نام و نام خانوادگی: درسا الهوئی

شماره دانشجویی: ۹۷۲۲۲۰۰۵

شماره تمرین: ۱,۱ **تاریخ تحویل: ۲۰ فروردین ۱۴۰۱**

عنوان تمرین: اعمال موارد گفته شده در صورت تمرین بر مجموعه داده مربوط به تلفن همراه

---

۱. مقدمه

در این پروژه هدف انجام بررسی آماری داده های مربوط به مشخصات و قیمت تلفن همراه می باشد. همچنین سعی بر آن است تا با به کارگیری مدلی قیمت تلفن همراه با استفاده از مشخصات داده شده پیشبینی شوند. چندین رویکرد نیز در جهت بهبود این مدل به کار گرفته شده است.

۲. بررسی و پاکسازی داده ها

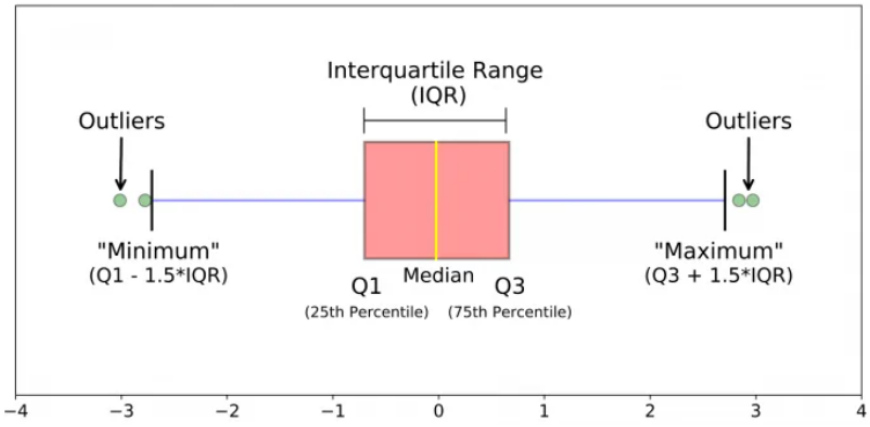
داده ها را بر اساس کلاس قیمت به دو قسمت آموزشی و آزمایشی تقسیم می کنیم. نسبت داده ها در قسمت آموزشی و آزمایشی با توجه به کلاس قیمت یکسان است. به این صورت که ۸۰٪ داده های هر کلاس قیمت را در قسمت آموزشی و ۲۰٪ باقی را در قسمت آزمایشی قرار می دهیم.

