نام و نام خانوادگی: یاس جابرانصاری

شماره دانشجویی : 97222018

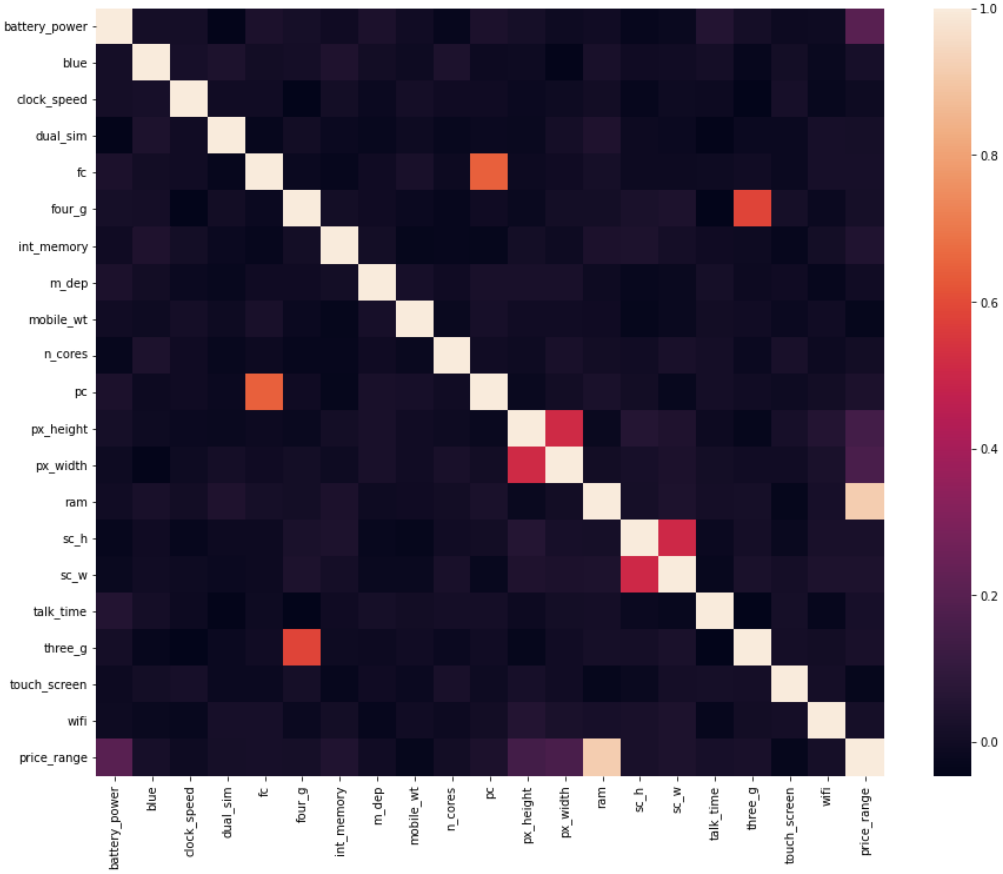
شماره تمرین : 1(عید 1401)

تحویل: 1401/01/20

تمرین شماره 1: mobile-price-classification

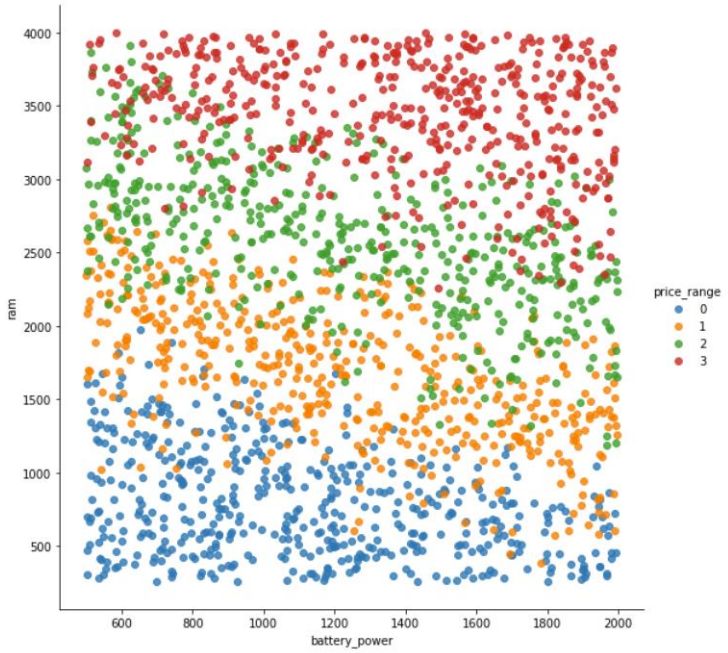
1. پس از خواندن داده اقدام به پاکسازی داده ها کردم. داده خالی و غیرعددی وجود نداشت پس اقدام به حذف موارد مشابه کردم که مورد مشابهی نیز یافت نشد. سپس داده های پرت را حذف کردم که طی این اقدام 12 داده حذف شد.
(به علت اینکه در بخش 7 به صورت جدا از scaling سوال شده بود در این قسمت از نرمالسازی داده ها و.. خودداری کردم)

2. کوریلیشن بین داده ها در یک heatmap بدست آوردم که به صورت زیر میباشد.

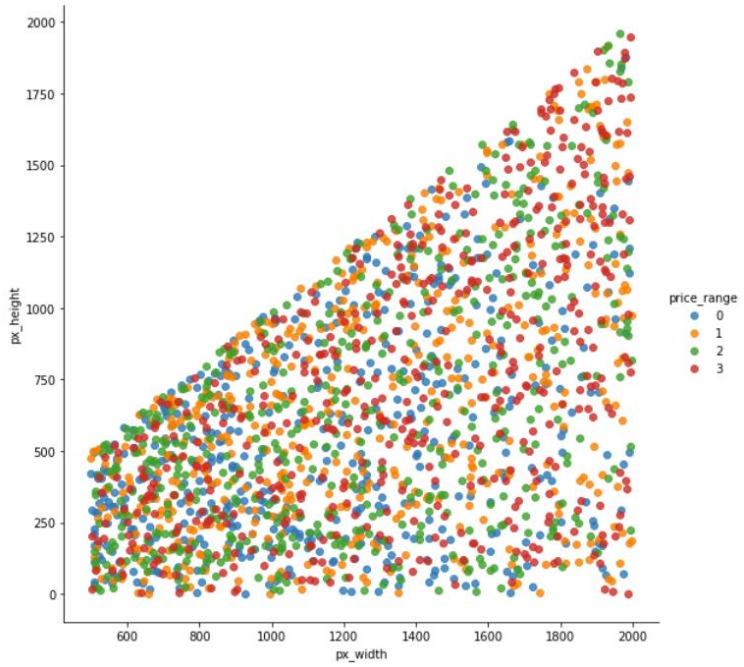


ستون price_range میتواند اعداد 0و1و2و3 را بگیرد.

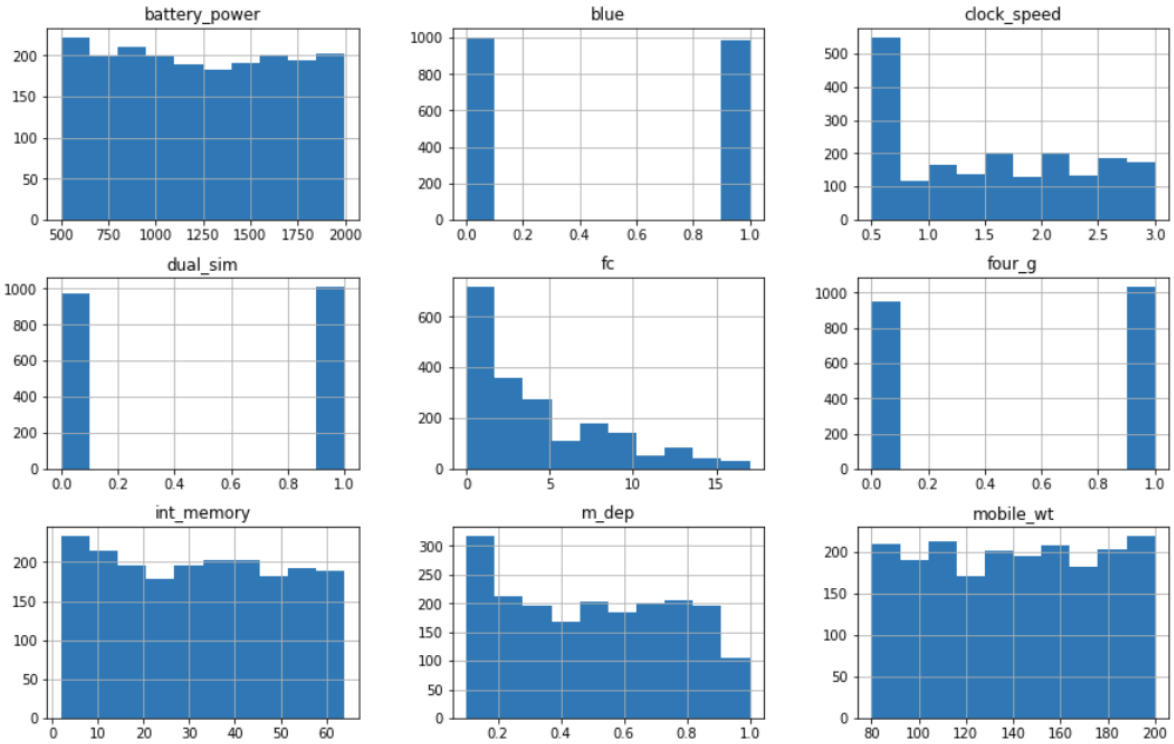
نمودار ram و battery_power را کشیدم و price_range را روی آن نمایش دادم که از آن نتیجه میشود که موبایل های با قیمت بیشتر(در محدوده ی 3) دارای رم بهتری هستند اما دارای هر قدرت باتری میتوانند باشند. از این نتیجه میشود ram بهتر ارتباط مستقیمی با قیمت بالاتر دارد.
همچنین تعداد بیشتری موبایل با محدوده قیمت بالاتر باتری بهتری دارند.



