نام و نام خانوادگی: درسا الهوئی

شماره دانشجویی: ۹۷۲۲۲۰۰۵

شماره تمرین: ۲,۱ **تاریخ تحویل: ۲۵ اردیبهشت ۱۴۰۱**

۱. مقدمه

هدف پیاده سازی الگوریتم های forward_selection و backward_selection بدون استفاده از پکیج و پیاده سازی تسک های خواسته شده می باشد.

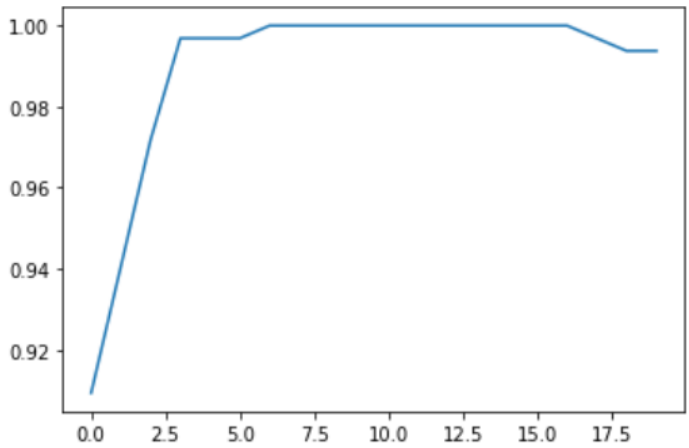
۲. تسک های خواسته شده

در گام اول ستون price_range را به دو حالت قیمت بالا (۱) و قیمت پایین (۰) تبدیل کرده و سپس آنها را به دو قسمت آموزشی و آزمایشی تقسیم می کنیم.

سپس از الگوریتم forward_selection استفاده می کنیم که فیچرهای زیر را به ما بر می گرداند.

{'features': ['ram', 'battery_power', 'px_height', 'px_width', 'blue', 'clock_speed', 'mobile_wt', 'fc', 'four_g', 'dual_sim', 'm_dep', 'n_cores', 'pc', 'int_memory', 'sc_h', 'sc_w', 'talk_time', 'three_g', 'touch_screen', 'wifi'], 'scores': [0.909375, 0.940625, 0.971875, 0.996875, 0.996875, 0.996875, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.996875, 0.99375, 0.99375], 'features_rank': range(0, 20), 'best_features': ['ram', 'battery_power', 'px_height', 'px_width', 'blue', 'clock_speed']}

طبق نمودار زیر score از ۲,۵ به بعد افزایش نمی آیند.



و طبق الگوریتم با داشتن فیچر های زیر الگوریتم خروجی بهتری خواهد داشت.

```
['ram', 'battery_power', 'px_height', 'px_width', 'blue', 'clock_speed']
```

در ادامه روی فیچر های بدست آمده LogisticRegression اجرا می کنیم. خلاصه مدل به صورت زیر است.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	190
1	1.00	1.00	1.00	210
accuracy			1.00	400
macro avg	1.00	1.00	1.00	400
weighted avg	1.00	1.00	1.00	400

Precision: مقدار پیشبینی های درست به کل پیشبینی ها.

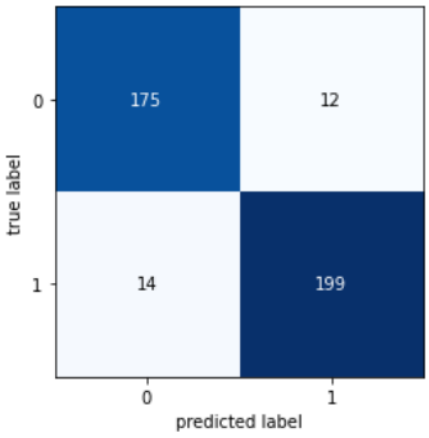
Recall: مقدار پیشبینی درست مدل به کل داده های درست.

F1-score: میانگین هندسی دو مورد بالا.

در ادامه روی داده ها PCA میزنیم و تعداد فیچرها را برابر ۶ قرار می دهیم (تعداد فیچرهایی که در بالا گفته شد) و با اجرای LogisticRegression نتایج زیر بدست می آید.

	precision	recall	f1-score	support
0	0.67	0.70	0.69	187
1	0.73	0.70	0.71	213
accuracy			0.70	400
macro avg	0.70	0.70	0.70	400
weighted avg	0.70	0.70	0.70	400

سپس ستون battery_power را bin بندی می کنیم. در اینجا چهار bin با نام های ۰، ۱، ۲، ۳ ایجاد می کنیم. حال مدل را به بخش آموزشی و آزمایشی تقسیم می کنیم. سپس با استفاده از فیچر جدید افزوده شد (battery_bin) متد svm را پیاده سازی می کنیم. ماتریس در همریختگی بدست آمده به صورت زیر است.



