تمرین قیمت گوشی

ابتدا کلاس های شماره ۰ و ۱ را به عنوان کلاس شماره ۰ و همچنین کلاس های شماره ۲ و ۳ ار به عنوان کلاس شماره یک در نظر می گیریم.

با اجرای الگوریتم forward selection مشاهده می شود که با ۵ فیچر بهترین auc_score بدست می آید.

در زیر نام فیچر ها و auc_score حاصل از اجرای آن فیچر ها تا تعداد ۹ بر روی رگرسیون مشاهده می شود.

```
{1: (0.9727250000000001, ['ram']),
 2: (0.9929680411331787, ['ram', 'battery_power']),
 3: (0.9978195488721805, ['ram', 'battery_power', 'px_height']),
 4: (0.9998998773497535, ['ram', 'battery_power', 'px_height', 'px_width']),
 5: (1.0, ['ram', 'battery_power', 'px_height', 'px_width', 'sc_h']),
 6: (0.9998999974999374,
  ['ram', 'battery_power', 'px_height', 'px_width', 'sc_h', 'int_memory']),
 7: (0.9998748873986588,
  ['ram',
   'battery_power',
   'px_height',
   'px_width',
   'sc_h',
   'int_memory',
   'mobile_wt']),
 8: (0.9791916766706683,
  ['ram',
   'battery_power',
   'px_height',
   'px_width',
   'sc_h',
   'int_memory',
   'mobile_wt',
   'm_dep']),
 9: (0.9706720672067206,
  ['ram',
   'battery_power',
   'px_height',
   'px_width',
   'sc_h',
   'int_memory',
   'mobile_wt',
   'm_dep',
   'clock_speed']),
```

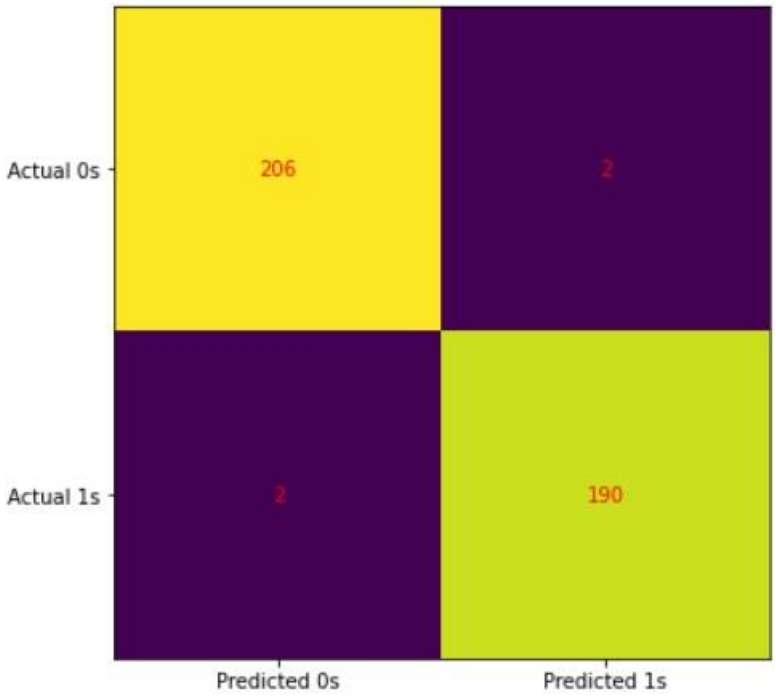
داده ها را به نسبت ۸ به ۲ برای ترین و تست تقسیم می کنیم و سپس با در نظر گرفتن ۵ فیچری که با کمک الگوریتم forward selection بدست آمده اند، مدل رگرسیون را اجرا می کنیم. این ۵ فیچر در زیر آمده اند:

```
features: ['ram', 'battery_power', 'px_height', 'px_width', 'sc_h']
```