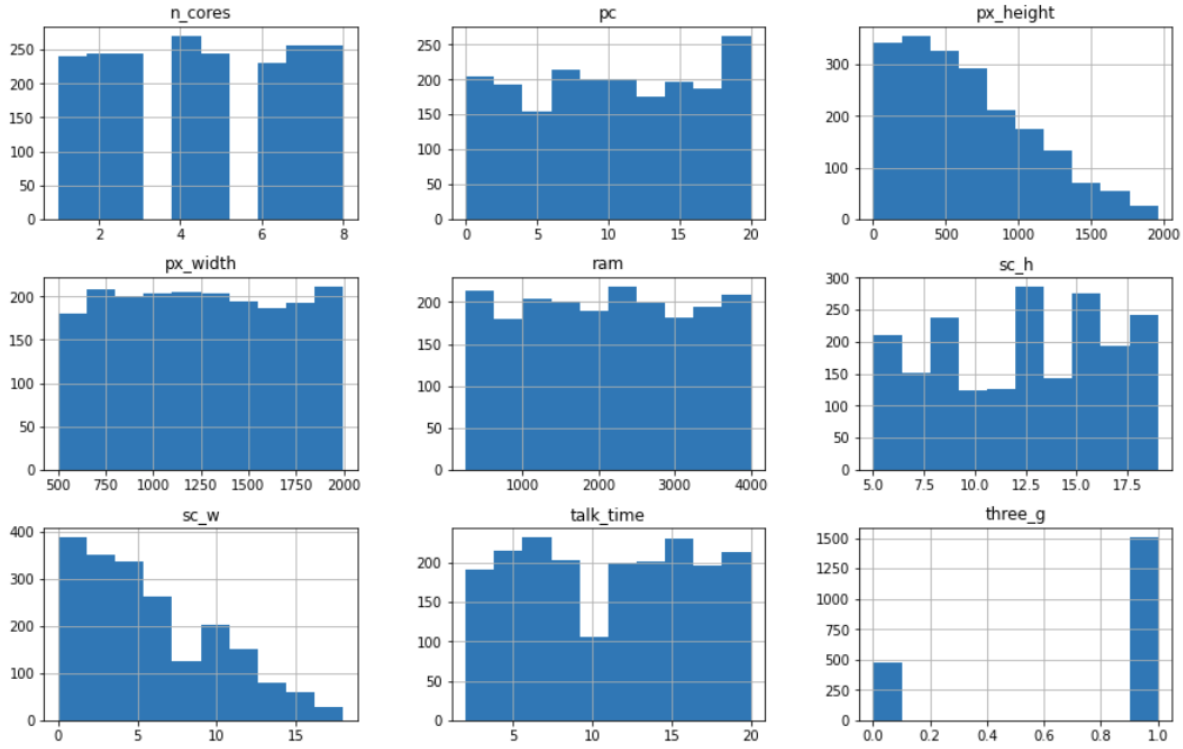
نمودار px_width و px_height را کشیدم و price_range را روی آن نمایش دادم.
از این نتیجه میشود موبایلهای با px_widthو px_height خیلی بالا جزو گروه 0 نیستند(قیمت پایین ندارند) (در نزدیکی قله نقطه آبی دیده نمیشود)



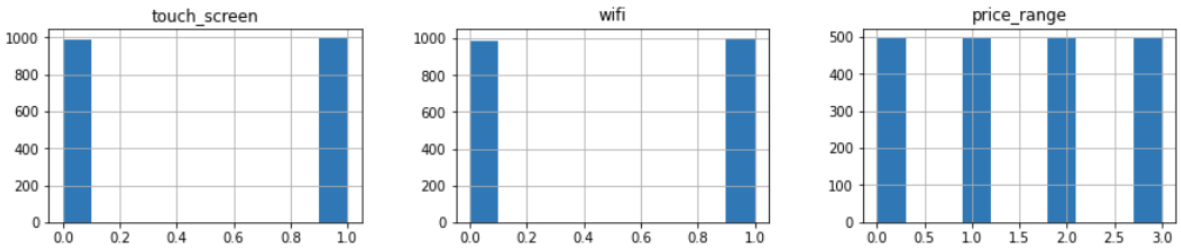
سپس هیستوگرام تمام ستونها را کشیدم که معرف پراکندگی آنهاست.



پراکندگی قدرت باتری تقریبا یکسان است.
موبایلهای بیشتری با قدرت ساعت 0.5-0.75 وجود دارد.
تعداد موبایلهای با 1 و 2 سیمکارت برابر است.
تعداد بیشتری موبایل با fc کم وجود دارد.
تعداد موبایلهای دارای قابلیت 4g بیشتر است.
تعداد موبایلهای با حافظه داخلی کمتر بیشتر است.
تعداد موبایلهای با m_dep کمتر ، بیشتر است.
تقریبا پراکندگی موبایلها بر اساس وزن موبایل یکسان است.

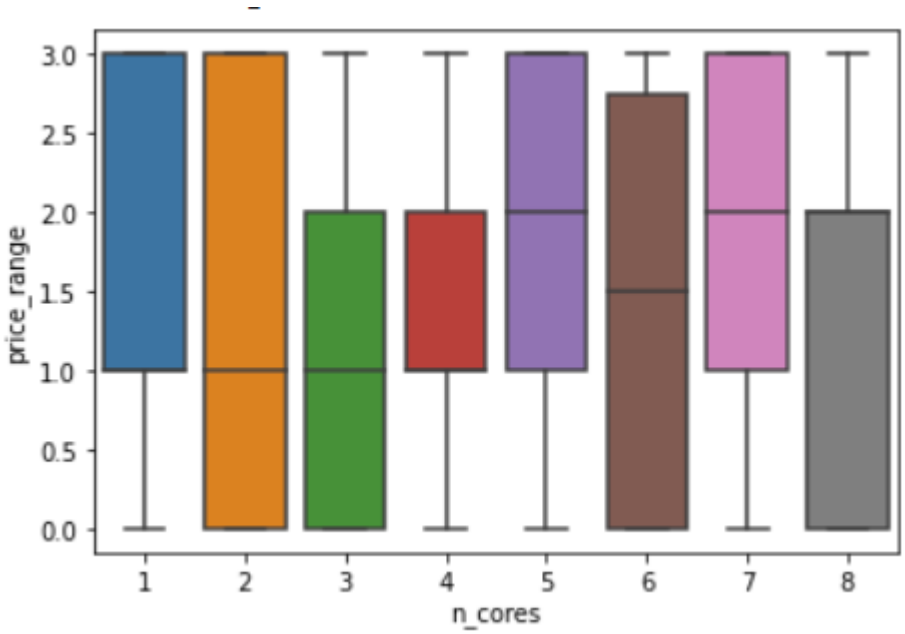


تعداد موبایلهای با 4 هسته داخلی بیشتر از بقیه است.
تعداد موبایلهای با 20 pc بیشتر است.
تعداد بیشتری موبایل px_height کمی دارند(رابطه نزولی)
تعداد موبایلها با px_width های مختلف تقریبا یکسان است.
تعداد موبایلهای با زمان مکالمه 10 از بقیه کمتر است.
تعداد موبایلهای دارای 3g 3 برابر موبایلهای فاقد این قابلیت است.



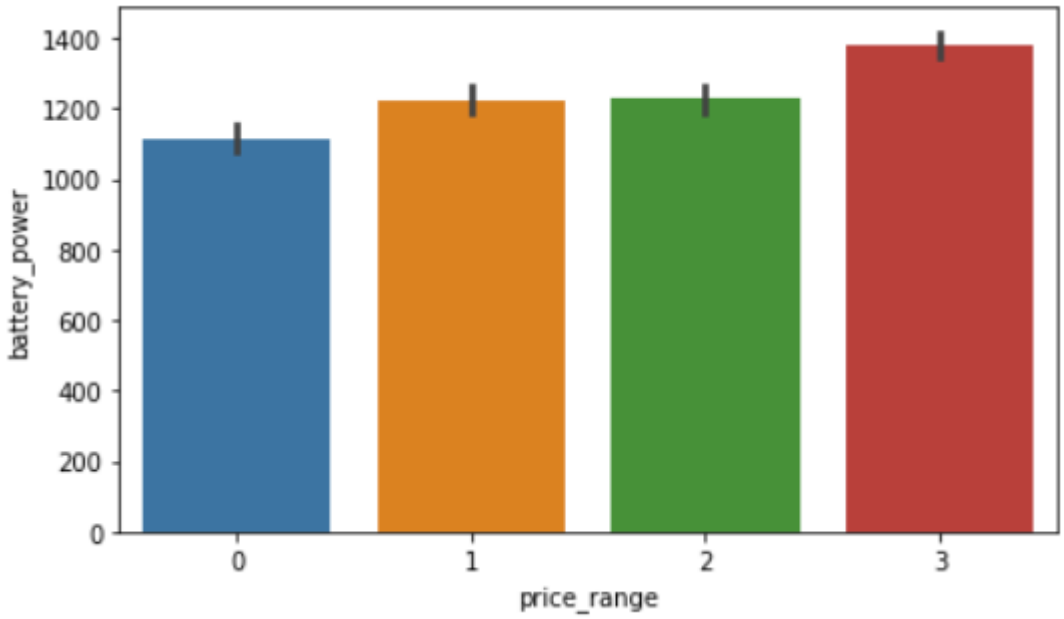
تعداد برابری موبایل دارای صفحه لمسی و بدون آن وجود دارند.
تعداد برابری موبایل دارای قابلیت و بدون قابلیت wifi وجود دارد.
تعداد موبایلهای موجود در هر محدوده قیمت برابر است.

