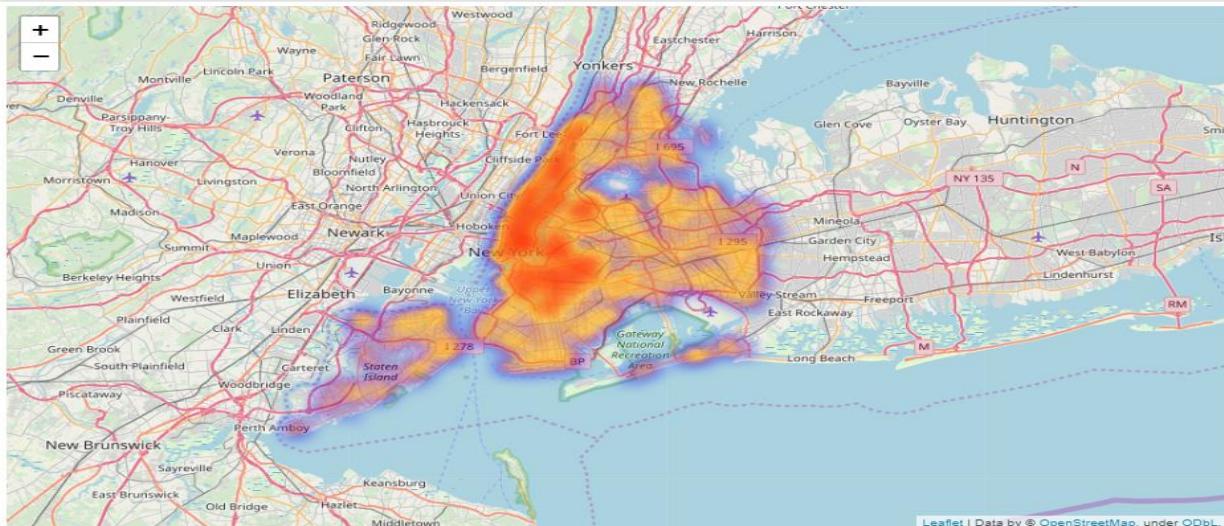
حال تنها دو ویژگی null وجود دارند name و last\_review؛ ویژگی name از نظر مدل آماری اهمیتی ندارد ولی می‌توان آن را برای بررسی این موضوع که، صاحبخانه‌ها از چه عناوینی برای نام گذاری مکان‌ها استفاده مکنند استفاده کرد لذا این ویژگی را فعلانگه میداریم ولی در فصل بعد و قبل از ساخت مدل؛ آن را حذف می‌کنیم. همینطور راجع به ویژگی last\_review.

### 4-3- نقشه پراکندگی برای مکان‌های اقامتی و مکان‌های پرطرفدار

به منظور ایجاد دید اولیه از dataset و ملموس‌تر شدن این چالش ابتدا مکان‌های اقامتی موجود در این dataset را با توجه به دو ویژگی طول و عرض جغرافیایی رسم می‌کنیم.

### where are this Hotel

```
import folium
from folium.plugins import HeatMap
m=folium.Map([40.7128,-74.0060],zoom_start=13)
HeatMap(airbnb[['latitude','longitude']].dropna(),radius=8,gradient=[0.2:'blue',0.4:'purple',0.6:'orange',1.0:'red']).add_to(m)
display(m)
```



شکل (9-3) نمودار HeatMap برای نمایش پراکندگی مکان های اقامتی موجود در dataset

همانطور که در شکل (9-3) مشهود است؛ قسمت هایی که مکانهای بیشتری برای اجاره دادن ارائه کرده اند به ترتیب با رنگ های قرمز، نارنجی، بنفش و آبی مشخص شده اند. که در ادامه، این قسمت ها را به تفکیک شهرها و محله ها بررسی خواهیم کرد.

حال به بررسی نمودار دیگری می پردازیم که پراکندگی مکان های اقامتی را براساس محبوبیت آنها در بین گردشگران نشان می دهد. هر دایره در شکل زیر تعداد خانه ها در یک محدود فرضی و فضای آبی اطراف آن، محدوده‌ی مورد نظر را نشان می دهد.

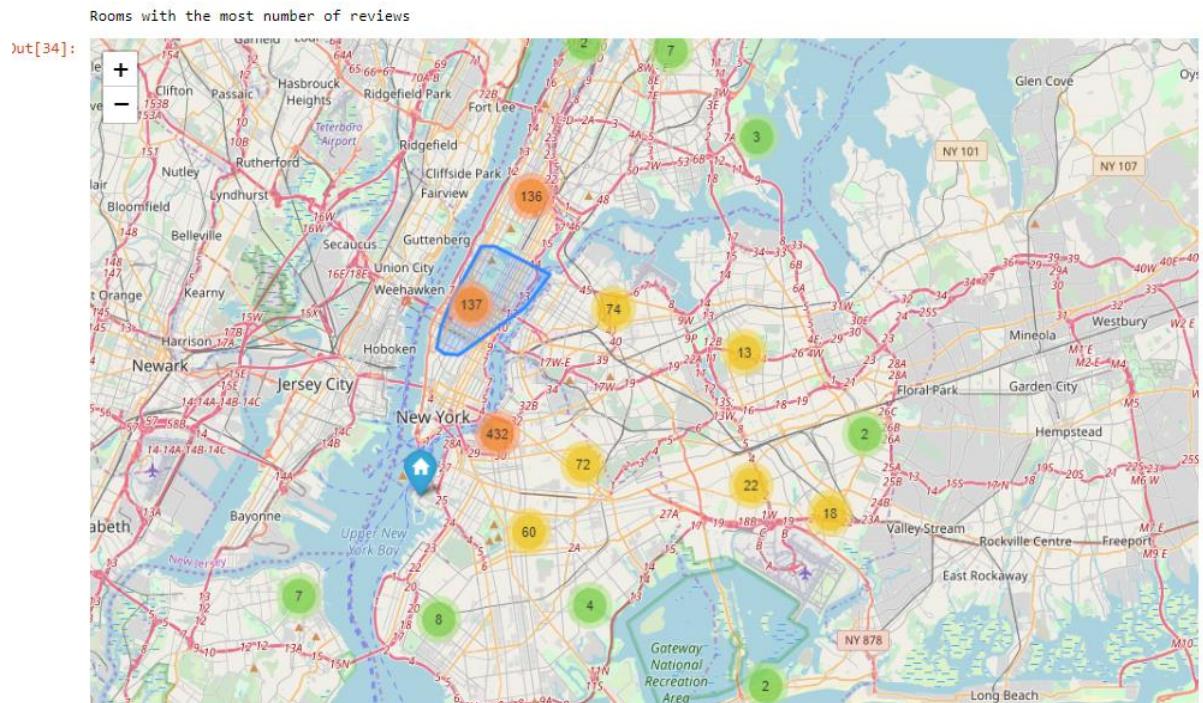
## which are top\_reviewed

```
import folium
from folium.plugins import MarkerCluster
from folium import plugins
print('Rooms with the most number of reviews')
Long=-73.80
Lat=40.80
mapdf1=folium.Map([Lat,Long],zoom_start=10,)

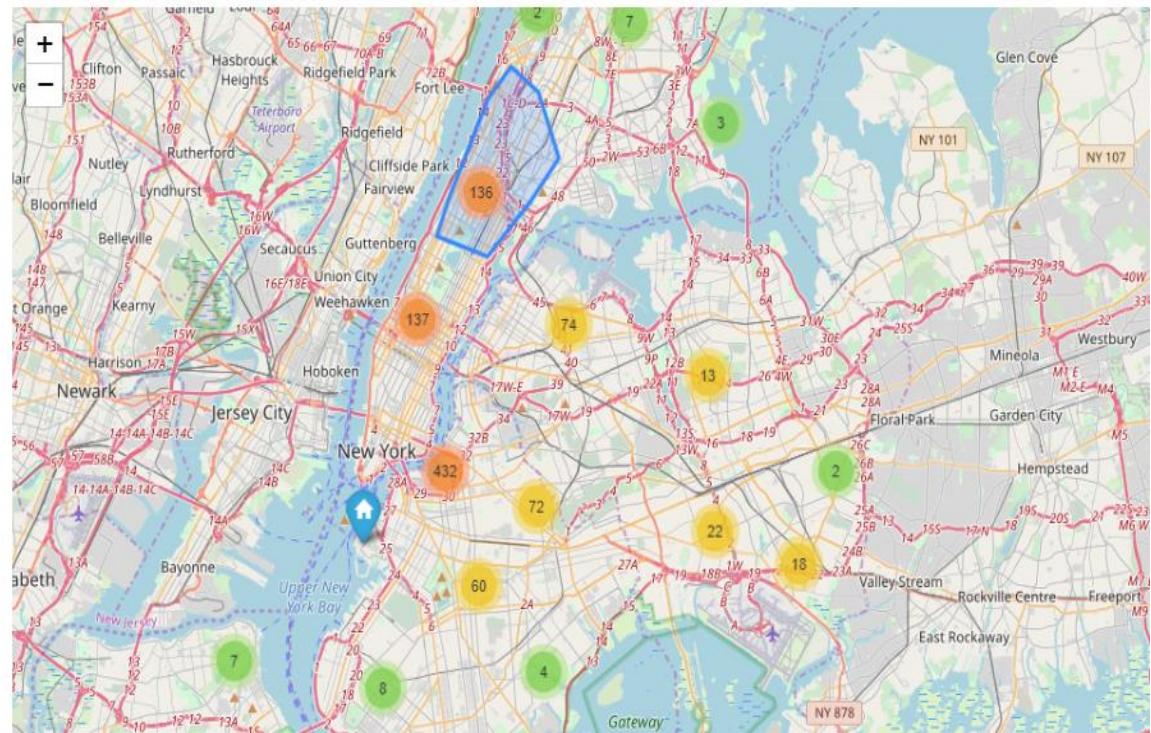
mapdf1_rooms_map=plugins.MarkerCluster().add_to(mapdf1)

for lat,lon,label in zip(top_reviewed_listings.latitude,top_reviewed_listings.longitude,top_reviewed_listings.name):
    folium.Marker(location=[lat,lon],icon=folium.Icon(icon='home'),popup=label).add_to(mapdf1_rooms_map)
mapdf1.add_child(mapdf1_rooms_map)

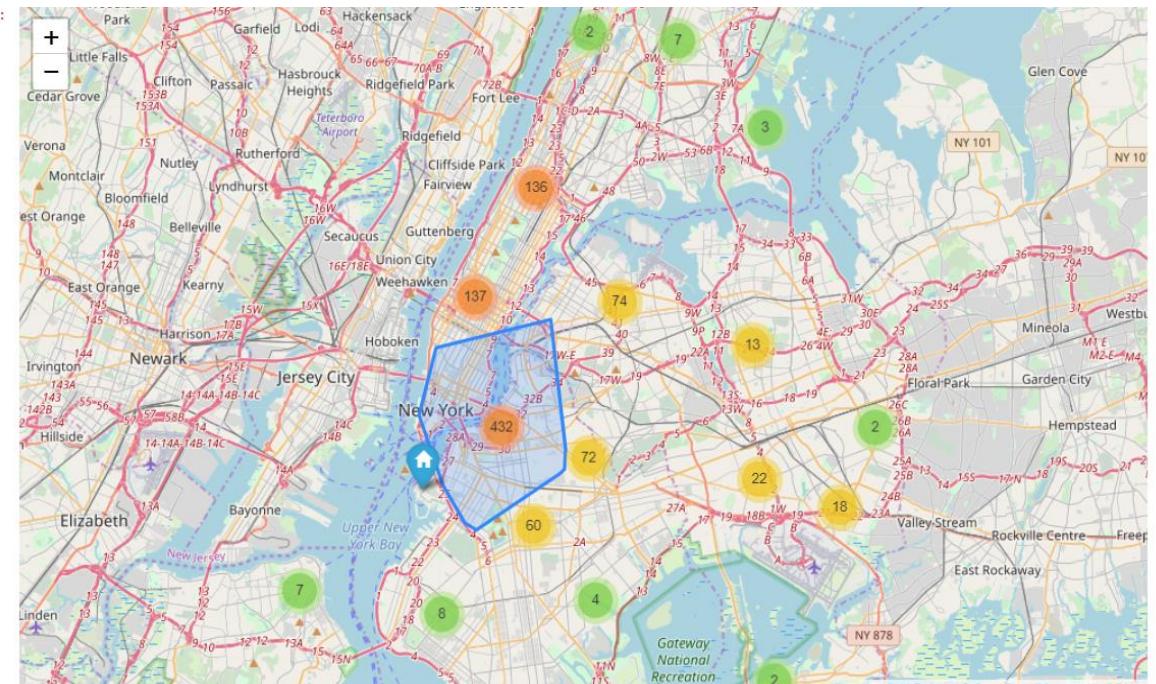
mapdf1
```



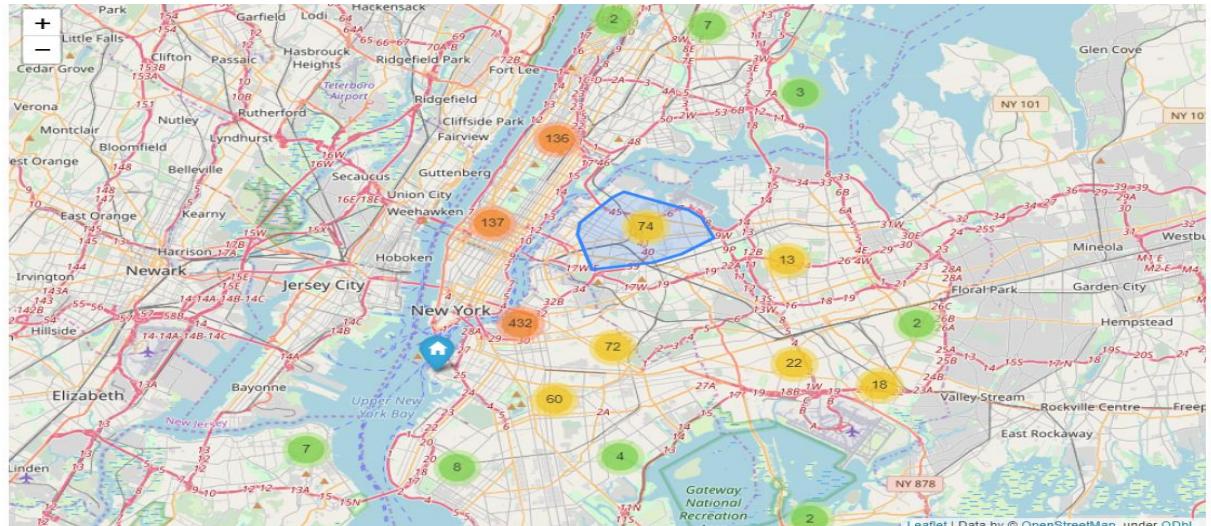
الف-3-9-



⌚-9-3



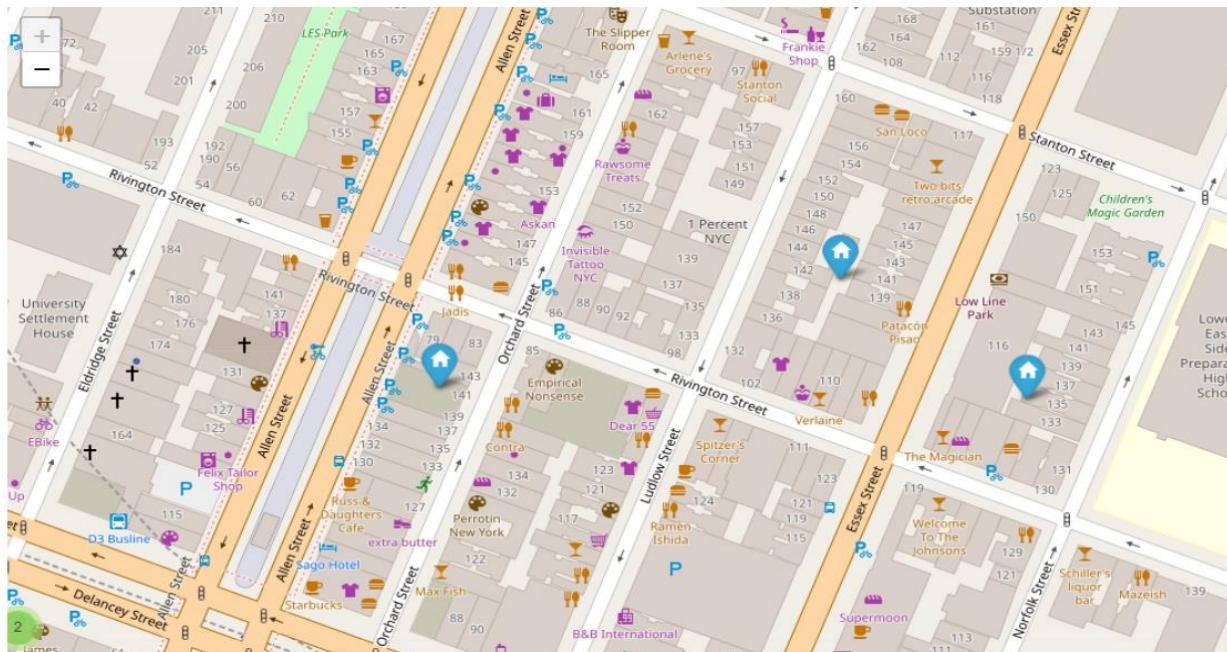
⌚-9-3



ب-9-3

شکل (9-3) نمایش پراکندگی مکان های اقامتی با محبوبیت بالا

با بزرگنمایی تصاویر میتوان تا حدودی از امکانات آن محدوده، دسترسی به خطوط رتباطی و ... مطلع شد که هر یک میتواند دلیلی بر محبوب بودن آن منطقه باشد.



شکل (10-3) نمایش پراکندگی مکان های اقامتی با محبوبیت بالا با جزئیات

به عنوان مثال در شکل بالا وجود چندین رستوران، نمایشگاه نقاشی و فروشگاه میتواند از عوامل محبوب بودن این منطقه باشد. حال به بررسی دقیق تر این **dataset** و فیلد های آن می پردازیم.

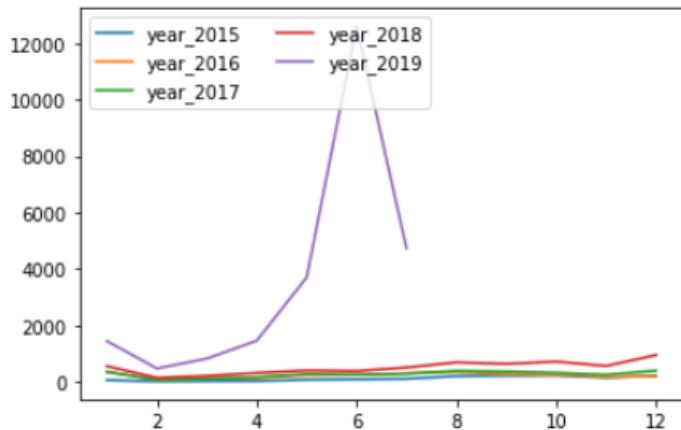
### 5-3- بررسی ویژگی last\_review

این **dataset** عموما شامل داده های سال 2019 میباشد هرچند از سال های دیگر هم داده هایی وجود دارد. حال ما نموداری را برای تعداد خانه هایی که در سال های 2019 و 2018 و 2017 و 2016 اجاره شده اند را به تفکیک ماه ها (محورX) و سال ها (نمودار های مختلف) را رسم می کنیم.

#### what can we find from last\_review

```
airbnb['last_review'] = pd.to_datetime(airbnb['last_review'],infer_datetime_format=True)
year_2019=[]
year_2018=[]
year_2017=[]
year_2016=[]
year_2015=[]
for date in airbnb['last_review']:
    if date.year == 2019:
        year_2019.append(date.month)
    if date.year == 2018:
        year_2018.append(date.month)
    if date.year == 2017:
        year_2017.append(date.month)
    if date.year == 2016:
        year_2016.append(date.month)
    if date.year == 2015:
        year_2015.append(date.month)
print("Number of Homes for Rent in 2019 ",len(year_2019))
print("Number of Homes for Rent in 2018 ",len(year_2018))
print("Number of Homes for Rent in 2017 ",len(year_2017))
print("Number of Homes for Rent in 2016 ",len(year_2016))
print("Number of Homes for Rent in 2015 ",len(year_2015))
year_2019 = pd.Series(year_2019)
year_2018 = pd.Series(year_2018)
year_2017 = pd.Series(year_2017)
year_2016 = pd.Series(year_2016)
year_2015 = pd.Series(year_2015)
df = pd.DataFrame({ 'year_2015':year_2015.value_counts(), 'year_2016':year_2016.value_counts(),'year_2017':year_2017.value_counts()})
plt.plot(df.index,df.values);
plt.legend(['year_2015','year_2016','year_2017','year_2018','year_2019'],ncol=2,loc='upper left');
```

Number of Homes for Rent in 2019	25209
Number of Homes for Rent in 2018	6050
Number of Homes for Rent in 2017	3205
Number of Homes for Rent in 2016	2707
Number of Homes for Rent in 2015	1393



شکل (11-3) نمودار تعداد خانه های رزرو شده به تفکیک ماه

یکی از مسائل و مشکلاتی که ما در تحلیل داده با آنها مواجه هستیم کمبود داده است؛ در این dataset هم تعداد نمونه هایی که از سال های 2016 و 2017 و 2018 داریم بسیار کم هستند و نمیتوان تحلیلی دقیقی داشت ولی با مشاهده نمودار سال 2019 در شکل (11-3) میتوان دریافت ماه ششم سال اوج مسافرت ها بوده، زیرا در ژوئن دمای هوای خیلی گرم و نه خیلی سرد است و نمایشگاه ها و امکانات فرداوانی در دسترس است و کنسرت ها و نمایش های رایگان خیابانی به فرداوانی وجود دارد. در حالی که قبل از آن هوای سرد تر است و بعد از هوای شدت گرم میشود و متاسفانه این شرکت هنوز اطلاعات خود برای 5 ماه اخر سال در اختیار قرار نداده است.

