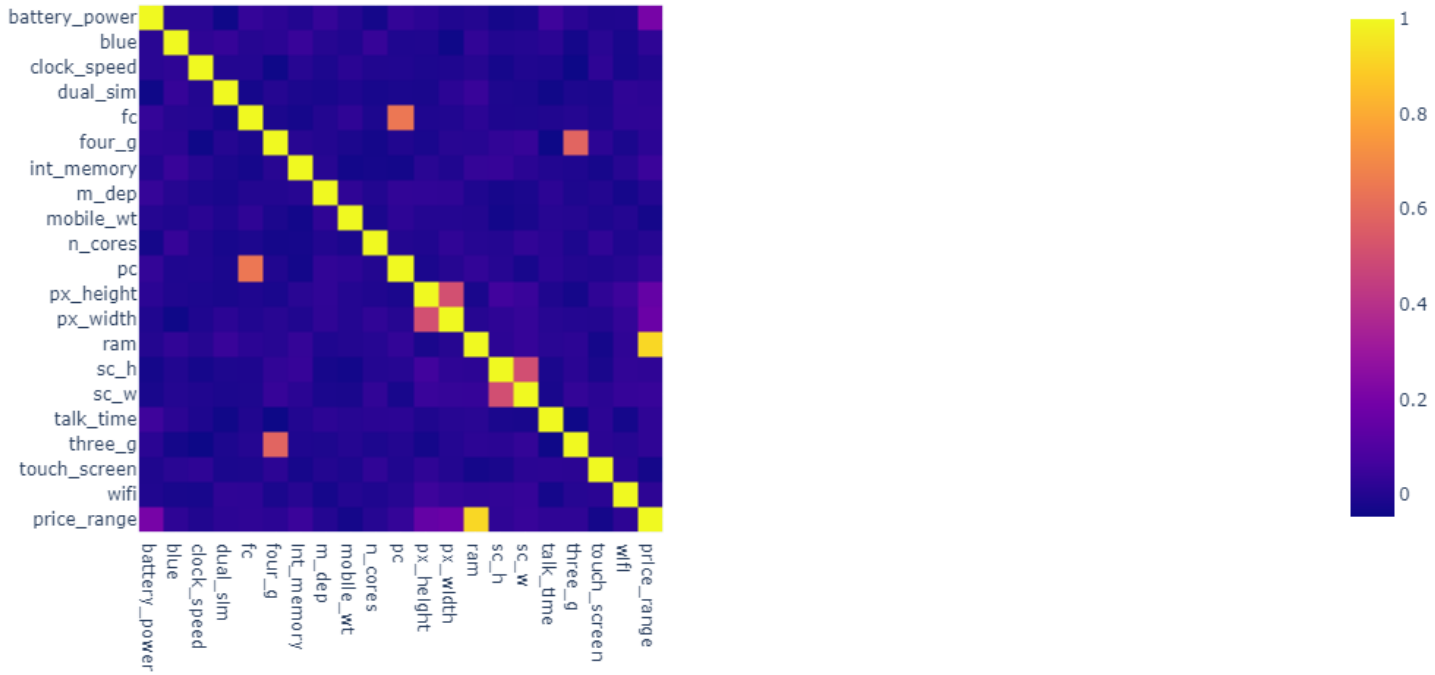
داده های این سری برای تخمین قیمت گوشی موبایل بر اساس ویژگی های آن بود.

قبل از شروع به انجام کارهای گفته شده و ساخت مدل ها موارد زیر را در مورد داده ها بررسی می کنیم.

در داده ها دیتای نادرست احتمالا وجود دارد که بهتر است آن ها را حذف کنیم. برای مثال گوشی های بسیار گرانی وجود دارند که دوربین ندارند. از طرفی بعضی از column ها نمی توانند مقدار منفی داشته باشند یا مثلا ستون ram نمی تواند هر مقداری داشته باشد.