سپس نمودار جعبه ای تعداد هسته ها و محدوده ی قیمت را کشیدم.



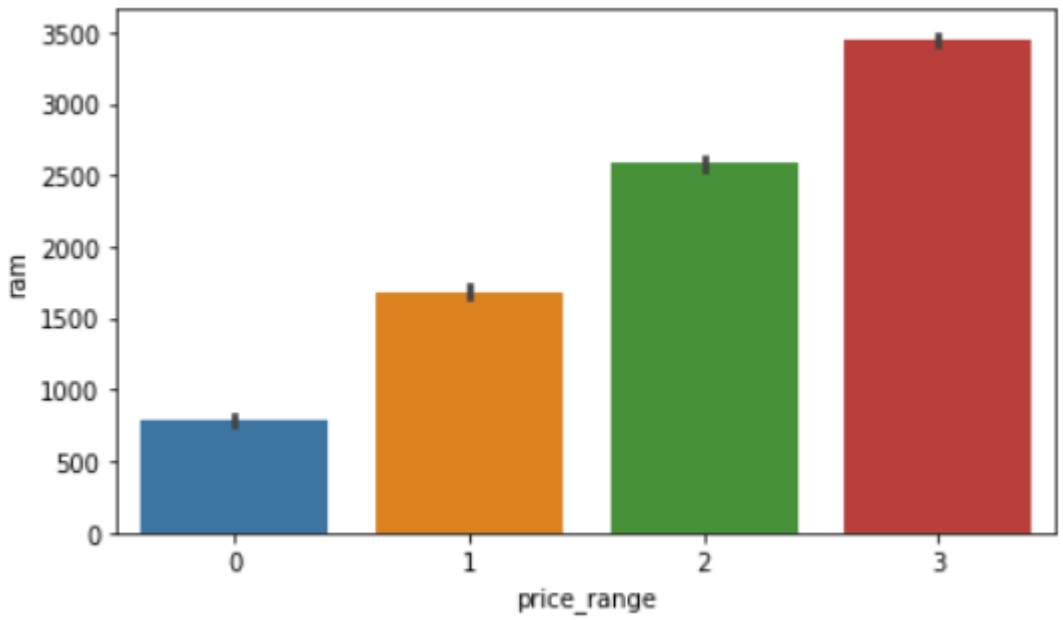
از این نمودار میتوان نتیجه گرفت که موبایلهای با تعداد هسته 5 و 7و 8 میانگین قیمت بیشتری دارند اما ماکسیمم و مینیمم قیمت موبایلهای در هر گروه(با هر تعداد هسته) برابر است اما میانگین و چارت متفاوتی دارند.

نمودار قدرت باتری و محدوده قیمت را کشیدم



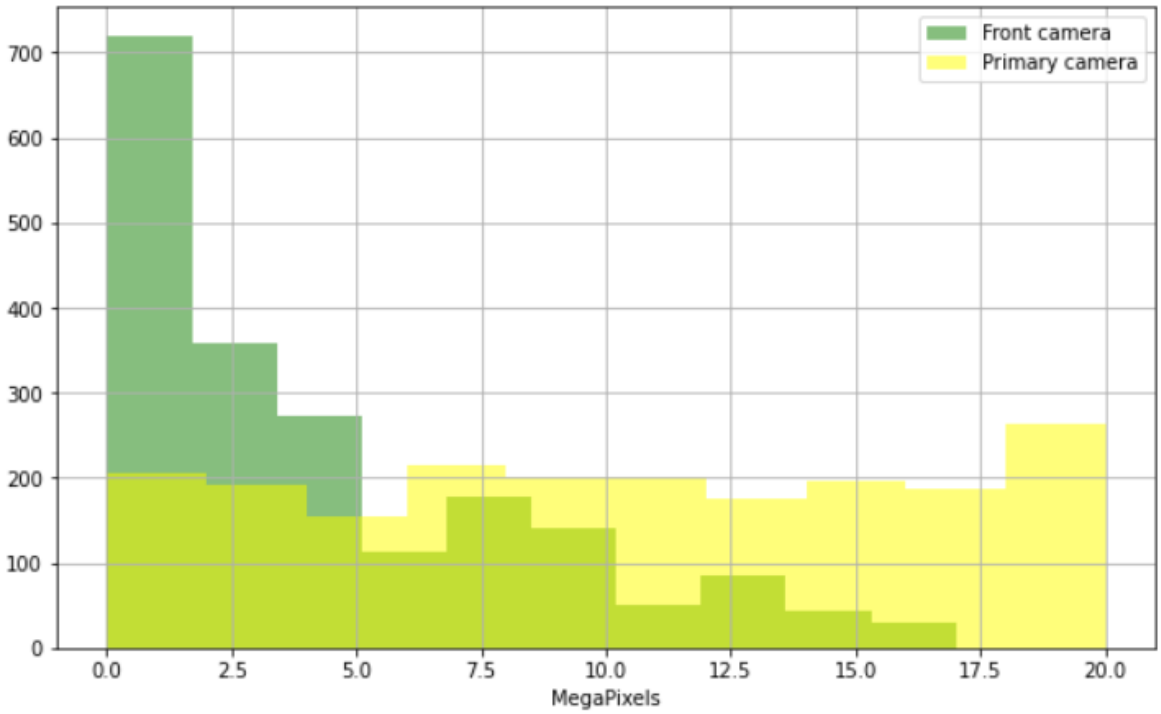
از این نتیجه میشود که موبایلهای با قیمت بالاتر به طور میانگین قدرت باتری بهتری دارند.

نمودار ram و محدوده قیمت را کشیدم.



از این نمودار نتیجه میشود که موبایلهای با قیمت بالاتر به طور میانگین ram بهتری دارند.

نمودار هیستوگرام دوربین اصلی و دوربین جلو را بر حسب مگاپیکسل کشیدم.



از این نتیجه میشود که تعداد موبایلهای با دوربین جلو ضعیف بیشتر است اما تعداد موبایلهای با دوربین اصلی قوی بیشتر است.
در نتیجه میتوان گفت اکثر موبایلها دوربین اصلی خوب و دوربین جلو ضعیفی دارند.

آزمون فرض

1. فرض کردم که میانگین سرعت ساعت در 1.5 است و آن را با استفاده از 1sample t-test آزمودم که pvalue = 0.23 بدست آمد از این رو که 0.23>0.05 بود فرض را پذیرفتم.

2.فرض کردم ارتباطی بین تعدا هسته ها و دوتایی بودن سیم کارت وجود دارد و آن را با استفاده از 2 sample t-test آزمودم که فرض رد شد و ارتباطی بین آنها وجود ندارد.

3. فرض کردم ارتباطی بین ram و حافظه داخلی وجود دارد و آن را با استفاده از z-test آزمودم که رد شد.

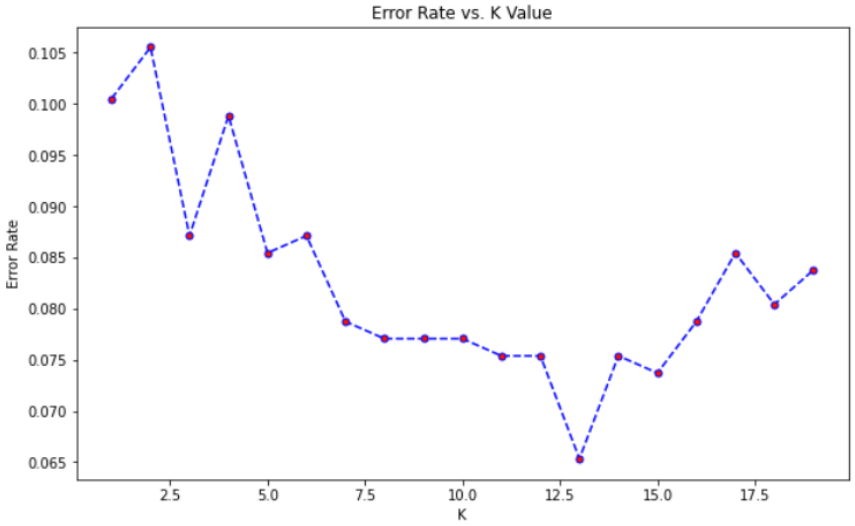
4. فرض کردم ارتباطی بین px_width و px_height وجود دارد و آن را با 2sample t-test آزمودم که رد شد.

5. فرض کردم که ارتباطی بین تعداد هسته ها و وزن موبایل وجود دارد و آن را با 2sample t-test آزمودم که رد شد.

ارائه مدل کلاسیفایر:

ابتدا داده ها را برای آموزش و تست به نسبت 70 به 30 جدا کردم بدین صورت که y = محدوده قیمت و x = باقی دیتاست.

ابتدا از k nearest neighbors استفاده کردم. برای تعیین تعداد همسایه ها نمودار خطا و تعداد همسایه ها رو کشیدم که بر حسب این نمودار تعداد 13 همسایه کمترین خطا را دارد.