### 6-3- بررسی ویژگی neighbourhood\_group

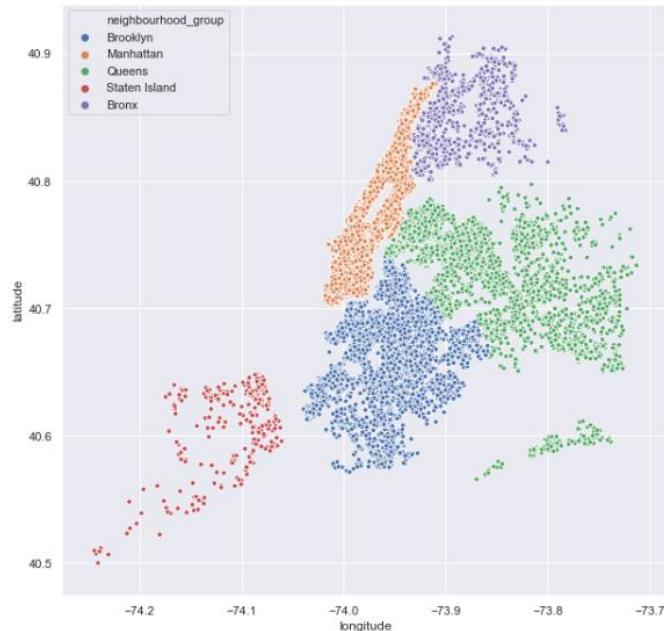
ابتدا به بررسی این موضوع میپردازیم که ویژگی neighbourhood\_group که شامل چه مقادیری است و سپس با رسم نمودار scatterplot براساس طول و عرض جغرافیایی مکان های موجود در dataset نقشه ای ابتدایی از شهر های New York City به دست می آوریم:

### what are neighbourhood\_group value

```
#examining the values of n_group
airbnb.neighbourhood_group.unique()

array(['Brooklyn', 'Manhattan', 'Queens', 'Staten Island', 'Bronx'],
      dtype=object)

plt.figure(figsize=(10,10))
sns.scatterplot(x='longitude', y='latitude', hue='neighbourhood_group', s=20, data=airbnb);
```



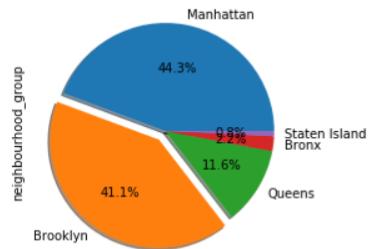
شکل (12-3) بررسی ویژگی neighbourhood\_group

باتوجه به شکل (12-3) Manhattan ,Queens ، Brooklyn دارای مقادیر neighbourhood\_group (12-3) و Bronx و Staten Island این 5 شهر را به تفکیک میتوانید در این شکل مشاهده کنید.

میخواهیم به بررسی این موضوع بپردازیم که سهم هریک از این شهرها در ما چقدر است و هر شهر چند درصد از مکانهای dataset را شامل میشود نمودار دایره‌ای<sup>۱</sup> در شکل زیر این موضوع را به وضوح نمایش میدهد:

### Sort cities by number of hotels and draw charts

```
airbnb['neighbourhood_group'].value_counts().plot.pie(explode=[0,0.1,0,0], autopct='%1.1f%%', shadow=True)
plt.show()
```



شکل (13-3) بررسی فراوانی خانه‌های ارائه شده در هر شهر (neighbourhood\_group)

همانطور که در شکل (13-3) مشاهده میشود شهر Manhattan با 44.3 درصد بیشترین مکان را برای کرایه دارد و بعداز آن Brooklyn با 41.1 درصد و Queens با 11.6 درصد و Bronx با 2.2 درصد و در نهایت Staten Island با 0.8 درصد کمترین مکان را برای کرایه دارد . حال این سوال مطرح است آیا ارائه مکان بیشتر ارتباطی با توریستی بودن این شهر دارد؟ که در ادامه به توضیح این موضوع می‌پردازیم .

### -7-3 بررسی ویژگی neighbourhood

#### 1. pie plot

ویژگی neighbourhood شامل محله های موجود در هر شهر است این ویژگی شامل 221 مقدار است که 20 مورد از این محله ها در شکل زیر قابل مشاهده اند:

## what are neighbourhood value?

```
#examining the values of neighbourhood
neighbourhood_values= airbnb.neighbourhood.unique()
print(neighbourhood_values[0:20], '....')
len(neighbourhood_values)
```

```
['Kensington' 'Midtown' 'Harlem' 'Clinton Hill' 'East Harlem'
 'Murray Hill' 'Bedford-Stuyvesant' "Hell's Kitchen" 'Upper West Side'
 'Chinatown' 'South Slope' 'West Village' 'Williamsburg' 'Fort Greene'
 'Chelsea' 'Crown Heights' 'Park Slope' 'Windsor Terrace' 'Inwood'
 'East Village'] .....
```

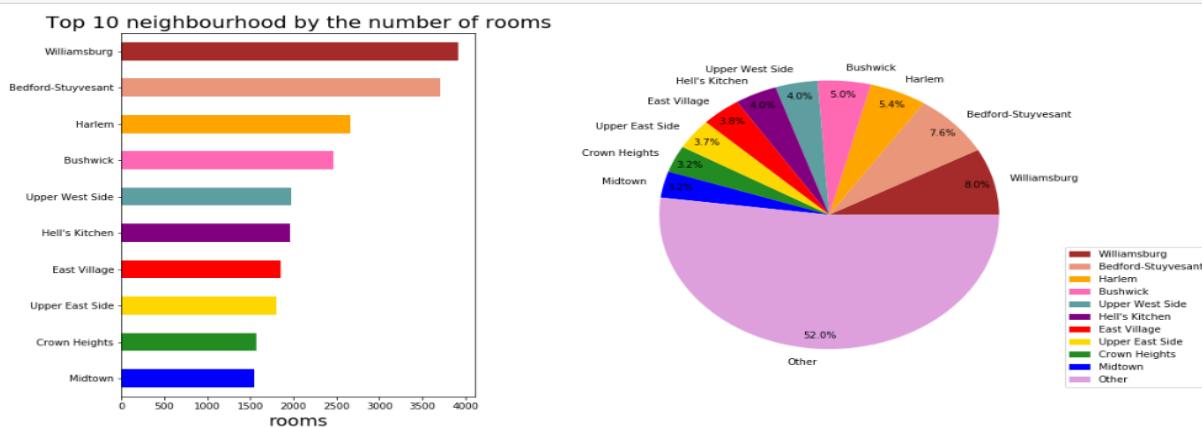
221

شکل (14-3) بررسی مقادیر در ستون neighbourhood

برای بررسی دقیق تر برروی این ویژگی ، ده محله ای که بیشترین مکان را برای اجاره ، ارائه داده اند در نموداری رسم می کنیم :

```
fig,ax=plt.subplots(1,2,figsize=(15,8))
clr = ("blue", "forestgreen", "gold", "red", "purple", 'cadetblue', 'hotpink', 'orange', 'darksalmon', 'brown')
airbnb.neighbourhood.value_counts().sort_values(ascending=False)[10].sort_values().plot(kind='barh',color=clr,ax=ax[0])
ax[0].set_title("Top 10 neighbourhood by the number of rooms",size=20)
ax[0].set_xlabel('rooms',size=18)

count=airbnb['neighbourhood'].value_counts()
groups=list(airbnb['neighbourhood'].value_counts().index)[:10]
counts=list(count[:10])
counts.append(count.agg(sum)-count[:10].agg('sum'))
groups.append('Other')
type_dict=pd.DataFrame({"group":groups,"counts":counts})
clr1=('brown','darksalmon','orange','hotpink','cadetblue','purple','red','gold','forestgreen','blue','plum')
qx = type_dict.plot(kind='pie', y='counts', labels=groups,colors=clr1, autopct='%.1f%%', pctdistance=0.9, radius=1.2,ax=ax[1])
plt.legend(loc=0, bbox_to_anchor=(1.15,0.4))
plt.subplots_adjust(wspace = 0.5, hspace = 0)
plt.ioff()
plt.ylabel('')
```



**شکل (15-3)** بررسی 10 محله که بیشترین مکان را برای اجاره ارائه داده اند

در شکل (15-3) میتوان این ده محله را مشاهده کرد برای مثال محله williamsburg ، محله ای است که با تقریبا 4000 اتاق یعنی 8 درصد از رکوردها ، بیشترین مکان را برای اجاره ارائه داده است .

### 8-3- بررسی ارتباط بین neighbourhood\_group و neighbourhood

آیا ارتباطی بین شهرهایی که مکان بیشتری برای اجاره ارائه داده اند شکل (13-3) و ده محله هایی که مکان بیشتری برای اجاره ارائه داده اند شکل (15-3) وجود دارد؟  
برای بررسی این موضوع این ده محله را به تفکیک شهرهایی که به آن تعلق دارند مشخص میکنیم :



**شکل (16-3)** بررسی 10 محله که بیشترین مکان را برای اجاره ارائه داده اند به تفکیک شهرهایی که به آن تعلق دارند .

همانطور که از تصویر مشخص است ده محله برتر جزو شهر های برتر هستند!! به عبارت دیگر در شکل (3-13) مشاهده کردیم که manhattan و Brooklyn دو شهری هستند که بیشترین مکان را ارائه داده اند و در این تصویر مشاهده میشود ده محله ای که بیشترین مکان را ارائه داده اند متعلق به این دو شهر هستند.

حال این سوال مطرح است که آیا محله هایی که بیشترین مکان را ارائه میدهد گردشگران بیشتری هم دارند؟؟؟ یا به عبارت دیگر محله ها و شهر های برتر از نظر گردشگران کدامند؟

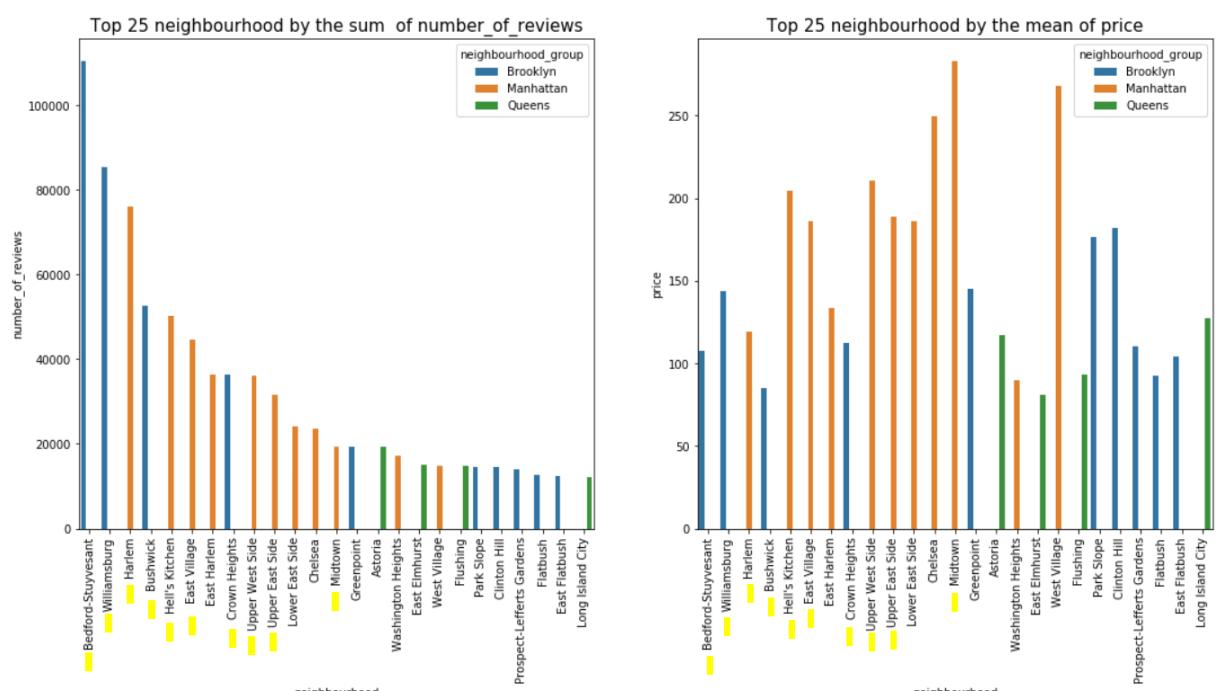
همانطور که در بخش 3-1 در توضیح فیلد های dataset اشاره شد؛ فیلد number\_of\_reviews تعداد نظرات گردشگران راجع به مکان کرایه شده را نشان میدهد یا به عبارت دیگر تعداد افرادی که آن خانه را رزرو کرده اند؛ حال اگر خانه ای اصلاً توسط گردشگران رزرو نشود این ستون برای آن سطر مقدار صفر خواهد داشت و هر شهر و محله ای که تعداد مسافر بیشتری داشته باشد مسلماً مکان بیشتری کرایه داده و نظرات بیشتری راجع به مکان های آن شهر وجود دارد. در قطعه کد زیرما dataset را براساس محله ها (neighbourhood) گروه بندی کردیم و به ازای هر گروه میانگین قیمت و مجموع تعداد نظرات به دست آوردیم؛ نمودار سمت چپ مجموع تعداد نظرات بر حسب محله هارا به تفکیک شهر ها و نمودار سمت راست میانگین قیمت را براساس محله ها به تفکیک شهر ها نشان میدهد.

## Price level and number of hits by neighborhood

```
df_top_prices_by_neighbourhood = airbnb.groupby('neighbourhood').agg({'price': 'mean', 'number_of_reviews' : 'sum','neighbourhood': 'count'})
f,ax=plt.subplots(1,2,figsize=(18,8))
viz_6=sns.barplot(x=df_top_prices_by_neighbourhood['neighbourhood'], y=df_top_prices_by_neighbourhood['number_of_reviews'], data=df_top_prices_by_neighbourhood)
viz_6.set_xlabel('neighbourhood')
viz_6.set_xticklabels(viz_6.get_xticklabels(), rotation=90);
#price
viz_7=sns.barplot(x=df_top_prices_by_neighbourhood['neighbourhood'], y=df_top_prices_by_neighbourhood['price'], data=df_top_prices_by_neighbourhood)
viz_7.set_xlabel('neighbourhood')
viz_7.set_xticklabels(viz_6.get_xticklabels(), rotation=90);
```

```
reviews' : 'sum','neighbourhood_group':'min'}}).sort_values('number_of_reviews' ,ascending=False).reset_index().head(25)
cod['number_of_reviews'], data=df_top_prices_by_neighbourhood,ax=ax[0],hue=df_top_prices_by_neighbourhood['neighbourhood_group'])

cod['price'], data=df_top_prices_by_neighbourhood,ax=ax[1],hue=df_top_prices_by_neighbourhood['neighbourhood_group'])
```



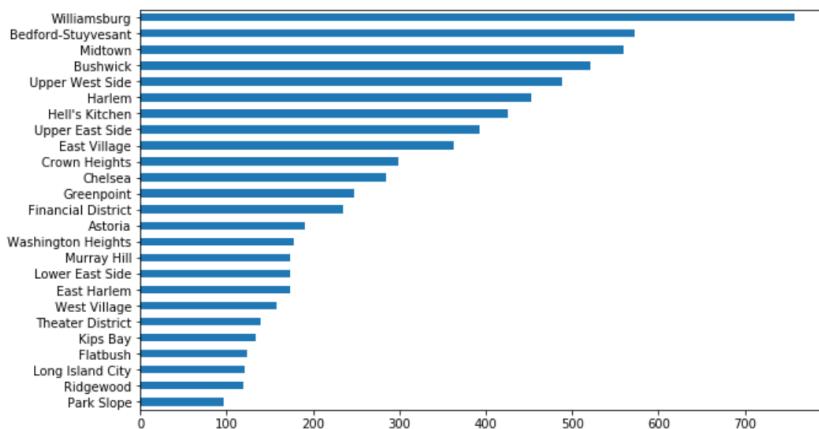
شکل (17-3) بررسی 25 محله برتر از نظر گردشگران

شکل سمت چپ 25 محله ای که بیشترین تعداد گردشگر را داشته اند به ترتیب نشان میدهد، ده محله برتر که در شکل (15-3) مشاهده کردید در این شکل با نوار زرد رنگ مشخص شده است. با مشاهده این شکل متوجه میشویم که ده محله ای که بیشترین تعداد مکان برای اجاره داشته اند از نظر گردشگران نیز محبوب اند. در شکل سمت راست میانگین قیمت به ازای هر محله نشان داده شده است. از این شکل میتوان متوجه شد که چهار محله برتر از نظر گردشگران دارای میانگین قیمت به نسبت پایینی بوده اند؛ محله های دیگر که در شهر manhatta واقع شده اند و پر طرفداراند به دلیل امکانات رفاهی و خدمات لوکس این شهر است که در ادامه توضیحات بیشتری ارائه خواهد شد.

موضوع قابل توجه دیگر در این شکل این است که سه شهر Lower East Side و Chelsea و East Side که جزو شهر های محبوب هستند، در لیست ده محله نیستند و ممکن است گردشگران در این سه شهر با مشکل کمبود مکان های اقامتی مواجه باشند یا به عنوان مثال دیگر محله Bedford-Stuyvesant از نظر گردشگران محبوب تر از Williamsburg است حال آنکه تعداد مکان های اقامتی

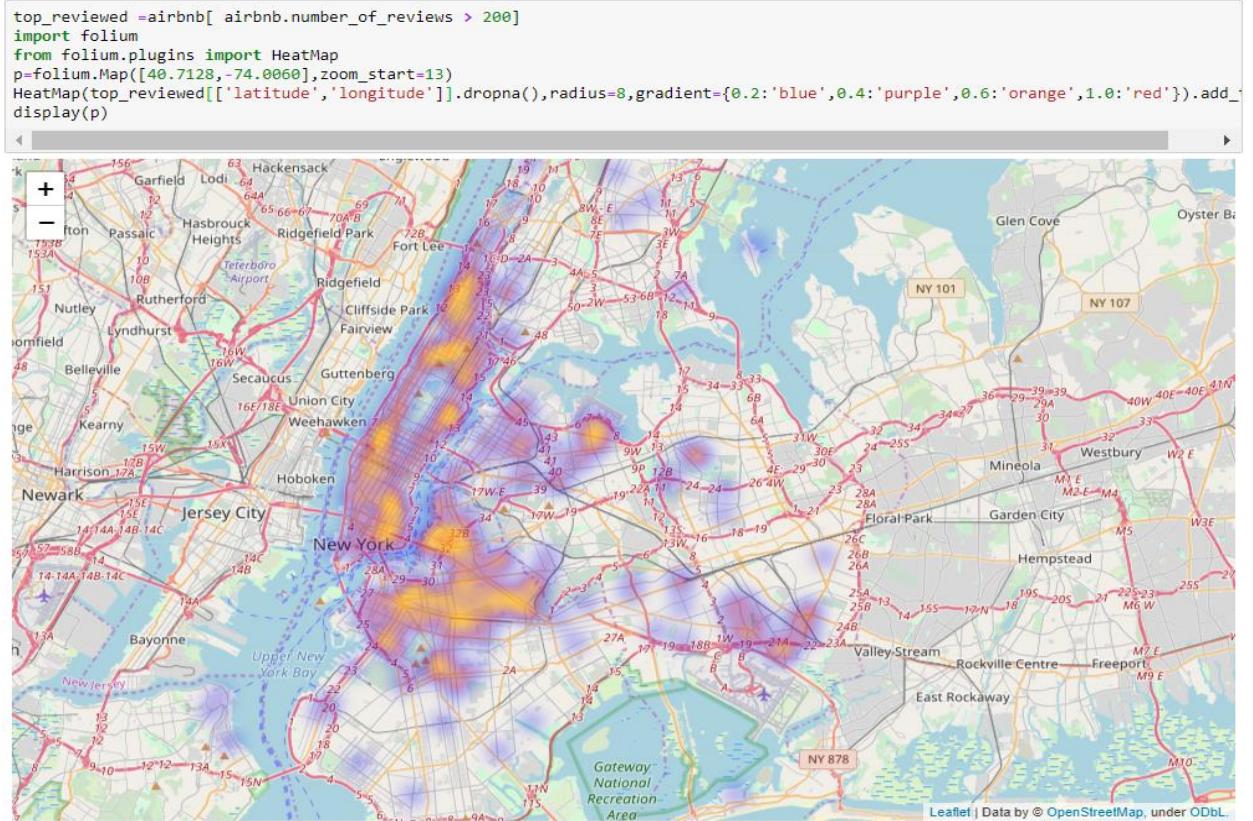
در Williamsburg بیشتر است و ممکن است مردم در محله Bedford-Stuyvesant با مشکل موافق باشند یا اینکه تعداد زیادی از خانه ها در محله Williamsburg خالی و بدون گردشگر باشند؛ برای روشن شدن این موضوع تعداد خانه هایی که تابه حال مراجعه کننده نداشتند را بررسی میکنیم :

```
plt.figure(figsize=(10,6))
sub_12 = airbnb[airbnb.number_of_reviews == 0]
sub_12.neighbourhood.value_counts().sort_values(ascending=False)[:25].sort_values().plot(kind='barh');
```