تعداد داده های آموزشی ۱۶۰۰ می باشد که این داده ها فاقد مقدار null می باشند.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1600 entries, 8 to 1017
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   battery_power  1600 non-null   int64
 1   blue           1600 non-null   int64
 2   clock_speed    1600 non-null   float64
 3   dual_sim       1600 non-null   int64
 4   fc             1600 non-null   int64
 5   four_g         1600 non-null   int64
 6   int_memory     1600 non-null   int64
 7   m_dep          1600 non-null   float64
 8   mobile_wt      1600 non-null   int64
 9   n_cores        1600 non-null   int64
 10  pc             1600 non-null   int64
 11  px_height      1600 non-null   int64
 12  px_width       1600 non-null   int64
 13  ram            1600 non-null   int64
 14  sc_h           1600 non-null   int64
 15  sc_w           1600 non-null   int64
 16  talk_time      1600 non-null   int64
 17  three_g        1600 non-null   int64
 18  touch_screen   1600 non-null   int64
 19  wifi           1600 non-null   int64
 20  price_range    1600 non-null   int64
dtypes: float64(2), int64(19)
memory usage: 275.0 KB
```

در گام بعد به حذف داده های پرت می پردازیم l; این کار را با به کارگیری روش IQR صورت می گیرد. با مشخص کردن چارک ها و محاسبه IQR (IQR = Q3 – Q1)، بازه ی نرمال داده ها را از Q1 – 1.5*IQR تا

Q3 + 1.5*IQR در نظر می گیریم و به این طریق سایر داده های خارج از این بازه حذف می شوند. در نتیجه این گام، تعداد داده ها از ۱۶۰۰ به ۱۲۰۸ داده کاهش می یابد.

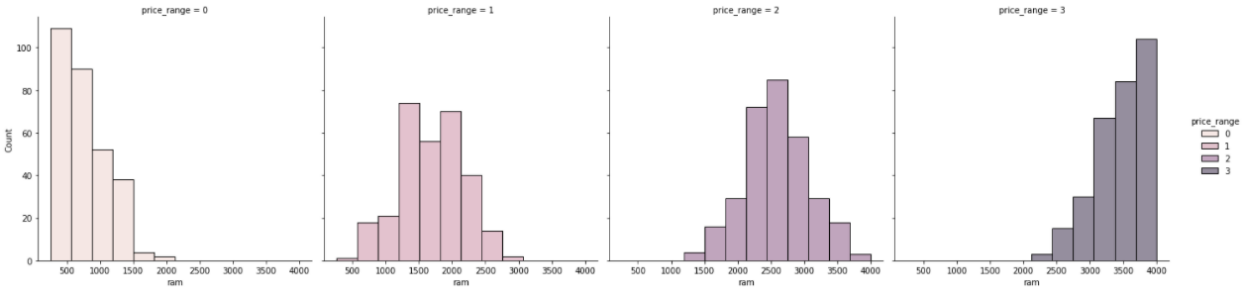


پس از حذف داده های پرت، داده ها را به منظور وجود داده تکراری و حذف آن بررسی می کنیم. با توجه به اینکه تعداد داده ها پس از این اعمال تغییر ثابت می ماند، در می یابیم که سطر تکراری ای در داده ها وجود ندارد.

۳. تصویرسازی و بررسی آماری داده ها

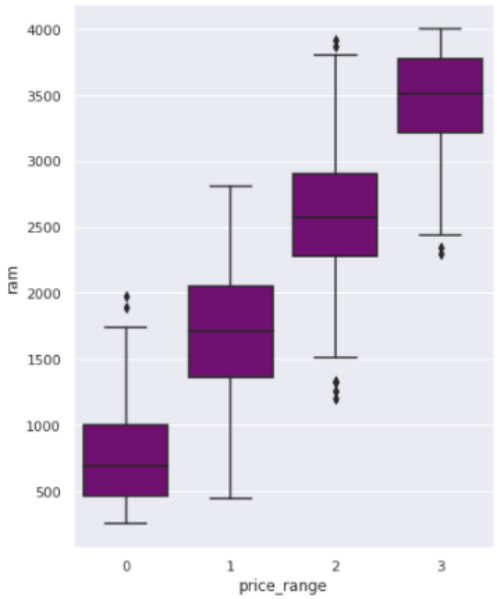
ابتدا مقادیر یکتای هر ستون (که هر یک بیانگر ویژگی ای خاص می باشند) را به منظور بررسی کلی چاپ می کنیم. با نگاهی اجمالی، می یابیم که غالب ستون ها مقادیر منحصر به فرد محدودی دارند.

تصویر سازی را با نمایش ارتباط بین ram و price_range آغاز می کنیم. به این صورت که در هر یک از کلاس های قیمت، تعداد تلفن های همراه را با توجه به ram آنها نمایش می دهیم.



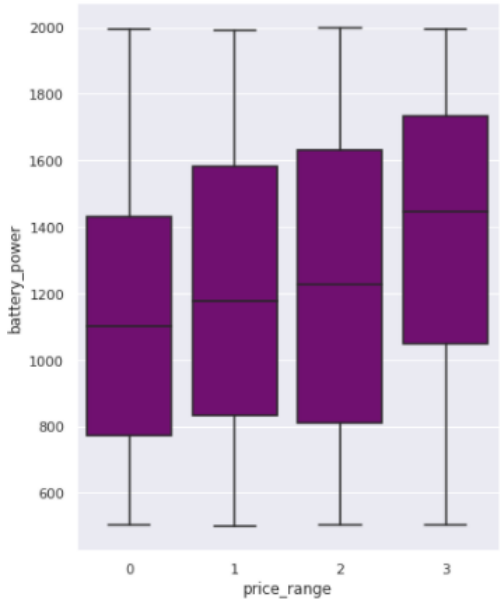
همانطور که در تصویر نمودار قابل مشاهده است، کلاس قیمت بالاتر از ram نسبتا بیشتری در مقایسه با کلاس قیمت پایین تر برخوردار است.

در ادامه نمودار جعبه ای ram در هر کلاس قیمت را رسم می کنیم. این نمودار روشی استاندارد برای نمایش توزیع داده ها می باشد.

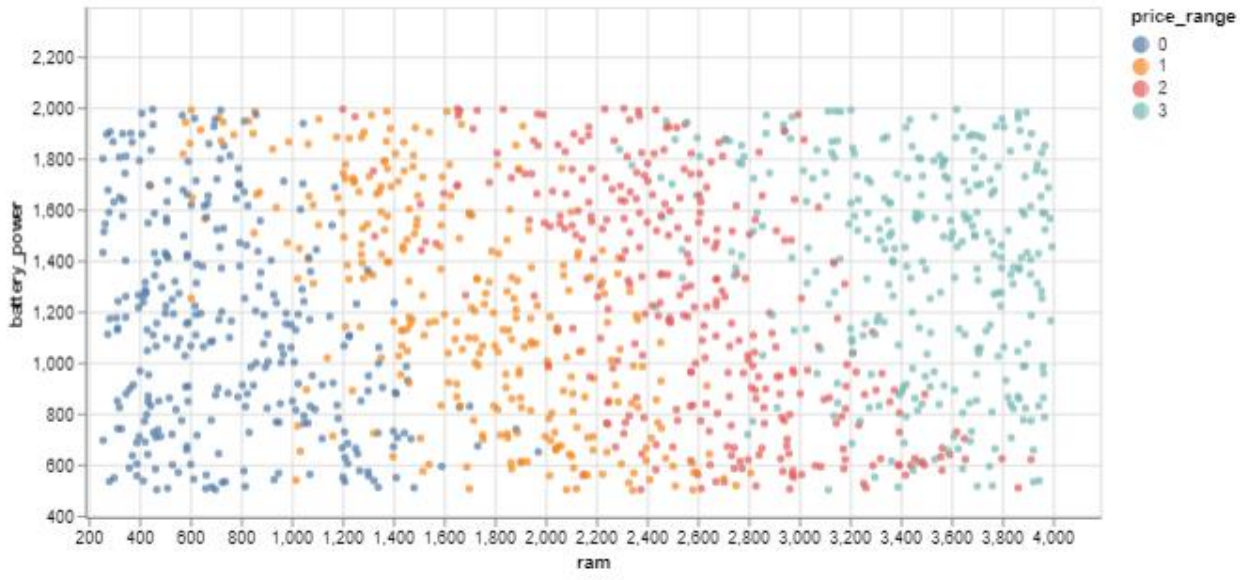