سپس داده ها را آموزش دادم و دقت آن را بدست آوردم که برابر 93.47% بود.

کلاسبندی knn از استراتژی one-vs-one استفاده میکند.

سپس از کلاسبندی decision tree استفاده کردم و داده ها را آموزش دادم که دقتی برابر 80.9% داد.

این کلاسبندی هم از استراتژی one-vs-one استفاده میکند.

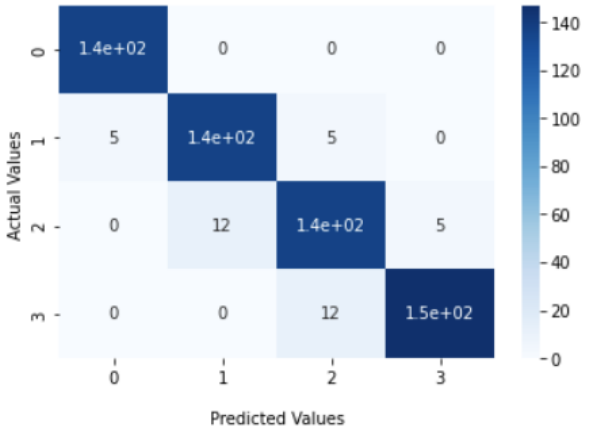
سپس از کلاسبندی random forest استفاده کردم و داده ها را آموزش دادم که دقتی برابر 87.27% داد.

این کلاسبندی نیز از استراتژی one-vs-one استفاده میکند.

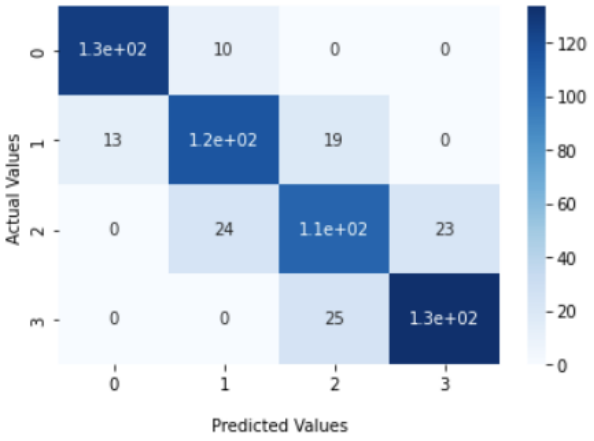
بررسی نتایج با confusion matrix :

نتایج بدست آمده بدین صورت است:

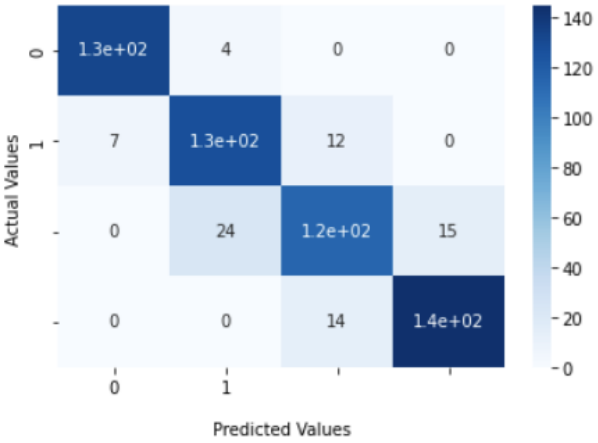
KNN :



DECISION TREE :

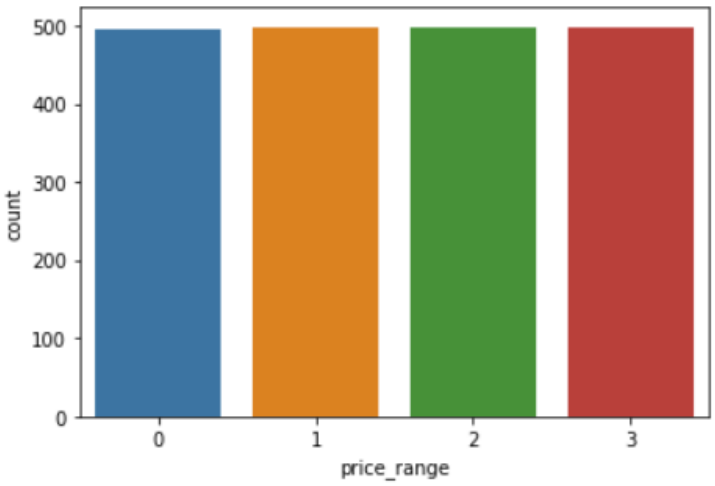


RANDOM FOREST :



مدل knn عملکرد بهتری داشته است نسبت به بقیه اما اختلافشان ناچیز است. چرا؟

6. نمودار پراکندگی مقدارهای مختلف price_range را رسم کردم و با توجه به نمودار داده ها متوازن میباشند:



در صورت نامتوازن بودن داده ها میتوان از 3 روش زیر استفاده کرد.

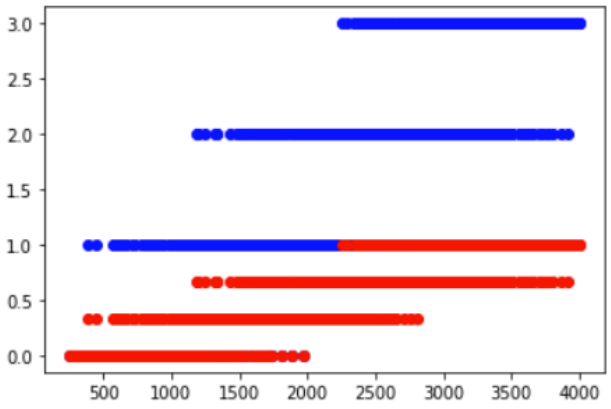
Under sampling : در اینجا تعداد داده های کلاس کمتر کافی است پس با نگهداشتن تمام نمونه های کلاس با داده کمتر، از کلاس با داده ی بیشتر تعداد کمتری نمونه برمیداریم تا داده ها متوازن شوند.

Over sampling : در اینجا تعداد داده های با کلاس کمتر کافی نیست و و ما نمیتوانیم از روش قبل استفاده کنیم پس باید به فکر افزایش تعداد داده های کلاس کمتر باشیم که این کار را با استفاده از روش boostrapping یا smote انجام میدهیم.

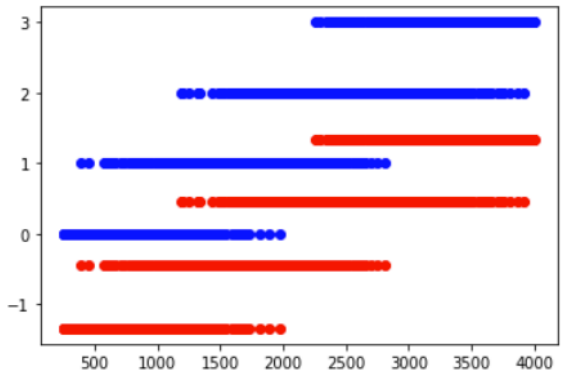
نمونه برداری مجدد از مجموعه داده های مختلف: تمام نمونه های کلاس کمتر را نگه میداریم و از کلاس بیشتر به تعداد نمونه های کلاس کمتر n نمونه برمیداریم(کار روش اول را n بار تکرا میکنیم) در این حالت n مدل مختلف را آموزش میدهیم.

7. scaling

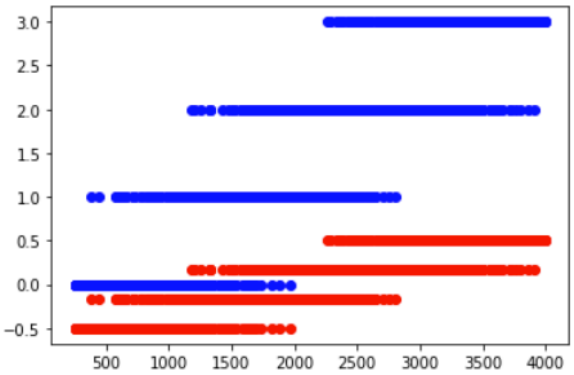
ابتدا از روش پیماش min max استفاده کردم و تفاوت داده های Y را روی نمودار به نمایش گذاشتم بدین صورت که داده های آبی داده های قدیمی و داده قرمز پس از پیمایش است.



سپس از روش استانداردسازی استفاده کردم.



سپس از نرمالسازی استفاده کردم.



سپس تاثیر این پیمایش ها را بر روی کلاسبندی knn سنجیدم.

پس از استاندارد سازی دقت مدل knn برابر 93.13 شد

پس از Min max scaler دقت مدل knn برابر 93.13 شد

پس از نرمالسازی دقت مدل knn برابر 93.97 شد.

8. داده ها را نسبت 80 به 20 جدا کردم.

دقت مدل ها به نسبت قبل افزایش یافت بدین صورت که :

با استفاده از استاندارد سازی روی کلاسبندی knn دقت برابر 94.97 شد.

