بسمه تعالی

دانشگاه شهید بهشتی
دانشکده علوم ریاضی
گروه علوم کامپیوتر

تمرین سری ۱ واحد درسی داده کاوی

سارا چرمچی
۴۰۰۴۲۲۰۶۶

استاد راهنما : جناب آقای دکتر فراهانی

دستیار آموزشی : آقای علی شریفی

فروردین ۱۴۰۱

مقدمه

کلیه تحلیل های این گزارش با استفاده از زبان برنامه نویسی پایتون اجرا شده است و کدها به پیوست موجود است.

تمرین دوم

در این تمرین از دیتاست یک پلتفرم مشهور المانی در زمینه اجاره خانه ها استفاده شده است. اطلاعات این دیتاست بسیار گسترده و تمیزنشده است. ابتدا کتابخانه های مورد استفاده را وارد می کنیم. سپس داده ها را به هر طریقی مناسب است می خوانیم (در اینجا داده های کگل را از طریق کولب فراخوانی می کنیم)

اندازه دیتاست : (۲۶۸۸۵۰, ۴۹) و ۴۹ ستون دارد شامل :

```
Index(['regio1', 'serviceCharge', 'heatingType', 'telekomTvOffer',
       'telekomHybridUploadSpeed', 'newlyConst', 'balcony', 'picturecount',
       'pricetrend', 'telekomUploadSpeed', 'totalRent', 'yearConstructed',
       'scoutId', 'noParkSpaces', 'firingTypes', 'hasKitchen', 'geo_bln',
       'cellar', 'yearConstructedRange', 'baseRent', 'houseNumber',
       'livingSpace', 'geo_krs', 'condition', 'interiorQual', 'petsAllowed',
       'street', 'streetPlain', 'lift', 'baseRentRange', 'typeOfFlat',
       'geo_plz', 'noRooms', 'thermalChar', 'floor', 'numberOfFloors',
       'noRoomsRange', 'garden', 'livingSpaceRange', 'regio2', 'regio3',
       'description', 'facilities', 'heatingCosts', 'energyEfficiencyClass',
       'lastRefurbish', 'electricityBasePrice', 'electricityKwhPrice', 'date'],
      dtype='object')
```

۱.پاک سازی داده ها :

پاک سازی داده ها شامل حذف ستون های هجو، حدف داده های تکراری ، بررسی داده های ناموجود و رسیدگی به داده های پرت.

مرحله اول پاک سازی داده ها : بررسی داده های nan و تصمیم در مورد پر کردن آن ها با روش fillna

ابتدا درصد داده های ناموجود در هر ستون را با دستور df.isna().sum()/len(df) نمایش می دهیم و میبینیم تعداد زیادی از ستون ها داده های ناموجود دارند که ممکن است در آینده باعث خطای مدل شوند پس چنین ستون هایی را حذف می کنیم.

جای گذاری داده های ناموجود در داده های عددی و داده های categorical که درصد بالایی نداشتند که حذف شوند.

داده های عددی: یکی از راه های پرکردن داده های عددی ناموجود، استفاده از میانگین داده های موجود آن ستون است.

داده های categorical: داده های ناموجود در هر ستون را با پرفرکانس ترین دادی آن ستون جای گذاری می کنیم

مرحله دوم پاک سازی داده ها: حذف داده های بی معنی و یا غیر مهم

در ستون livingSpace داده هایی با مقدار ۰ داریم که بی معنی است چرا که آپارتمانی با متراژ صفر نداریم

در ستون totalRent داده هایی با مقدار ۰ داریم که بی معنی است چرا که آپارتمانی با اجاره بهای صفر نداریم

در ستون noRooms داده هایی با مقدار بیش از ۱۰۰ داریم که بی معنی است چرا که آپارتمانی با این تعداد اتاق منطقی نیست .

در ستون livingSpace داده هایی با مقدار بیش از ۶۰ متر مربع داریم که اجاره بهای زیر ۳۰ یورو دارند که بی معنی است چرا که آپارتمانی با این متراژ چنین قیمتی ندارد.