این نمودار، آنچه که از نمودار قبلی بدست آوردیم را تایید می کند. یعنی با افزایش ram، دامنه قیمت نیز افزایش می یابد.

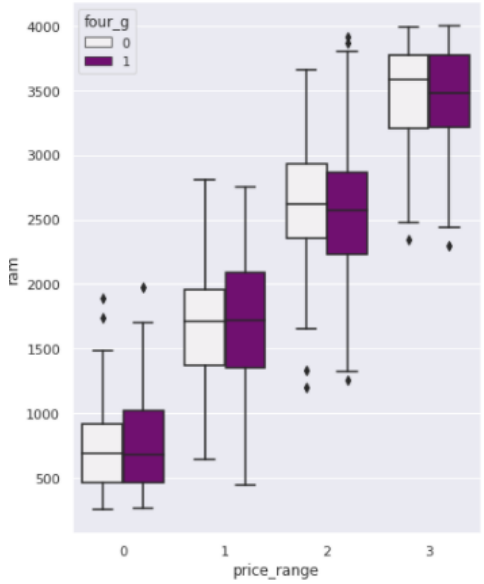
نمودار جعبه ای بعدی مربوط به battery_power در هر کلاس قیمت می باشد. با نگاهی کلی می توان دریافت که در کلاس قیمتی بالا تر، حجم بیشتری از پراکندگی داده ها در battery_power بالاتر می باشد.



نموداری به تصویر کشیده شده در زیر، پراکندگی قیمت را با توجه به دو ویژگی ram و battery_power نمایش می دهد. با دقت به این نمودار می توان دریافت که این دو ویژگی با قیمت رابطه مثبتی دارند. بدین صورت که با فزایش این دو، تلفن همراه در کلاس قیمتی بالا تری قرار می گیرد.



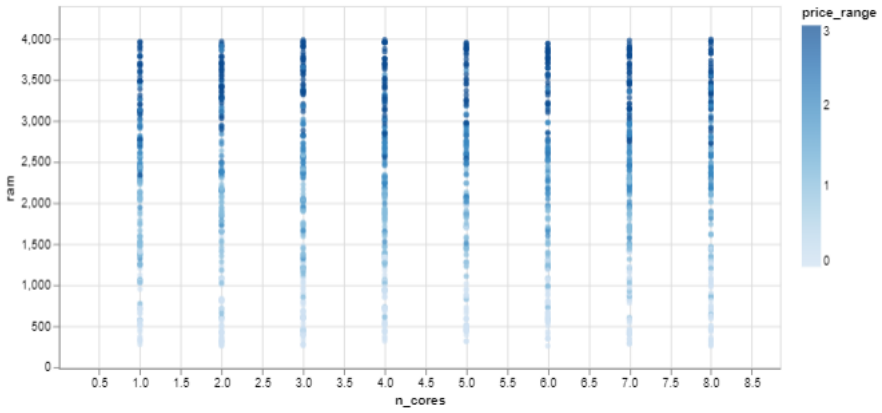
نمودار جعبه ای زیر با استفاده از دو ویژگی ram و four_g در کلاس های قیمت رسم شده است. پیش تر رابطه بین ram و کلاس های قیمت را مورد بررسی قرار دادیم. اما با ترکیب آن با ویژگی four_g نتیجه بیشتری حاصل نمی شود. چراکه برخوردار نبودن از ویژگی four_g تاثیر محسوسی بر نمودار ندارد.



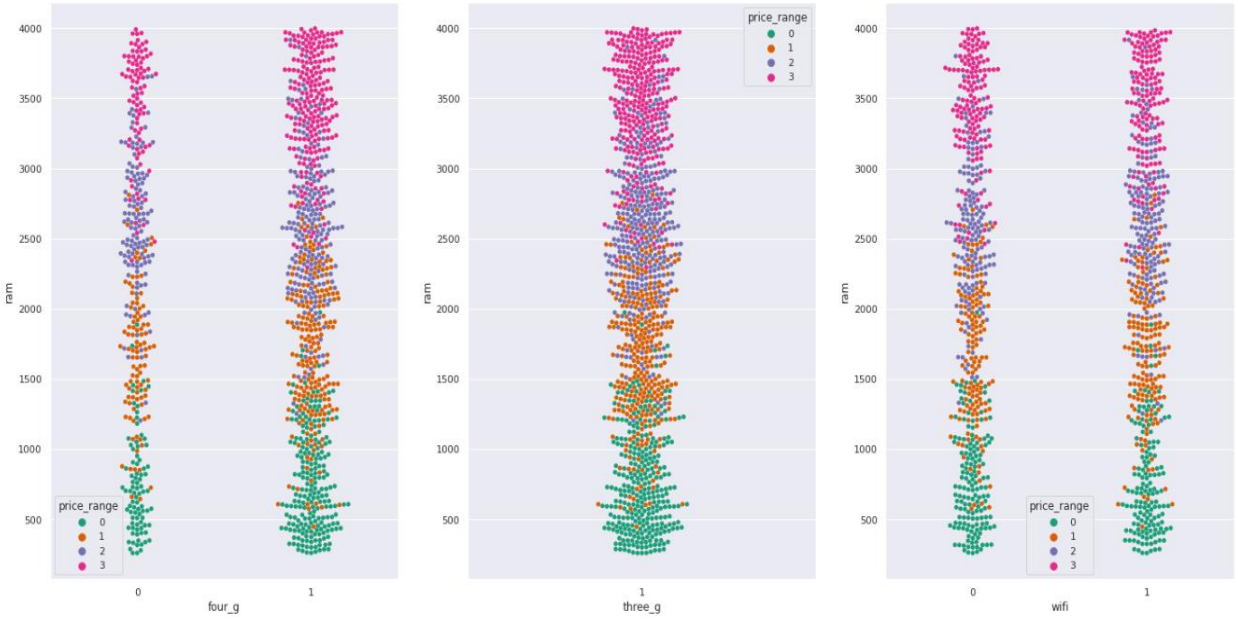
به منظور دسترسی به جزییات بیشتر، دو ویژگی قبلی را با بهره گیری از نمودار ویولن به تصویر می کشیم. با در نظر گرفتن تعداد کم داده های در دسترس این موضوع کع نتیجه محسوسی حاصل نشده است، قابل درک است.

نمودار ویولن ترکیبی از نمودار جعبه ای و نمودار چگالی می باشد که شکل توزیع داده ها را نمایش می دهد. هر چه تعداد داده ها در یک محدوده خاص بیشتر باشند، ویولن برای آن دامنه نیز بزرگتر است. در واقع هر چه عرض ویولن در یک قسمت بیشتر باشد، به این معنی است که نمونه های مجموعه داده با احتمال بیشتری در آن قسمت قرار می گیرند.

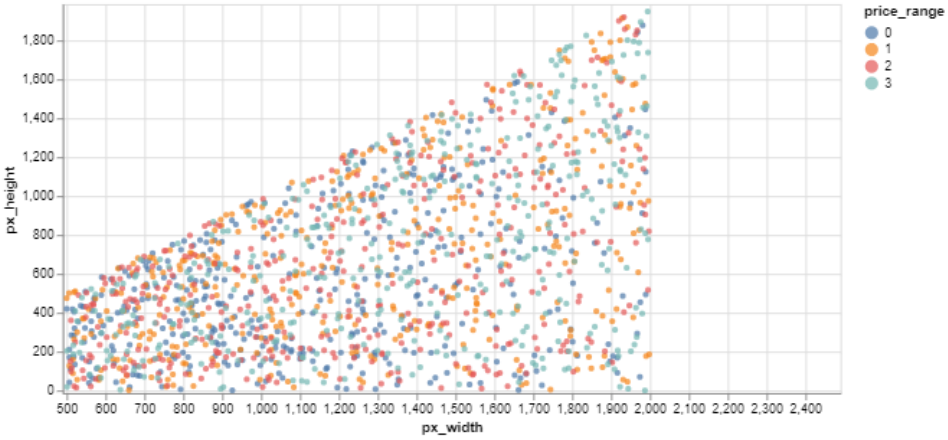
در نمودار زیر پراندگی قیمت با توجه به دو ویژگی ram و n_cores نمایش داده شده است. به صورت کلی مشاهده می شود که گراکنگی قیمت بالاتر در تلفن همراه با ram و n_cores بیشتر، وجود دارد (به تنالیته رنگ در هر ستون دقت کنید).



هر یک از سه نمودار زیر حجم پراکندگی قیمت را با توجه به ویژگی ram در ترکیب با ویژگی های wifi، three_g و four_g نمایش می دهند. برای مثال نمودار میانی نشان می دهد که تعداد حجم بیشتری از تلفن های همراه ارائه شده از ویژگی four_g برخوردارند و همچنین با افزایش ram قیمت آنها نیز افزایش می یابد.



طبق نمودار زیر به طور کلی می توان مشاهده کرد که تلفن های همراه با ابعاد کوچکتر در کلاس قیمت کمتر، تراکم بیشتری دارند.



۴. آزمون های فرض

اولین تست آماری به کار گرفته شده، آزمون یک نمونه ای t-test می باشد. میانگین n_cores در کلاس قیمت ۳ را به صورت شهودی ۶ عدد در نظر گرفته ایم که این فرض صفر ما می باشد. حال با اجرای این تست، درستی فرض را بررسی می کنیم. با توجه به اینکه مقدار p-value بدست آمده کمتر از ۰,۰۵ می باشد، فرض صفر رد می شود.