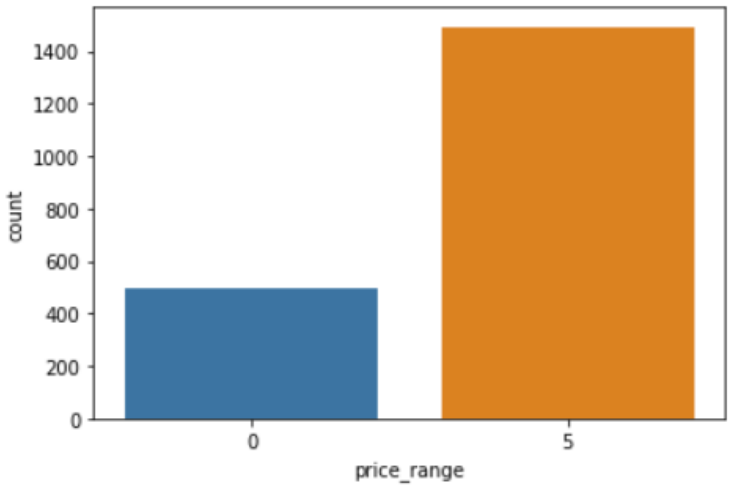
با استفاده از Min max scaler روی کلاسبندی knn دقت برابر 94.47 شد.

با استفاده از Normalization روی کلاسبندی knn 94.72 شد.

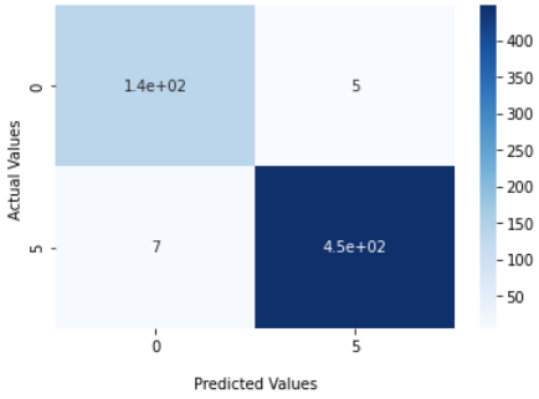
با اینکه دو روش Min max و نرمالسازی یک روش پیمایش هستند تفاوت در دقت به علت متفاوت بودن داده های آموزش و تست است.

9. متاسفانه قادر به انجام این بخش نبودم.

10. لیبل کلاسهای 1و2و3 را به 5 تغییر دادم یعنی محدوده قیمت فقط مقدار 0 و 5 میتواند بگیرد. در این صورت داده ها نامتوازن شدند

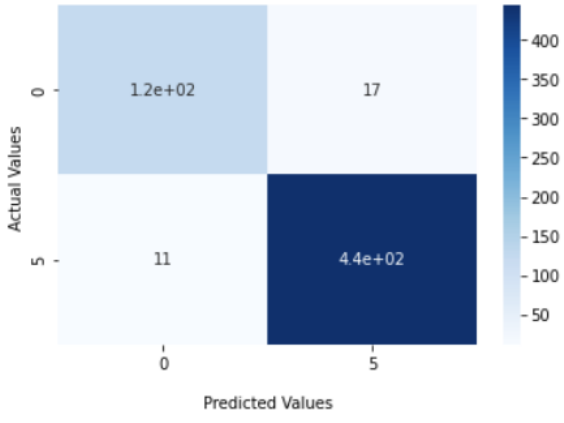


و به خاطر نامتوازن شدن داده ها دقت مدلها نیز افزایش میابد. مخصوصا مدلی مانند knn . دقت برابر 97.99 شد با ماتریس کوریلیشن زیر:

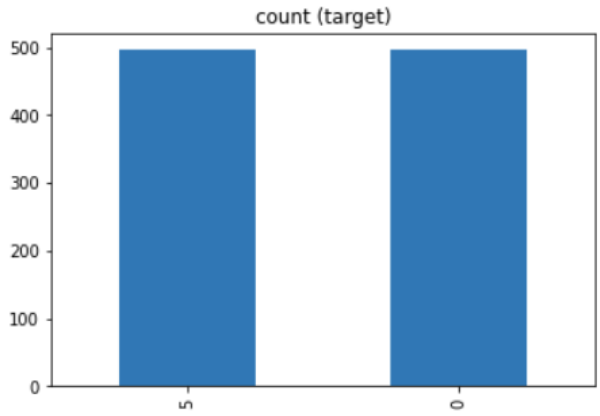


این دقت و ماتریس را برای مدل decision tree نیز بررسی کردم:

دقت برابر 95.31 شد



حال به متوازن کردن داده ها از روش Under sampling پرداختم. قابل ذکر است که مقدار داده ها از 2000 به 1000 کاهش یافت.



سپس دوباره دقت مدل knn و decision tree را بررسی کردم که به صورت 96.64 و 94.3 بود.

از این میتوان نتیجه گرفت که نامتوازن بوده داده ها میتواند باعث افزایش دقت شود زیرا مدل به نفع یک کلاس بیشتر میتواند کلاسبندی کند اما خطای کلاسبندی کلاس کوچکتر بیشتر رخ میدهد.

تمرین2 : apartment-rental-offers-in-germany

پس از خواندن داده ها به پاکسازی آن پرداختم. چون هدف نهایی total_rent بود هر داده ای که این ستونش خالی بود را حذف کردم. سپس هر داده ای که اجاره ی کل برابر 0 یا کوچکتر از 0 داشت را حذف کردم. همچنین هر داده ای که living_space برابر یا کوچکتر از صفر داشت را حذف کردم. سپس هر داده ای که سال ساخت(year_constructed) بزرگتر از 2022(امسال) را داشت حذف کردم و هر داده ای که base_rent کوچکتر از صفر داشت را حذف کردم.

سپس ستونهایی را که بیشتر از 60% خالی داشتند را حذف کردم.

پس از این مراحل در مجموع 9 ستون و 40811 ردیف حذف شدند.

داده هایی که هنوز خالی داشتند را به نمایش گذاشتم. در مجموع 17 ستون بودند.

برای ردیف هایی که گزینه newlyConst آنها صحیح بود اما condition ای نداشتند ستون condition آنها را برابر "اولین بار استفاده" گذاشتم.
بقیه ردیفهایی که ستون condition آنها خالی بود را برابر گزینه جدید "other" قرار دادم.

داده هایی که سال ساخت آنها خالی بود بر اساس condition گروهبندی کردم و و ستون سال ساخت آنها را بر اساس میانگین پر کردم.

ارتباط بین سال ساخت و محدوده ی سال ساخت را بدست آوردم و با توجه به آن داده هایی که محدوده سال ساخت آنها خالی بود را پر کردم.

سپس داده های تکراری را شناسایی کردم و یکی از آنها را نگهداشتم.

بعد از آن چند ستون که کاربردی برایم نخواهند داشت را حذف کردم.

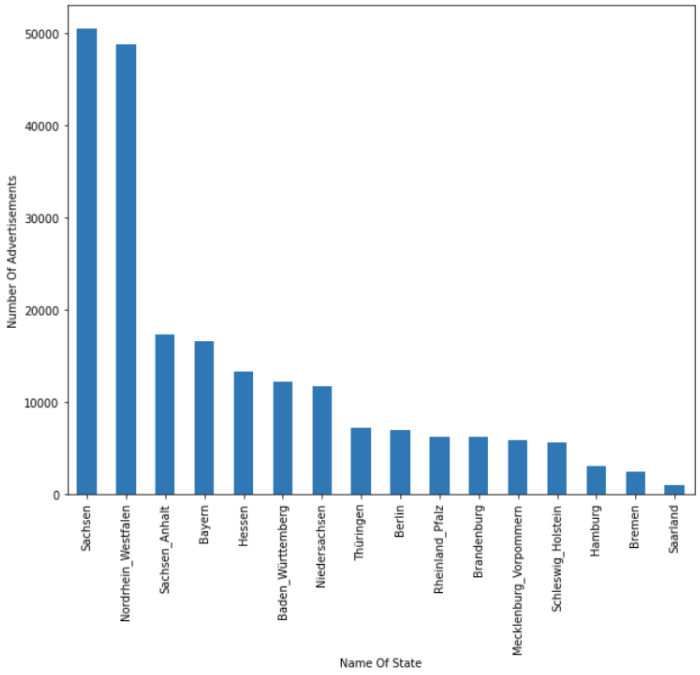
داده های پرت ستونهای عددی را حذف کردم.

ستون های خالی داده های عددی را با میانگین پر کردم.

به علت اینکه جلوتر قصد استفاده از one-hot vector رو دارم در اینجا داده های غیرعددی که بیش از 100 تنوع دارند را حذف کردم که در پیامد آن regio2 regio3 حذف شدند که به ترتیب برابر شهر و منطقه ی آگهی تبلیغاتی بودند.