برخی داده ها باعث شلوغ شدن دیتاست می شود و در اینجا نیازی نداریم پس خذف می کنیم :

```
df_g.drop(columns=['livingSpaceRange','street','description','facilities','geo_krs','geo_plz','scoutId','telek
omUploadSpeed','telekomTvOffer','pricetrend','regio3','noRoomsRange','picturecount','geo_bln','date',\
    'houseNumber','streetPlain','firingTypes','yearConstructedRange'],inplace=True)
```

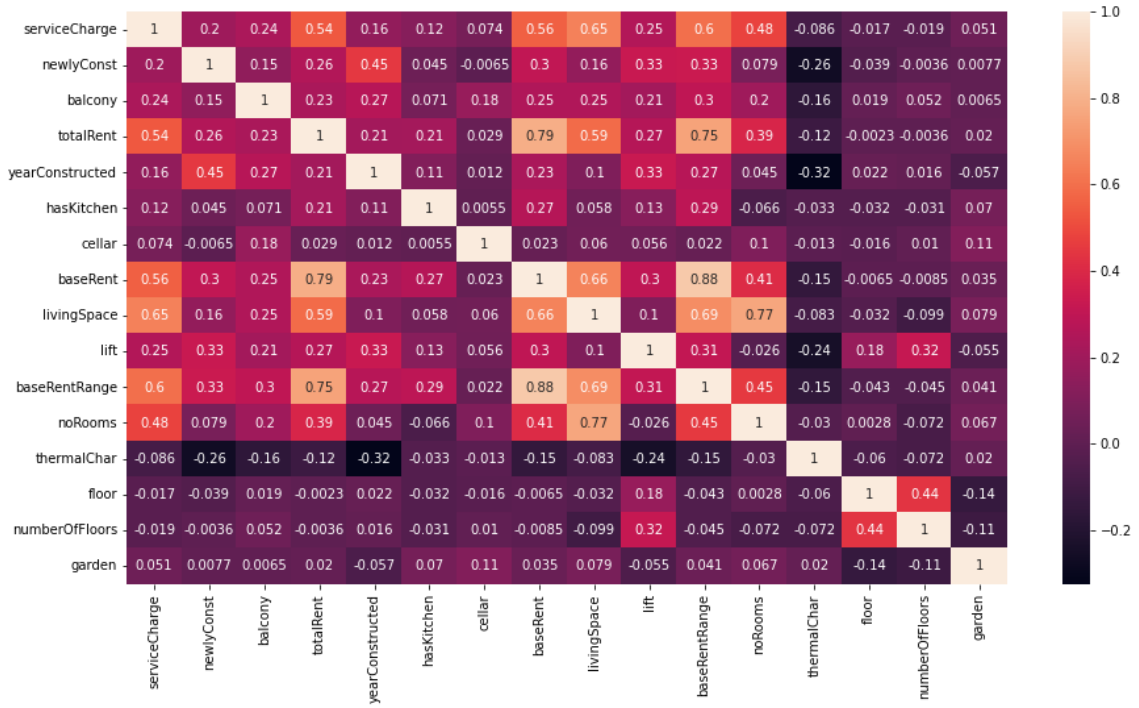
مرحله سوم پاک سازی داده ها: نرمالایز کردن داده ها با روش سنتی میانگین و واریانس : در این مورد، میانگین به اضافه یا منهای ۳ برابر انحراف معیار است که دادههای بزرگتر از میانگین به اضافه ۳ برابر انحراف معیار و کوچکتر از میانگین منهای ۳ برابر انحراف معیار، پرت محسوب می شوند.

مرحله چهارم پاک سازی داده ها: برخی داده های categorical را به خاطر تعدد موارد unique می توان کاهش داد، شمارشی از انواع داده ها در فیچر ایالت می کنیم و چند مورد آخر که کمترین فروانی را داشتند در نوع جدید other ذخیره می کنیم.

۲. مصور سازی داده ها

ارایه اطلاعات کلی در حالت تجمیعی در خصوص آگهی ها.