```
p value: 4.2960675997042104e-24
```

تست دوم نیز مانند آزمون قبل، تست یک نمونه ای t-test است. میانگین battery_power برابر ۱۲۴۱ را فرض صفر در نظر می گیریم (طبق توضیحات داده ها ای عدد تقریبا با میانگین برابر است بنابراین انتظار می رود فرض صفر تائید شود). همانطور که مشاهده می شود مقدار p-value از ۰,۰۵ بیشتر است و در نتیجه فرض صفر پذیرفته می شود.

```
p value: 0.8410815339994593
```

	battery_power
count	1600.000000
mean	1241.006250

تست سوم chi2 است. فرض صفر این است که price_range و touch screen که ویژگی های غیرعددی هستند، به یکدیگر مرتبط نمی باشند. با توجه به مقدار p-value این فرض درست می باشد.

```
p value: 0.9454723168495832
```

تست چهارم مانند حالت قبل است با این فرض که price_range و bluetooth با یکدیگر ارتباطی ندارند. طبق مقدار بدست آمده برای p-value این فرض درست می باشد.

```
p value: 0.9998946315164954
```

تست پنجم، تست spearman می باشد که این تست همبستگی دو متغیر عددی را نمایش می دهد. فرض صفر در اینجا نا مرتبط بودن ram و int_memory می باشد.که با توجه به مقدار p-value فرض درستی است.