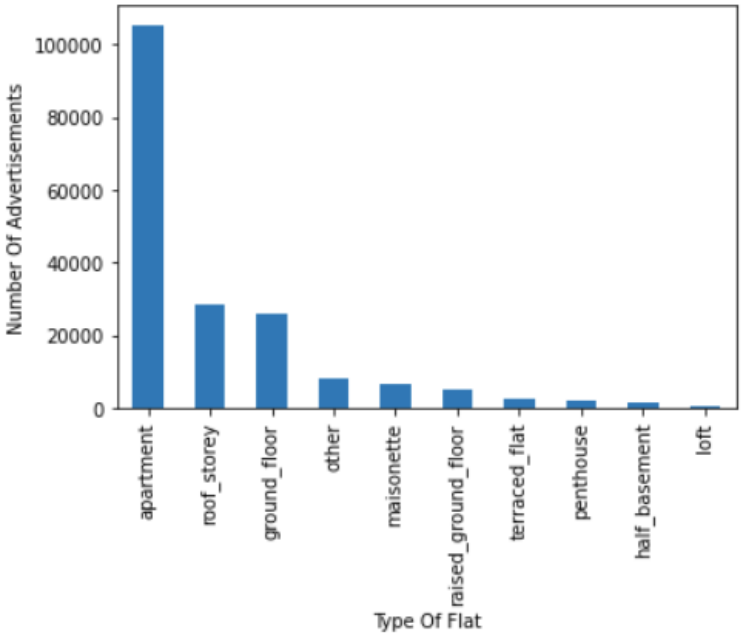
Visualization

تعداد آگهی های تبلیغاتی در هر ایالت را به نمایش گذاشتم



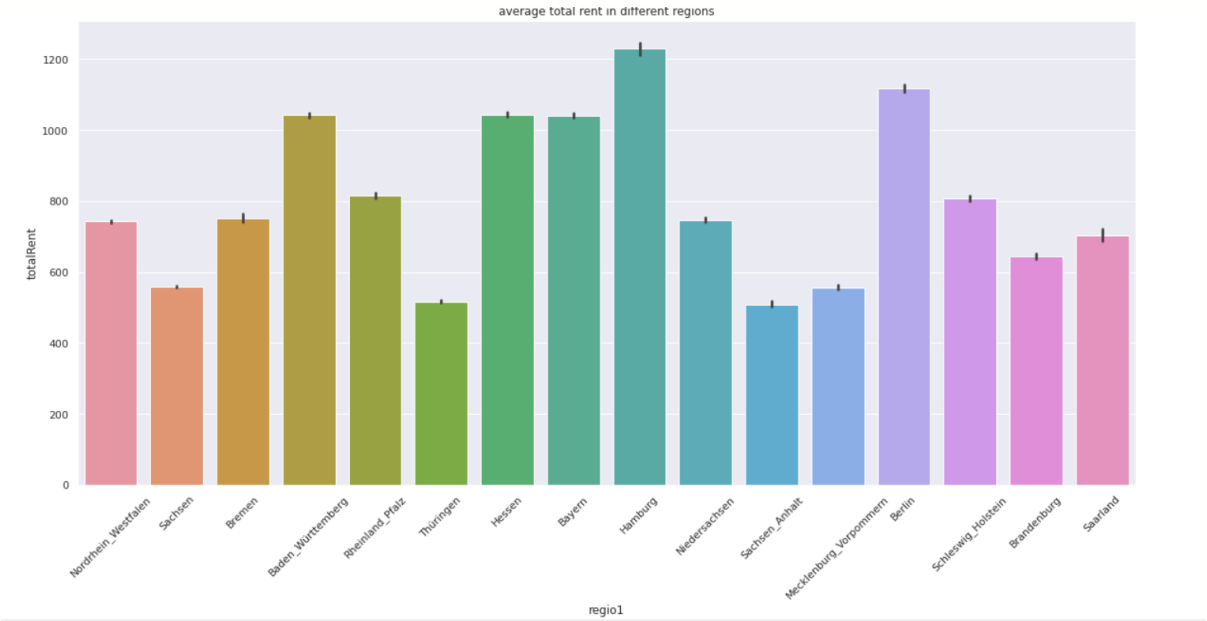
از این نمودار نتیجه میشود بیشترین تعدا آگهی در ایالت Sachsen و سپس nordrhein_westfalen میباشد.

سپس مقدار آگهی ها بر اساس نوع ساختمان را به نمایش گذاشتم.



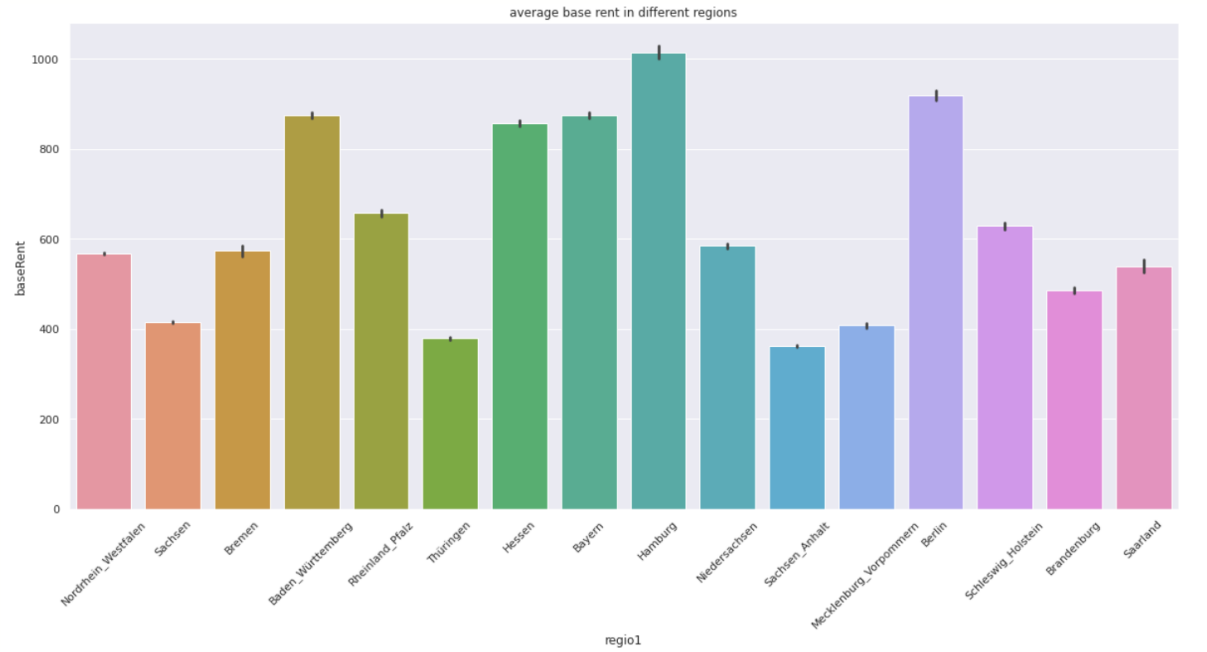
که از این نتیجه میشود بیشترین آگهی در مجموع با اختلاف برای آپارتمان است.

سپس نمودار میانگین اجاره ی کل را برای ایالتهای مختلف رسم کردم.



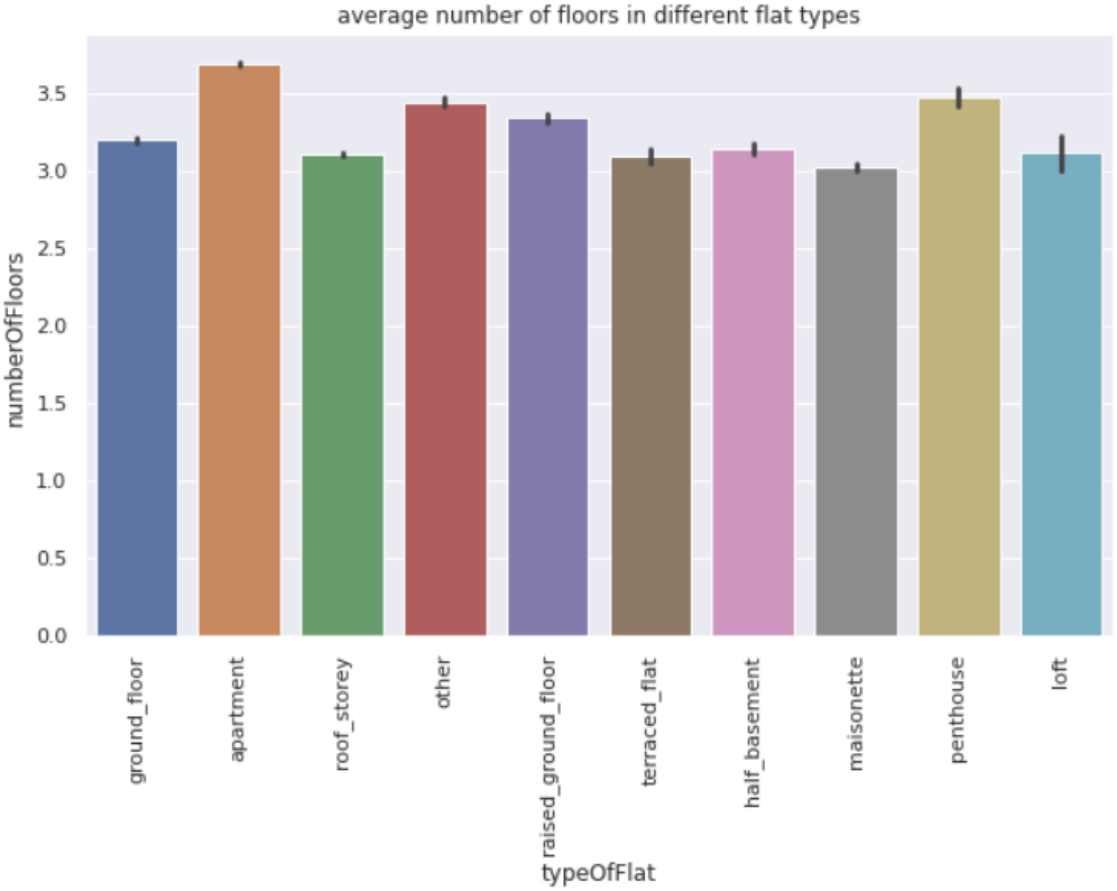
که نتیجه میدهد میانگین اجاره کل در ایالت hamburg و سپس berlin از همه بیشتر است.

سپس نمودار میانگین اجاره ی پایه را برای ایالتهای مختلف رسم کردم.