پاک سازی این موارد در قسمت ئاکسازی داده ها می تواند انجام بشود. (در تمرین سری قبلی به آن پرداخته شده)

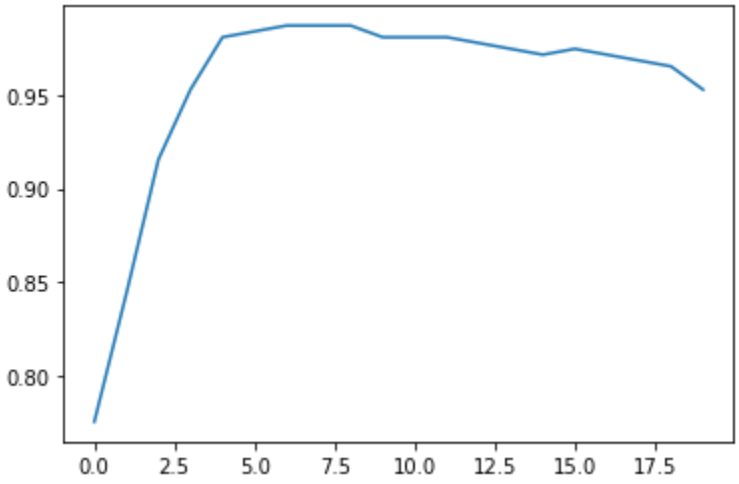
ابتدا به correlation داده ها نگاه کنیم.



همان طور که می بینیم بین بعضی از ستون ها ارتباطاتی وجود دارد. در انتخاب ویژگی ها قبل از ساخت مدل می توانیم از این موارد بهره ببریم.

ساخت و انتخاب مدل

داده ها را sample گیری می کنیم و به دو قسمت validation و train تقسیم می کنیم. سپس scaling انجام می دهیم و داده ها را به تابع forward_selection برای انتخاب feature ها می دهیم. در این روش یکی یکی همه ویژگی ها را اضافه میکنیم و بر اساس آن مدل logistic reggression را ساخته و معیار auc را حساب می کنیم. بهترین حالت را پیدا می کنیم و به همین صورت هر مرحله تعداد ویژگی ها را زیاد می کنیم. خروجی به این صورت خواهد شد. از بین همه ویژگی ها 6 ویژگی انتخاب شدند که شامل ram, battrey_power, px-height, px-weight, mobile-wt, sc-h هستند. نمودار score را در زیر مشاهده می کنید.



حال با ویژگی های انتخاب شده مدل را train می کنیم. خروجی مدل بدین صورت است.

percision	0.9893617	0.93814433	0.98958333	1
recall	0.97894737	0.98913043	0.95959596	0.99122807
f1score	0.98412698	0.96296296	0.97435897	0.99559471

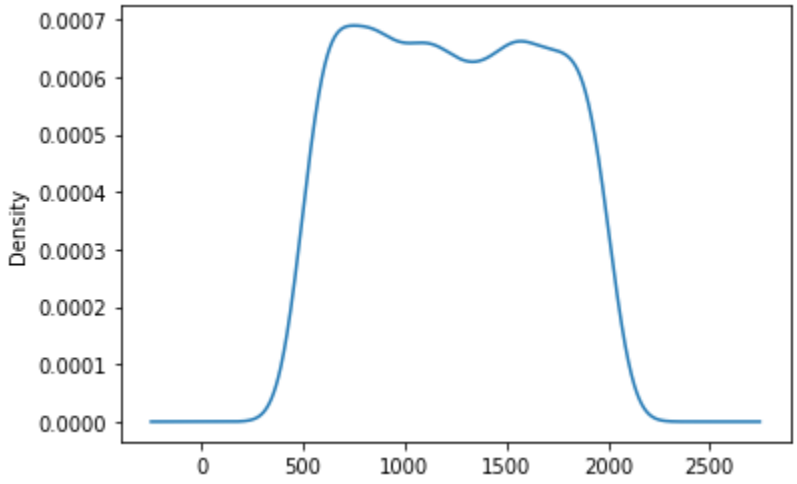
حال با pca داده ها را تبدیل به 6 کلاس می کنیم و دوباره مدل ها را می سازیم. خروجی بدین صورت می شود.

percision	0.54285714	0.86666667	1	0.61621622
recall	1	0.14130435	0.25252525	1
f1score	0.7037037	0.24299065	0.40322581	0.76254181

مشاهده می کنیم که نتایج بسیار متفاوت است. انتخاب روش کاری در موارد مختلف می تواند متفاوت باشد. برای مثال وقتی با تعداد زیاد feature سر کار داریم شاید چاره ای جز dimention reduction نداشته باشیم. اما در کل در این سوال مشاهده می کنیم که روش forward selection بهتر عمل کرده است.