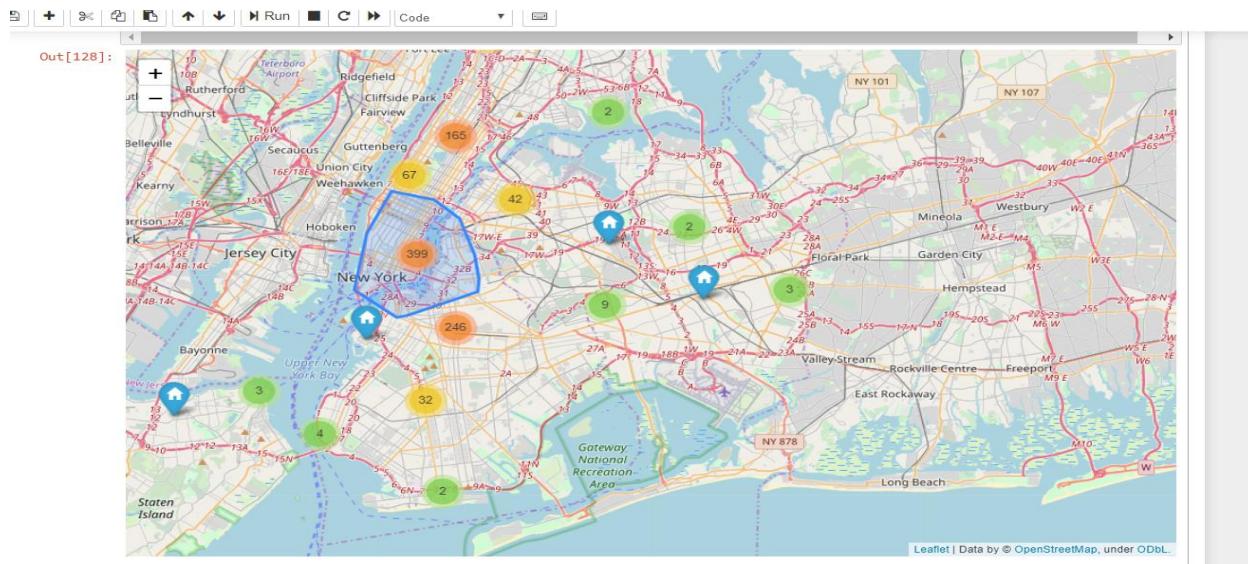
شکل (18-3) بررسی مکان های اقامتی بدون گردشگر

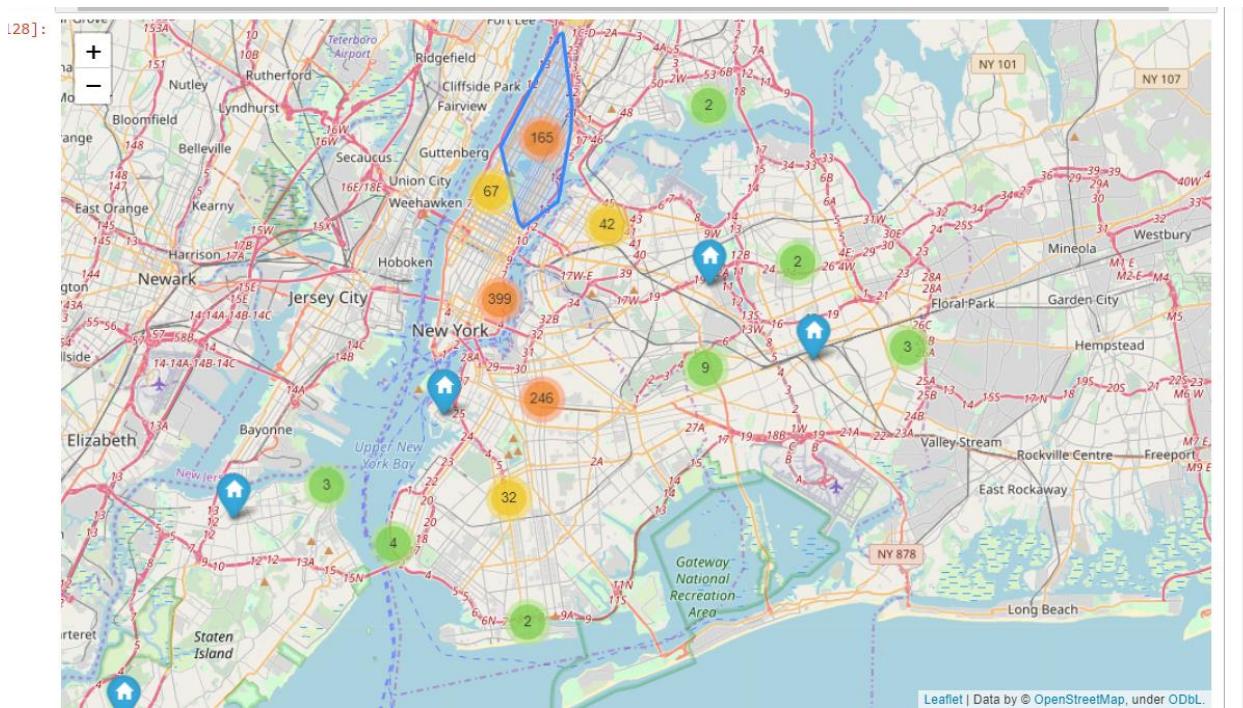
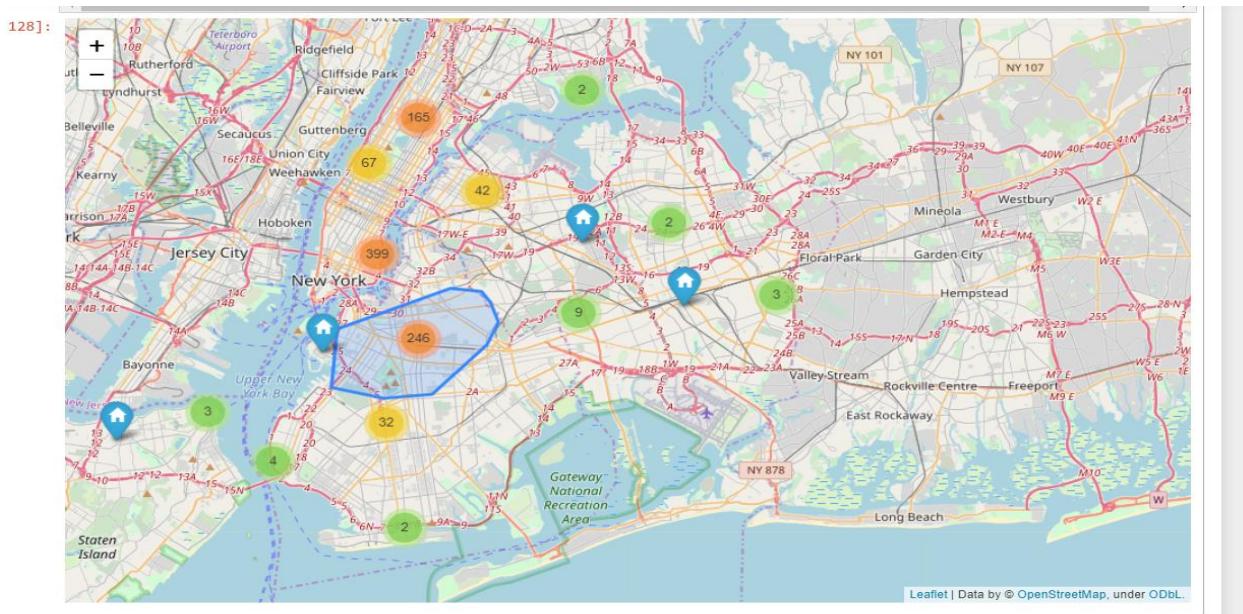
با توجه به شکل (18-3) در محله williamsburg تقریبا 750 مکان اقامتی وجود دارد که تا به حال استفاده نشده است و همینطور راجع به بقیه محله ها البته باید پراکندگی این مکان های خالی و نیز پراکندگی مکان های پرگردشگر را نیز بررسی کرد شاید این مکان ها در بعضی مناطق دردسترس نباشند.



شکل (3-19) بررسی مکان های اقامتی با گردشگر بالا

در شکل بالا پراکندگی مکان های پرطرفدار از نظر گردشگران را مشاهده میکنید حال به بررسی پراکندگی مکان های خالی می پردازیم





**شکل (20-3)** بررسی مکان های اقامتی بدون گردشگر

همانطور که در این چند تصویر مشاهده میشود این خانه ها تقریبا تمامی نواحی پر طرفدار را پوشش می دهند پس میتوان گفت کمبود مکان در این

شهر ها وجود ندارد هر چند برای اثبات این ادعا بررسی های بیشتری نیاز است مثل اینکه این مکان در چه سالی اضافه شده و آیا هنوز نیز در دسترس است؟ آیا در زمان اوج مسافرت ها؛ که در بخش 5-3 مورد بررسی قرار گرفت؛ در دسترس هستند؟ ... که به اطلاعات بیشتری نیاز دارد.

### 9-3- room\_type بررسی ویژگی

همانطور که در قطعه کد زیر نمایان است این ویژگی سه مقدار Private room و Shared room و Entire home/apt را دارد.

#### what are room\_type values?

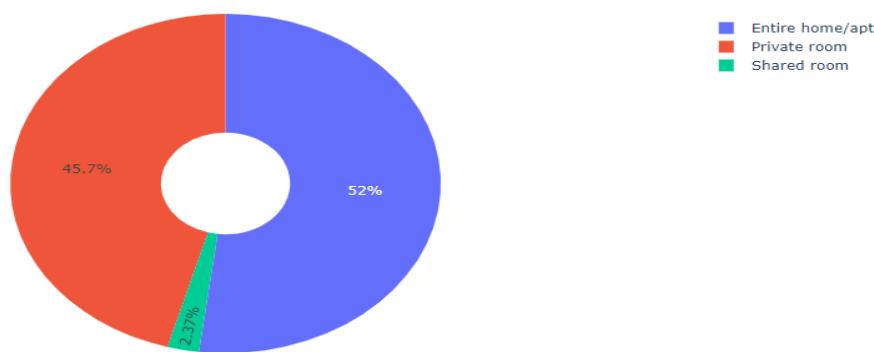
```
#examining the unique values of room_type
airbnb.room_type.unique()
array(['Private room', 'Entire home/apt', 'Shared room'], dtype=object)
```

شکل (21-3) بررسی مقادیر ویژگی room\_type

و فراوانی هر دسته نیز در شکل زیر نشان داده شده است:

```
import plotly.offline as pyo
import plotly.graph_objs as go
print('type of rooms')
roomdf = airbnb.groupby('room_type').size()/airbnb['room_type'].count()*100
labels = roomdf.index
values = roomdf.values
# Use `hole` to create a donut-like pie chart
fig = go.Figure(data=[go.Pie(labels=labels, values=values, hole=.3)])
fig.show()
```

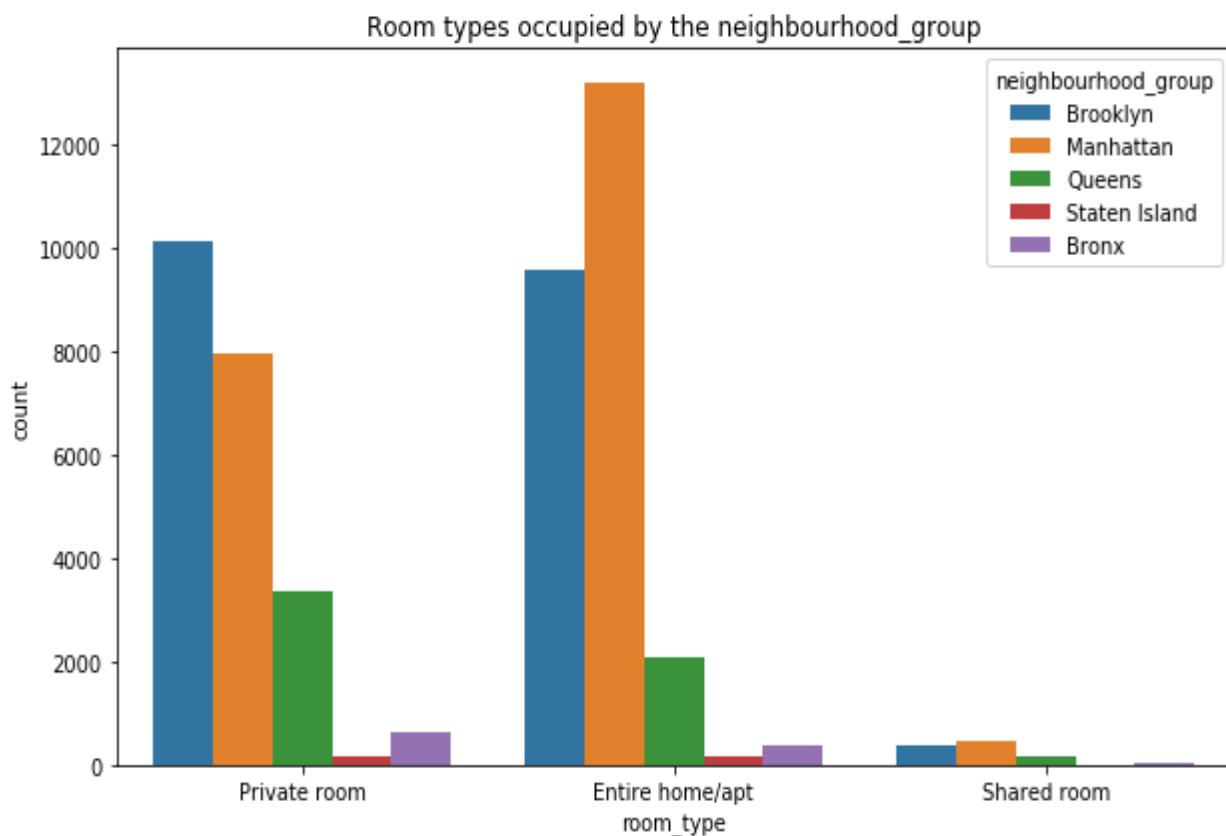
type of rooms



شکل (22-3) بررسی فراوانی مقادیر room\_type

همانطور که در شکل (22-3) نشان داده شده است Entire home/apt بیشترین درصد اتاق های ارائه شده را تشکیل می دهد؛ بعد از آن Private room با 45.7 درصد و کمترین فراوانی مربوط به Shared room با 2.37 درصد است. حال این فراوانی را به تفکیک شهر ها بررسی می کنیم تا دید بهتری از مکان های ارائه شده داشته باشیم .

```
plt.figure(figsize=(10,6))
sns.countplot(x = 'room_type',hue = "neighbourhood_group",data = airbnb)
plt.title("Room types occupied by the neighbourhood_group")
plt.show()
```



شكل (3-3) بررسی فراوانی مقادیر room\_type به تفکیک شهر ها

باتوجه به شکل (23-3) بیشتر مکان های ارائه شده در شهر Manhattan هستند و Shared room کمترین تعداد ارائه را داشته است این موضوع را برای تمامی شهر ها می توان بیان کرد؛ مگر در مورد Staten Island و Bronx زیرا برای این شهرها به اندازه کافی نمونه در dataset موجود نیست؛ نکته دیگر اینکه Shared room در تمامی شهر ها کمترین میزان ارائه را داشته است. حال باید دید مکان های پر طرفدار از نظر گردشگران در کدام دسته قرار میگیرند و آیا نیازی برای ارائه بیشتر مکان های Shared room هست یا خیر.

```
#let's grab 10 most reviewed listings in NYC
top_reviewed_listings=airbnb.sort_values(by=['number_of_reviews'],ascending=False).head(1000)
top_reviewed_listings['room_type'].value_counts()

Private room      510
Entire home/apt   480
Shared room        10
Name: room_type, dtype: int64
```

شکل (24-3) بررسی room\_type در مکان های پر گردشگر

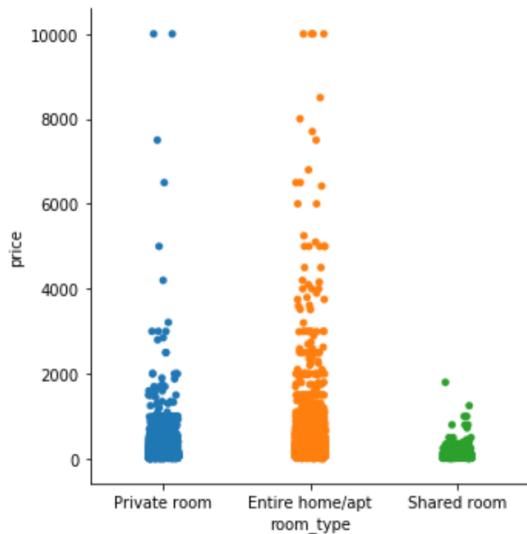
باتوجه به قطعه کد بالا Private room و بعد از آن Entire home/apt محبوب ترین room\_type ها از نظر گردشگران هستند و Shared room ها همانطور که ارائه کمتری دارند طرفدار کمتری نیز دارند و نیازی برای افزایش آنها نیست.

### 10-3- بررسی ویژگی های price و room\_type

میخواهیم به بررسی این موضوع بپردازیم که آیا قیمت و نوع اتاق ارائه شده ارتباط معنا داری با هم دارند؟ برای این منظور نمودار زیر را رسم میکنیم.

```
#catplot room type and price
plt.figure(figsize=(10,6));
sns.catplot(x="room_type", y="price", data=airbnb);
plt.ioff();
room_type_and_price = airbnb.groupby('room_type').agg({'price': 'mean'})
room_type_and_price
```

room_type	price
Entire home/apt	211.794246
Private room	89.780973
Shared room	70.127586



شکل (25-3) بررسی ویژگی های price و room\_type

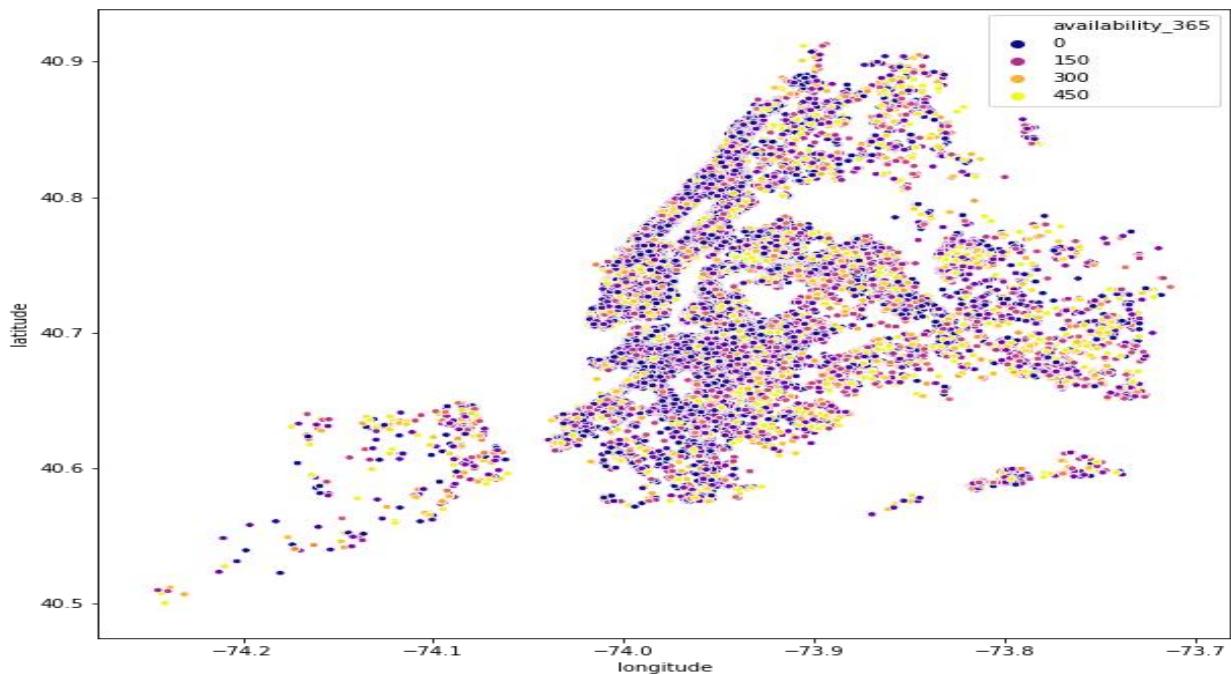
باتوجه به شکل بالا میتوان دریافت که کمترین قیمت را دارند؛ حال آنکه Entire home/apt ها تقریباً گران قیمت ترین خانه ها هستند و Shared room ها با میانگین قیمت 89.78 دلار در جایگاه دوم هستند.

### 11-3- availability\_365 بررسی ویژگی

ویژگی دیگری که در این dataset وجوددارد availability\_365 است این ویژگی نشان می‌دهد می‌داند صاحبخانه چند روز از 365 روز سال می‌تواند خانه‌ی ثبت شده را در اختیار گردشگران قرار دهد ما این ویژگی را در شکل (26-3) نمایش داده ایم به این صورت که هر خانه با یک دایره کوچک نشان داده شده است که رنگ این دایره با توجه به ویژگی availability\_365 برای آن خانه و راهنمای جدول تعیین می‌شود.

### availability\_365

```
plt.figure(figsize=(10,10))
sns.scatterplot(x='longitude', y='latitude', hue='availability_365', s=20, data=airbnb ,palette='plasma' );
```



شکل (26-3) بررسی ویژگی های availability\_365

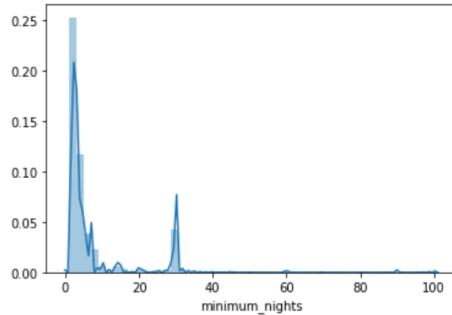
با مراجعه به شکل (19-3) و یادآوری پراکندگی مکان های پرطرفدار در میابیم صاحبان خانه در مکان های پرطرفدار خانه های خود را در بازه زمانی کمتری در دسترس قرار میدهند حال آنکه در مکان های با محبوبیت کمتر از نظر گردشگران خانه ها در تمام طول سال در دسترس هستند اتفاقی که خیلی هم دور از تصور نیست زیرا افراد در مناطق پر گردشگر در زمان هایی که حجم سفر به این مناطق افزایش می یابد برای کسب درآمد بخشی از فضای در اختیار خود را که استفاده کمتری دارد در اختیار گردشگران قرار می دهند.