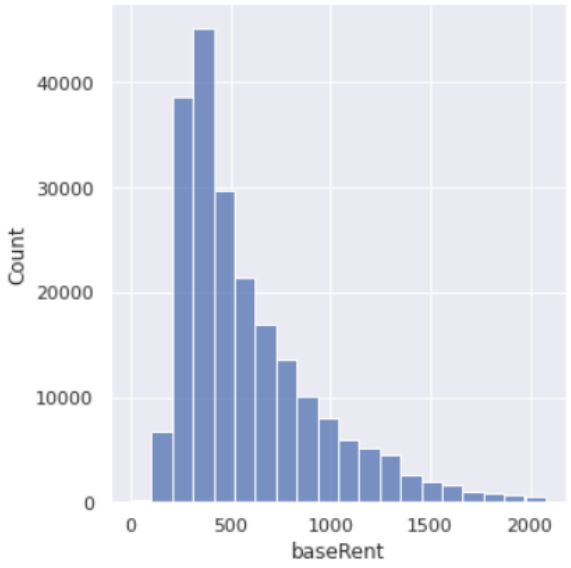


که در اینجا نیز دو ایالت hamburg و سپس berlin از همه بیشتر است. تقریبا اختلاف اجاره کل و پایه اجاره برای همه ایالت ها حدود 200 است.

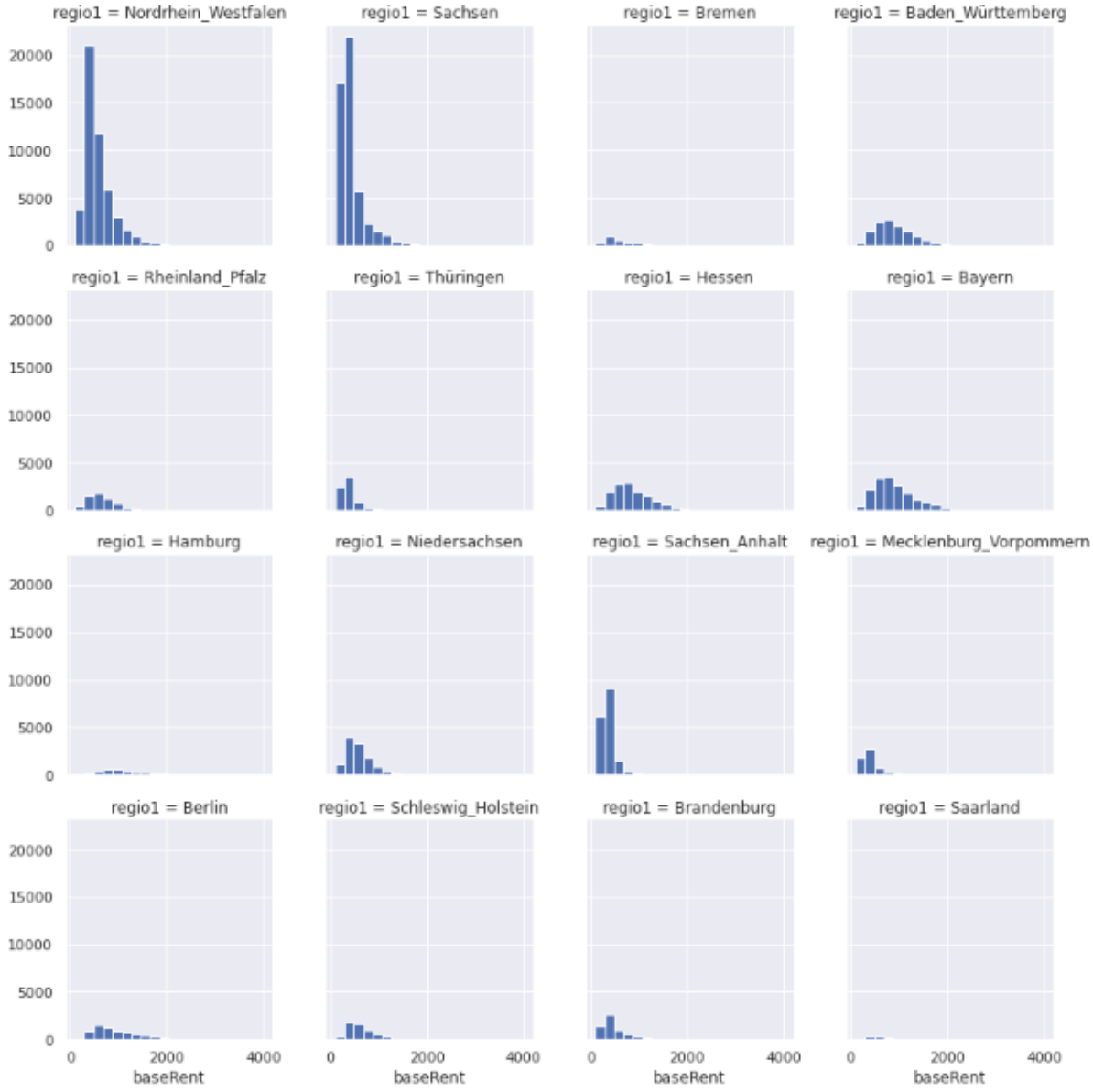
سپس نمودار میانگین تعداد طبقات در مدلهای مختلف ساختمان را رسم کردم.



سپس تعداد آگهی های با اجاره ی پایه مختلف را کشیدم که از این نتیجه میشود بیشترین تعداد آگهی برای خانه با اجاره پایه 400-500 و سپس برای 300-400 است و هرچه اجاره پایه افزایش میابد تعدا آگهی های منتشر شده کاهش یافته است.

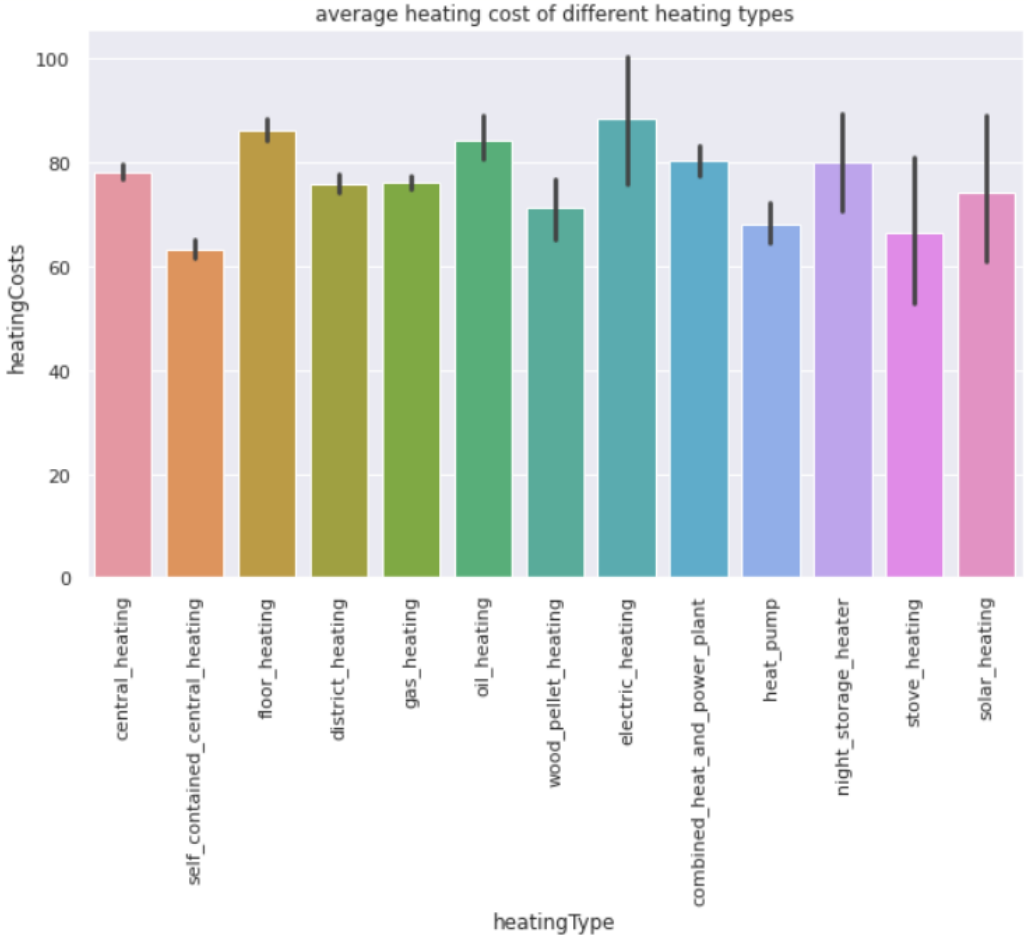


سپس نمودار هیستوگرام اجاره پایه را به تفکیک ایالتها رسم کردم. تا بدست آورم که بیشترین آگهی در نمودار بالا مربوط به کدام ایالت بود که با توجه به نمودار زیر متعلق به ایالت Sachsen و nordrhein_westfalen میباشد.