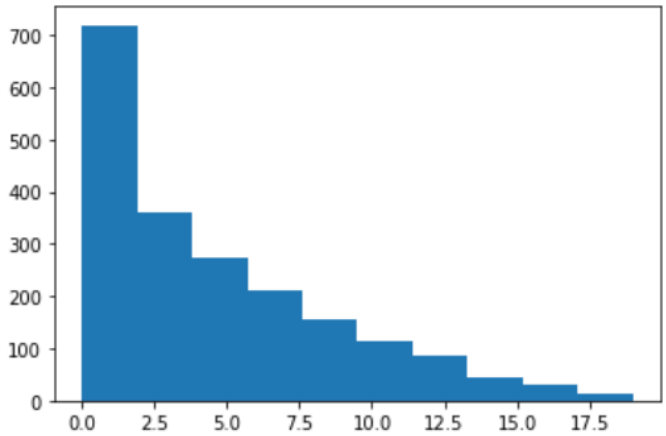
bin بندی دیگری انجام میدهیم و ۱۰ بازه ایجاد می کنیم که بازه ها به صورت زیر هستند.

```
(499.503, 650.7]    180
(800.4, 950.1]      175
(650.7, 800.4]      167
(1848.3, 1998.0]    165
(950.1, 1099.8]     158
(1548.9, 1698.6]    157
(1698.6, 1848.3]    153
(1399.2, 1548.9]    151
(1249.5, 1399.2]    148
(1099.8, 1249.5]    146
Name: battery_power, dtype: int64
```

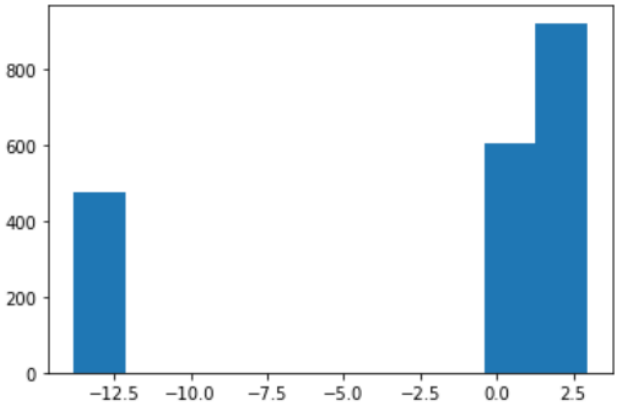
bin بندی بعدی را با تعیین نقطه شروع و پایان بازه ها به صورت دستی ایجاد می کنیم. بازه ها در تصویر زیر قابل مشاهده اند.

```
(1600, 1998]    410
(700, 1000]     345
(1300, 1600]    311
(1000, 1300]    302
(501, 700]      230
Name: battery_power, dtype: int64
```

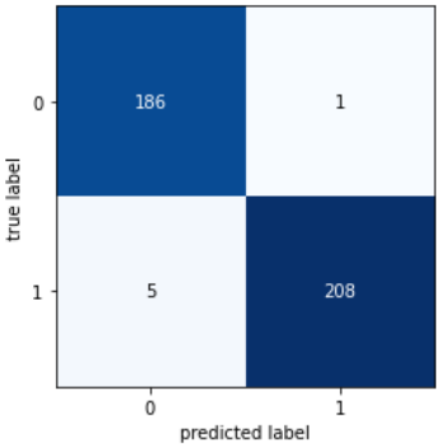
حال پراکندگی داده ها در ستون fc را رسم می کنیم.



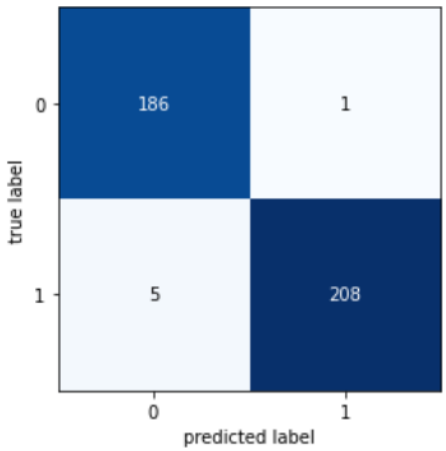
با اعمال log_transform پراکندگی داد ها به صورت زیر می شود. از انجایی که لگاریتم ۰ تعریف نشده است مقدار ۰,۰۰۰۰۰۱ را به آن اضافه می کنیم. استفاده از log_transform برای نرمال سازی داده ها است که البته در اینجا جواب نمی دهد.