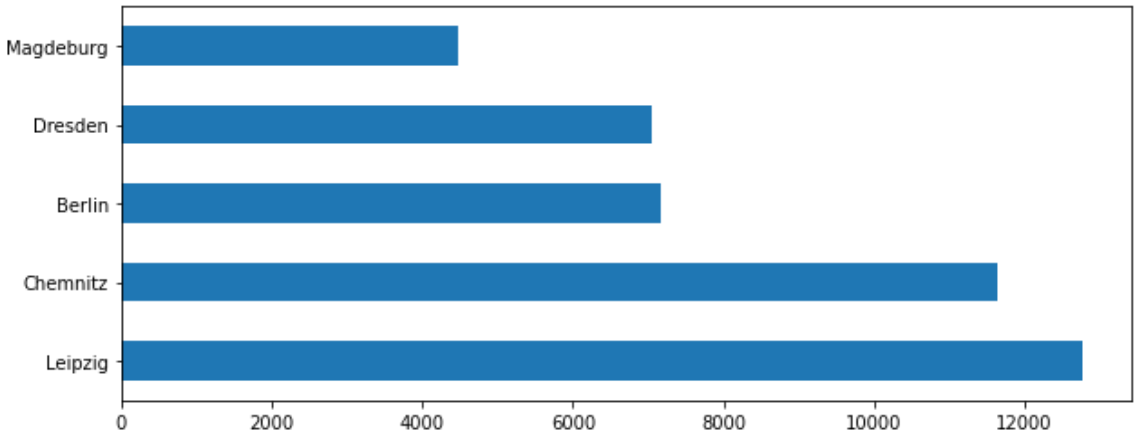
ابتدا رابطه کلی داده های عددی را نمایش می دهیم :



بدیهی است که رابطه ی هر مولفه با خودش حداکثری است (۱) و تا تیره ترین حالت که به صفر میل کرده و عدم ارتباط دو مولفه را نشان می دهد مانند رابطه تعداد نظرات و شناسه.

سپس به بررسی و مصور سازی هر ویژگی می پردازیم:

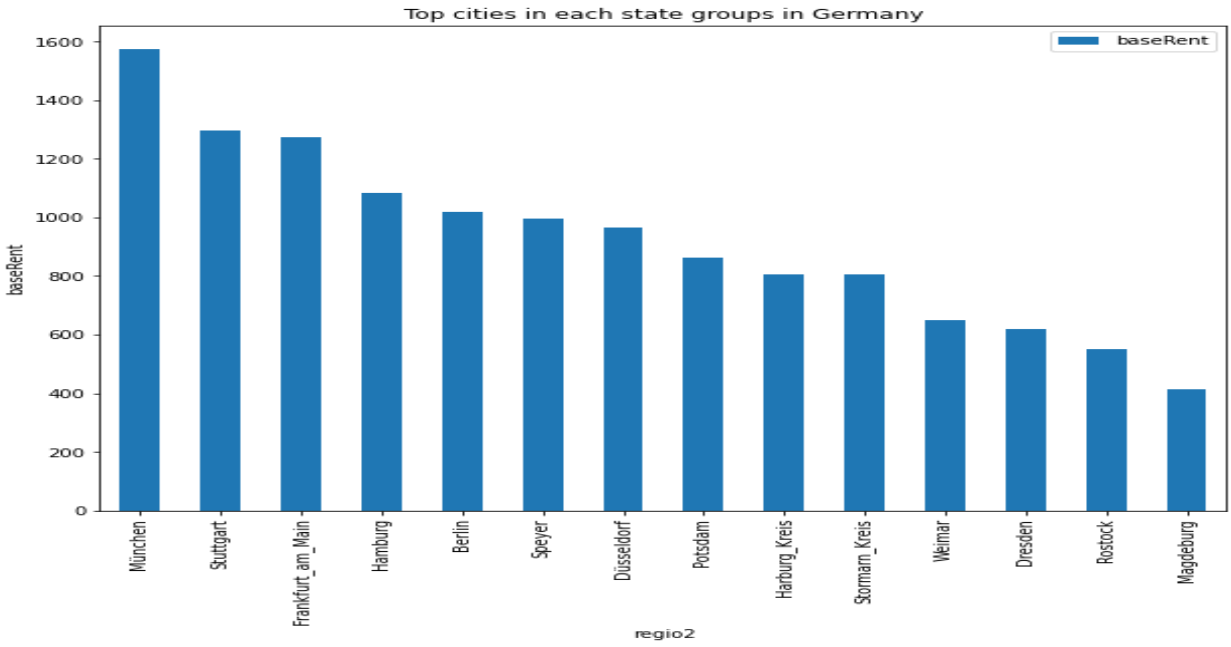
تعداد آگهی ها : تعداد اگهی ها بر اساس منطقه جغرافیایی را با دستور value_counts() بدست آورده و نمودار را رسم می کنیم(۵ شهر اول) :



میبینیم شهر Leipzig با بیش از ۱۲۰۰۰ آگهی در رتبه اول قرار دارد.