```
p value: 0.2378298469482512
```

۵. رگرسیون لجیستیک

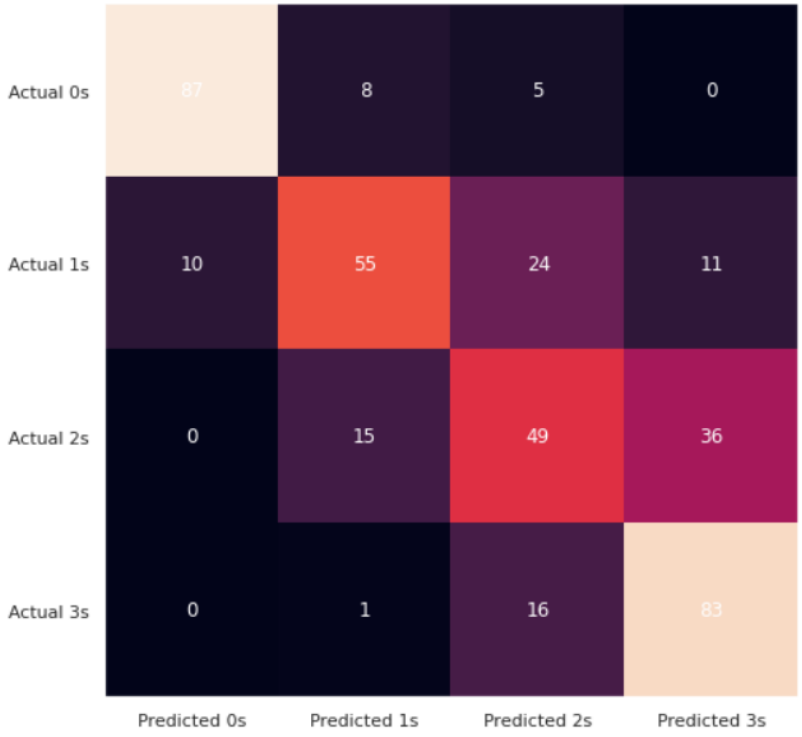
رگرسیون لجیستیک، یکی از الگوریتم های یادگیری ماشین است که از آن غالبا در مسائل طبقه بندی که متغیرهای کیفی مطرح هستند استفاده می شود. به طور کلی خروجی در رگرسیون لجستیک به شکل ۰ یا ۱ می باشد. زمانی که تعداد کلاس های خروجی برابر ۲ باشد، به آن طبقه بندی باینری گفته می شود. تفاوت رگرسیون لجستیک با رگرسیون خطی این است که در رگرسیون خطی، معادله خط می تواند هر مقداری داشته باشد در حالی که در رگرسیون لجستیک مقادیری که در خروجی داریم احتمالا متعلق به یکی از دو کلاس ۰ یا ۱ است. در رگرسیون خطی هدف پیش بینی مقادیر پیوسته در خروجی است در حالی که هدف رگرسیون لجستیک پیشبینی مقادیر گسسته است و در آن قصد داریم داده ها را به دو یا چند کلاس مشخص طبقه بندی کنیم. بنابراین مقادیر باید بین ۰ و ۱ باشند که این موضوع با معادله خطی امکان پذیر نیست. تابعی که در رگرسیون لجستیک استفاده می شود، تابع سیگموید یا همان لجستیک می باشد. مرز تصمیم گیری نیز به ما کمک می کند تا دو کلاس را از یکدیگر متمایز کنیم. این مرز تصمیم گیری را تابع لجستیک مشخص می کند.

در اینجا با تغییر پارامترهای مدل در جهت بهبود دقت، چندبار از مدل استفاده شده است.

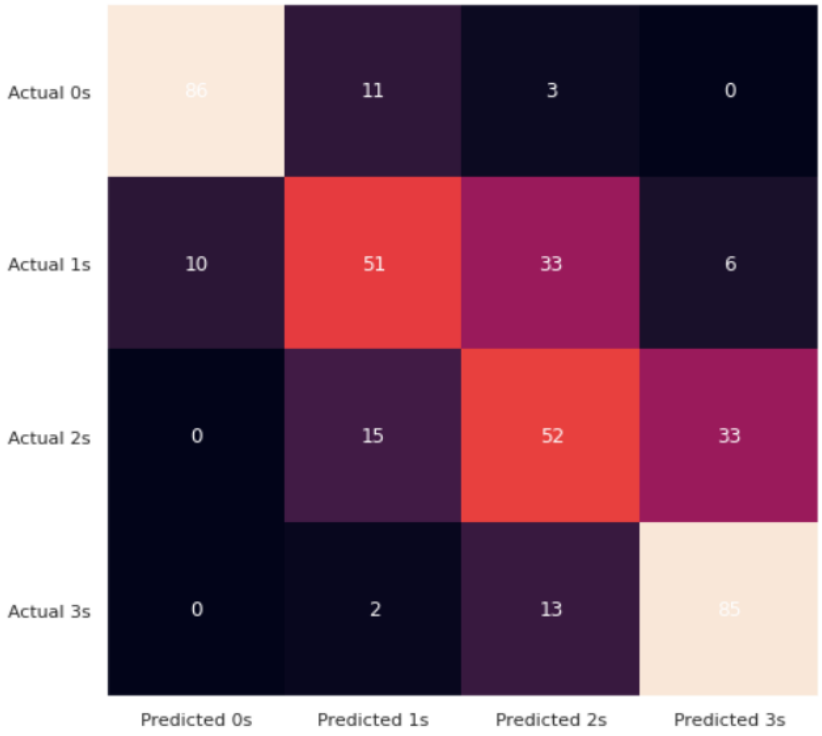
OvO (One-vs_One) به این صورت است که دو کلاس (باینری) برای پیش بینی وجود دارد که یک کلاس را مثبت و دیگری را منفی در نظر می گیریم. اما در OvA (One-vs-All) بیشتر از یک کلاس برای طبقه بندی وجود دارد و طبقه بندی اینگونه صورت می گیرد که کلاس مورد نظر برای دسته بندی، مثبت و سایر کلاس ها، منفی در نظر گرفته می شوند. مشکلی که این روش دارد این است که تعداد داده های یک کلاس (کلاس مثبت) در مقایسه با سایر کلاس ها (کلاس های منفی) بالانس نیست. برای مثال اگر سه کلاس دسته بندی هریک با ۱۰۰ نمونه داشته باشیم، برای دسته بندی کلاس ها طبق OvA، ۱۰۰ نمونه از یک کلاس و ۲۰۰ نمونه از سایر کلاس ها داریم و زمانی که تعداد داده های آموزشی کم باشند مدل نمی تواند به خوبی پیش بینی انجام دهد. اگر هم مدل را روی تعداد زیادی از نمونه های کلاس اول و تعداد کمی از نمونه های سایر کلاس ها آموزش دهیم، مدل برای سایر کلاس ها پیش بینی خوبی ندارد. بنابراین به نظر می رسد روش OvO رویکرد بهتری باشد که برای دسته بندی بیشتر از دو کلاس، مثلا سه کلاس، می توان اینگونه عمل کرد که پیش بینی را روی هر جفت کلاس انجام داد یعنی کلاس ۱ و کلاس ۲، کلاس ۲ و کلاس ۳ ، کلاس ۱ و کلاس ۳، در این حالت همواره تعداد داده های هر دو کلاس با یکدیگر برابرند.

اولین بار برای به کار گیری مدل، پارامتر multi_class را برابر OvR قرار می دهیم (حالت پیش فرض نیز OvR یا همان OvA می باشد). دقت مدل در این حالت برابر ۶۸,۵٪ می شود.

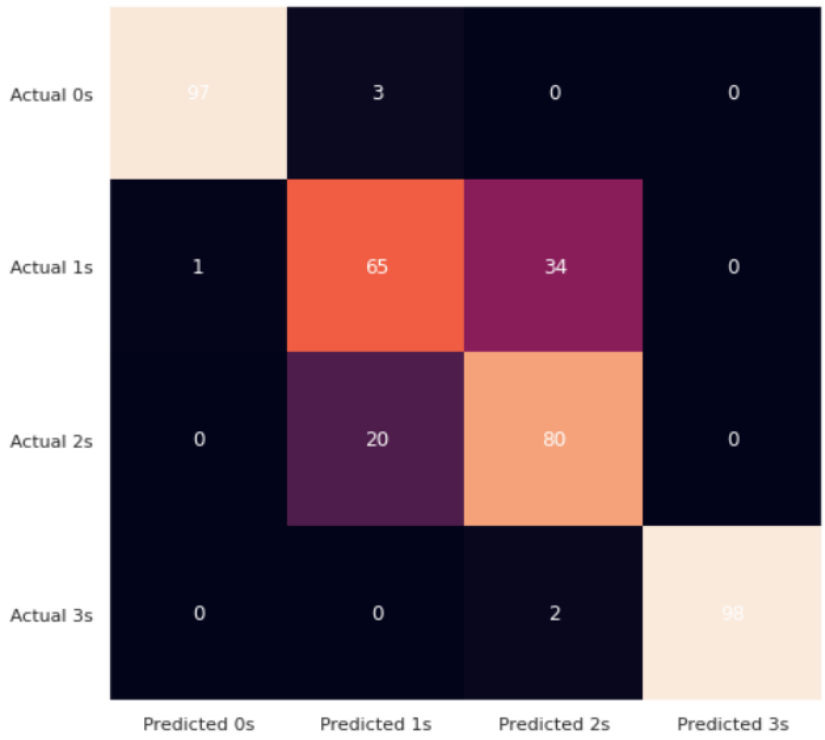
ماتریس در هم ریختگی نیز به شکل زیر است.



با تغییر پارامتر max_iter به ۵۰۰ (حالت پیش فرض آن ۱۰۰ تکرار می باشد) مجددا مدل را بر روی داده ها اجرا می کنیم. این بار به دقت ۶۸,۵٪ می رسیم. ماتریس در هم ریختگی مدل نیز در تصویر زیر قابل مشاهده است.



در این مرحله پارامتر solver که برای کاهش تابع هزینه می باشد را به مدل می افزاییم. مقدار پیش فرض این پارامتر liblinear می باشد که آن را با newton-cg جایگزین کرده ایم. این روش از ماتریس هسین استفاده می کند و بر روی مجموعه داده های بزرگ، عملکرد کندی دارد چرا که برای بدست آوردن ماتریس به مشتقات دوم نیازمندیم. با تغییر این پارامتر به دقت ۸۵٪ میرسیم. ماتریس در هم ریختگی نیز به صورت زیر می باشد.



۶. بررسی توازن داده ها

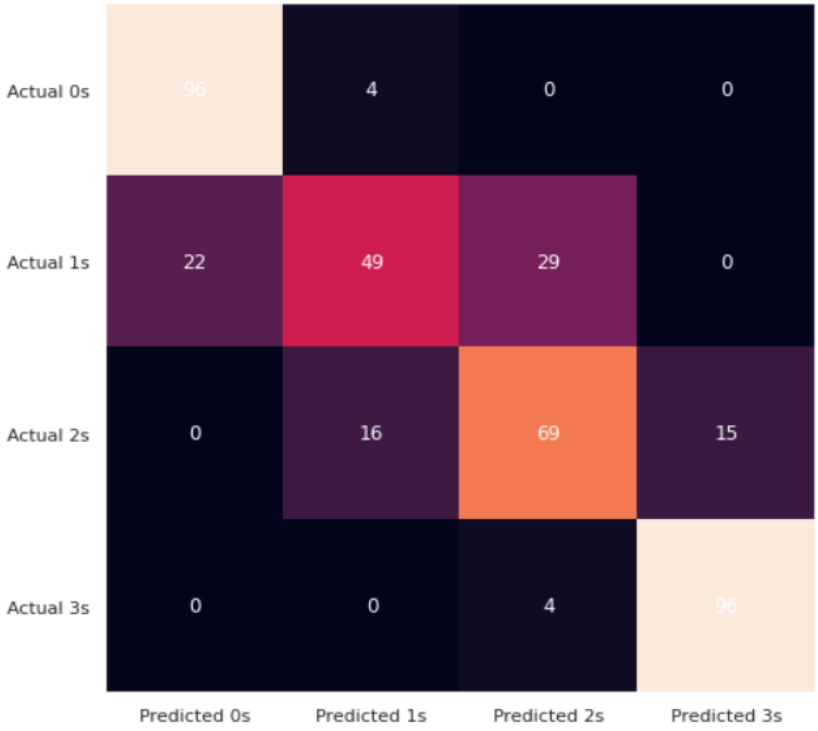
با چاپ کردن داده های هر یک از کلاس های قیمت، متوجه می شویم که پراکندگی داده ها در هر کلاس از توازن نسبی ای برخوردار است.