سپس متد svm را پیاده سازی می کنیم و ماتریس درهمریختگی را رسم می کنیم.



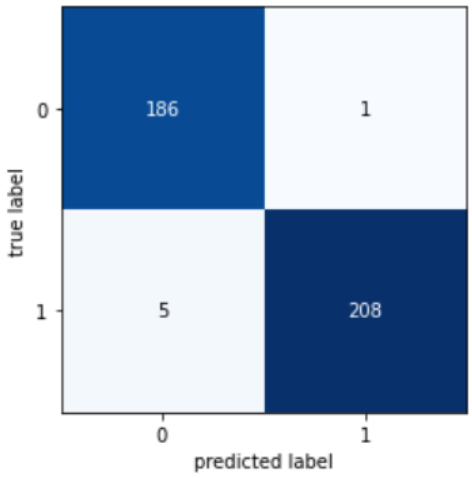
در گام بعدی ستون area که بیانگر مساحت گوشی همراه است را به داده ها اضافه می کنیم. متد svm را روی آن پیاده سازی کرده و ماتریس درهمریختگی را رسم می کنیم.



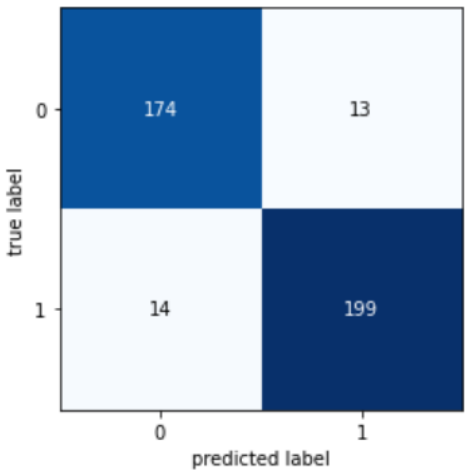
در گام بعد ستون هایی که به صورت باینری اند (کتگوریکال) را به صورت one_hot در می آوریم. ستون های کتگوریکال در تصویر زیر آورده شده اند.

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

با این کار تعداد ستون های دیتا به ۳۰ افزایش می یابد. حال بعد از پیاده سازی svm ماتریس درهمریختگی را رسم می کنیم.



در نهایت روی تمام داده ها که هر مرحله به آن فیچر جدیدی افزودیم، متد svm را پیاده سازی می کنیم. ماتریس درهمریختگی به صورت زیر است.



۳. پاسخ به سوالات توضیحی

۸. Bootstrapping چیست و چه تفاوتی با Cross Validation دارد؟ چه زمانی از Bootstrapping استفاده می شود؟

Bootstrapping یک تکنیک نمونه گیری شامل گرفتن داده ها از منبع با استفاده از جایگزینی می باشد. منظور از جایگزینی این است که داده ای یکسان ممکن است چندین بار در مجموعه داده نمونه ما شامل شود به زبان ساده تر یعنی انتخاب مجدد یک داده که قبل تر انتخاب شده است، امکان پذیر می باشد. Bootstrapping بر اساس قانون اعداد بزرگ است که می گوید با تعداد داده کافی، توزیع تجربی، تقریب خوبی از توزیع واقعی خواهد بود. این روش نمونه گیری کمک می کند تا از overfitting جلوگیری شود و همچنین به ثبات الگوریتم های یادگیری ماشین نیز کمک می کند.

در Bootstrapping مدل را روی نمونه بدست آمده آموزش می دهیم و روی مجموعه اولیه منهای مجموعه بدست آمده تست میکنیم. چرا که برخی از داده های موجود در مجموعه اولیه در نمونه بدست آمده وجود ندارند. این کار را میتوان روی تمام نمونه های بدست آمده از مجموعه اولیه انجام داد. با این کار به تعداد نمونه ها، خطا تخمین زده می شود و میانگین خطاها تخمین خطای Bootstrap را به ما می دهد. میتوان نشان داد که تخمین Bootstrap واریانس کمتری از تخمین خطا روی مجموعه داده اولیه که با تقسیم تصادفی آن به دو قسمت داده های آموزشی و آزمایشی صورت می گیرد، دارد.