Feature engineering

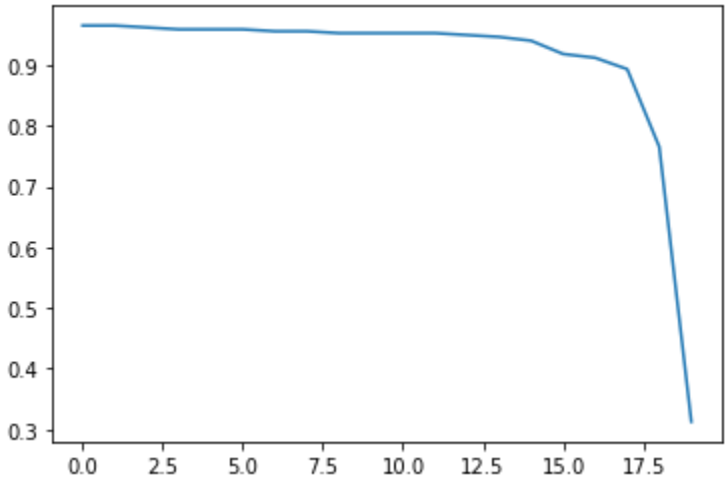
در این قسمت با روش binning برای battrey-power پیش می رویم. ابتدا نمودار زیر را مشاهده کنید.



عکس بالا نمودار distribution داده ها هست. با توجه به نمودار این ستون را به 3 قسمت کوچکتر از 800، بزرگتر از 1700 و بین این 2 مقدار تبدیل می کنیم. (مثلا تعداد کلاس کوچکتر از 800 مقدار 422 عدد می شود.)

backward selection

این روش را هم پیاده سازی کردیم. در اجرا با استفاده از این روش مجموعه بهترین feature های ما خالی شد.
نمودار زیر مقدار score ها در طول این فرآیند است.



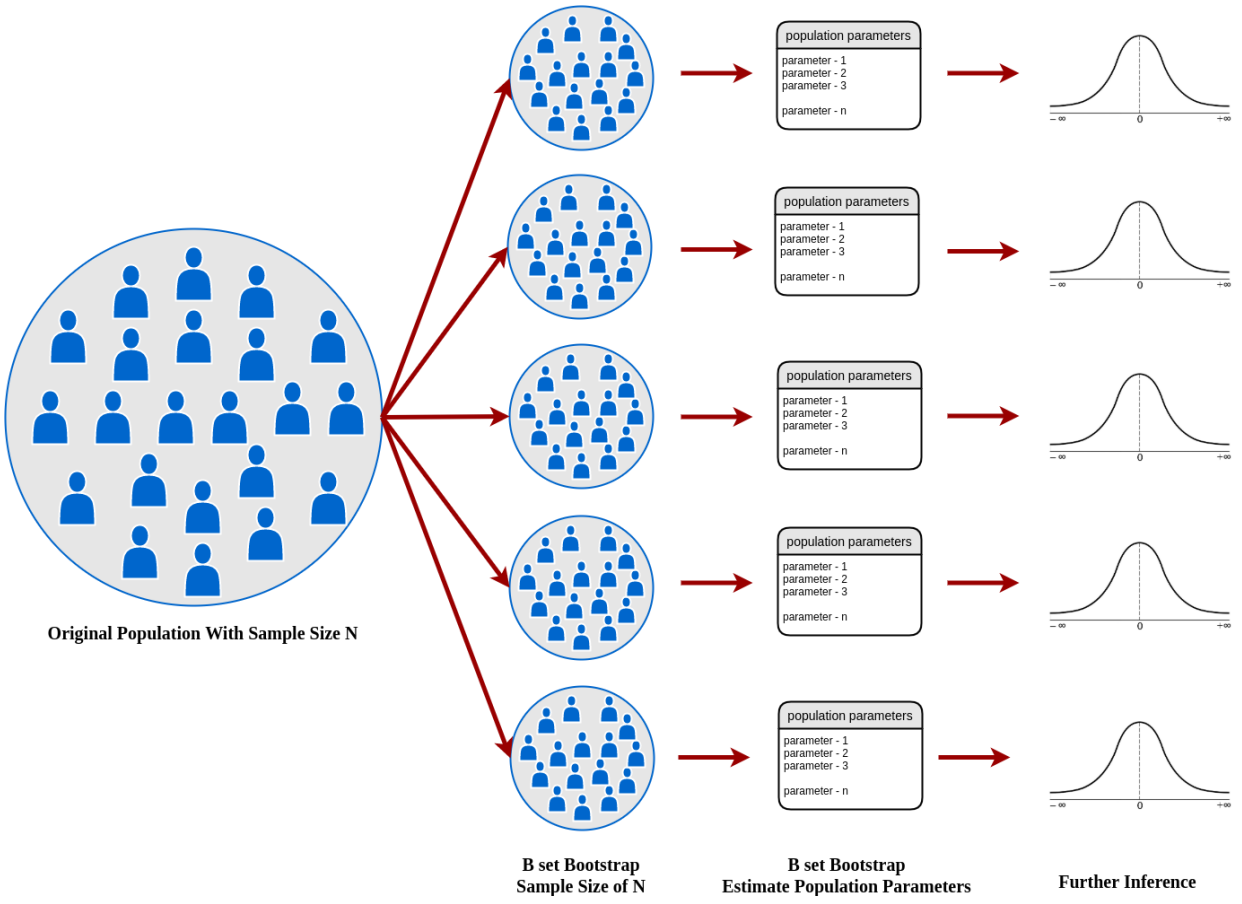
مشاهده می کنیم که نمودار همواره کم شده است.یعنی معیاری که برای انتخاب ویژگی انتخاب کردیم با همه ویژگی ها بهتر است و کلا هر چه تعداد ویژگی بیشتری داشته باشیم بهتر جواب می دهد. البته که داده ها و تمیز نبودن آن هم به طور کلی خیلی تاثیر دارد.