### minimum\_nights - بررسی ویژگی 12-3

ویژگی دیگری که به بررسی آن می پردازیم `minimum_nights` است این ویژگی بیانگر این است که در صورت انتخاب یک مکان اقامتی حداقل چند شب باید آن مکان را رزرو کنیم.

### `minimum_nights`

```
|: sns.distplot(airbnb[(airbnb['minimum_nights'] <= 100) & (airbnb['minimum_nights'] > 0)]['minimum_nights'])
plt.ioff()
```

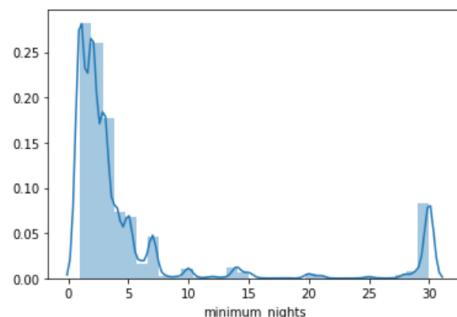


شکل (27-3) بررسی ویژگی های `minimum_nights`

همانطور که در شکل (27-3) مشاهده میشود تقریباً بعد از 30 روز داده چندانی وجود ندارد لذا برای بررسی دقیق تر ما بازه بررسی خود را کوچکتر میکنیم به این کار رایج در داده کاوی حذف مقادیر خارج از محدوده می گویند.

### `minimum_nights`

```
sns.distplot(airbnb[(airbnb['minimum_nights'] <= 30) & (airbnb['minimum_nights'] > 0)]['minimum_nights'], bins=31)
plt.ioff()
```



شکل (28-3) بررسی ویژگی های `minimum_nights`

با مشاهده نمودار فوق میتوان در یافت اکثر صاحب خانه ها این شرط را دارند که شما حداقل یک روز یا دوروز در مکان ارائه شده از سوی آن ها اقامت کنید. این موضوع را میتوان به راحتی درک کرد که حذف مقادیر خارج از محدوده چقدر میتواند به روشن و واضح شدن نمودار کمک کند.

### 13-3- بررسی همبستگی و ارتباط ویژگی ها

موضوع مهم دیگری که در داده کاوی مورد اهمیت است حذف ستون های اضافی است. فرض کنید شما در آمد هر ماه ودر آمد سالانه را در dataset خود دارید؛ حال آنکه نیازی به نگهداری درآمد سالانه نیست زیرا شما می توانید با محاسبه ای ساده آنرا به دست آورید. برای کشف این ارتباط ها بین ویژگی ها ابزار هایی از جمله محاسبه correlation در اختیار است. در شکل زیر میتوانید این محاسبه را برای dataset جاری مشاهده کنید.

#### correlation between features

```
#No strong correlation except number_of_reviews vs reviews_per_month
airbnb.corr().style.background_gradient(cmap='coolwarm')
```

	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month
host_id	1	0.0202242	0.127055	0.0153091	-0.0173643	-0.140106	0.209783
latitude	0.0202242	1	0.0847884	0.0339387	0.0248693	-0.0153888	-0.0187577
longitude	0.127055	0.0847884	1	-0.150019	-0.0627471	0.0590943	0.138516
price	0.0153091	0.0339387	-0.150019	1	0.0427993	-0.0479542	-0.0505641
minimum_nights	-0.0173643	0.0248693	-0.0627471	0.0427993	1	-0.0801161	-0.124905
number_of_reviews	-0.140106	-0.0153888	0.0590943	-0.0479542	-0.0801161	1	0.589407
reviews_per_month	0.209783	-0.0187577	0.138516	-0.0505641	-0.124905	0.589407	1
calculated_host_listings_count	0.15495	0.0195174	-0.114713	0.0574717	0.12796	-0.0723761	-0.0473121
availability_365	0.203492	-0.0109835	0.0827307	0.0818288	0.144303	0.172028	0.163732

#### correlation between features

```
#No strong correlation except number_of_reviews vs reviews_per_month
airbnb.corr().style.background_gradient(cmap='coolwarm')
```

	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	availability_365
host_id	1	0.0202242	0.127055	0.0153091	-0.0173643	-0.140106	0.209783	0.15495	0.203492
latitude	0.0202242	1	0.0847884	0.0339387	0.0248693	-0.0153888	-0.0187577	0.0195174	-0.0109835
longitude	0.127055	0.0847884	1	-0.150019	-0.0627471	0.0590943	0.138516	-0.114713	0.0827307
price	0.0153091	0.0339387	-0.150019	1	0.0427993	-0.0479542	-0.0505641	0.0574717	0.0818288
minimum_nights	-0.0173643	0.0248693	-0.0627471	0.0427993	1	-0.0801161	-0.124905	0.12796	0.144303
number_of_reviews	-0.140106	-0.0153888	0.0590943	-0.0479542	-0.0801161	1	0.589407	-0.0723761	0.172028
reviews_per_month	0.209783	-0.0187577	0.138516	-0.0505641	-0.124905	0.589407	1	-0.0473121	0.163732
calculated_host_listings_count	0.15495	0.0195174	-0.114713	0.0574717	0.12796	-0.0723761	-0.0473121	1	0.225701
availability_365	0.203492	-0.0109835	0.0827307	0.0818288	0.144303	0.172028	0.163732	0.225701	1

### شکل (28-3) بررسی همبستگی و ارتباط ویژگی ها

اگر مقدار این محاسبه برای دو ویژگی کمتر از 0/33 باشد ارتباطی بین این دو ویژگی وجود ندارد که در اکثر خانه های جدول فوق عدد به دست آمده این موضوع را نشان می دهد. اگر 0/33 تا 0/66 باشد یعنی ارتباطی بین review\_per\_month و number\_of\_reviews آنها برقرار است که در شکل فوق دو ویژگی بازدید باهم ارتباط دارد و این مسئله پر واضح است زیرا هرچه تعداد کل بازدید ها بیشتر باشد به تبع آن بازدید ها در هر ماه نیز بیشتر است. و اگر بیش از 0/66 باشد یعنی ارتباط قوی وجود دارد و می توان یکی را حذف کرد که در این dataset چنین ارتباطی وجود ندارد.

### 14-3- صاحبخانه های پر مشغله

یکی از موضوعاتی که این چالش قصد بررسی آن را دارد؛ این است که کدام یک از صاحبخانه ها پر مشغله تر هستند. برای این منظور قطعه کد زیر را در نظر گرفته ایم که در آن 10 صاحبخانه ای که بیشترین گردشگر را داشته اند نمایش داده شده است. ما در اینجا dataset را براساس صاحبخانه ها گروه بندی کرده ایم و سپس تعداد گردشگر را برای هر صاحبخانه محاسبه کرده و ده صاحبخانه برتر را نمایش داده ایم.

#### Which of the ten hosts has the most reviews?

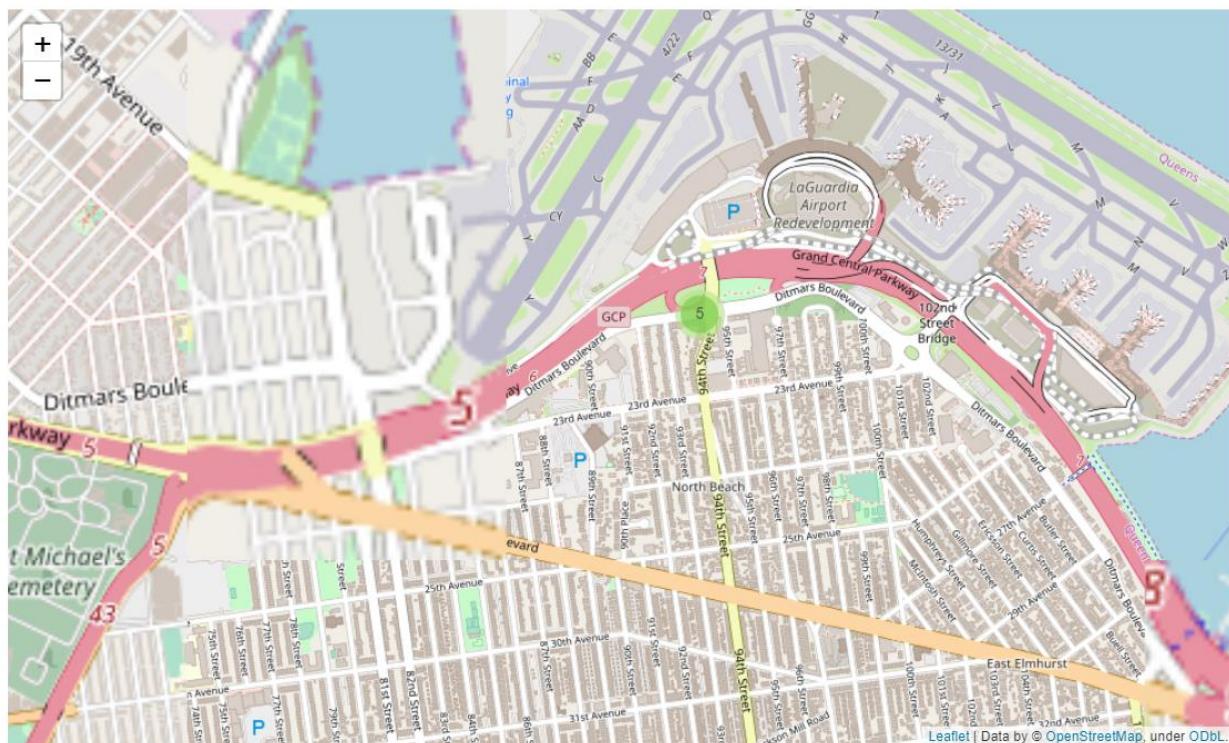
```
airbnb_for_busy_host=pd.read_csv('C:\\\\Users\\\\PCROOM\\\\Documents\\\\python data sets\\\\AB_NYC_2019.csv')
mean_of_price=airbnb_for_busy_host.price.mean()
top_host = airbnb_for_busy_host.groupby('host_id').agg({'host_name': 'max','number_of_reviews' : 'sum','price': 'mean','neighbourhood_group': 'count'})
print(mean_of_price)
top_host
```

152.7206871868289

host_id	host_name	number_of_reviews	price	neighbourhood_group	calculated_host_listings_count
37312959	Maya	2273	42.600000	Queens	5
344035	Brooklyn& Breakfast -Len-	2205	74.615385	Brooklyn	13
26432133	Danielle	2017	47.200000	Queens	5
35524316	Yasu & Akiko	1971	186.818182	Manhattan	11
40176101	Brady	1818	74.714286	Brooklyn	7
4734398	JJ	1798	49.000000	Manhattan	3
16677326	Alex And Zeena	1355	85.000000	Manhattan	12
6885157	Randy	1346	56.733333	Brooklyn	15
219517861	Sonder (NYC)	1281	253.195719	Manhattan	327
23591164	Angela	1269	65.000000	Queens	4

### شکل(29-3) صاحب خانه هایی با بیشترین گردش

همانطور که در شکل (29-3) نشان داده شده است maya با 2273 گردشگر؛ محبوب ترین صاحب خانه بوده که هزینه هر شب اقامت در خانه های او به طور متوسط 42 دلار است که نسبت به متوسط قیمت هر شب اقامت در کل dataset 152 دلار؛ بسیار پایین است و جزو خانه ای ارزان قیمت به حساب می آید. البته باید توجه داشت تعداد خانه های ارائه شده توسط هر صاحبخانه با محبوبیت او ارتباط تنگاتنگی ندارد. مثلا Sonder با 327 خانه در رتبه نهم قرار گرفته، حال آنکه maya تنها 5 خانه ارائه کرده است نکته دیگر اینکه این صاحب خانه ها در شهر های محبوب و پرطرفدار هستند. نکته دیگری که باید بررسی کرد امکاناتی است که در نزدیکی هر خانه وجود دارد؛ ما این کار را برای خانه های ارائه شده توسط maya انجام داده ایم.



شکل(30-3) خانه های ارائه شده توسط maya

با توجه به شکل بالا نزدیک بودن به فرودگاه و دسترسی به مسیر های اصلی نیز می تواند از عوامل محبوبیت خانه های این صاحب خانه باشد.

### 15-3- لغت های پرتکرار در معرفی مکان های اقامتی

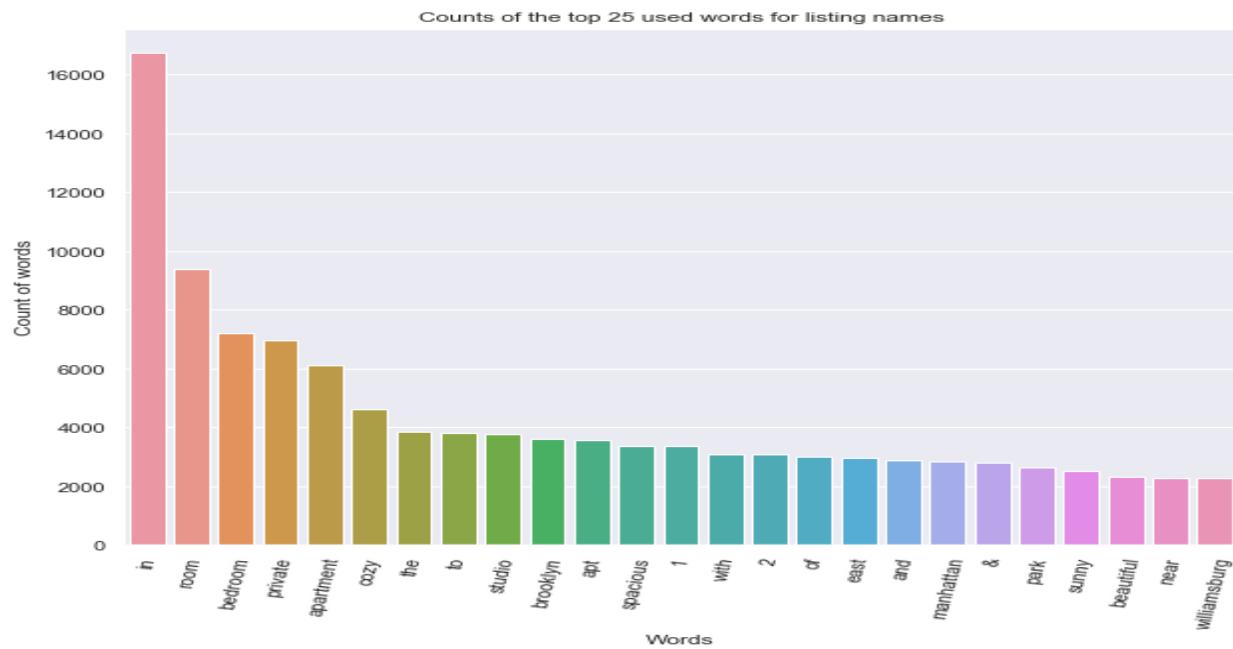
ما در این بخش میخواهیم تحلیلی هرچند مختصر بر ستون `name` داشته باشیم و ببینیم صاحبان مکان های اقامتی بیشتر از چه واژه هایی برای معرفی خانه خود استفاده می کنند. لذا در قطعه کد زیر ما ابتدا لغات موجود در ستون `name` را مورد بررسی قرار داده و فراوانی هر لغت را به دست می آوریم. و سپس برای درک بهتر، نمودار میله ای که فراوانی هر لغت را نشان دهد رسم میکنیم. باید توجه داشت که بعضی از این لغات بار معنایی ندارند مثل حروف اضافه، & و ...

#### which word are more use in column name

```
#initializing empty list to put value of name column to it each record in one cell
_names_=[]
#getting name strings from the column and appending it to the list
for name in airbnb.name:
    _names_.append(name)
#setting a function that will split those name strings into separate words
def split_name(name):
    spl=str(name).split()
    return spl
#initializing empty list where we are going to have words counted
_names_for_count_=[]
#getting name string from our list and using split function, later appending to list above
for x in _names_:
    for word in split_name(x):
        word=word.lower()
        _names_for_count_.append(word)

# use counter:A counter is a container that stores elements as dictionary keys, and their counts are stored as dictionary values
from collections import Counter
#let's see top 25 used words by host to name their listing
_top_25_w=Counter(_names_for_count_).most_common(25)#list of tupel
#convert it to DataFrame
sub_w = pd.DataFrame(_top_25_w,columns=['Words','Count'])

#we are going to use barplot for this visualization
viz_5=sns.barplot(x='Words', y='Count', data=sub_w)
viz_5.set_title('Counts of the top 25 used words for listing names')
viz_5.set_ylabel('Count of words')
viz_5.set_xlabel('Words')
viz_5.set_xticklabels(viz_5.get_xticklabels(), rotation=80);
```



شکل (31-3) لغات پر تکرار در dataset

### 16-3- price - تحلیل ویژگی

یکی از مهم ترین ویژگی های این dataset، ویژگی price است، که قیمت هر شب اقامت در آن مکان اقامتی را نشان می دهد. ما در این قسمت سعی داریم تحلیل جامعی براین ویژگی داشته باشیم.

در این بخش ما ابتدا dataset را براساس شهرها به پنج قسمت جداگانه تفکیک کردیم (هر قسمت شامل داده های یکی از پنج شهری است که در قسمت های قبل معرفی شده است) و با استفاده از تابع describe برخی مولفه های آماری را برای ویژگی price در هر یک از این بخش ها محاسبه کردیم.

## Check prices for places by neighbourhood\_group

```
#Brooklyn *****
sub_1 = airbnb.loc[airbnb['neighbourhood_group'] == 'Brooklyn' ]
sub_1 = sub_1[sub_1['price'] != 0]
#Brooklyn's list price
price_sub1=sub_1[['price']]
#Manhattan*****
sub_2 = airbnb.loc[airbnb['neighbourhood_group'] == 'Manhattan' ]
sub_2 = sub_2[sub_2['price'] != 0]
#Manhattan's list price
price_sub2=sub_2[['price']]
#Queens ****
sub_3 = airbnb.loc[airbnb['neighbourhood_group'] == 'Queens' ]
sub_3 = sub_3[sub_3['price'] != 0]
#Queens's list price
price_sub3=sub_3[['price']]
#Staten Island ****
sub_4 = airbnb.loc[airbnb['neighbourhood_group'] == 'Staten Island' ]
sub_4 = sub_4[sub_4['price'] != 0]
#Staten Island's list price
price_sub4=sub_4[['price']]
#Bronx ****
sub_5 = airbnb.loc[airbnb['neighbourhood_group'] == 'Bronx' ]
sub_5 = sub_5[sub_5['price'] != 0]
#Bronx's list price
price_sub5=sub_5[['price']]
#making list of price by neighbourhood_group
price_list_by_n=[price_sub1, price_sub2, price_sub3, price_sub4, price_sub5]
```

```
#creating an empty list that we will append later with price distributions for each neighbourhood_group
p_l_b_n_2=[]
#creating list with known values in neighbourhood_group column
nei_list=['Brooklyn', 'Manhattan', 'Queens', 'Staten Island', 'Bronx']
#creating a for loop to get statistics for price ranges and append it to our empty list
for x in price_list_by_n:
    i=x.describe()
    i.reset_index(inplace=True)# index become a column and new index (0,1,...) replaced dataframe modify not new
    i.rename(columns={'index':'Stats'}, inplace=True)#if inplace is true the original dataframe Change not creating new dataframe
    p_l_b_n_2.append(i)
#changing names of the price column to the area name for easier reading of the table
p_l_b_n_2[0].rename(columns={'price':nei_list[0]}, inplace=True)
p_l_b_n_2[1].rename(columns={'price':nei_list[1]}, inplace=True)
p_l_b_n_2[2].rename(columns={'price':nei_list[2]}, inplace=True)
p_l_b_n_2[3].rename(columns={'price':nei_list[3]}, inplace=True)
p_l_b_n_2[4].rename(columns={'price':nei_list[4]}, inplace=True)
#finalizing our dataframe for final view
stat_df=p_l_b_n_2 #stat_df has 5 dataframes
stat_df=df.set_index('Stats') for df in stat_df:#df is each dataframe we want to make stats's column as general index one stats
stat_df=stat_df[0].join(stat_df[1:])
print('This DataFrame shows the statistical descriptions for the price of the list in each neighbourhood_group ')
stat_df
```

This DataFrame shows the statistical descriptions for the price of the list in each neighbourhood\_group

	Brooklyn	Manhattan	Queens	Staten Island	Bronx
Stats					
count	20095.000000	21660.000000	5666.000000	373.000000	1090.000000
mean	124.438915	196.884903	99.517649	114.812332	87.577064
std	186.896837	291.386838	167.102155	277.620403	106.725371
min	10.000000	10.000000	10.000000	13.000000	10.000000
25%	60.000000	95.000000	50.000000	50.000000	45.000000
50%	90.000000	150.000000	75.000000	75.000000	65.000000
75%	150.000000	220.000000	110.000000	110.000000	99.000000
max	10000.000000	10000.000000	10000.000000	5000.000000	2500.000000

### شکل (31-3) بررسی ویژگی قیمت به تفکیک شهرها

سطر count در این dataset تعداد رکورد های موجود در dataset اصلی به ازای هر شهر را نشان می دهد. همانطور که دربخش های قبل نیز گفته شده بود manhattan با 21661 رکورد، بیشترین مکان را برای جاره دارد.