در ادامه شهرهای برتر از نظر میانگین اجاره را بررسی می کنیم :

ابتدا دیتافریم را بر اساس شهر و ایالت و میانگین قیمتی با دستور groupby تفکیک می کنیم و با ماکزیموم کردن و سورت کردن به ترتیب نزولی می چینیم، برای شهود بهتر با کتابخانه matplotlib آن را رسم می کنیم :



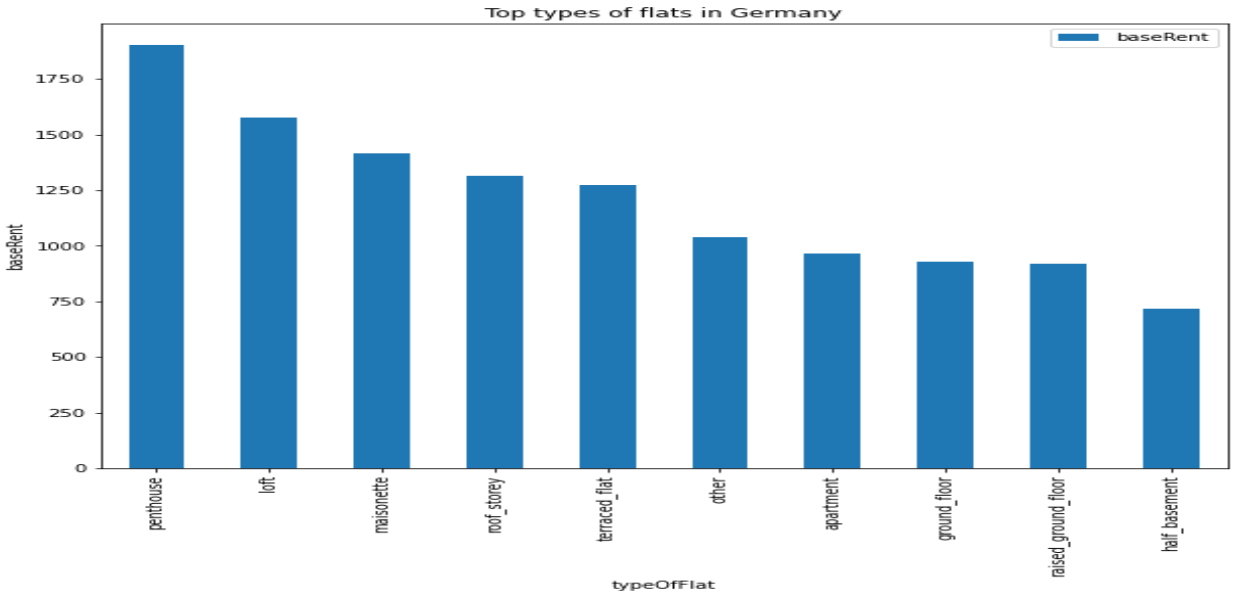
همانطور که انتظار می رفت شهر Munchen یا مونیخ که از شهر های گران محسوب می شود در تحلیل داده ها نیز نسبت به سابرین میانگین قیمت اجاره بالاتری را بدست اورد.

آیا ارتباطی میان نوع خانه مورد اجاره و قیمت ان وجود دارد؟ از روش قبلی مجددا استفاده می کنیم و نوع خانه را بر اساس قیمت و ایالت تحلیل می کنیم :

به وضوح خانه های پینت هاوس در صدر جدول قرار دارند

	typeOfFlat	regio1_edit	baseRent
86	penthouse	Berlin	1903.851778
44	loft	Berlin	1574.337931
58	maisonette	Berlin	1415.626783
114	roof_storey	Berlin	1315.278607
128	terraced_flat	Berlin	1273.992258
72	other	Berlin	1040.130694
2	apartment	Berlin	962.908263
16	ground_floor	Berlin	929.009412
102	raised_ground_floor	Hessen	920.475919
41	half_basement	other	718.684186

و جالب است که با وجود آنکه گران ترین شهر مونیخ بوده است اما اگر نوع خانه مدنظر قرار گیرد، گران ترین خانه ها در برلین هستند.