```
2    314
3    303
1    296
0    295
Name: price_range, dtype: int64
```

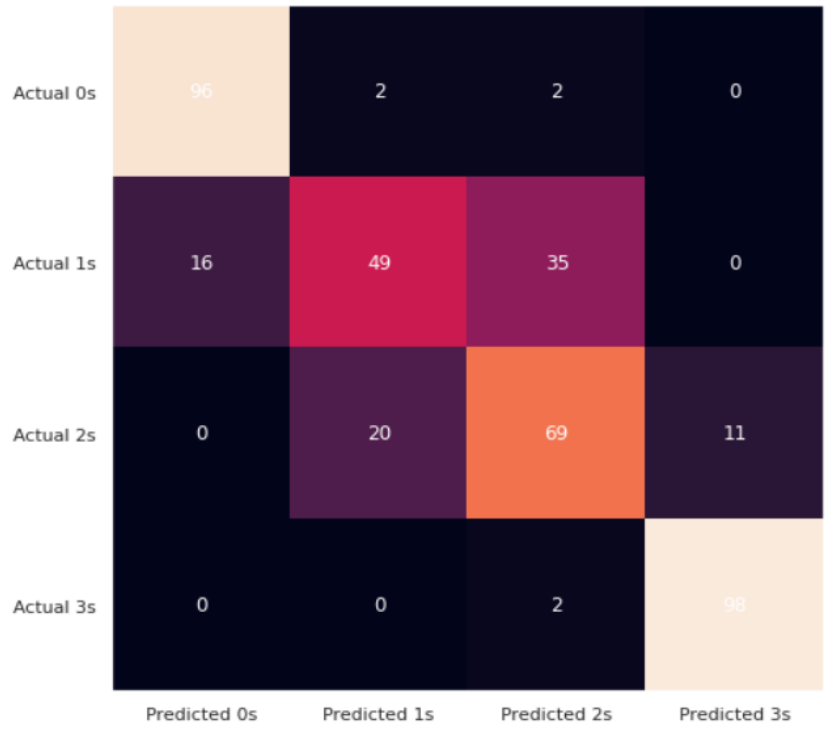
۷. مقیاس بندی داده ها

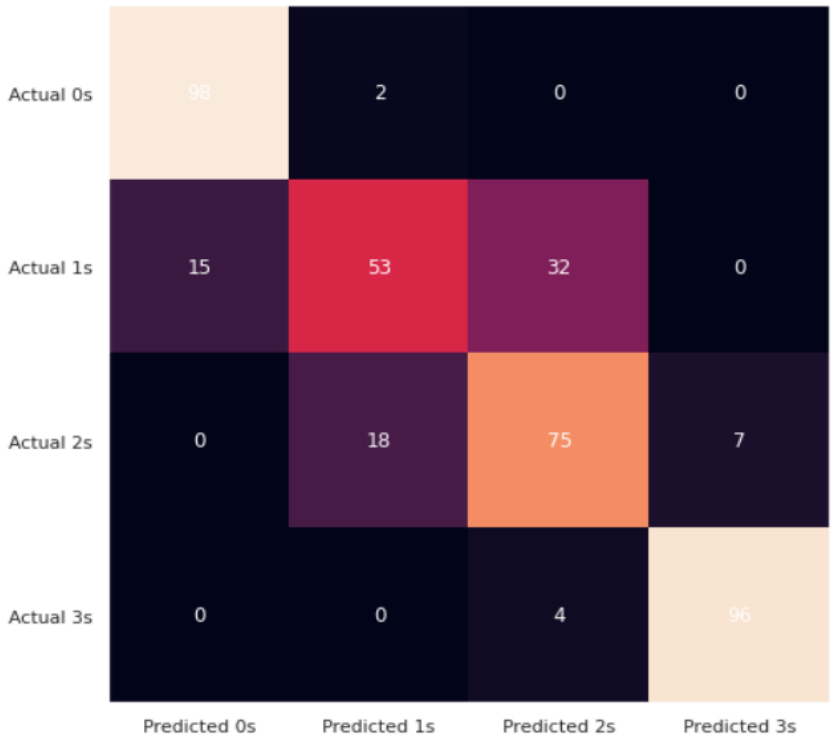
در این مرحله هدف نرمال سازی داده ها و سپس اجرای مدل روی آن و بررسی نتیجه عملکرد مدل می باشد. به این منظور از چندین scaler مختلف استفاده کرده ایم.

اولین scaler مورد استفاده، min-max scaler می باشد. در این مقیاس بندی داده ها به اعداد در بازه ۰ و ۱ تبدیل می شوند. پس از اعمالاین تغییر داده ها را به بهترین مدلی که از قسمت قبل بدست آوردیم می دهیم و در نتیجه دقت مدل به ۷۷,۵٪ می رسد. ماتریس در هم ریختگی مربوط نیز در زیر قابل مشاهده است.



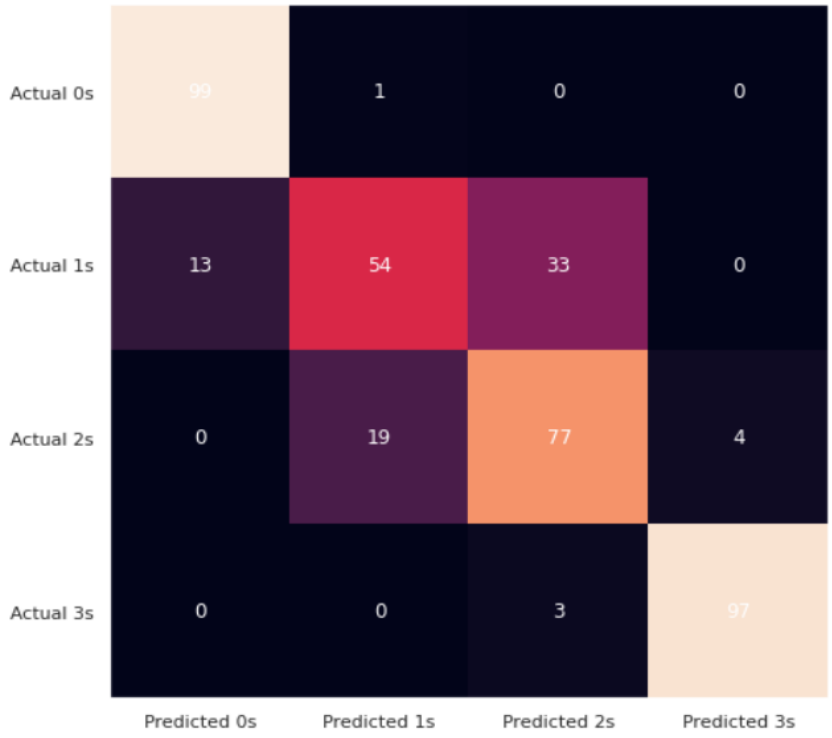
بار دیگر مقیاس بندی داده ها را با استفاده از quantile scaler انجام می دهیم. این روش باعث می شود تا ستون ها (ویژگی ها) از یک توزیع نرمال پیروی کنند و در نتیجه در هر ستون مقادیر پرتکرار و شایع تر تاثیر بیشتری خواهند داشت و از تاثیر داده های پرت کاسته می شود. پس از مقیاس بندی داده ها به این روش، مدل اعمال شده به دقت ۷۸٪ دست می یابد که با حالت قبل تفاوت محسوسی ندارد. ماتریس در هم ریختگی نیز به صورت زیر است.





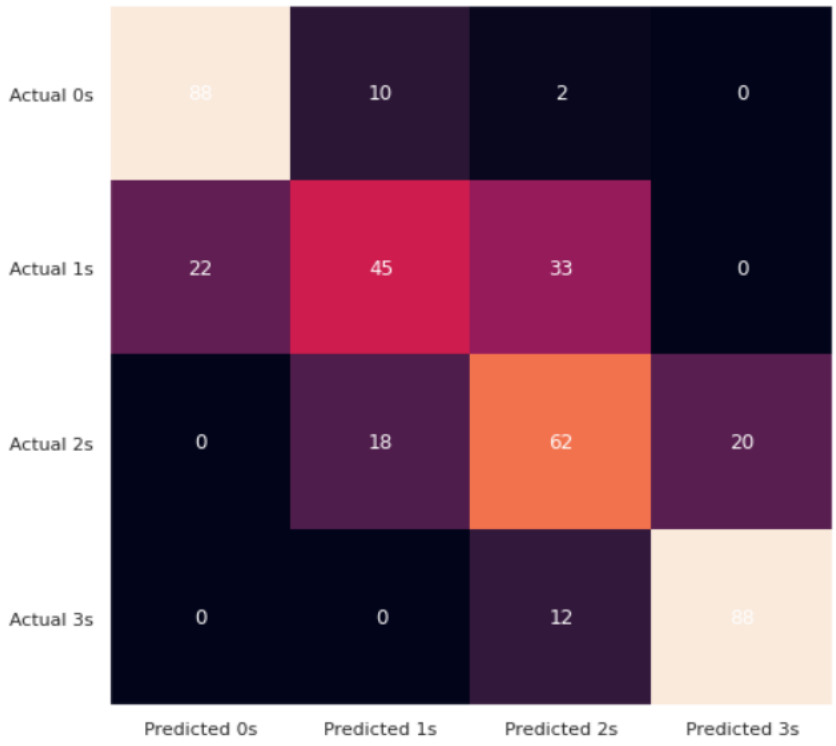
سومین scaler مورد استفاده، robust scaler است. در این روش، مقیاس بندی داده ها با استفاده از چارک های اول، دوم و سوم صورت می گیرد. این روش مقیاس بندی زمانی کارآمد تر است که داده های پرت داشته باشیم. پس از دادن داده های مقیاس بندی شده به مدل دقت آن به ۸۰,۵٪ می رسد.

آخرین scaler مورد استفاده، standard scaler می باشد. در این روش از میانگین و واریانس داده ها استفاده می شود و پس از نرمال سازی، میانگین داده ها برابر ۰ و واریانس آنها برابر ۱ می شود. پس از اجرای مدل روی داده هایی که به این روش مقیاس بندی شده اند، به دقت ۸۱,۷۵٪ می رسیم که از حالات قبل بیشتر است.

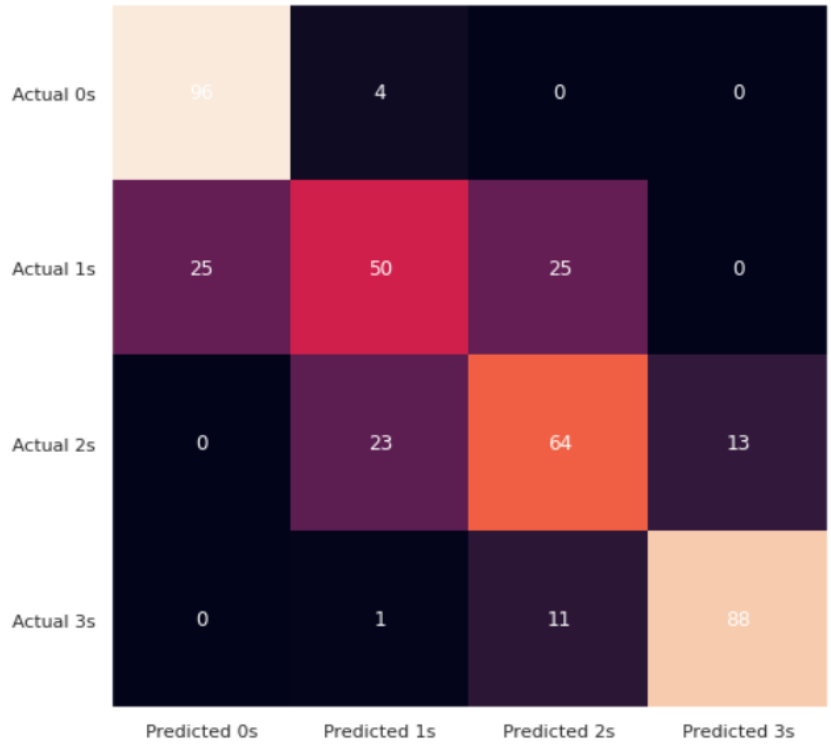


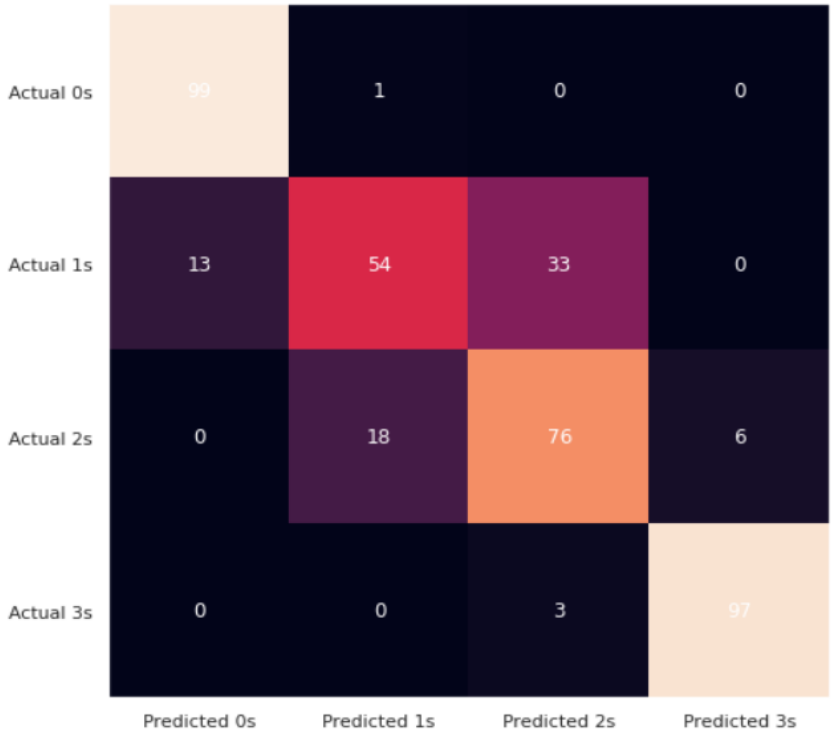
۸. کاهش ویژگی ها

به منظور کاهش ویژگی در این مرحله از الگوریتم PCA (آنالیز مولفه اصلی) با pov های داده شده که هر کدام باعث کاهش ویژگی ها به تعداد ویژگی معینی می شوند (هرچه pov بیشتر باشد، اطلاعات کمتری از دست می رود.)، استفاده می کنیم. سپس بهترین مدلی که تا الان به آن دست یافته ایم را بر روی آن پیاده سازی می کنیم. اولین مرتبه pov را برابر ۰,۷۵ قرار می دهیم. تعداد ویژگی ها به ۱۳ عدد کاهش می یابد و دقت مدل برابر ۷۰,۷۵٪ می شود. ماتریس در هم ریختگی نیز به صورت زیر است.