نتیجه اجرای این مدل در زیر مشاهده می شود:

	precision	recall	f1-score
0	0.99	0.99	0.99
1	0.99	0.99	0.99

با کمک confusion matrix در می یابیم که مدل در هر کلاس فقط دو مورد اشتباه داشته است:



برای اجرای pca ابتدا داده ها را با کمک standard scaler نرمال سازی می کنیم. حاصل اجرای مدل رگرسیون بر روی داده های pca را در زیر می بینیم:

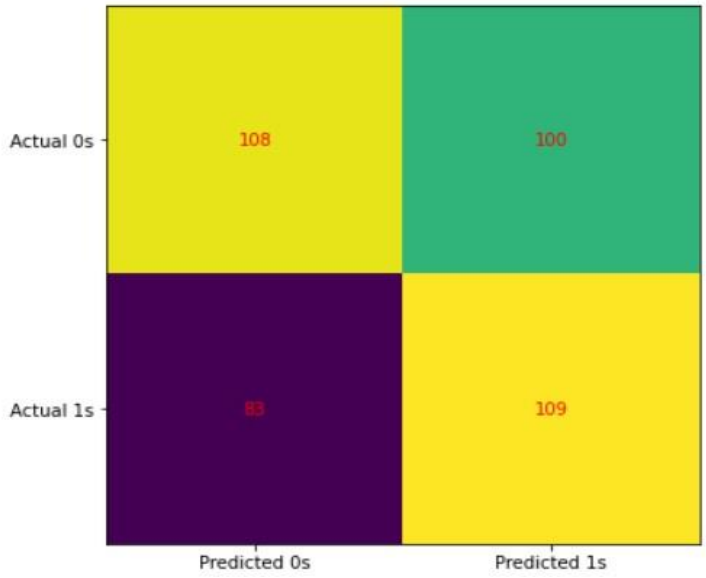
	precision	recall	f1-score
0	0.57	0.52	0.54
1	0.52	0.57	0.54

در کلاس شماره یک، ۵۷٪ از داده ای که واقعا متعلق به این کلاس بودند درست تشخیص داده شده اند و ۵۲٪ از داده هایی که برای این کلاس تشخیص داده شده اند، واقعا متعلق به این کلاس هستند.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad \text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$= \frac{True\ Positive}{Total\ Actual\ Positive} \qquad = \frac{True\ Positive}{Total\ Predicted\ Positive}$$

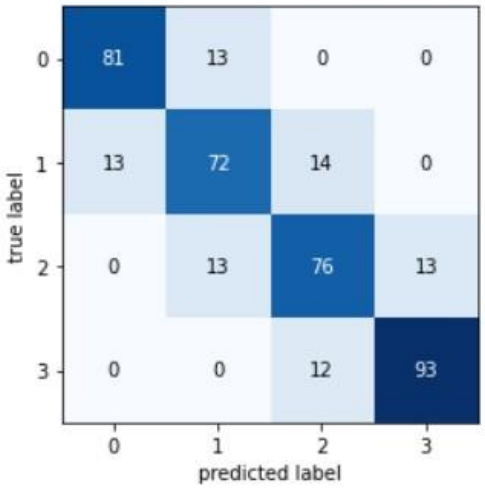
در شکل زیر نیز تعداد داده هایی که برای هر کلاس به درستی تشخیص داده شده اند را می بینیم:



مقادیر فیچر battery_power را به ۳ قسمت تقسیم می کنیم که تعداد داده های متعلق به هر گروه به صورت زیر است:

0	697
2	662
1	641

حاصل اجرای مدل svm بر روی دیتا با این حالت به score=80.5% می رسد و confusion matrix مربوط به این حالت نیز به شکل زیر است:



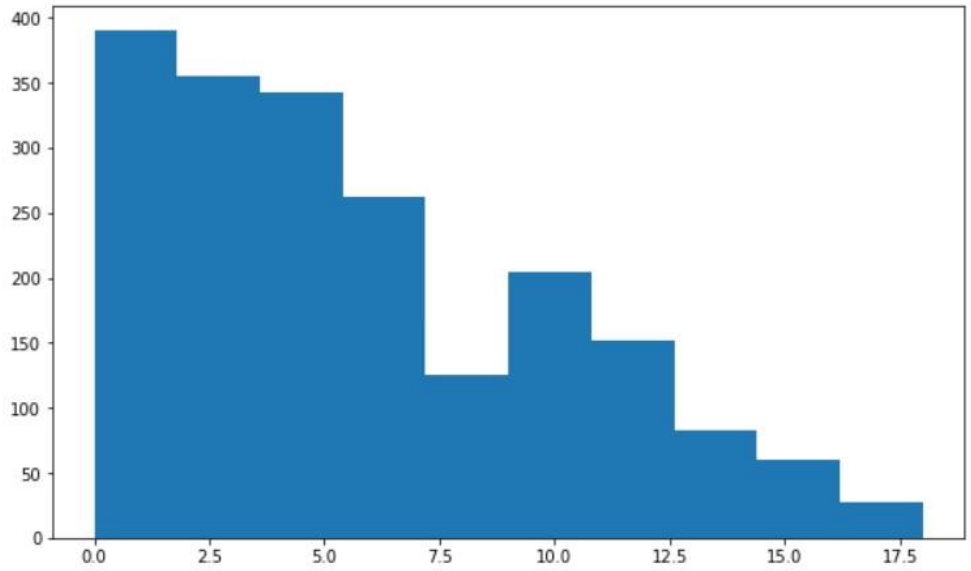
یک بار نیز این فیچر را به ۷ قسمت تقسیم می کنیم و تعداد داده های هر بازه به صورت مقابل است:

```
(499.503, 714.857]      311
(714.857, 928.714]      297
(1784.143, 1998.0]      287
(1570.286, 1784.143]    277
(928.714, 1142.571]     276
(1142.571, 1356.429]    276
(1356.429, 1570.286]    276
```

دو بازه بندی قبلی به گونه ای بود که اندازه بازه ها برابر باشد، حال یک بار بازه بندی را با بازه های نابرابر انجام می دهیم که در شکل زیر اندازه بازه ها و تعداد داده های هرکدام مشخص هستند.

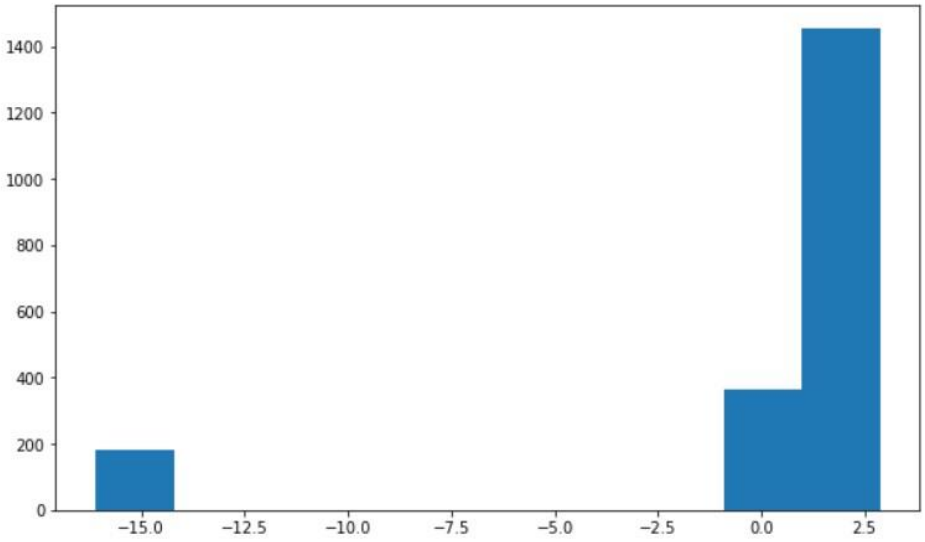
```
(1400, 1998]    794
(700, 1000]     411
(1000, 1250]    326
(501, 700]      284
(1250, 1400]    183
```