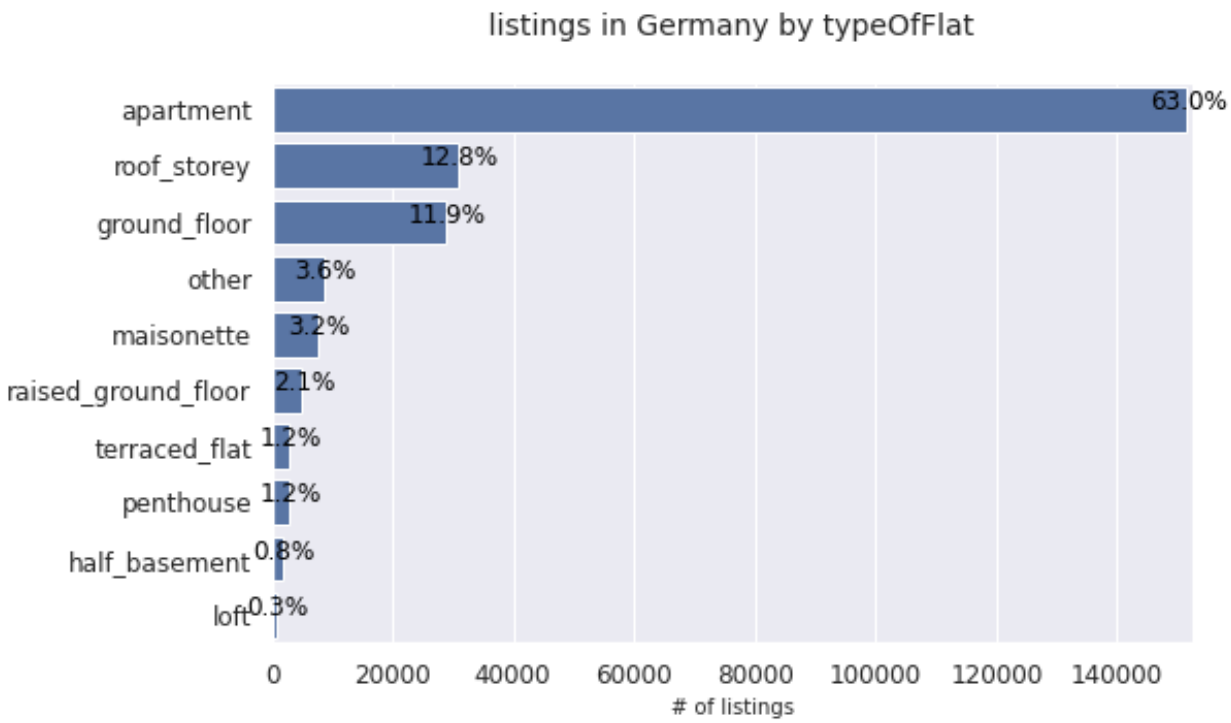


آیا ارتباطی بین قیمت خانه و متراژ ان با در نظر گرفتن تعداد اتاق ها وجود دارد ؟ (در اینجا داده تعداد اتاق ها را گروه بندی کردیم و به عنوان داده categorical درر نظر گرفتیم) :

به وضوح با افزایش متراژ خانه باید قیمت آن نیز افزایش یابد ،
همانطور که میبینیم در نمودار نیز با افزایش متراژ قیمت افزایش یافته از طرفی با افزایش متراژ تعداد اتاق ها نیز افزایش داشته است.

داده های روشن تر با تعداد اتاق کتر از دو تا در متراژ کمتر از ۱۰۰ متر مربع قرار دارند که منطقی است.