پاسخ به سوال های در سند

1. درباره one hot encoding. به طور کلی هر موقع با دیتا های categorical سر و کار داریم می توانیم به این روش فکر کنیم.در این روش هر حالت از یک ویژگی به یک ویژگی جدید تبدیل می شود و به صورت binary داشتن آن حالت را در یک داده نشان می دهیم. در حالاتی که تعداد حالت های ما زیاد است نمی توان خیلی یه این روش اتکا کرد چون ما نمی خواهیم تعداد ویژگی هایمان خیلی زیاد بشود. در مورد زمان استفاده، در مورد داده های categorical که معلوم است ما نیاز به داده عددی داریم. اما در حالتی که داده ها عددی هستند. با یک مثال پیش برویم. ساعت 4 عصر ممکن است با ساعت 6 عصر خیلی فرق نکند ولی صبح بودن یا عصر بودن تفاوت زیادی دارد. در این حالت زمان برای ما خیلی مهم نیست و در یک بازه خاصی بودن مهم است. این یک مدل استفاده از از این روش است.

2. در مورد log transform و exponential transform. بعضی اوغات توزیع داده ها normal است اما در مواردی که توزیع log normal است با log گرفتن می توان آن را به normal distribution تبدیل کرد. پس در کل این تبدیل ها برای این کار هستند و نرمال شدن توزیع معادل اضافه شدن یک سری فرض های آماری است که فرایند یادگیری را می تاوند خیلی راحت تر کند.
3. در مورد bootstraping و cross validation.

در bootstrapping چند sample با جایگذاری انتخاب می کنیم. سپس در معیار آماری در هر کدام حساب می شود و میانگین همه آن ها بهترین مقدار ها را می دهد. عکس زیر به خوبی نشان دهنده است.