شکل زیر نحوه توزیع داده های فیچر sc_w را نشان می دهد:

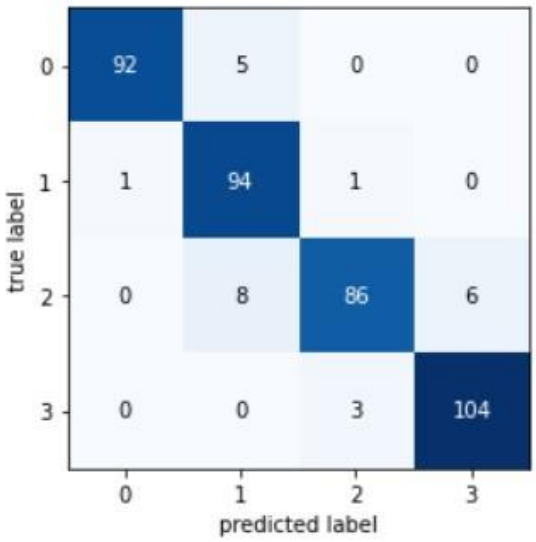


در کل از data transforming استفاده می کنیم چون اگر داده ها دچار چولگی باشند، احتمال بایاس شدن الگوریتم زیاد است و همچنین تبدیل داده ها در یک scale یکسان باعث می شود که الگوریتم بتواند ارتباط بین داده ها را بهتر تشخیص بدهد. یکی از این تبدیلات Log transform می باشد که از آن کمک می گیریم تا چولگی داده ها را از بین ببریم. برای این که این تبدیل خوب عمل کند و به توزیع نرمال برسیم، باید داده ها از توزیع log-normal پیروی کنند.

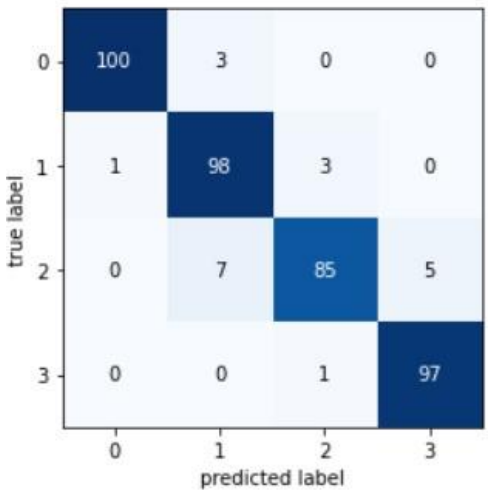
با توجه به نمودار زیر، log transform تاثیری در بهبود توزیع داده ها و نرمال کردن آن نداشته:



عملکرد مدل پس از اجرا بر روی لگاریتم این فیچر نسبت به حالتی که فیچر battery_power را بازه بندی کردیم، بهتر بوده و در زیر آورده شده:



یک فیچر جدید به نام volume نشکیل می دهیم که حجم گوشی را نشان می دهد. در این حالت عملکرد مدل مشابه حالت قبلی بوده و تفاوت زیادی در کل نداشته البته که در کلاس شماره ۰ بسیار عملکرد بهتری را ثبت کرده:

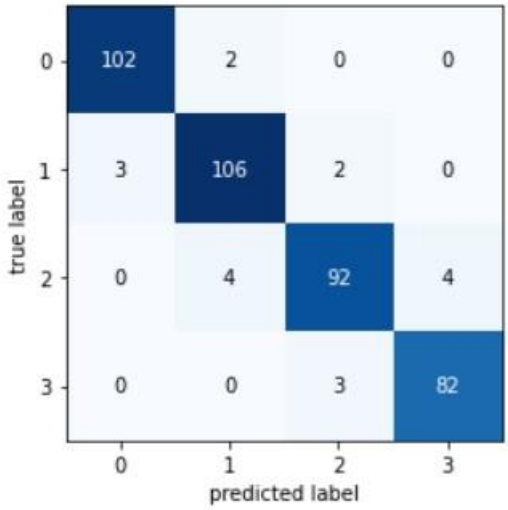


فیچر هایی که دارای حداکثر ۳ داده یونیک هستند را شناسایی می کنیم و به عنوان داده های کتگوریکال در نظر می گیریم. لیست لاین فیچر ها در زیر آمده است که تعداد آن ها ۷ تا می باشد:

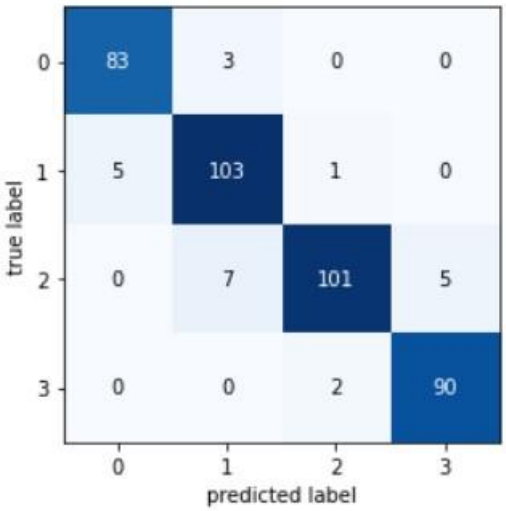
```
['blue',
 'dual_sim',
 'four_g',
 'three_g',
 'touch_screen',
 'wifi',
 'battery_bin']
```

مدل های یادگیری ماشین نمی توانند با داده های کتگوریکال کار کنند و به همین علت ما این داده ها را به داده های عددی تبدیل می کنیم. برای این کار می توان از روش های مختلفی استفاده کرد که هر کدام از این روش ها خوبی ها و مشکلات مخصوص به خود را دارند و هر کدام از آن ها تحت شرایطی مناسب هستند. یکی از این روش ها one hot encoding می باشد. این روش زمانی مناسب است که تعداد فیچر های کتگوریکال زیادی نداشته باشیم و همچنین تعداد داده های یونیک زیاد نباشد چون به ازای هر داده یونیک هر فیچر، یک ستون به دیتاست اضافه می شود و موجب بزرگ شدن دیتاست می شود.

با کمک one hot encoding این فیچر ها را از حالت کتگوریکال خارجج می کنیم. دقت مدل در این حالت نسبت به حالات قبلی بهتر می شود و به ۹۵/۵٪ می رسد. از شکل زیر می توان دریافت که عملکرد مدل در این حالت نسبت به حالت قبلی که فیچر volume را اضافه کردیم، بجز آخرین کلاس در بقیه کلاس ها بهبود داشته است.



در حالتی که همه تغییرات قبلی باهم اعمال شوند، مدل به دقت ۹۴٪ می رسد و confusion matrix آن به شکل زیر می باشد:



۸-

Bootstrapping را می توانیم به صورت زیر تعریف کنیم:

روشی شامل نمونه گیری تکراری همراه با جایگزینی، از داده های منبع به منظور تخمین پارامتر جمعیت. به منظور فهم بهتر تعریف، اجزای جمله را شرح می دهیم.

نمونه گیری: فرایند انخاب یک زیرمجموعه از مجوعه گستره ای از داده ها به منظور آسان سازی کار کردن با آنها که این زیر مجموعه رفتار جامعه اصلی را حفظ می کند.

نمونه گیری با جایگزینی: به این معنی است که یک داده می تواند مجددا در نمونه ای که ایجاد کرده ایم پدیدار شود.

این روش کمک می کند تا از overfitting جلوگیری کنیم.

دلیل استفاده از Bootstrapping این است که زمانی که داده کمی داریم حتی انتخاب تصادفی داده ها این تضمین را نمی دهد که نمونه ای چراکه نمونه ای که به این طریق به دست می آید الزاما جامعه اصلی را به خوبی توصیف نمی کند.