در ادامه به درصد فروانی انواع خانه های مورد اجاره را بررسی می کنیم، می بینیم که خانه هایی مانند پینت هاوس که قیمت بالایی دارند درصد فراوانی بسیار اندکی در بین خانه ها دارند اما خانه ها از توع آپارتمان متداول ترین نوع خانه های اجاره ای است:

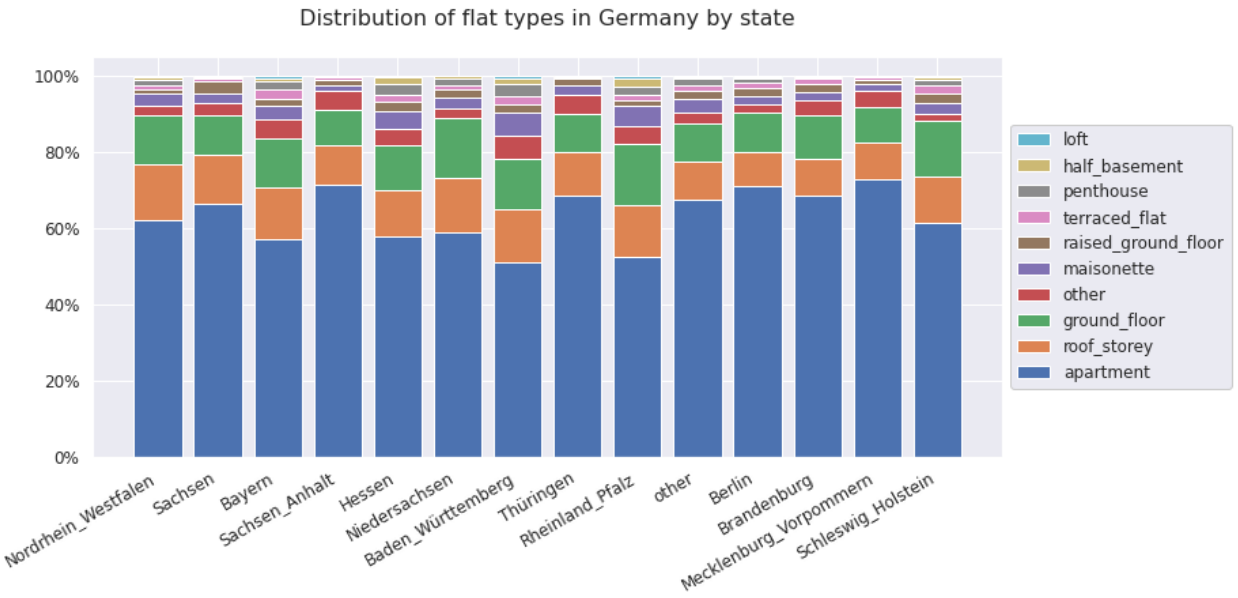


سپس با استفاده از میانگین و میانه برای داده های categorical مانند داده وجود یا عدم وجود آشپزخانه داده ها را تکیک کردیم و برای خانه های دارای آشپرخانه و بدون اشپرخانه میانگین و انحراف از معیار اجاره بها را محاسبه کرده و مشخصا :

خانه های دارای آشپزخانه میانگین اجاره بهای بالاتری دارند

در نهایت نگاه جامعی به فروانی نوع خانه ها در ایالات مختلف می اندازیم :

از تکنیک مشابهی استفاده شده است اما نوع نمایش نمودار تکنیک متفاوتی استفاده شده است که منبع ذکر شده است.



۳. تلاش در ساخت مدل جهت پیش بینی پارامترهایی مانند قیمت

ایا می توان مدلی ارایه کرد که براساس شاخصه های دیتاست، اجاره بهای انواع خانه ها را پیش بینی کند؟

از مدل رگرسیون خطی استفاده می کنیم. اما پیش از مدل سازی، بخش پیش پردازش داده ها را تکمبل می کنیم تا مدل پاسخ بهتری دهد. منظور از پیش پردازش داده ها encoding است. مدل های یادگیری ماشین نیاز دارند متغیرهای عددی به عنوان ورودی دریافت کنند بنابراین اگر در دیتاست متغیرهای categorical داشته باشیم برای فیت کردن و ارزیابی مدل پیش از مدل سازی داده ها را encode می کنیم.

ابتدا با روش sklearn.preprocessing.LabelEncoder برچسب را بین 0 and n_classes-1 کد گذاری می کنیم. و فیت می کنیم.

به این ترتیب داده های categorical مورد نیاز به داده عددی تبدیل می شوند .