Bootstrapping و Cross Validation

- Bootstrapping نمونه گیری با بهره گیری از جایگزینی می باشد (معمولا داده های جانشین جدید با تعداد داده های مجموعه داده اولیه برابر است). با توجه به روش جایگزینی، نمونه داده بدست آمده با این روش ممکن است حاوی چندین نمونه یکسان از یک داده در مجموعه اولیه باشد و همچنین نمونه ای از یک داده در مجوعه اولیه در نمونه بدست آمده نباشد و حذف شده باشد.
- در Cross Validation نمونه گیری بدون جایگزینی صورت می گیرد و در نتیجه اندازه نمونه بدست آمده از مجموعه داده اولیه کوچکتر است. استفاده از این مدل زمانی توصیه می شود که حجم داده ها زیاد باشد.
- در Cross Validation داده ها را به k قسمت تقسیم می کنیم و با k-1 بخش مدل را آموزش می دهیم و با آن ۱ بخش باقی مانده مدل را تست میکنیم. اما برای Bootstrapping گفتیم مدل را روی نمونه های بدست آمده آموزش می دهیم و روی تفاضل نمونه ها با مجموعه اصلی تست می کنیم.

۹. در این روش ۵ بار Two Fold Cross Validation را اجرا می شود. استفاده از Two Fold موجب می شود تا اطمینان داشته باشیم هر مشاهده در مجموعه داده آموزشی یا آزمایشی تنها یکبار برای آزمودن مدل استفاده می شوند.

۱۰. طبق شکل بایاس تا قبل ۳ بدون هیچ الگویی و بعد ۳ با الگویی خطی کاهش می یابد. روش elbow روشی برای تشخیص k در الگوریتم kmeans می باشد. ما به منظور پیدا کردن k بهینه روی مدل، k های مختلف را روی آن تست می کنیم. معیار WCSS مجموع مربعات فاصله نقاط یک کلاستر در الگوریتم kmeans را با مرکز آن داده ها می یابد. نمودار WCSS بر حسب k مشابه نمودار بایاس می باشد. طبق روش elbow اولین نقطه ای که از آن k به صورت خطی شروع به کم شدن می کند، بهترین k است. بنابراین طبق این روش در نمودار داده شده می توان کمترین خطا را یافت. خطا روی داده تست که مدل تا کنون آن را ندیده است برابر است بامجموع واریانس مدل، بایاس مدل و واریانس اپسیلون (که نمی توان آن را کاهش داد). اگر مدل بایاس زیاد و واریانس کم داشته باشد، آنگاه مدل underfit است و اگر بایاس کم و واریانس زیاد باشد، مدل overfit شده است. به علت اینکه خطا به بایاس و واریانس بسته است، باید نقطه بهینه ای بیابیم که در آن جمع بایاس و واریانس کمترین مقدار باشد که این نقطه معمولا جایی بدست می آید که بایاس در حال کم شدن و واریانس در حال زیاد شدن است (سرعت کم شدن بایاس از سرعت زیاد شدن واریانس کمتر است). به همین علت می توان از روش elbow استفاده کرد اما نه همیشه چرا که بایاس همواره به صورت خطی کم نمی شود.

سوالات امتیازی

۱. با پیاده سازی backward_selection فیچر های زیر بدست می آید.


```
{'features': ['int_memory', 'clock_speed', 'fc', 'four_g', 'm_dep', 'talk_time', 'three_g', 'blue', 'dual_sim', 'n_cores', 'pc', 'sc_h', 'sc_
w', 'wifi', 'touch_screen', 'mobile_wt', 'px_width', 'px_height', 'battery_power', 'ram'], 'scores': [1.0, 0.996875, 0.996875, 0.996875, 0.99
6875, 0.996875, 0.996875, 0.99375, 0.99375, 0.99375, 0.99375, 0.99375, 0.99375, 0.990625, 0.984375, 0.98125, 0.975, 0.91875, 0.55937
5], 'features_rank': range(0, 20), 'best_features': []}
```


سپس مدل LogisticRegression را روی فیچر هایی که با استفاده از backward_selection بدست آمد، اعمال می کنیم. نتایج بدست آمده به صورت زیر است.

	precision	recall	f1-score	support
0	0.98	0.97	0.98	190
1	0.97	0.99	0.98	210
accuracy			0.98	400
macro avg	0.98	0.98	0.98	400
weighted avg	0.98	0.98	0.98	400