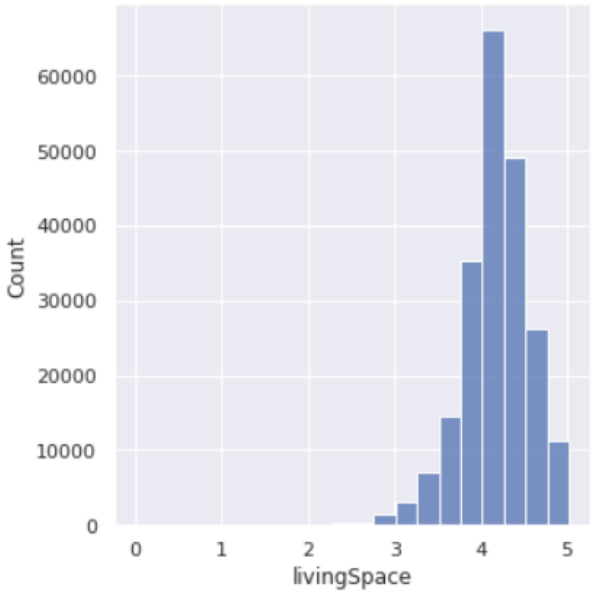


سپس نمودار میانگین اجاره پایه بر اساس سال ساخت را کشیدم تا ارتباط این 2 مولفه را بیابیم. از ایم نتیجه میشود که به طور حدودی از سال 1990(محدوده سال ساخت 5) به بعد هرچه ساخت خانه جدیدتر بوده است اجاره پایه آن نیز بیشتر بوده و خانه هایی که بین 1980 و 1990 ساخته شده اند به طور میانگین اجاره پایه کمتری دارند.

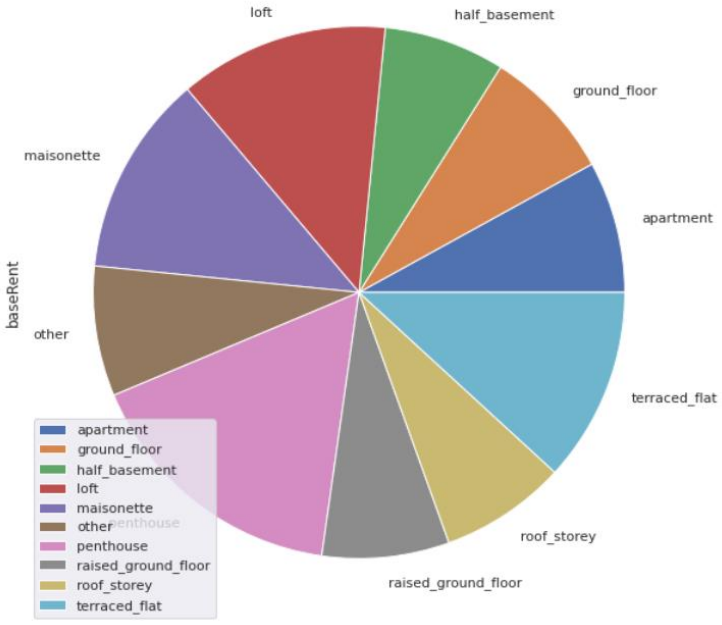
به دنبال پیدا کردن ارتباطی بین نوع وسیله گرمایشی و هزینه های گرمایشی بودم که نمودار زیر را رسم کردم و متوجه شدم که بیشترین میانگین هزینه برای مدل گرمایش الکتریکی میباشد که ماکسیمم هزینه نیز برای همین مدل است و کمترین هزینه برای self-contained-central-heating میباشد . اما در مجموع میانگین هزینه ی مدلهای مختلف نزدیک است و اختلاف فاحشی مشاهده نمیشود.

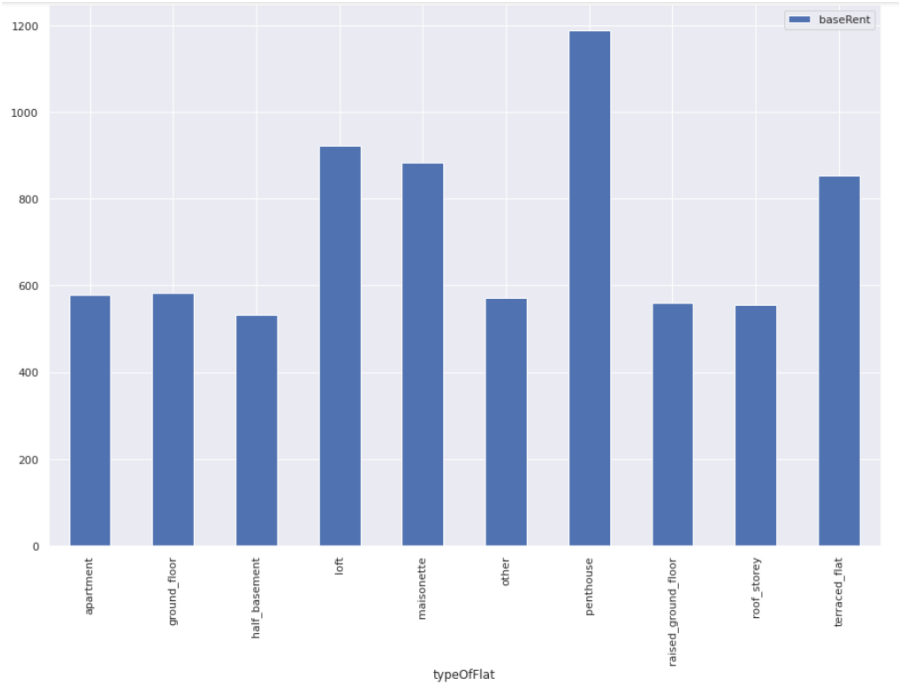


نمودار تعداد آگهی ها بر حسب living_space را رسم کردم که از آن میتوان نتیجه گرفت بیشترین آگهی ها مربوط به محیط زندگی حدودا برابر 4 بوده است.



نمودار اجاره پایه بر اساس نوع ساختمان را به دو صورت رسم کردم.



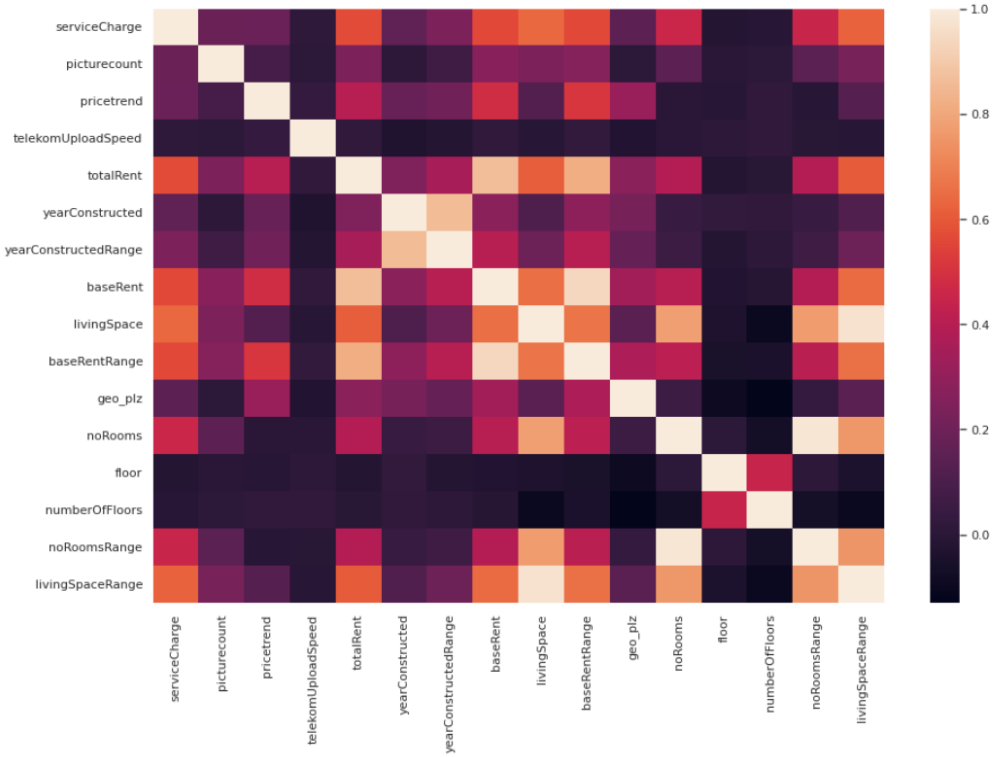


که از این 2 نتیجه میشود بیشترین اجاره پایه مربوط به penthouse و کمترین مربوط به half-basement میباشد.

مدل سازی قیمتها:

داده ها را با استفاده از Min max scaler نرمالسازی کردم.

سپس heatmap داده های عددی آن را کشیدم.



سپس داده های غیر عددی آن را با استفاده از one-hot endcoding به داده های عددی تبدیل کردم. همچنین داده های bool را به 0 و 1 تبدیل کردم.

سپس داده ها را برای آموزش و تست جدا کردم به نسبت 75 به 25 و با استفاده از linear regression آنها را مدلسازی کردم که دقتی برابر 94.46% داد.

Multi processing

یک تابع به نام dataCleaning ساختم و عواملی که برای پاکسازی داده های عددی در بخش اول استفاده کردم را در آن پیاده سازی کردم.

سپس با استفاده از pool و با گروهبندی بر اساس regio1 به صورت بازگشتی و موازی تابع dataCleaning را فراخوانی کردم و زمان runtime آن را محاسبه کردم که برابر 9.86 شد.

سپس بدون استفاده از multi processing و به صورت بازگشتی تابع dataCleaning را فراخوانی کردم که زمان runtime آن برابر 7.68 شد.

گمان میکنم علت اینکه اجرای موازی بیشتر طول کشید به علت صرف کردن زمانی برای تقسیم کارهاست.

Dask

با استفاده از dask به پاکسازی داده ها بر اساس همان تابع قسمت قبل پرداختم و زمانی بسیار کمتر و برابر 0.041 ثانیه به طول انجامید.

Pyspark

با استفاده از pyspark به پاکسازی داده ها بر اساس همان تابع قسمت قبل پرداختم و زمانی بسیار کمتر و برابر 0.135 ثانیه به طول انجامید.