بعد از آماده سازی داده ها، مدل سازی را شروع می کنیم. داده‌ها را به دو بخش آموزشی و آزمایشی تفکیک می کنیم. همانطور که مشخص است ٪۸۰ از داده‌ها برای بخش آموزش و ٪۲۰ برای آزمایش در نظر گرفته شده‌اند. به کمک کتابخانه sklearn و کلاس LinearRegression، محاسبات مربوط به مدل رگرسیونی در متغیرl_reg ذخیره می کنیم. و مدل را train می کنیم.

سپس کارایی مدل را ارزیابی می کنیم. سهمی که مدل رگرسیونی از تغییرات متغیر وابسته دارد را با مربع آر (R square) می‌شناسند. مقدار R^2 در بازه ۰ تا ۱ اندازه گیری می شود و هر چه به یک نزدیک تر باشیم کارایی مدل بهتر است. R^2 برابر صفر بدترین حالت و خطای مدل را نشان می دهد.

سایر معیارهای ارزیابی مانند : میانگین خطای مطلق (mean absolute error - MAE) که مجموع میانگین اختلاف مطلق بین مقادیر واقعی و پیش بینی است و خطای میانگین مربع (mean square error - MSE) که شبیه MAE است، با این تفاوت که مربع تفاوت مقادیر خروجی واقعی و پیش بینی شده را به جای استفاده از مقدار مطلق، قبل از محاسبه مجموع همه آنها محاسبه می کند.

با استفاده از کتابخانه sklearn بدون محاسبات اضافه می توان این مقدار را بدست آورد : sklearn.metrics.r2_score