۲. یکی از مباحث مهم در یادگیری ماشین، انتخاب بهترین مدل است. به طور کلی یک آزمون فرض آماری برای مقایسه نمونه ها، احتمال مشاهده دو نمونه داده را با توجه به این که نمونه ها توزیع یکسانی دارند، نشان می دهد. فرض آزمون آماری را فرض صفر گویند که با محاسبه به معیارهای آماری و تفسیر آنها، فرض صفر را قبول و یا رد می کنیم. در مسئله انتخاب مدل بر اساس توانایی تخمین آن، علاقه مندیم بدانیم آیا تفاوتی معنادار بین دو مدل وجود دارد یا خیر. اگر نتیجه تست نشان دهد که شواهد کافی برای رد فرضیه صفر وجود ندارد، آنگاه هر تفاوت مشاهده شده در توانایی مدل به دلیل شانس آماری است و اگر نتیجه تست نشان دهد شواهد کافی برای رد فرض صفر وجود دارد، آنگاه هر تفاوت مشاهده شده در توانای مدل به دلیل تفاوت در مدل ها می باشد. لازم به ذکر است که نتیجه تست احتمالی است.

مقایسه مدل های یادگیری ماشین از طریق statistical significance tests انتظاراتی را تحمیل می کند که بر انواع آزمون های آماری قابل استفاده تاثیر می گذارد. به عنوان مثال:

- تخمین مهارت. معیار خاصی از توانایی و مهارت مدل باید انتخاب شود. این معیار می تواند دقت طبقه بندی (یک نسبت) یا MAE باشد که نوع تست های قابل استفاده را محدود می کند.

- تخمین های مکرر. نمونه ای از امتیازات توانایی برای محاسبه آماری مورد نیاز است. آموزش و تست مکرر یک مدل معین بر روی داده های یکسان یا متفاوت،بر نوع آزمونی که می توان استفاده کرد تاثیر خواهد داشت.

- توزیع تخمین ها. مشخص می کند آیا می توان از آزمون های پارامتریک یا ناپارامتریک استفاده کرد.

- گرایش مرکزی. توانایی مدل اغلب با استفاده از یک خلاصه آماری مانند میانگین یا میانه، بسته به توزیع امتیاز توانایی، مقایسه و توصیف می شود.

نتایج یک آزمون آماری معمولا test static و p-value اند. که هردو را می توان تفسیر کرد و در ارائه نتایج به منظور کمی کردن سطح اطمینان یا معنی دار بودن تفاوت بین مدل ها استفاده می شوند. این اجازه می دهد تا ادعاهای قوی تری به عنوان بخشی از انتخاب مدل نسبت به عدم استفاده از تست های فرض آماری مطرح شوند.

معنی دار بودن آماری به عدم احتمال وقوع میانگین تفاوت های مشاهده شده در نمونه به دلیل خطای نمونه گیری اشاره دارد. با یک نمونه به اندازه کافی بزرگ، علیرغم تفاوت های به ظاهر ناچیز جمعیت، همچنان ممکن است statistical significance یافت شود. از سوی دیگر practical significance به این می پردازد که آیا تفاوت به اندازه ای بزرگ است که از نظر عملی ارزش داشته باشد. در حالی که statistical significance به طور دقیق تعریف شده است، practical significance شهودی و ذهنی تر است.

تست های statistical significance برای مقایسه عملکرد مدل های ML با این فرض طراحی شده اند که از توزیع یکسان گرفته شده اند. اگر فرض صفر رد شود، نشان می دهد که تفاوت در امتیاز توانایی از نظر آماری معنا دار است.

۳. MCC یک ابزار آماری است که برای ارزیابی مدل استفاده می شود. وظیفه آن اندازه گیری تفاوت بین مقادیر پیش بینی شده و مقادیر واقعی است و برای کلاس بندی های باینری کاربرد دارد. این معیار توسط chi squared بدست می آید و خروجی آن سه حالت دارد. اگر خروجی عدد ۱ باشد، پیش بینی درست و اگر ۱- باشد، پیش بینی غلط بوده است. خروجی ۰ نیز به معنی تصادفی بودن پیش بینی است.

۴. منابع

https://carpentries-incubator.github.io/machine-learning-novice-python/۰۸-bootstrapping/index.html#:~:text=In٪۲۰statistics٪۲۰and٪۲۰machine٪۲۰learning,our٪۲۰resampled٪۲۰dataset٪۲۰multiple٪۲۰times

https://machinelearningmastery.com/statistical-significance-tests-for-comparing-machine-learning-algorithms/

https://www.kdnuggets.com/2019/01/comparing-machine-learning-models-statistical-vs-practical-significance.html

https://www.voxco.com/blog/matthewss-correlation-coefficient-definition-formula-and-advantages/