زمانی که داده های کمی داریم از Bootstrapping استفاده می کنیم چراکه با این روش می توان تعداد نمونه های زیادی ایجاد کرد.

زمانی که داده های زیادی داریم از Cross Validation استفاده می کنیم. این روش برای حجم کم داده ها کاربرد ضعیف تری دارد.

هر دو آنها چندین نمونه داده می سازند که با اجرای مدل بر رویی آنها می توانیم مدل را آموزش دهیم.

در Bootstrapping چندین نمونه از داده های اولیه ساخته می شود و سپس مدل روی هر یک از آنها آموزش داده می شود و روی تفاضل هر یک از جامعه اصلی ارزیابی می شود و در نهایت برای ارزیابی مدل از نتایج میانگین گیری می شود. در Cross Validation داده را به k قسمت تقسیم می کنیم و مدل را روی k-1 قسمت آموزش می دهیم و با استفاده از یک قسمت باقی مانده ارزیابی می کنیم که این کار k مرتبه صورت می گیرد و در نهایت از نتایج میانگین گیری می شود.

منبع:

https://www.analyticsvidhya.com/blog/۰۲/۲۰۲۰/what-is-bootstrap-sampling-in-statistics-and-machine-learning/

-۹

۲*۵ یعنی ۵ بار 2-Fold Cross Validation انجام می دهد.

-۱۰

ایده اصلی روش های خوشه بندی مبتنی بر تقسیم مانند k-means به دست آوردن تعداد خوشه ها به نحوی است که مجموع فواصل درون خوشه ای داده ها (یا مجموع مربعات فواصل درون خوشه ای) حداقل شود.

مجموع فواصل درون خوشه ای داده ها، میزان فشردگی خوشه بندی انجام شده را نشان می‌دهد و هدف، حداقل سازی آن تا جای ممکن است. روشelbow ، مجموع فواصل درون خوشه ای داده ها را به عنوان تابعی از تعداد خوشه ها در نظر می‌گیرد. این روش برای تعیین تعداد صحیح خوشه‌ها در یک دیتاست_استفاده می‌شود. به این ترتیب که تعداد خوشه ها به نحوی انتخاب می‌شوند که افزودن یک خوشه دیگر، بهبودی در حداقل‌سازی WSS ایجاد نکند (هدف، یافتن k ای است که برای هر خوشه واریانس را زیاد افزایش ندهد). طبق این روش k مناسب در نقطه ای بدست می آید که واریانس در حال افزایش و بایاس در حال کاهش است. با این حال همواره نمی توان از این روش بهره برد، خصوصا در جایی که داده ها خیلی خوشه ای نباشند.

سوالات امتیازی بخش ۱

۱. با کمک الگوریتم backward selection به این نتیجه می رسیم که با تعداد ۵ فیچر به بهترین نتیجه می رسیم. این فیچر ها و score را در زیر می بینیم:

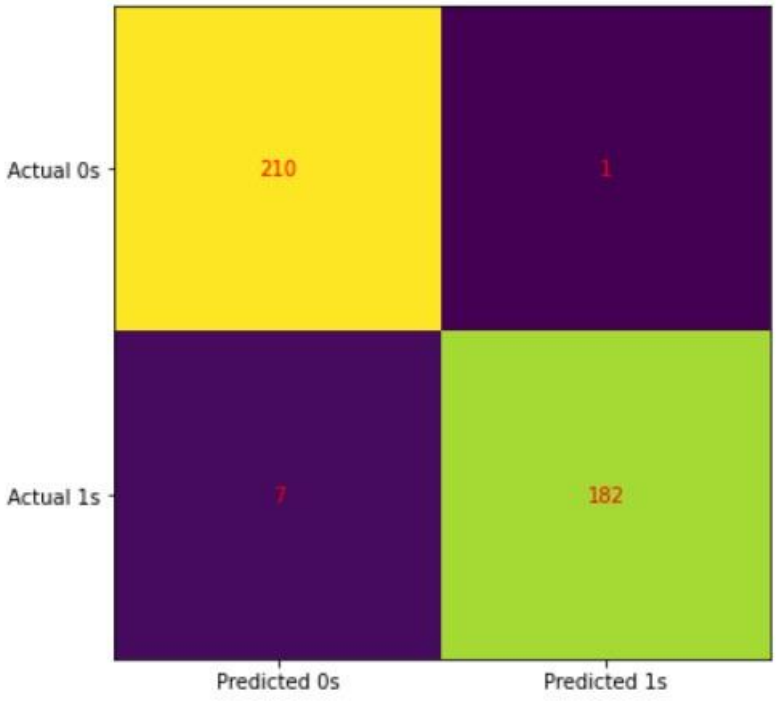
```
score: 0.9998491514770584
features: ['battery_power', 'px_height', 'px_width', 'ram', 'touch_screen']
```