Mean Squared Error: 186.1745819698833

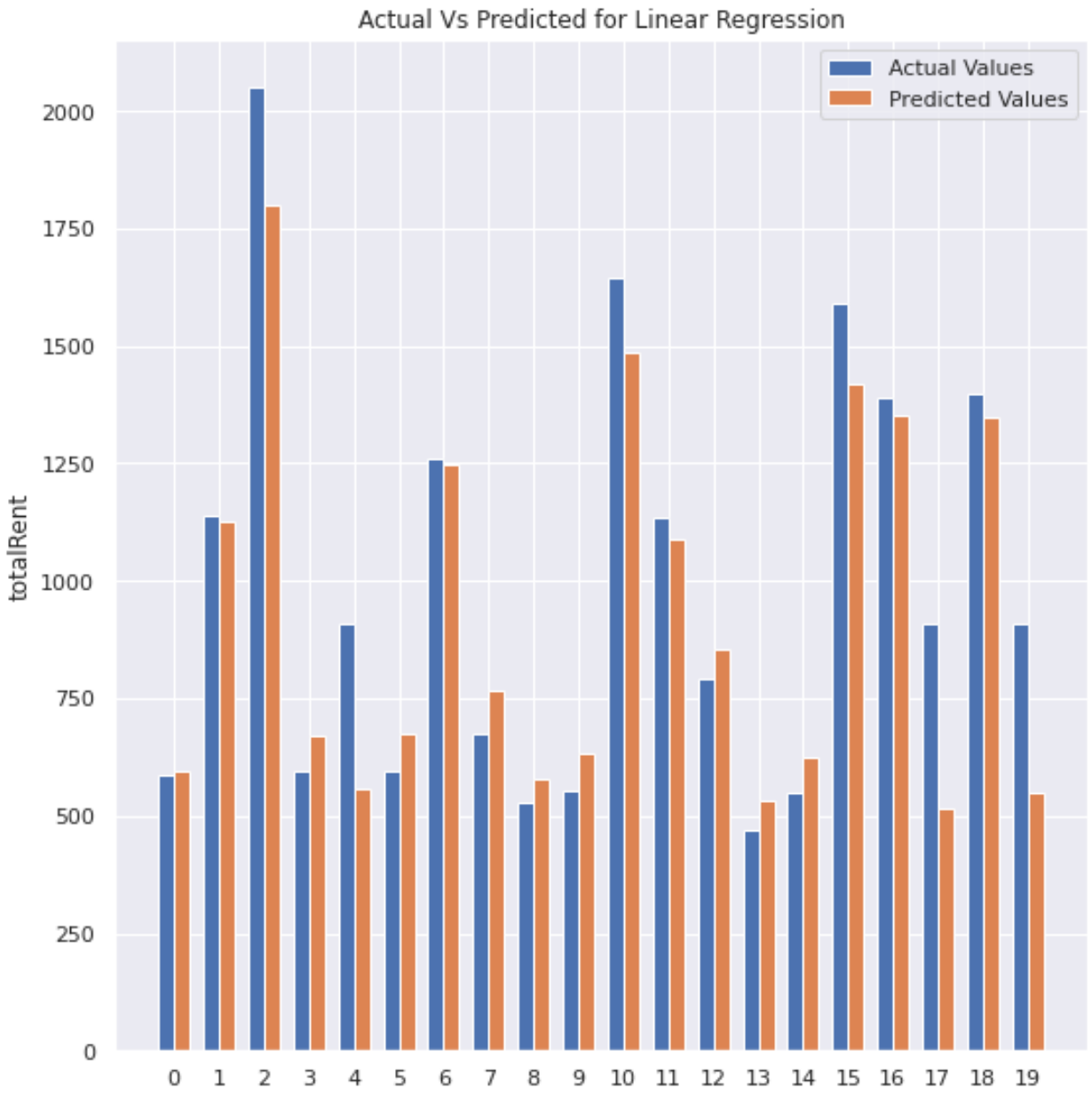
R2 Score: 80.75206235908281

Mean Absolute Error: 110.94225582755931

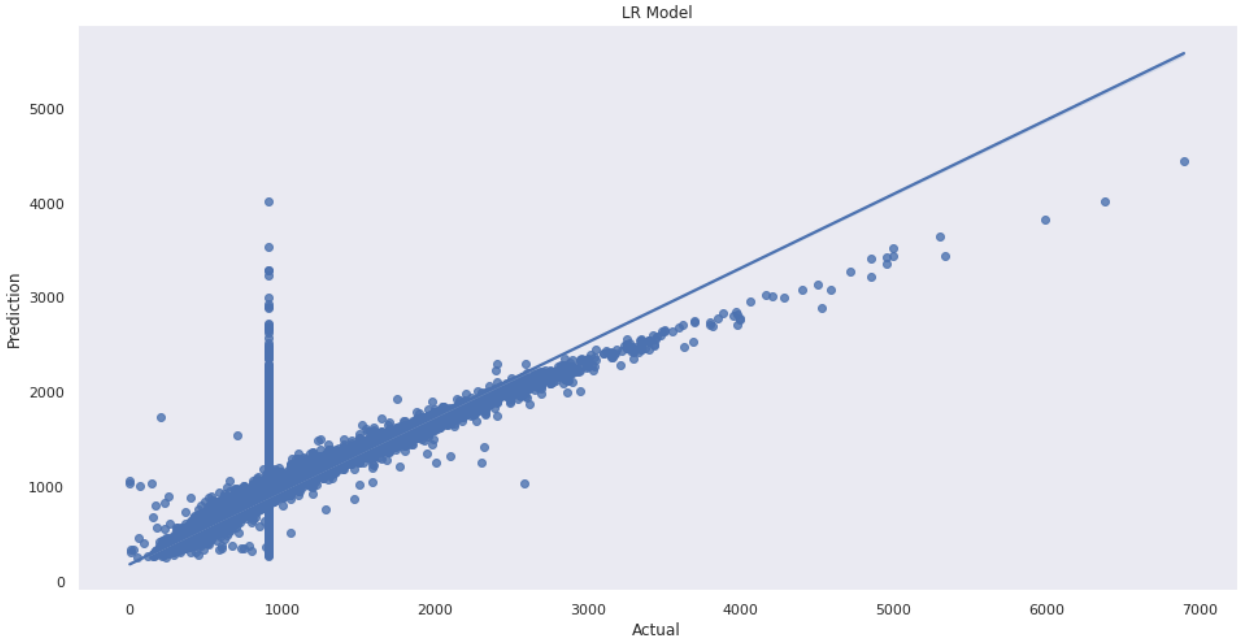
Mean Squareroot Error: 34660.97497166079

دقت مدل حدود ۸۰ ٪ درصد است.

در ادامه نمودار میله ای تفاوت مقادیر واقعی قیمت و مقادیر پیش بینی شده ی ۲۰ داده اول را نمایش می دهیم و با کتابخانه seaborn نمودار داده ها و مقدار فیت بودن مدل را مصور می کنیم :



و مصور سازی نهایی :


LR Model
Prediction
Actual
0
1000
2000
3000
4000
5000
6000
7000