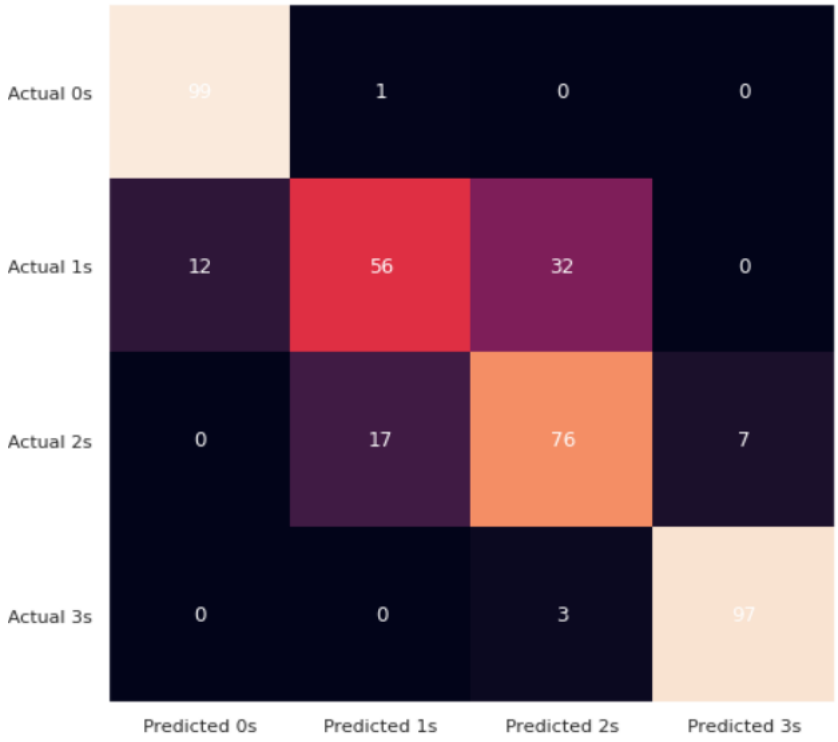
در مرتبه دوم pov را برابر ۰,۸ قرار می دهیم. تعداد ویژگی ها به ۱۴ عدد کاهش می یابد و دقت مدل برابر ٪۷۴,۵ می شود. ماتریس در هم ریختگی مربوط به آن نیز قابل مشاهده است.

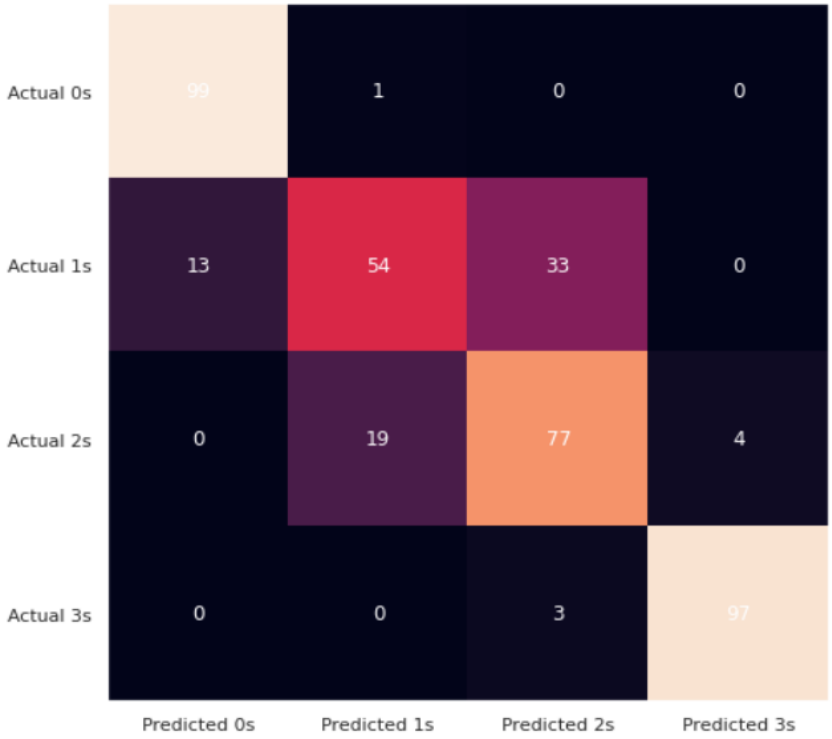




در این مرحله pov را برابر ۰,۹ قرار می دهیم. تعداد ویژگی ها به ۱۶ عدد کاهش می یابد و دقت مدل به ٪۸۱,۵ می رسد. ماتریس در هم ریختگی را نیز می توانید مشاهده کنید.

در مرتبه چهارم pov را برابر ۰,۹۵ قرار می دهیم. تعداد ویژگی ها برابر ۱۷ و دقت مدل برابر ۸۲٪ می شود.





در آخرین مرتبه pov را برابر ۰,۹۹ قرار می دهیم. تعداد ویژگی ها برابر ۱۹ و دقت مدل برابر ۸۱,۷۵٪ می شود.

دلیل استفاده از الگوریتم PCA کاهش ابعاد و ویژگی های مجموعه داده مورد بررسی می باشد چراکه این مسئله باعث کاهش محاسبات و سبک تر شدن کار مدل می شود. این الگوریتم از برداری که در آن واریانس داده ها بیشینه می شود استفاده می کند و ابعاد آن ها را اهش می دهد و به فضایی با ابعاد کمتر تبدیل می کند. لازم به ذکر است که اگر همبستگی بین ویژگی ها در مجموعه داده، کم باشد الگوریتم PCA عملکرد ضعیفی خواهد داشت همچنین اگر ابعاد داده را خیلی کاهش دهیم احتمال از دست دادن اطلاعات ارزشمند افزایش می یابد و در نتیجه دقت مدل با احتمال خوبی کاهش خواهد یافت.

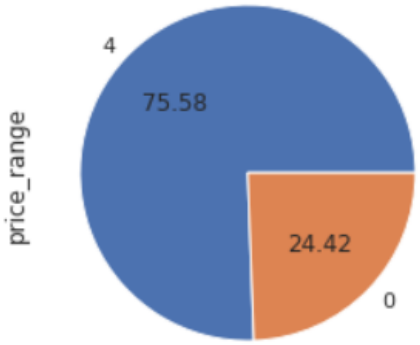
در اینجا زمانی که تعداد ویژگی ها را از ۱۳ به ۱۷ افزایش می دهیم، دقت مدل نیز افزایش می یابد. این مسئله که با افزایش ویژگی ها دقت افزایش می یابد ممکن است به دو علت باشد: ۱. بین داده ها همبستگی قوی وجود ندارد ۲. ویژگی های زیادی از مدل حذف شده اند و در نتیجه اطلاعات ارزشمندی را از دست داده ایم.

۹. نامتوازن سازی داده

اگر تعداد داده های کلاس های مورد بررسی داریم متوازن نباشند (تعداد داده های یک کلاس چندین برابر کلاس دیگر باشد) مدل به خوبی آموزش نمی بیند. چراکه با داده های یک کلاس کمتر سروکار داشته و در نتیجه در هنگام پیش بینی، احتمال کمتری وجود دارد که داده ناشناخته را به کلاسی با نمونه های خیلی کم در مقایسه با سایر کلاس ها نسبت بدهد. در نتیجه قبل از اعمال مدل بر روی داده ها لازم است که آنها را متوازن سازی کنیم.

در اینجا داده های کلاس قیمت ۱و۲و۳ را با کلاس قیمت جدید ۴ جیگزین میکنیم. در نتیجه دو کلاس قیمت داریم: ۰ و ۴. توزیع داده ها در این دو کلاس، متوازن نیست.