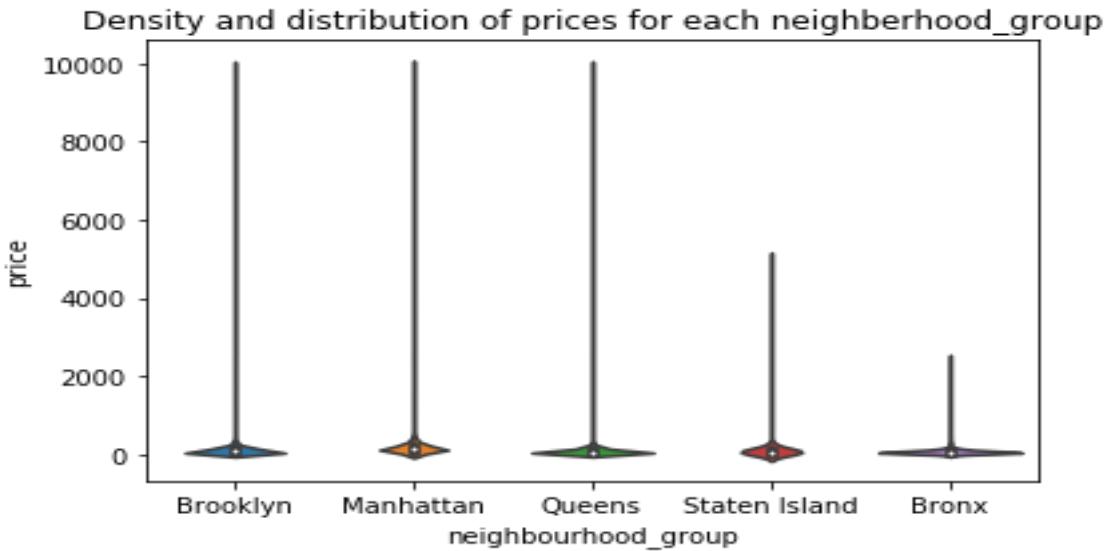
در سطر بعدی میانگین قیمت ها برای مکان های اقامتی در هر شهر محاسبه شده، همانطور که مشاهده میشود شهر manhattan با میانگین قیمت 196 دلار پرهزینه ترین خانه ها را دارد و پس از آن Brooklyn با میانگین قیمت 124 دلار قرار دارد. پس این دو شهر در عین حال که پرگردشگر و محبوب هستند گران قیمت هم هستند !! البته این اتفاقی رایج در شهر های توریستی است

با میانگین قیمت 87-دلار ارزان ترین شهر در ارائه مکان است.

در سطر بعدی انحراف معیار برای قیمت های هر شهر محاسبه شده است. هر چه این عدد بیشتر باشد یعنی قیمت های این شهر پراکندگی بیشتری از میانگین دارند این مقدار در شهر manhattan بیشینه است یعنی قیمت ها در این شهر از میانگین فاصله زیادی دارند؛ به عبارت دیگر خانه های زیادی وجود دارند که خیلی ارزان قیمت تر از میانگین هستند و همچنین خانه های زیادی هم هستند که گران قیمت تر از میانگین هستند ولی در Bronx داده ها به میانگین نزدیک تر اند و پراکندگی قیمت کمتر است. همچنین در این dataset چارک های داده ها هم محاسبه شده اند؛ مثلا در شهر Staten Island 25 درصد داده ها کمتر از 50 دلار هستند و نیمی از داده ها کمتر از 75 دلار و 75 درصد داده ها کمتر از 110 دلار به عبارت دیگر 75 درصد از داده ها از میانگین کوچک تر اند و تنها 25 درصد از داده ها با فاصله زیادی از میانگین بیشتر اند و همین عامل باعث انحراف معیار زیاد در این شهر شده است. برای بررسی بیشتر این جدول از نمودار violinplot که نمودار جعبه ای را به همراه چگالی داده ها نشان میدهد استفاده کرده ایم.

#### Check without dropping noise data

```
viz=sns.violinplot(data=airbnb, x='neighbourhood_group', y='price')
viz.set_title('Density and distribution of prices for each neighborhood_group');
```



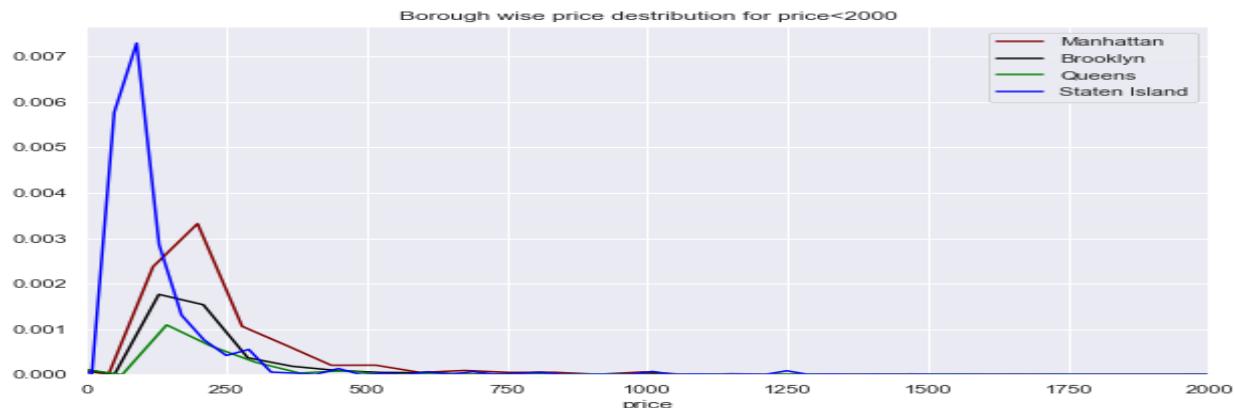
شکل (32-3) بررسی ویژگی قیمت با وجود داده های نویزی

همانطور که در قسمت قبلی گفته شد 75 درصد داده ها در تمامی شهر ها کوچکتر یا نزدیک به میانگین اند و 25 درصد باقی مانده فاصله زیادی از میانگین دارند که در شکل بالا نیز این مسئله قابل مشاهده است (خط رسم شده بعد از قسمت رنگی فاصله ی 75 درصد داده ها تا ماکزیمم داده ها را نشان می دهد یعنی 25 درصد انتهایی داده ها) و درک الگوی داده ها را سخت کرده همانطور که در بخش های قبل نیز گفته شد به این داده ها خارج از محدوده یا Outlier می گویند زیرا با الگوی قالب که اکثر داده ها از آن پیروی می کنند تفاوت زیادی دارند و در بررسی های آماری برای فهم بهتر، از مسئله کنار گذاشته می شوند. در این بخش ما با رسم نمودار شکل زیر دریافتیم که داده های آماری پس از 500 دلار تقریبا ناچیز هستند و میتوان آنها را نادیده گرفت؛

```

plt.figure(figsize=(10,6))
sns.distplot(airbnb[airbnb.neighbourhood_group=='Manhattan'].price,color='maroon',hist=False,label='Manhattan')
sns.distplot(airbnb[airbnb.neighbourhood_group=='Brooklyn'].price,color='black',hist=False,label='Brooklyn')
sns.distplot(airbnb[airbnb.neighbourhood_group=='Queens'].price,color='green',hist=False,label='Queens')
sns.distplot(airbnb[airbnb.neighbourhood_group=='Staten Island'].price,color='blue',hist=False,label='Staten Island')
sns.distplot(airbnb[airbnb.neighbourhood_group=='Long Island'].price,color='lavender',hist=True,label='Long Island')
plt.title('Borough wise price distribution for price<2000')
plt.xlim(0,2000)
plt.show()

```

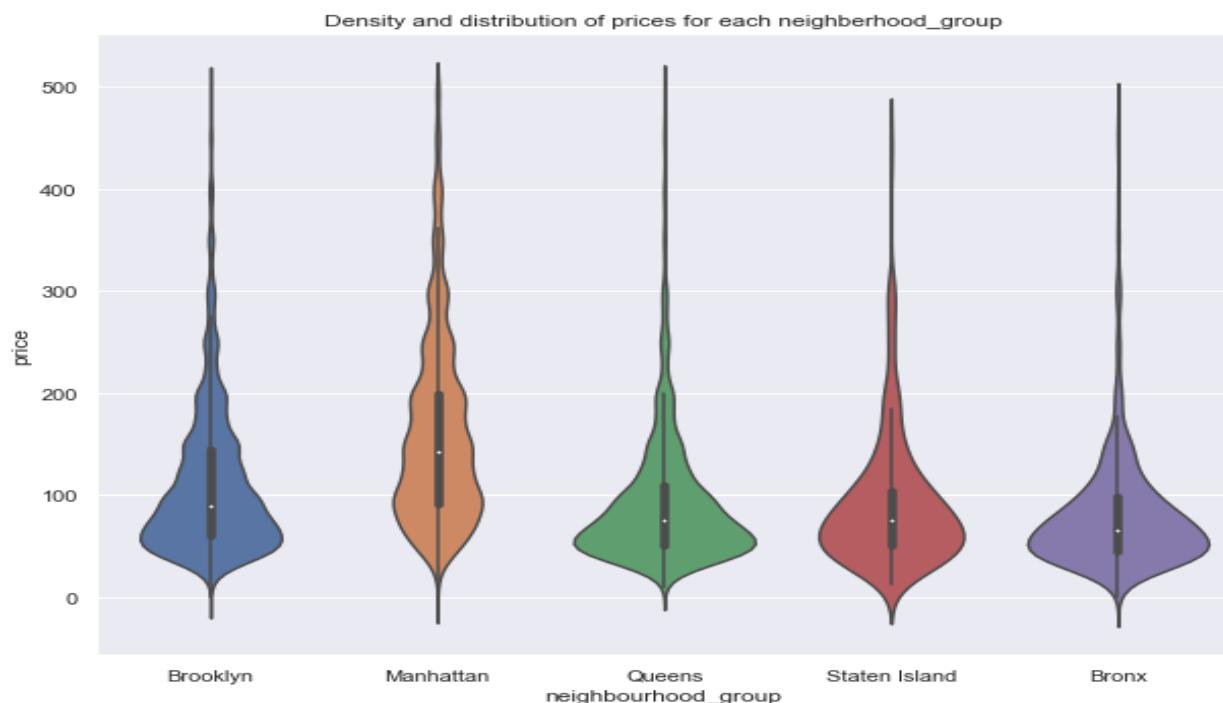


شکل (33-3) تشخیص داده نویزی

لذا با محدود کردن رنج قیمت نمودار واضح تری خواهیم داشت:

#### Check with dropping noise data

```
#we can see from our statistical table that we have some extreme values, therefore we need to remove them
#creating a sub-dataframe with no extreme values / less than 500
sns.set(rc={'figure.figsize':(10,8)})
sub_6=airbnb[airbnb.price < 500]
#using violinplot to showcase density and distribution of prices
viz_2=sns.violinplot(data = sub_6 , x='neighbourhood_group', y='price')
viz_2.set_title('Density and distribution of prices for each neighborhood_group');
```



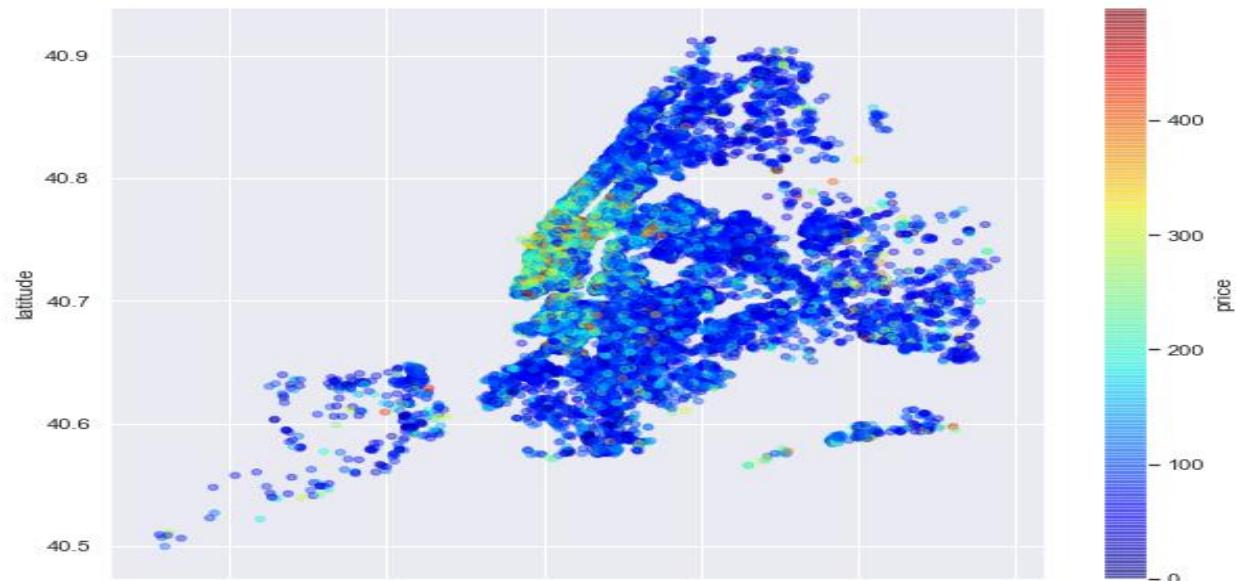
شکل (34-3) بررسی ویژگی قیمت با حذف داده های نویزی

پر واضح است که درک و بررسی این نمودار به مراتب راحت تر از نمودار شکل (32-3) است. کشیدگی نمودار شهر staten istland و manhattan نشان دهنده ای انحراف معیار زیاد در این دو شهر است و همانطور که در بخش قبلی به آن اشاره شد، کوتاه و پر بودن نمودار شهر Bronx دلیلی بر نزدیک بودن مقادیر داده ها در این شهر است. نقطه سفید رنگ در مرکز هرشکل میانه رانشان میدهد و مشاهده میشود در شهر manhattan میانه داده ها بیشینه است و همانطور که قبل از آن اشاره شد این شهر خانه های گران قیمت تری نسبت به سایر شهر ها دارد.

وحال میخواهیم به بررسی این موضوع بپردازیم که خانه های گران قیمت در کدام طول و عرض جغرافیایی قرار گرفته اند و پراکندگی آنها چگونه است. در نمودار شکل زیر این مسئله به خوبی نشان داده شده است.

## plot of place and price

```
# what we can do with our given longitude and latitude columns
viz_4=sub_6.plot(kind='scatter', x='longitude', y='latitude', c='price',
                  cmap=plt.get_cmap('jet'), colorbar=True, alpha=0.4, figsize=(10,8))
```



شکل (34-3) بررسی پراکندگی با توجه به قیمت

## فصل 4 :

پیشگویی قیمت با الگوریتم های  
یادگیری ماشین

در این فصل سعی داریم مدلی برای پیش بینی قیمت ارائه کنیم. لذا پس از انجام پیش پردازش برروی داده ها و آماده سازی `dataset`، به معرفی دو الگوریتم از الگوریتم های یادگیری ماشین می پردازیم و با استفاده از این دو الگوریتم قیمت را پیش بینی خواهیم کرد.

همانطور که در فصل 2 نیز اشاره شد؛ کتابخانه `sklearn` شامل بسیاری از الگوریتم های یادگیری ماشین است و از کتابخانه های بسیار مهم در این زمینه به شمار می آید. در این بخش ما از متدهای مختلفی از این کتابخانه، جهت پیش پردازش و به کار گیری الگوریتم های یادگیری ماشین استفاده کرده ایم.

## Price prediction with Machine Learning

```
#machine learning module
import sklearn
from sklearn import preprocessing
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
from sklearn.ensemble import GradientBoostingRegressor
```

شكل (1-4) اضافه کردن کتابخانه `sklearn`

## 1-4-پیش پردازش داده ها (2)

همانطور که در بخش های قبل نیز اشاره شد ویژگی `name` تاثیر چندانی در مدل پیش بینی ما نخواهد داشت لذا برای جلوگیری از تاثیر سوء این ویژگی در مدل پیش بینی آن را حذف خواهیم کرد.

**Removing the name attribute has no effect on the model build**

```
# Preparing the data
airbnb.drop(['name'],axis=1,inplace=True)
```

شكل (2-4) حذف کردن ویژگی `name`

## 1-1-4 کد گذاری برخی از ویژگی ها

یکی از کارهای رایج در پیش پردازش داده ها کد گذاری ویژگی هایی است که این امکان در آنها وجود داشته باشد؛ زیرا این اقدام کار پردازش را برای کامپیوتر راحت تر خواهد کرد.

در واقع در ویژگی که مقادیر آن ها محدود هستند؛ برای مثال ستون room\_type هر چند داده های زیادی دارد ولی این داده ها تنها سه مقدار private room , Entire home/apt ,shared room private room تمامی مقادیر جایگزین میکنیم . برای مثال در ویژگی room\_type تمامی مقادیر با 1 جایگزین میشوند. این کار بار پردازشی را کمتر خواهد کرد. در این قسمت ما سه ویژگی neighbourhood و neighbourhood\_group و room\_type را کد گذاری می کنیم. پس از اعمال این تغییر ستون room\_type دارای سه مقدار 0 و 1 و 2 خواهد بود. همانطور ستون neighbourhood\_group دارای پنج مقدار عددی از 0 تا 4 خواهد بود. همانطور که واضح است این اقدام در مورد ویژگی های دیگر محدود نیست .

#### Encoding neighbourhood\_group , neighbourhood , room\_type

```
'''Encode labels with value between 0 and n_classes-1.'''
le = preprocessing.LabelEncoder() # Fit Label encoder
le.fit(airbnb['neighbourhood_group'])
airbnb['neighbourhood_group']=le.transform(airbnb['neighbourhood_group']) # Transform Labels to numbers

le = preprocessing.LabelEncoder()
le.fit(airbnb['neighbourhood'])
airbnb['neighbourhood']=le.transform(airbnb['neighbourhood'])

le = preprocessing.LabelEncoder()
le.fit(airbnb['room_type'])
airbnb['room_type']=le.transform(airbnb['room_type'])

airbnb.sort_values(by='price',ascending=True,inplace=True)

airbnb.head()
```

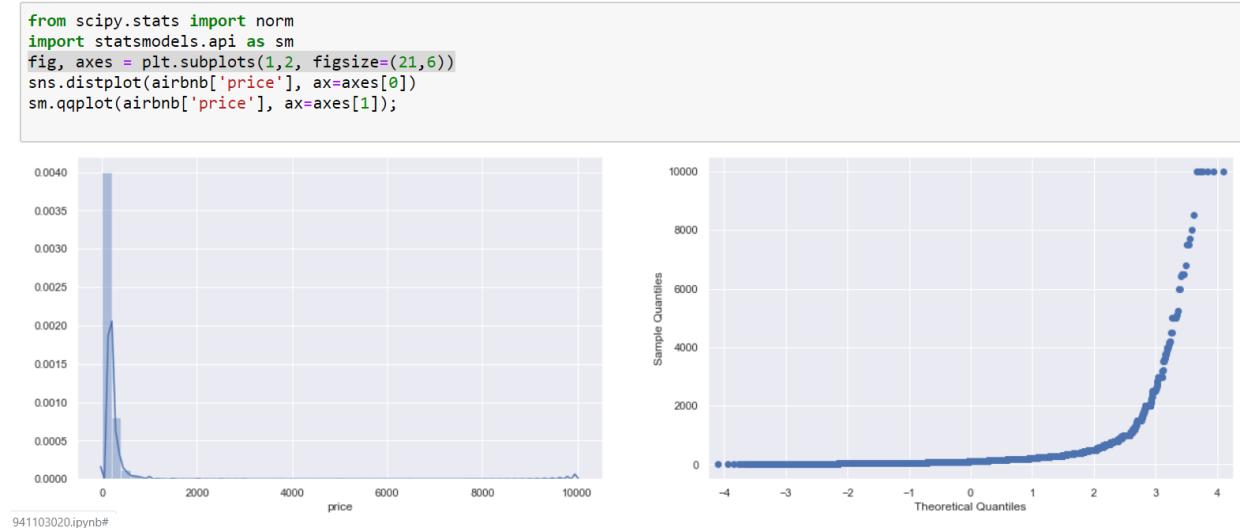
host_id	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews
19761	16494382	3	166	40.70735	-73.89343	1	1.397363	7
21196	2092314	1	60	40.64471	-73.94950	2	1.397363	30
41066	51596474	1	89	40.58737	-73.96882	2	1.397363	7
39805	1532337	0	203	40.84047	-73.87127	2	1.397363	14
21695	117941939	1	214	40.71269	-73.96409	1	1.397363	14
21733	86108833	0	173	40.84034	-73.83007	2	1.397363	1
21382	40134417	1	91	40.73076	-73.95575	0	1.397363	4
40200	129022496	1	13	40.68093	-73.94922	1	1.397363	10
37093	222098649	3	105	40.68547	-73.79063	0	1.397363	1
43894	274333	2	94	40.80298	-73.95375	2	1.397363	1

Encoding neighbourhood\_group , neighbourhood , room\_type (3-4) شکل

## 2-1-4 نرمال کردن ویژگی price

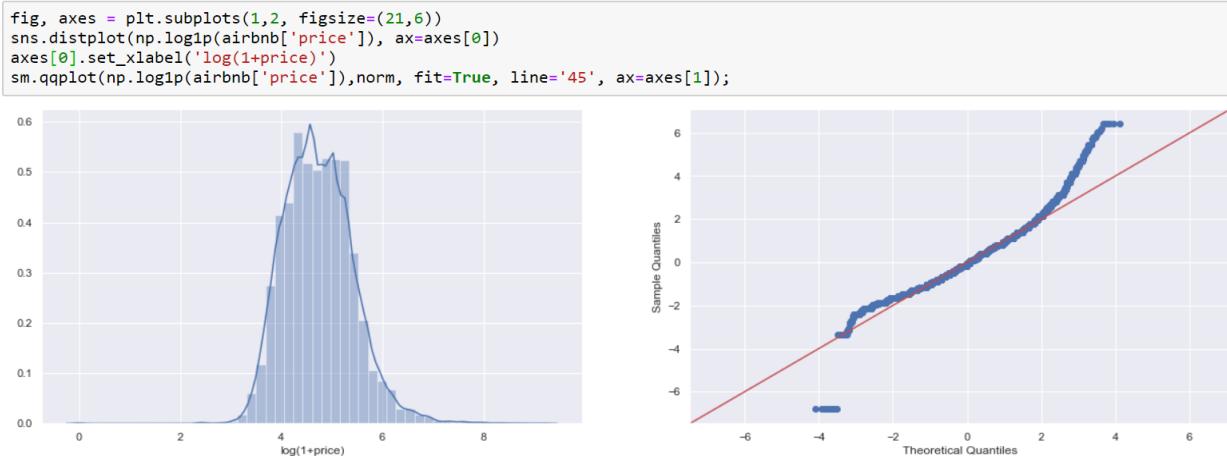
در این بخش قصد داریم توزیع فعلی ویژگی `price` را مشاهده کرده و سپس در صورت نیاز آن را با استفاده از تبدیل های مختلف به توزیع نرمال نزدیک کنیم.