تنها تفاوت این فیچر ها با فیچر هایی که از الگوریتم forward selection بدست آمده در این است که اینجا فیچر touch_screen بجای sc_w انتخاب شده است.

از شکل زیر و مقدار recall در کلاس شماره صفر می فهمیم که تمام مقادیری که متعلق به این کلاس هستند به صورت درست پیش بینی شده اند.

	precision	recall	f1-score
0	0.97	1.00	0.98
1	0.99	0.96	0.98

در زیر نیز confussion matrix آمده است که از بین ۲۱۱ داده ای که متعلق به کلاس شماره صفر هستند فقط یک داده اشتباه تشخیص داده شده که معادل تحلیلی است که از مقدار recall کلاس صفر داشته ایم.



۲. در رابطه با هر فرضی بین دو متغیر دو سوال مطرح است:

یک) احتمال وجود رابطه چقدر است.

دو) اگر رابطه ای وجود دارد، چقدر قوی است.

دو نوع ابزار برای پاسخگویی به این سوالات استفاده می شود: اولی با تست statistical significance بررسی می شود. و دومی توسط Measures of Association مورد توجه قرار گرفته است. تست statistical significance برای پاسخ دادن به این سوال استفاده می شود: احتمال اینکه چیزی که ما از آن به عنوان رابطه بین دو متغیر یاد می کنیم تصادفی باشد، چقدر است.

تست های statistical significance می گویند احتمال اینکه رابطه ای که فکر می کنیم پیدا کرده ایم تصادفی باشد چقدر است. آنها به ما می گویند که اگر فرض کنیم رابطه ای وجود دارد، احتمال اشتباه ما چقدر است.

statistical significance به این معناست که شانس خوبی وجود دارد که در یافتن رابطه بین دو متغیر به درستی عمل کنیم اما معنای آن با practical significance متفاوت است. می توانیم یافته های عملی

معناداری داشته باشیم اما پیامد آن یافته ممکن است کاربرد عملی نداشته باشد. محقق باید همواره هم statistical significance و هم practical significance را در نظر داشته باشد.

گام های statistical significance test:

۱. بیان فرض
۲. بیان فرض صفر
۳. انتخاب احتمال سطح خطا (آلفا)
۴. انتخاب و محاسبه تست برای اهمیت آماری
۵. تفسیر نتایج

منبع:

https://home.csulb.edu/~msaintg/ppa۶۹۶/۶۹۶stsig.htm

۳. MCC یک روش برای کلاس بندی باینری است که خروجی آن یکی از اعداد ۰، ۱، ۱- می باشد. ۱ زمانی خروجی داده می شود که پیش بینی به واقعیت نزدیک باشد. ۱- زمانی خروجی داده می شود که پیش بینی خلاف واقعیت باشد و ۰ نیز زمانی خروجی داده می شود که پیش بینی دقیقی نداشته باشیم.

منبع:

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.matthews_corrcoef.html

امیرحسین باباجانی ۹۷۲۲۲۰۰۹