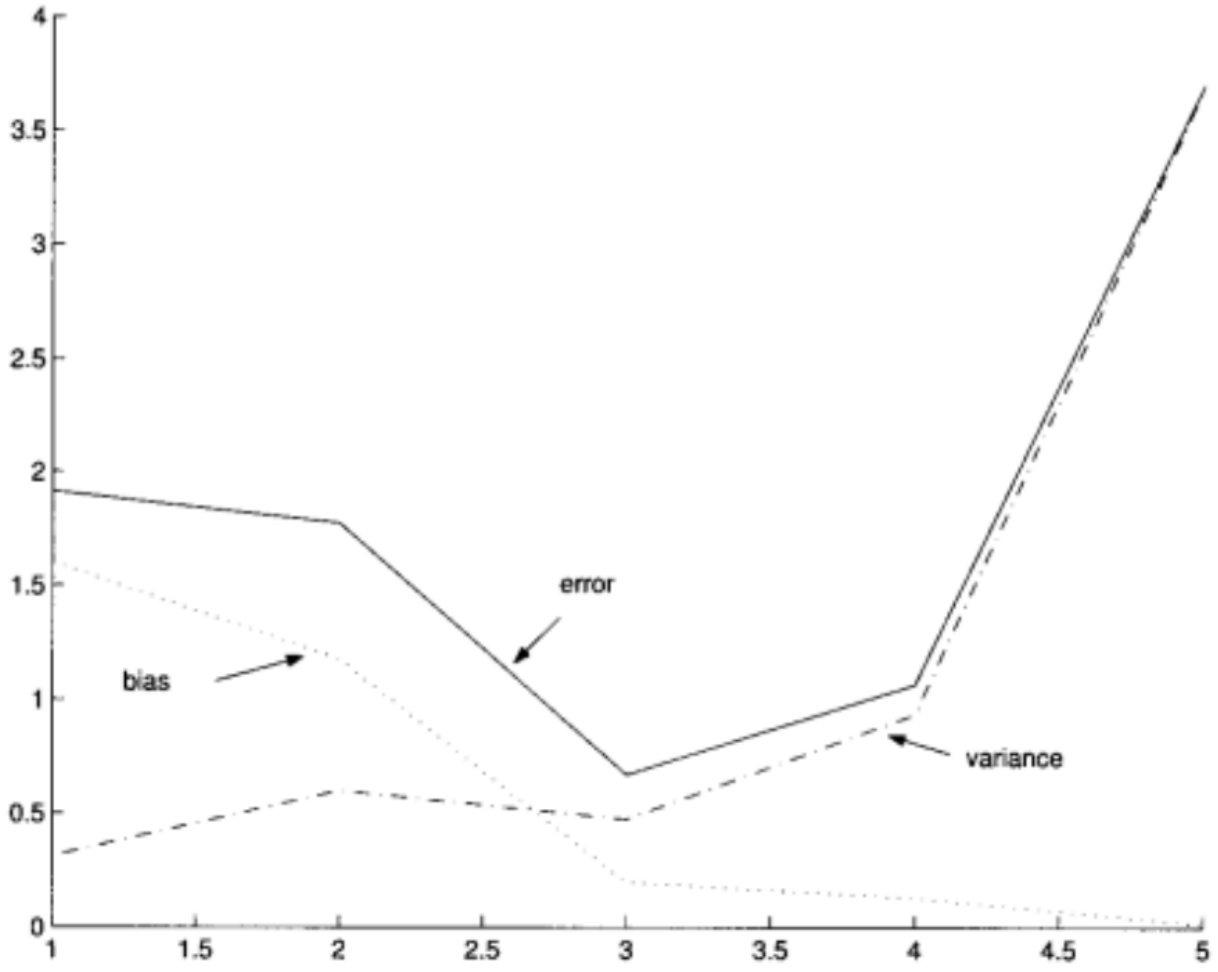


در cross validation در واقع sample های مختلف ساخته نمی شوند بلکه دیتا ما چندین بار به train و validation تقسیم می شود.

4. Cross fold 5x2

در این روش عملا 5 بار 2 fold را تکرار می کنیم. یعنی 5 بار داده ها را به train و validation تقسیم کرده ایم. موارد استفاده وقتی است که می خواهیم bias را کم کنیم. یعنی مواردی که bias زیاد است می تواند این کار به ما کمک کند.

5. در مورد سوال آخر



شکل ۱: نمودار بایاس و واریانس بنا بر مرتبه های مختلف مدل

بله. در الگوریتم k-means مقداری به نام inertia وجود دارد که برابر مجموع مربعات فاصله ها از نزدیک ترین مرکز خوشه است. وقتی نمودار مقادیر k بر حسب inertia را رسم کنیم از یک k مشخصی به بعد (مثلا 3) نمودار به صورت خطی کم می شود. اما قبل آن نمایی است. به این نقطه نقطه elbow می

گویند. و مقدار بهینه برای k همین نقطه است. در نمودار بالا نیز چنین الگویی وجود دارد. در نمودار هم مشاهده می کنیم از مرتبه 3 تا 5 مقدار خطی است.

خیر. چرا که رفتار تغییر بایاس همیشه به صورت خطی نیست اما با مشاهده واریانس و افزایش بیشتر واریانس نسبت به کاهش بایاس می توان نقطه بهینه برای انتخاب پیچیدگی مدل را پیدا کرد