در قطعه کد زیر ابتدا نمودار هیستوگرام ویژگی `price` را رسم می کنیم؛ در صورت نرمال بودن این ویژگی نمودار فرم زنگوله ای به خود خواهد گرفت و سپس برای اطمینان بیشتر از نمودار `qqplot` برای نمایش توزیع آن استفاده خواهیم کرد در صورت نرمال این نمودار این خط را نشان خواهد داد.



شکل (4-4) بررسی نرمال بودن ویژگی `price`

همانطور که در شکل فوق قابل مشاهده است این ویژگی توزیع نرمال نداشته و به اصطلاح دارای `skewd` مثبت است. برای نرمال کردن ویژگی هایی که دارای `skewd` هستند عموماً از تبدیل `log` استفاده میشود. تا بتوان آن ها را به توزیع نرمال نزدیک کرد.



شکل (5-4) بررسی نرمال بودن ویژگی price

همانطور که مشاهده میکنید این نمودار به فرم نمودارهای نرمال نزدیک شد و نقاط در نمودار qqplot نیز به یک خط راست نزدیک تر شده اند. اما هنوز بعضی از نقاط از خط راست فاصله دارند، این نقاط همان مقادیر outliers هستند که در شکل (3-32) نیز مشاهده کردید. در این قسمت برای اینکه مدل دقیق تری برای پیشگویی قیمت داشته باشیم باید این مقادیر خارج از محدوده را از dataset حذف کنیم.

```

airbnb = airbnb[np.log1p(airbnb['price']) < 8]
airbnb = airbnb[np.log1p(airbnb['price']) > 3]

```

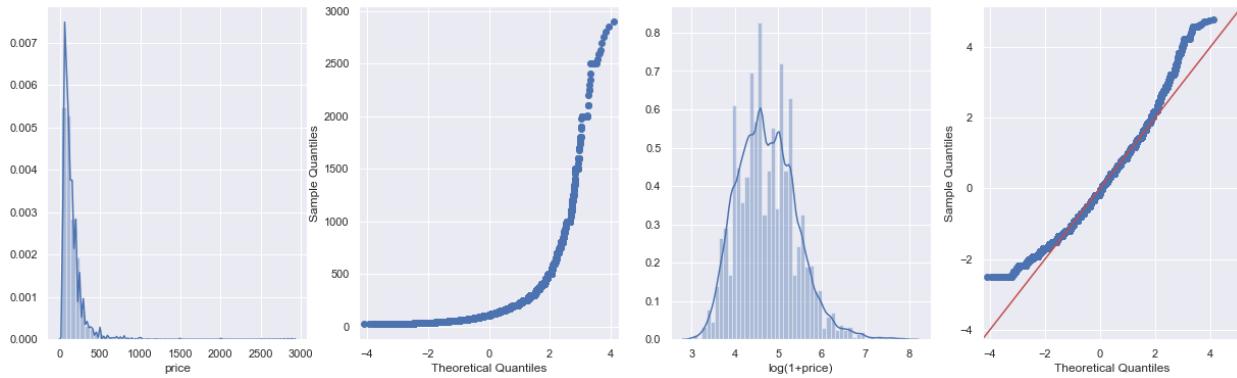
شکل (6-4) حذف مقادیر خارج از محدوده

حال چهار نمودار فوق را مجددا رسم می کنیم.

```

fig, axes = plt.subplots(1,4, figsize=(21,6))
sns.distplot(airbnb['price'], ax=axes[0])
sm.qqplot(airbnb['price'], ax=axes[1]);
sns.distplot(np.log1p(airbnb['price']), ax=axes[2])
axes[2].set_xlabel('log(1+price)')
sm.qqplot(np.log1p(airbnb['price']), norm, fit=True, line='45', ax=axes[3]);

```



شکل (7-4) بررسی نرمال بودن ویژگی price با حذف مقادیر خارج از محدوده

حال نقاط باقی مانده بیشترین تطابق را با خط راست دارند. بررسی نمودارها نشان داد که تبدیل لگاریتمی می‌تواند داده‌ها را به فرم نرمال نزدیک کند لذا این تبدیل را بر روی مقادیر ویژگی price اعمال می‌کنیم.

```
airbnb['price'] = np.log1p(airbnb['price'])
```

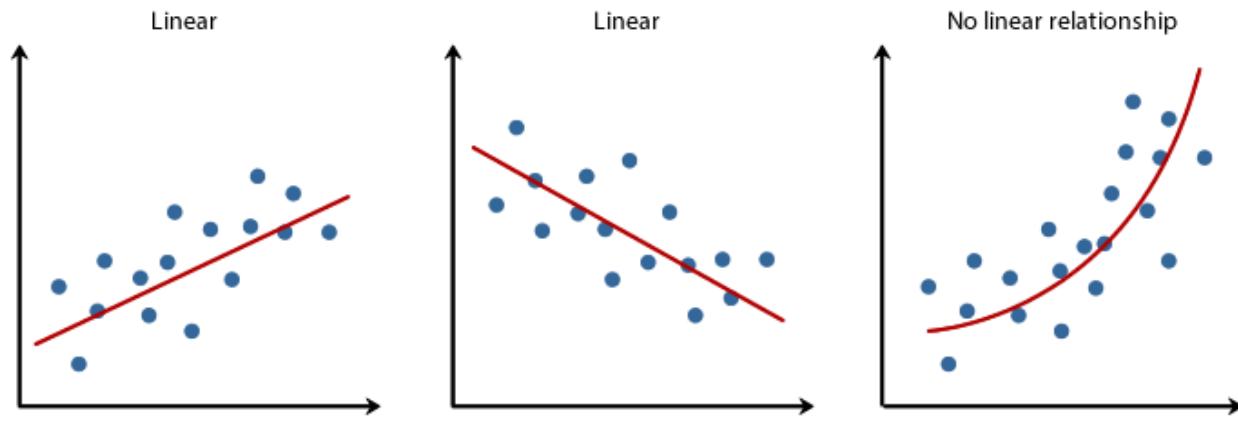
شکل (8-4) اعمال تبدیل  $\log$

## 4-2- بررسی دو الگوریتم رگرسیون

در این پروژه قصد داریم تا با استفاده از دو الگوریتم LinearRegression و Gradient Boosted Regressor به پیش‌بینی قیمت بپردازیم؛ لذا در این قسمت توضیح مختصری از نحوه عملکرد این دو الگوریتم ارائه خواهیم کرد. البته لازم به ذکر است که هر دو الگوریتم از مجموعه الگوریتم‌های رگرسیون هستند.

### 4-2-1 بررسی الگوریتم Regression

رگرسیون یک مدل پیش بینی کننده است که درباره ای رابطه بین متغیرهای مستقل و وابسته تحقیق می کند. تجزیه و تحلیل رگرسیون شامل رسم یک خط برروی مجموعه نقاط داده است، که این خط به شکل کلی داده ها نزدیک باشد و بتوان از این خط به عنوان الگویی برای پیشگویی استفاده کرد. این خط میتواند به صورت منحنی ، راست و .... باشد.



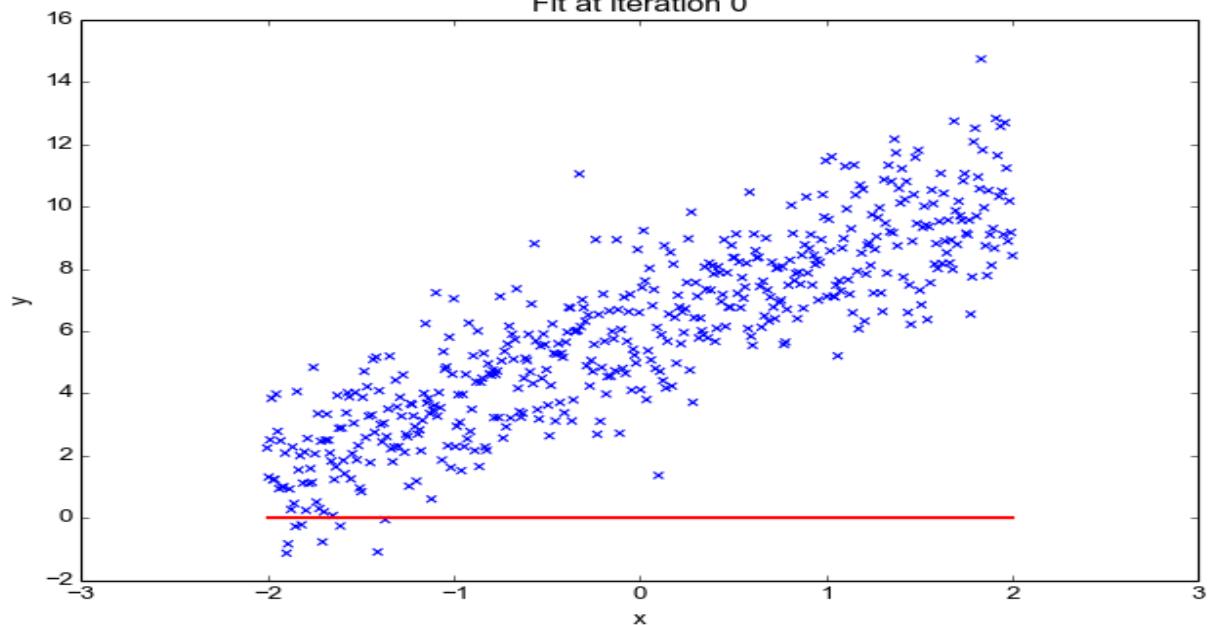
شکل (9-4) انواع رگرسیون

## 2-2-4 بررسی الگوریتم LinearRegression

در این الگوریتم ما به دنبال رابطه خطی بین متغیرها هستیم . برای سادگی فرض کنید `dataset` مایک متغیر مستقل (محور  $x$ ) و یک متغیر وابسته (محور  $y$ ) دارد. ما مقادیر این `dataset` را در یک مختصات دو بعدی رسم کرده سپس یک خط مستقیم را طوری پیدا میکنیم که بیشترین انطباق را با داده ها داشته باشد.

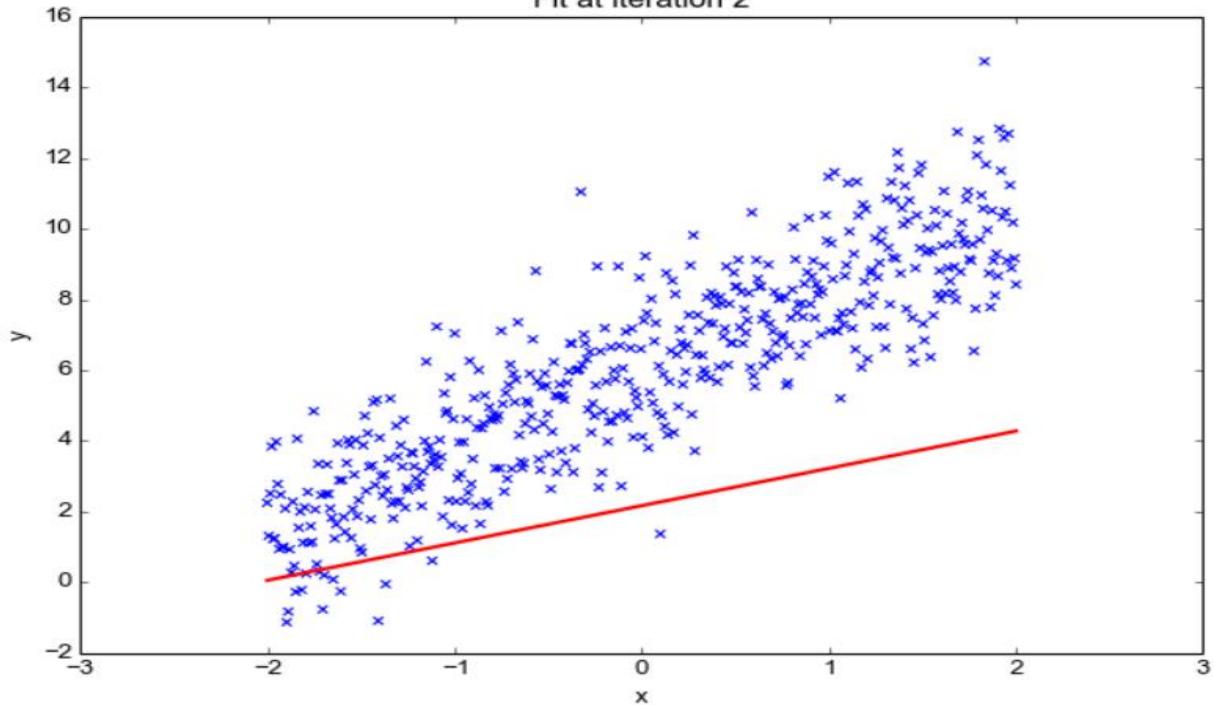
پس از یافتن این خط از معادله آن برای پیش بینی های بعدی استفاده میکنیم .