```
4    913
0    295
```



ستون داده های تست را نیز به کلاس قیمت ۰ و ۴ تغییر می دهیم.

برای متوازن کردن داده های آموزشی می توان روش های مختلف استفاده کرد. با توجه به آنچه خواسته شده، چند روش را به دلخواه توضیح می دهیم:

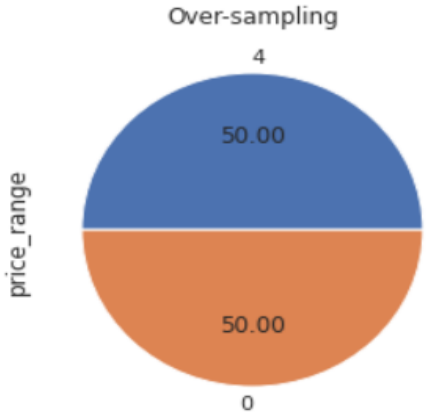
- اولین راه حل ممکن که به ذهن می رسد، جمع آوری داده های بیشتر می باشد.

- Random over-sampling: به صورت تصادفی از داده های کلاس با عضو کمتر کپی ایجاد می کنیم تا تعداد داده ها در دو کلاس متوازن شوند.

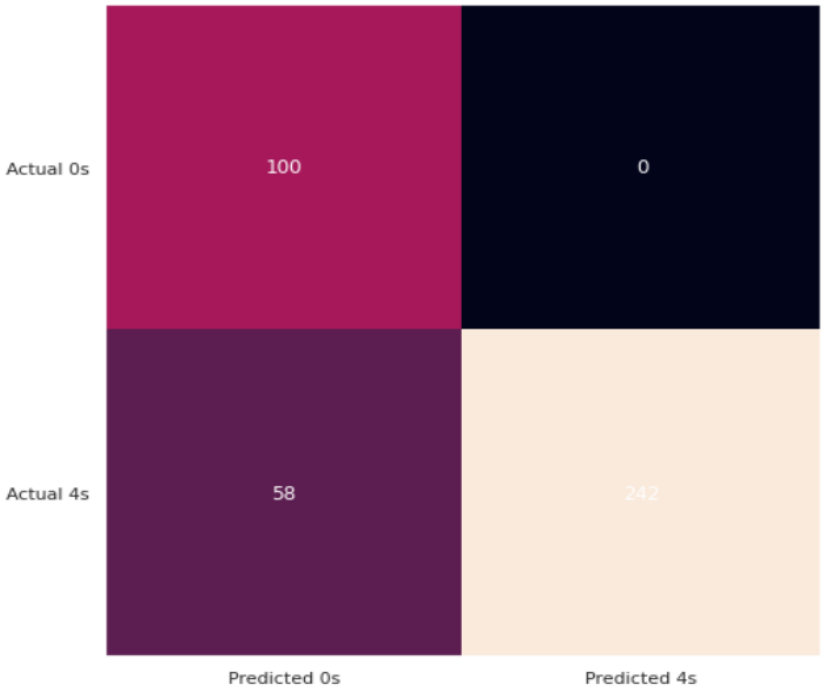
- Random under-sampling: به صورت تصادفی داده هایی را از کلاس با تعداد داده های کمتر حذف می کنیم تا تعداد داده های دو کلاس نسبت به یکدیگر متوازن شوند.

- راه حل دیگری که وجود دارد این است که به جای پی کردن داده های کلاس با تعداد عضو کمتر، با توجه به آنچه در دسترس است داده هی جدیدی که به کلاس کوچکتر تعلق داشته باشند بسازیم.

در اینجا از روش تصادفی oversampling استفاده می کنیم. و در نتیجه تعداد داده های آموزشی در هر یک از دو کلاس برابر می شوند.



سپس داده های آموزش متوازن شده را با استفاده از standard scaler مقیاس بندی می کنیم و مدل را روی آن آموزش می دهیم. حال داده های تست که متوازن نیستند را به مدل می دهیم. دقت مدل برابر ۸۵,۵٪ می شود که در مقایسه با دقت های بدست آمده در قسمت مقیاس بندی، عدد بزرگتری است. ماتریس در هم ریختگی مدل نیز به صورت زیر است.



۱۰. منابع

https://www.coursera.org/lecture/machine-learning/multiclass-classification-one-vs-all-68Pol

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

https://www.coursera.org/lecture/supervised-machine-learning-classification/upsampling-and-downsampling-w19MV

https://github.com/dataprofessor/imbalanced-data/blob/main/imbalanced_learn.ipynb