Fit at iteration 0



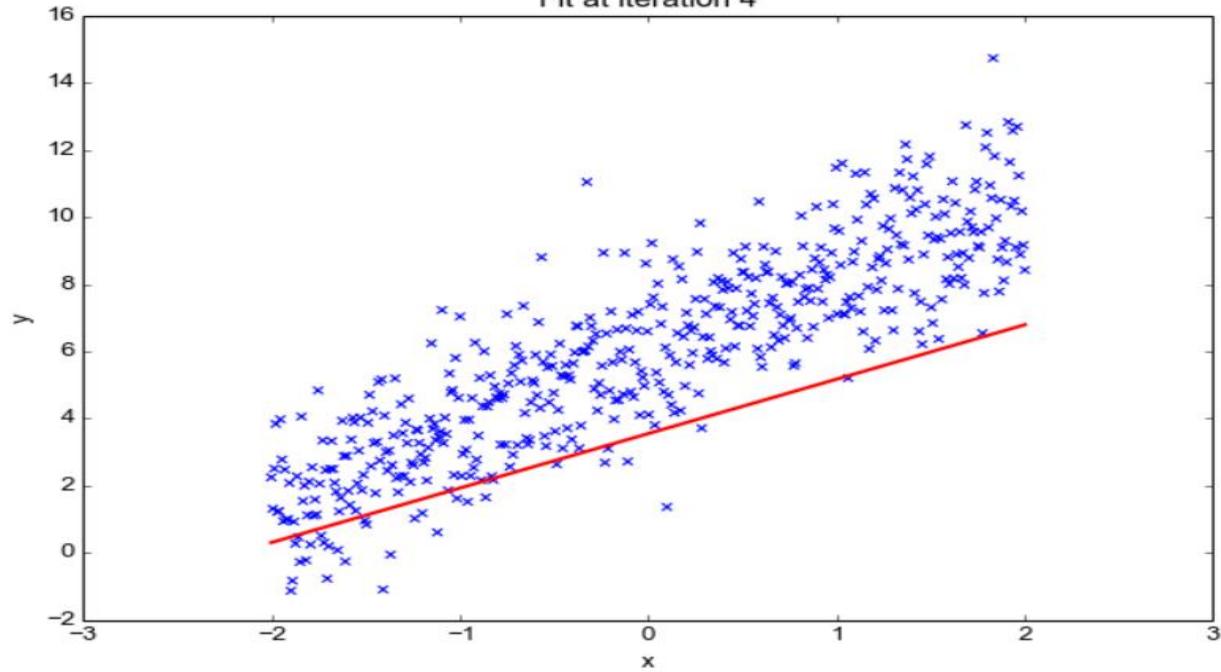
شكل (10-4) - الف

Fit at iteration 2



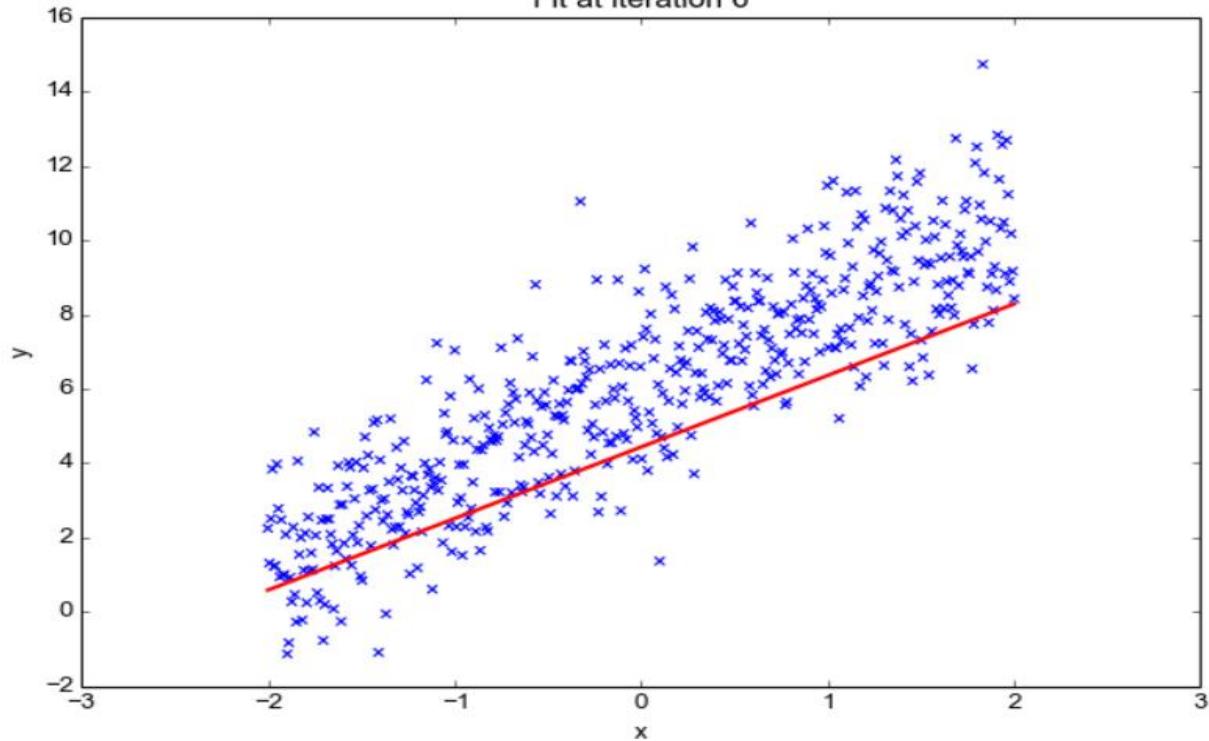
شكل (10-4) - بـ

Fit at iteration 4



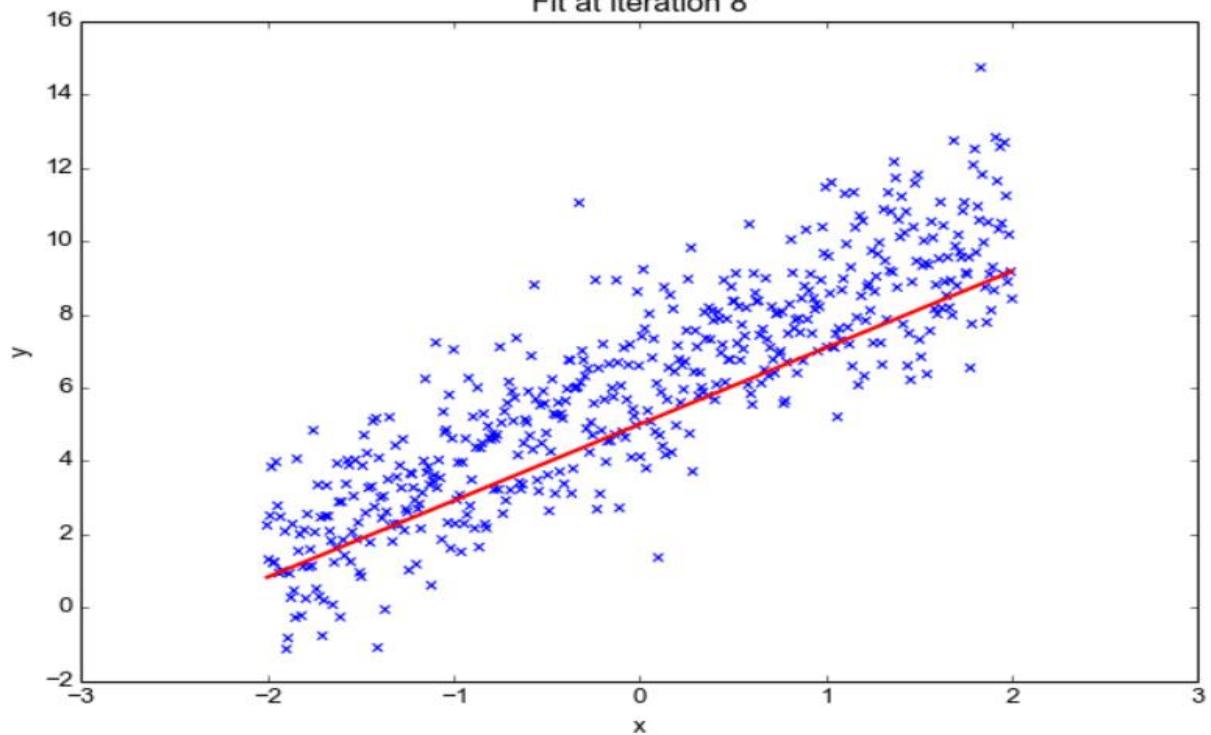
شكل (10-4) - ب

Fit at iteration 6



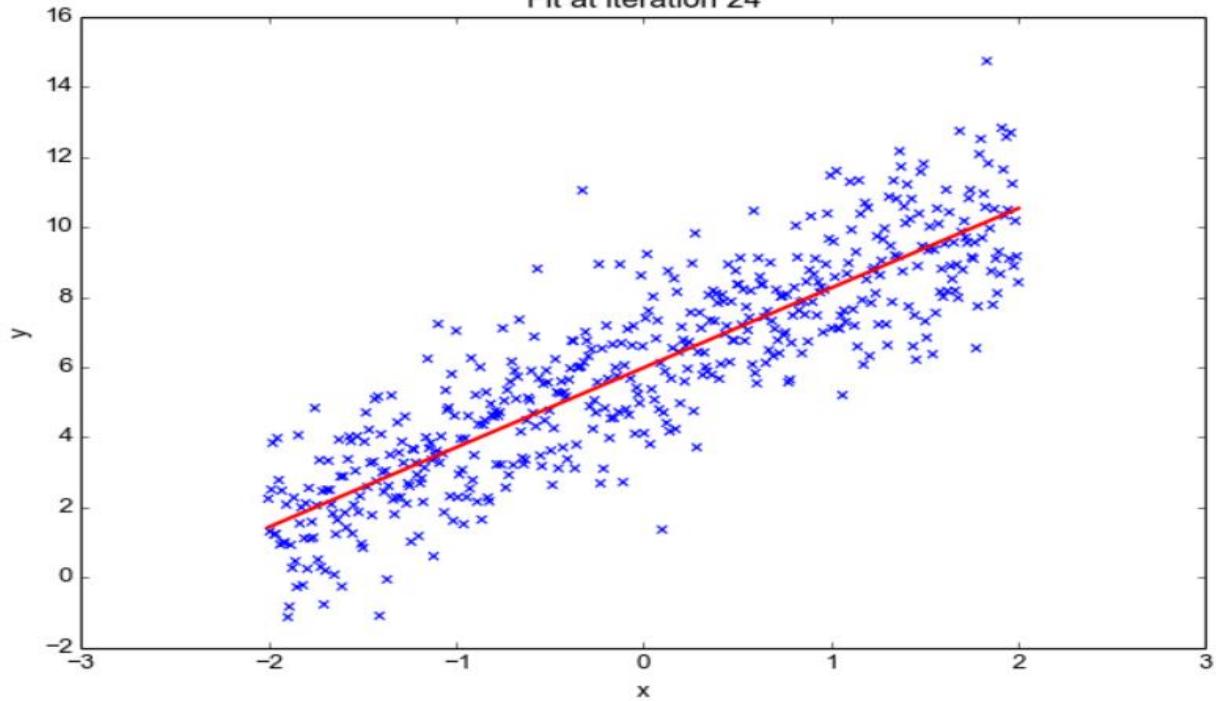
شكل (10-4) - ث

Fit at iteration 8



شكل (10-4)-ج

Fit at iteration 24



شكل (10-4)-ح

شکل (10-4) رگرسیون خطی تک متغیره

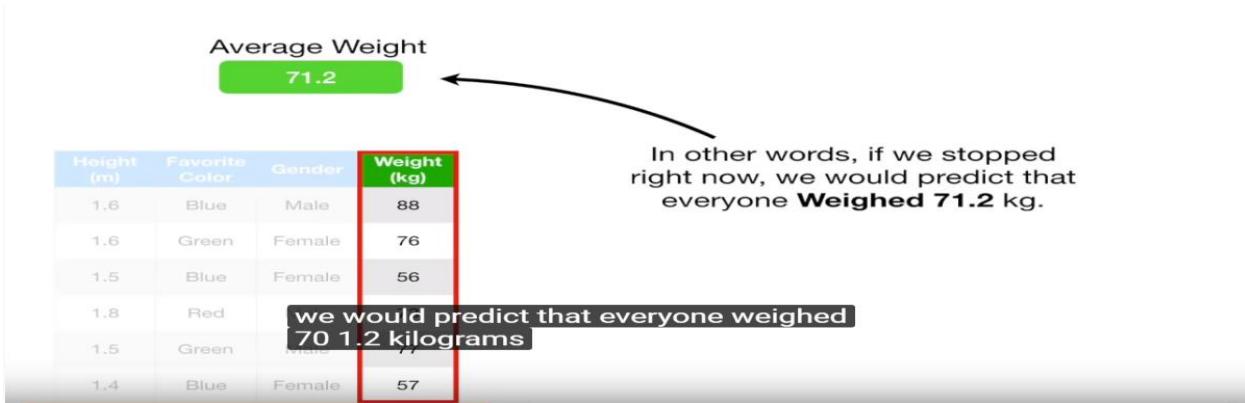
خط رسم شده در شکل (10-4)-ح بیشترین انطباق با داده ها را دارد و فاصله هر نقطه تا خط رسم شده (خطای مدل پیش بینی شده) حداقل است. اغلب مسائلی که ما در دنیای واقعی با آنها سرو کار داریم دارای بیش از یک متغیر مستقل هستند. لذا الگوریتم رگرسیون خطی، برای هر ویژگی یک ضریب به دست آورده و یک معادله، از متغیر های مستقل برای پیش بینی متغیر وابسته می سازد.

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p + \varepsilon$$

شکل (11-4) رگرسیون خطی چند متغیره

#### 3-2-4 بررسی الگوریتم Gradient Boosted Regressor

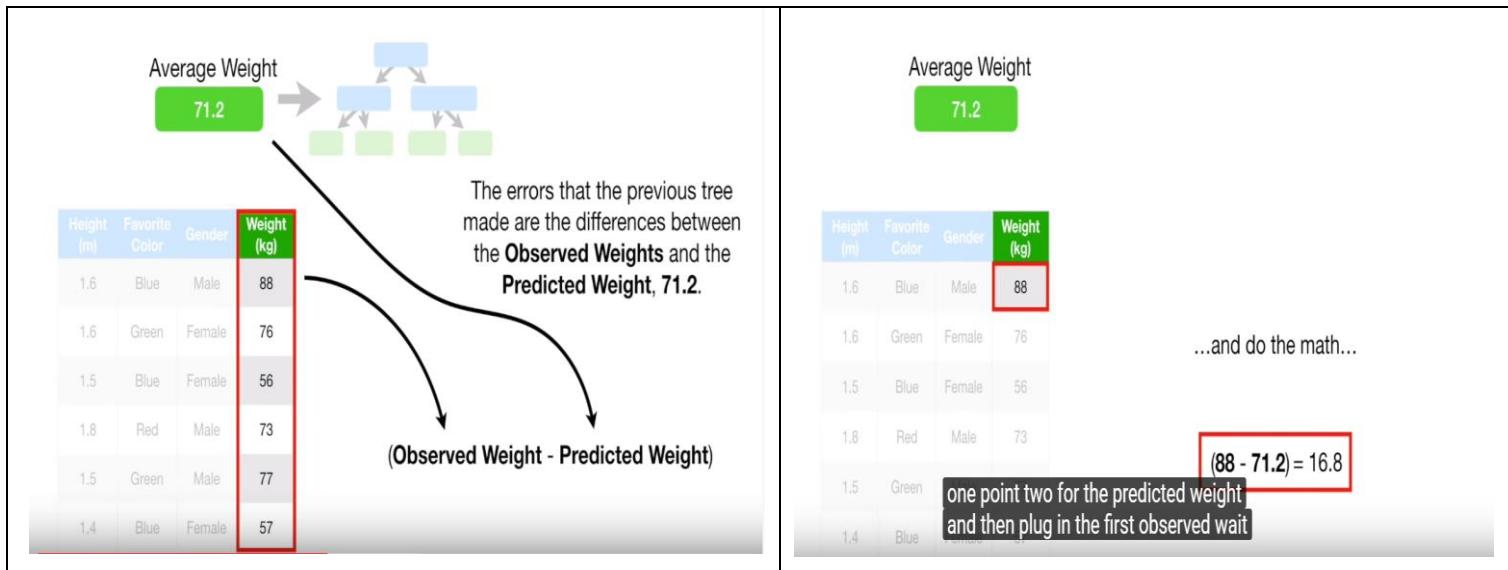
برای بررسی این الگوریتم، یک dataset را در نظر میگیریم که دارای چهار ویژگی وزن، قد، رنگ مورد علاقه و جنسیت است. و ما میخواهیم الگویی برای پیش بینی وزن براساس سه ویژگی دیگر بسازیم. در مرحله اول میانگین متغیر هدف (وزن) را محاسبه کرده و به عنوان مدل پیش بینی اولیه در نظر می گیریم. یعنی به ازای هر ورودی این مقدار را برای وزن پیش بینی میکنیم.



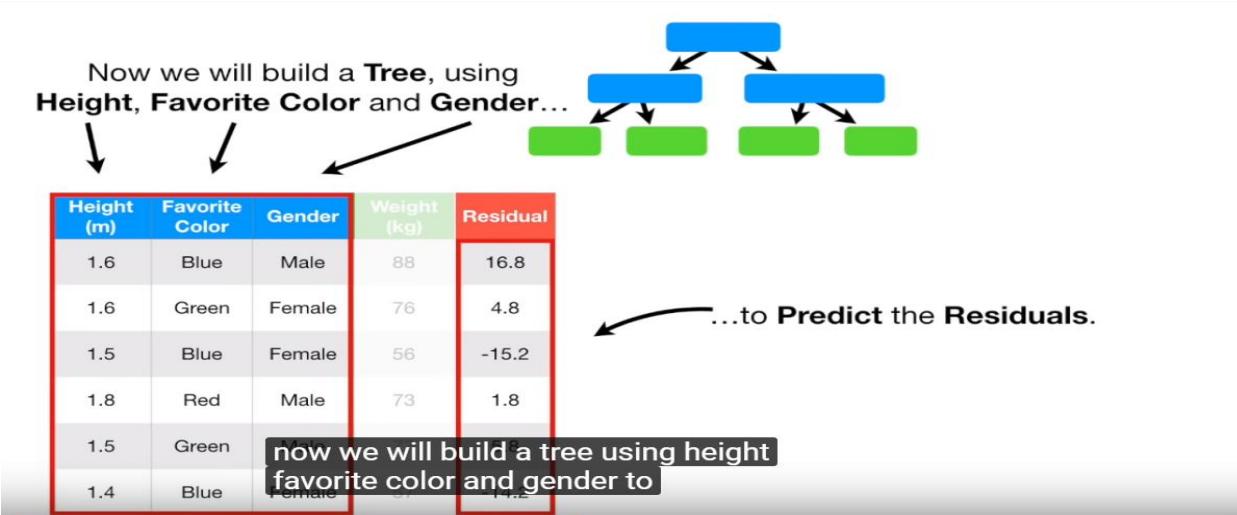
شکل (12-4) ایجاد مدل پیش بینی اولیه

حال باید خطاهای مدل قبلی را بدست آوریم. منظور از خطاء، اختلاف بین مقدار واقعی وزن و مقدار پیش بینی شده است.

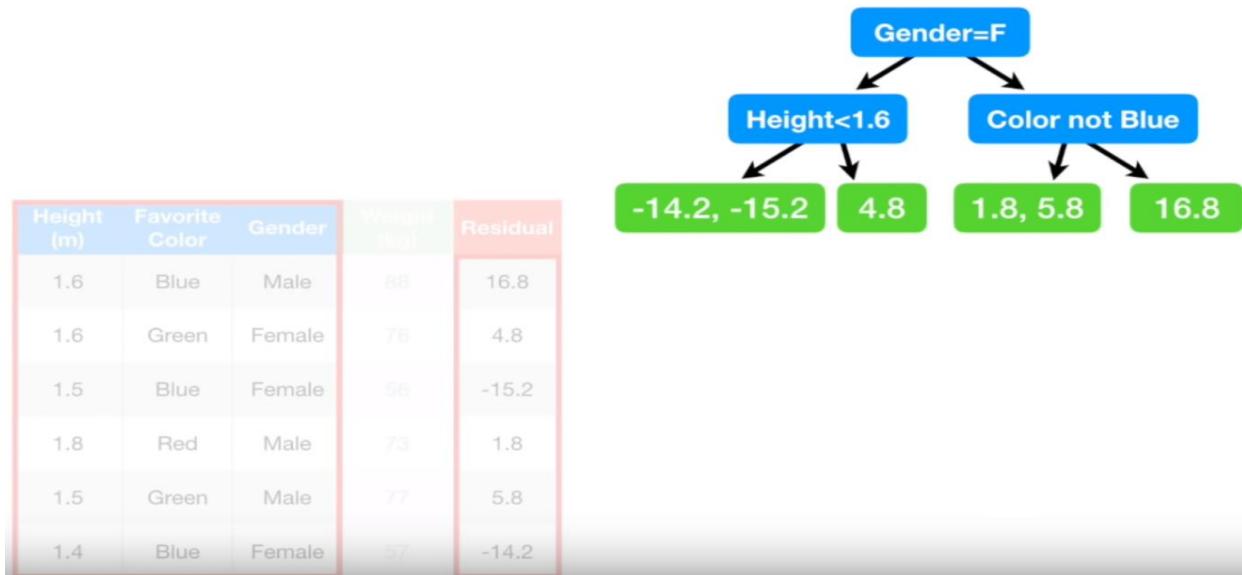
شکل (13-4) بدست آوردن خطاهای مدل اولیه



حال این مقدار محاسبه شده به ازای تمامی رکورد ها را در ستون جدیدی ذخیره می کنیم و در مرحله بعد سعی داریم درختی بسازیم که با استفاده سه ویژگی قد و رنگ مورد علاقه و جنسیت این اختلاف را پیش بینی کنیم.

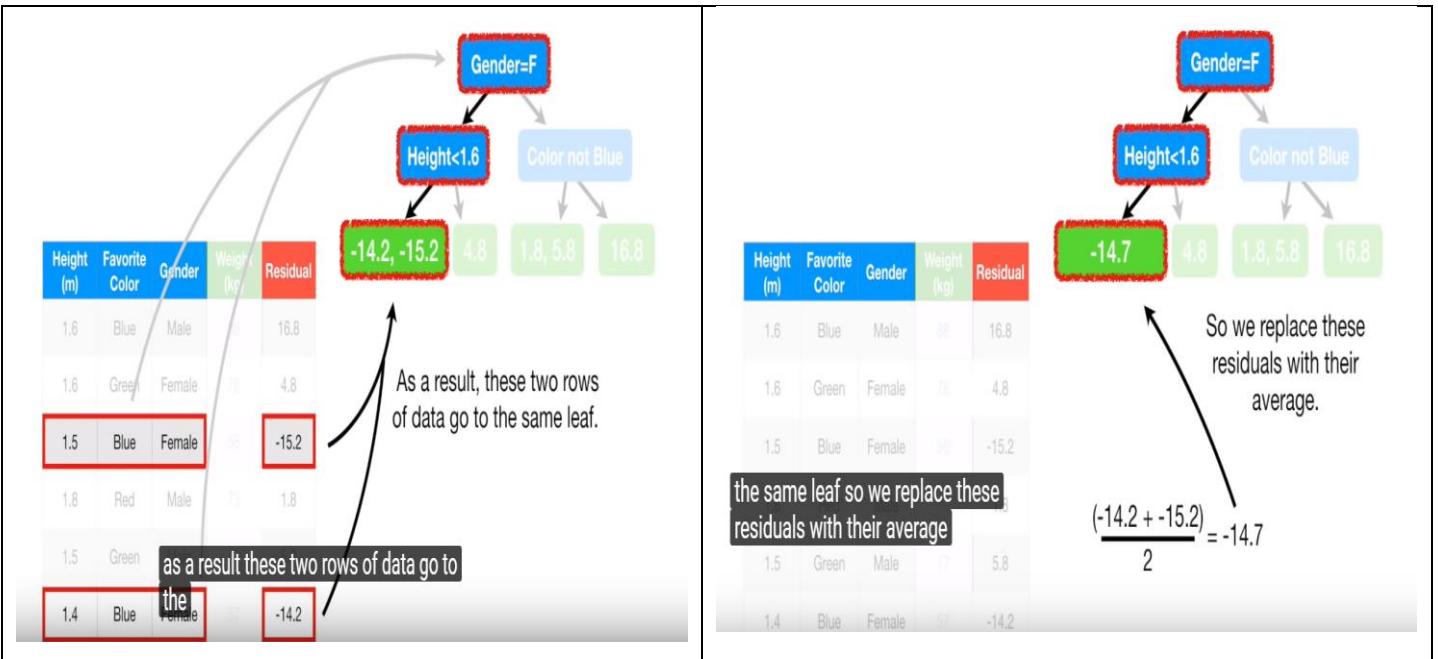


شکل (14-4) افزودن خطاهای به dataset



شکل (15-4) ایجاد درخت برای پیش‌بینی

در این شرایط بعضی از رکوردها به یک برگ می‌رسند زیرا تعداد برگ‌ها محدود است و بستگی به مسئله دارد. لذا به جای دو مقدار در یک برگ میانگین آنها را قرار می‌دهیم.



شکل (4-16) تصحیح درخت برای پیش بینی

حال این مدل پیش بینی را با مدل پیش بینی اولیه ترکیب میکنیم تا مدل دقیق تری بسازیم. همانطور که در بخش قبلی نیز به آن اشاره شد مدل پیش بینی اولیه وزن را 71.2 کیلوگرم پیش بینی میکند و مدل پیش بینی دوم نیز اختلاف وزن را برای نفر اول 16.8 پیش بینی میکند مسلماً با جمع این دو مقدار باید وزن واقعی شخص به دست آید.

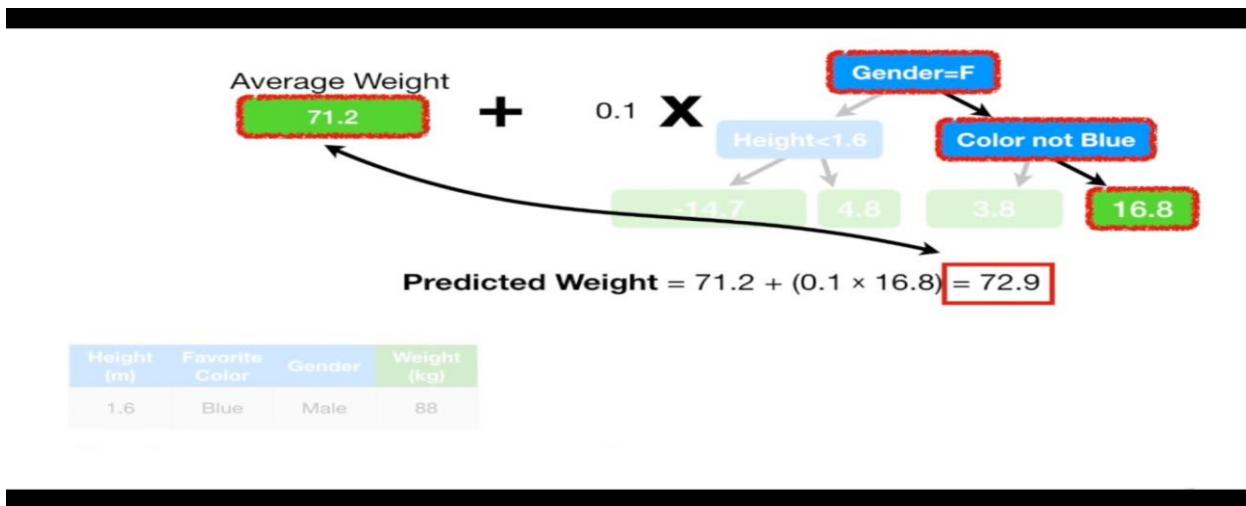


$$\dots \text{so the Predicted Weight} = 71.2 + 16.8 = 88$$

Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88

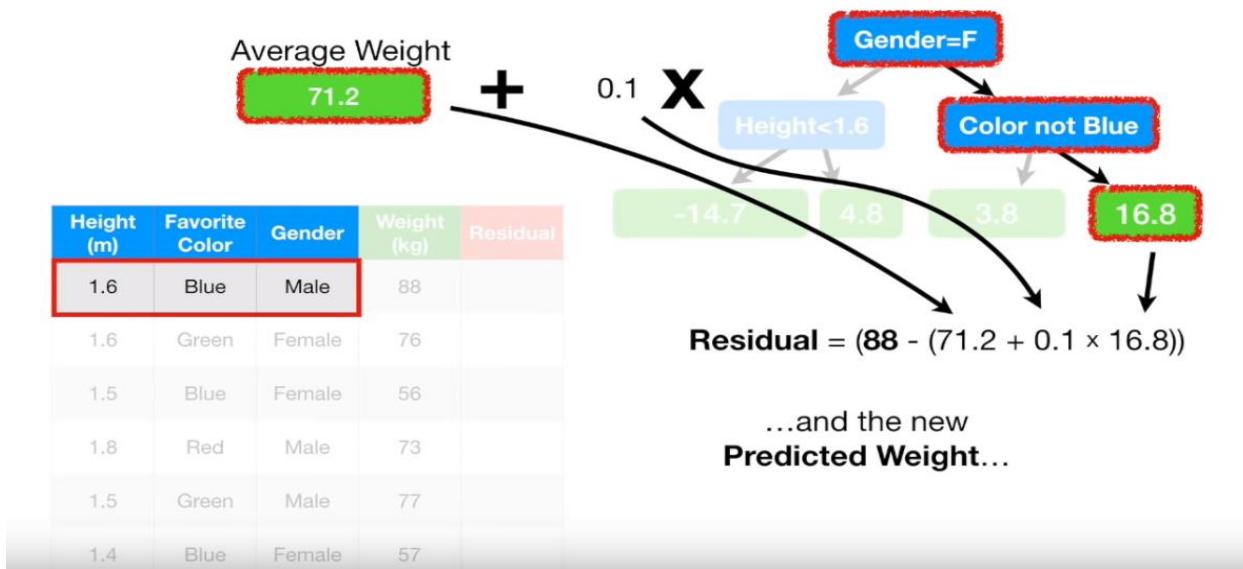
شکل (17-4) ادغام دو مدل پیش بینی

همانطور که در شکل (17-4) نیز مشخص است وزن اولیه به دست آمد ولی این مدل خوبی نیست زیرا هرچند وزن را دقیقا بدست آوردیم و خطای کمی داریم؛ ولی مبتنی بر **dataset** است و ممکن است برای داده هایی که خارج از **dataset** و برای پیش بینی به مدل داده می شوند به خوبی عمل نکند. ما برای رفع این مشکل و ارائه یک مدل خوب یک ضریب برای هر درخت در نظر میگیریم به اسم **learning rate** تا به مدل بگوییم سهم هر درخت در پیش بینی چقدر است. این ضریب مقداری بین صفر تا یک است. در این مثال ما مقدار 0.1 را در نظر گرفتیم و بار دیگر وزن را برای نفر اول پیش بینی کردیم. این مقدار هرچند به خوبی پیش بینی قبلی در شکل (17-4) نیست اما از مدل پیش بینی اولیه دقیق تر است.



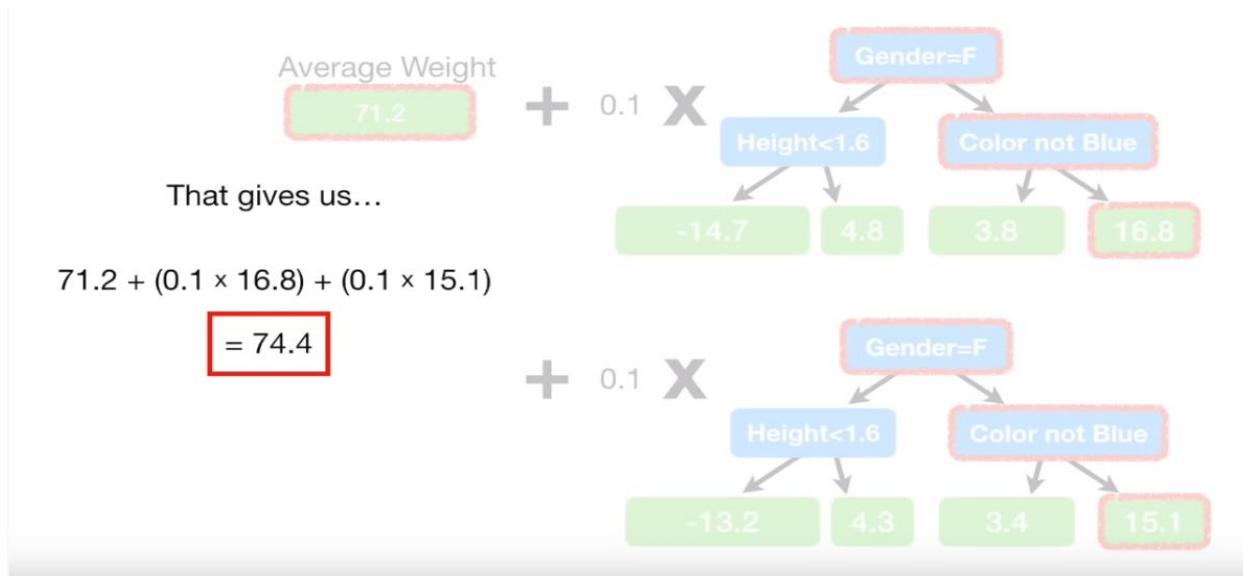
شکل (18-4) پیش بینی با استفاده از هر دو مدل

حال مراحل فوق را بار دیگر با مدل جدید تکرار میکنیم تا درخت دیگری برای پیش بینی بسازیم. یعنی با استفاده از مدل حاصل وزن را برای تمامی رکوردها پیش بینی کرده و اختلاف مقدار واقعی از مقدار پیش بینی شده را به عنوان ستون جدیدی به **dataset** اضافه میکنیم.



شکل (19-4) ساخت مدل جدید

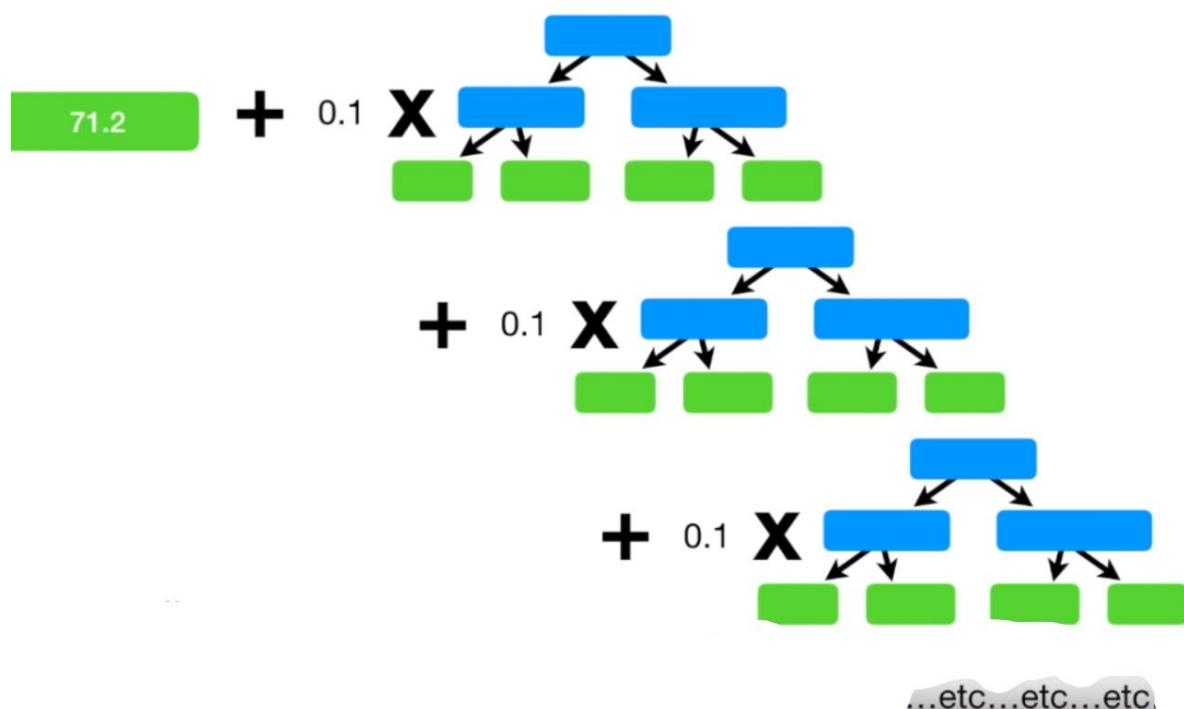
حال با بررسی ستون Residual در شکل (19-4) و ستون Residual در شکل (14-4) میتوان دریافت اختلاف مقادیر پیش بینی شده با مقادیر واقعی کمتر شده در واقع با هر با ساخت درخت جدید یک قدم پیشرفت خواهیم داشت. حال درخت جدید میسازیم و با همان ضریب یادگیری به مدل قبلی اضافه میکنیم و مقدار وزن را برای نفر اول مجدداً محاسبه میکنیم.



شکل (20-4) ساخت مدل جدید

مشاهده میشود این مقدار، از مقدار قبلی بهتر شده و به وزن اصلی شخص نزدیک شده است حال اگر ستون Residual را برای این مدل نیز به دست آوریم مشاهده میشود که این اختلاف باز هم کمتر شده است.

در واقع در این روش ما از یک برگ شروع کرده و با ساختن درخت هایی براساس خطای مدل قبلی و در نهایت ادغام این درخت ها با هم با یک درصد یادگیری؛ سعی در ایجاد یک مدل پیش بینی دقیق داریم.



شکل (21-4) طرح کلی مدل Gradient Boosted Regressor

#### 4-3-4- پیاده سازی الگوریتم های پیش بینی قیمت

گام بعدی پس از پیش پردازش داده ها مشخص کردن متغیر های مستقل ( $X$ ) و متغیر وابسته ( $Y$ ) است. زیرا می خواهیم با استفاده از متغیر های مستقل الگویی برای پیش بینی متغیر وابسته پیدا کنیم.

در مرحله بعد با استفاده از کتابخانه `dataset` و `sklearn` را به دو بخش `train` و `test` تقسیم میکنیم (`X_train, X_test, y_train, y_test`) در این پروژه ما 20 درصد از رکورد هارا برای `test` و 80 درصد را برای مجموعه `train` در نظر گرفتیم.

```
X = airbnb[['neighbourhood_group','neighbourhood','latitude','longitude','room_type','minimum_nights','number_of_reviews','calculated_host_listings_count','reviews_per_month','availability_365']]
y = airbnb['price']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=101)
```

**شکل (22-4)** آماده کردن مجموعه `train` و `test`

در واقع مدل پیش گویی ما الگوی خود را با استفاده از مجموعه `train` پیدا می کند. سپس `X_test` را به عنوان ورودی دریافت کرده و قیمت را برای هر رکورد در این مجموعه پیش بینی می کند. در مرحله بعد با مقایسه مقادیر پیش بینی شده توسط مدل و مقادیر واقعی (`y_test`) مدل را ارزیابی میکنیم.

### 1-3-4 پیاده سازی الگوریتم LinearRegression

همانطور که در قسمت قبل نیز گفته شد، بخشی از داده ها را که به عنوان مجموعه `train` در نظر گرفتیم، برای یادگیری در اختیار مدل مورد نظر قرار میدهیم `fit(X_train,y_train)`. حال این مدل برای هر ویژگی یک ضریب بدست می آورد؛ مطابق شکل (11-4)؛ تا یک معادله خطی برای پیش گویی قیمت ایجاد کند این ضرایب در شکل زیر تحت عنوان `coefficient` نشان داده شده است.

## LinearRegression

```
regressor = LinearRegression()
regressor .fit(X_train,y_train);
#To retrieve the intercept:
print(regressor.intercept_)
#For retrieving the slope:
coeff_airbnb = pd.DataFrame(regressor.coef_, X.columns, columns=['Coefficient'])
coeff_airbnb
```

-291.7668084477174

	Coefficient
neighbourhood_group	0.051842
neighbourhood	0.000560
latitude	0.955278
longitude	-3.485512
room_type	-0.721593
minimum_nights	-0.001925
number_of_reviews	-0.000611
calculated_host_listings_count	0.000205
reviews_per_month	-0.007336
availability_365	0.000753

شکل (23-4) ساخت معادله خطی به عنوان مدل

عدد 291.766 - عرض از مبدا این خط را نشان میدهد . حال متغیرهای مستقل در مجموعه test را (X\_test)، در اختیار مدل قرار میدهیم تا با استفاده از معادله به دست آمده در مرحله قبل قیمت را برای هر رکورد در این مجموعه پیش بینی کند. و سپس با استفاده از دو معیار r2\_score و mean\_squared\_error به تحلیل کارآمدی این مدل می پردازیم .

```
'''Get Predictions & Print Metrics'''
predicts = regressor .predict(X_test)
print("""
    Mean Squared Error: {}
    R2 Score: {}
""".format(
    np.sqrt(mean_squared_error(y_test, predicts)),
    r2_score(y_test,predicts),
))
)
```

Mean Squared Error: 0.4905424848707436  
R2 Score: 0.4736123787524762

شکل (24-4) بررسی کارآمدی مدل

معیار Mean Squared Error یکی از چندین معیاری است که به بررسی خطای مدل پیش بینی شده می‌پردازد و r2\_score نیز میزان مطابقت مدل پیش بینی شده با مقدار واقعی را نشان می‌دهد.

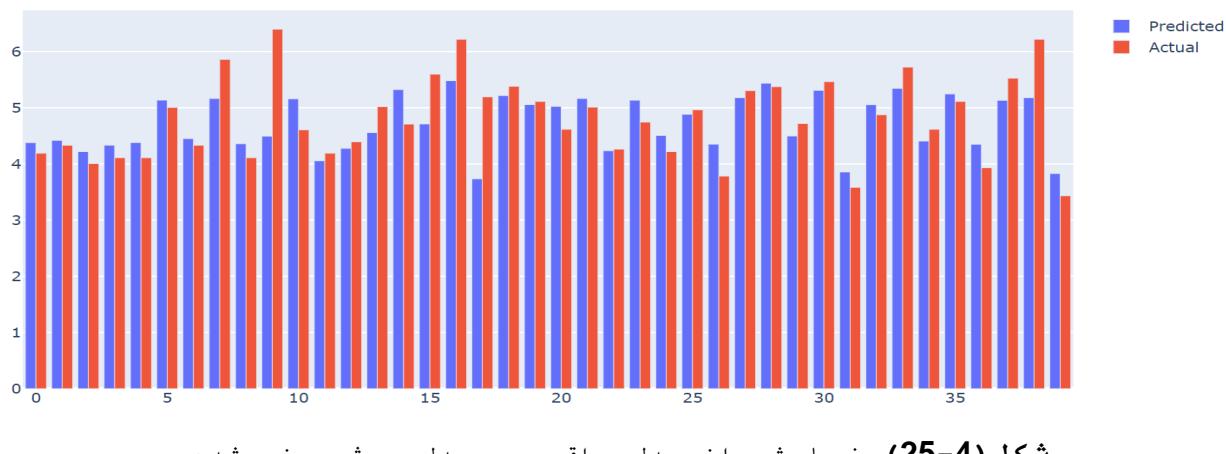
براساس معیار r2\_score اگر مقدار به دست آمده برای آن بین 0.4 تا 0.59 باشد مدل طراحی شده نسبتاً خوب است و اگر بین 0.6 تا 0.79 باشد مدل پیش بینی شده قوی است و اگر بیش از این مقدار باشد مدل بسیار به واقعیت نزدیک است در نتیجه مدل به دست آمده در این بخش مدل نسبتاً خوبی است.

مادر ادامه قصد داریم این دو مقدار را با مقادیر به دست آمده در مدل Gradient Boosted Regressor مقایسه کرد و مدل بهتر را معرفی کنیم. برای این که درک بهتری از مطابقت مدل پیش بینی شده و مقدار واقعی داشته باشید این دو را در دو شکل به صورت زیر رسم کرده ایم.

```
error_airbnb = pd.DataFrame({
    'Actual Values': np.array(y_test).flatten(),
    'Predicted Values': predicts.flatten()}).head(40)

title=['Pred vs Actual']
fig = go.Figure(data=[
    go.Bar(name='Predicted', x=error_airbnb.index, y=error_airbnb['Predicted Values']),
    go.Bar(name='Actual', x=error_airbnb.index, y=error_airbnb['Actual Values'])
])

fig.update_layout(barmode='group')
fig.show()
```

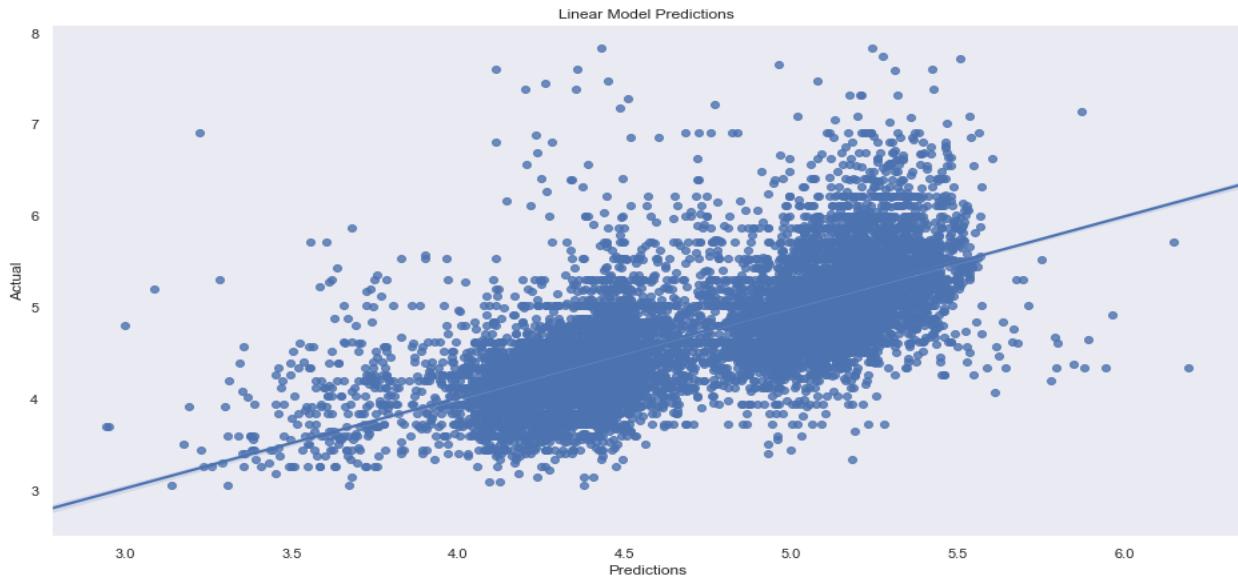


شکل (25-4) نمایشی از مدل واقعی و مدل پیش بینی شده

```

plt.figure(figsize=(16,8))
sns.regplot(predicts,y_test)
plt.xlabel('Predictions')
plt.ylabel('Actual')
plt.title("Linear Model Predictions")
plt.grid(False)
plt.show()

```



شکل (26-4) نمایشی از مدل واقعی و مدل پیش بینی شده

#### 3-4-2 پیاده سازی الگوریتم GradientBoostingRegressor

در این قسمت با وارد کردن مجموعه train مدلی برای پیش بینی قیمت با استفاده از الگوریتم GradientBoostingRegressor میسازیم.

در این نمونه 3000 درخت ساخته میشود و درصد یادگیری مدل از هر درخت 0.01 است.

```

GBoost = GradientBoostingRegressor(n_estimators=3000, learning_rate=0.01)
GBoost.fit(X_train,y_train);

```

شکل (27-4) ساخت مدل GradientBoostingRegressor

حال به بررسی دو معیار  $r^2$  score و Mean Squared Error می پردازیم.

```

'''Get Predictions & Metrics'''
predicts2 = GBoost.predict(X_test)

print("""
    Mean Squared Error: {}
    R2 Score: {}
""".format(
    np.sqrt(mean_squared_error(y_test, predicts2)),
    r2_score(y_test,predicts2) ,
))

```

Mean Squared Error: 0.4247580296079012  
R2 Score: 0.6053286426564966

#### شکل (28-4) بررسی کارآمدی مدل

با مقایسه این دو معیار در این الگوریتم و الگوریتم پیشین مشاهده میشود که این الگوریتم خطای کمتری داشته و با توجه به بازه های ارائه شده برای معیار `r2_score` مدل ساخته شده توسط الگوریتم GradientBoostingRegressor یک مدل قوی است و پیشگویی های آن تا حد زیادی به واقعیت نزدیک است.

## فصل 5: جمع بندی و پیشنهاد ها

## ۱-۵- مقدمه

یکی از مزیت های زبان پایتون وجود کتابخانه های قدرتمند و کارآمد آن است. در این پروژه در بخش های مختلف اعم از پیش پردازش، تحلیل، مصور سازی و ساخت مدل، ما از این کتابخانه ها استفاده کردیم. و توانستیم تحلیلی هر چند مختصر بر روی اکثر فیلد ها داشته باشیم. در این پروژه ما چالش مطرح شده در سایت kaggle را مورد بررسی قرار دادیم وسعی کردیم بارسم انواع نمودار ها و جداول، داده ها را به وضوح بررسی کنیم و به استخراج اطلاعات و روابط پنهان و آشکار در dataset سایت airbnb پرداختیم و دو مدل برای پیشگویی قیمت پیشنهاد کردیم البته در این پروژه از نظرات و تحلیل های دیگر تحلیل گران نیز استفاده شده تا تحلیلی جامع و درخور شان شما خوانندگان ارائه شود. در واقع سعی کردیم نمونه ای هر چند کوچک از پروژه های مطرح در زمینه ای داده کاوی را ارائه کنیم.

## ۲-۵- پیشنهادهایی برای کارهای آتی

در کنار همه تحلیل هایی که این پروژه ارائه میکند، میتوان به بررسی دیگر الگوریتم های رگرسیون برای پیش بینی قیمت پرداخت تا بتوان مدلی را ارائه کرد که بیشترین سازگاری را با داده ها دارد و بهتر از این دو مدل به پیش بینی قیمت بپردازد.

## مراجع

- [1] [www.kaggle.com](http://www.kaggle.com)
- [2] [www.python.org](http://www.python.org)
- [3] [www.docs.scipy.org](http://www.docs.scipy.org)
- [4] [www.pandas.pydata.org](http://www.pandas.pydata.org)
- [5] [www.matplotlib.org](http://www.matplotlib.org)
- [6] [www.seaborn.pydata.org](http://www.seaborn.pydata.org)
- [7] [www.wikipedia.org](http://www.wikipedia.org)
- [8] [www.maktabkhooneh.org](http://www.maktabkhooneh.org)
- [9] [www.youtube.com](http://www.youtube.com)

**Abstract:**

This documentary is a report based on a dataset analysis of the Airbnb website, a website where people rent accommodation, using various charts and tables. Python and its highly efficient and powerful libraries - designed for analysis and visualization - such as Numpy, Pandas, seaborn, plotly, folium, have been used to provide comprehensive and accurate analysis.

After scrutinizing the data in the dataset, we present two predictive models for price proposition to users who intend to register new residences, using the sklearn and scipy libraries.

**Keywords:** graph, python, analysis, regression, residency, dataset.



**Shariaty University**

# **Analysis and visualization of airbnb site data and predictive model**

**A Thesis Submitted in Partial Fulfillment of the Requirement for the Degree  
of Bachelor of Science in Software Engineering**

**By:  
Mahnaz Divargar**

**Advisor:  
Ms. Zahra Zamanzadeh**

**February 2020**