به نام خدا

تمرین شماره 2 داده کاوی

علی نصرتی

400422200

**دیتاست شماره 1:**

1- در این جا برای پیاده سازی forward selection از تابع forwards استفاده می کنیم. می دانیم که در این الگوریتم باید ابتدا با یک مجموعه خالی شروع کرده و سپس هر فیچری که ویژگی خاصی داشت(که در این جا این ویژگی auc انتخاب شده است) به مجموعه اضافه کنیم تا به یک حالت پایانی برسیم. برای این موضوع می توان دو موضوع را مد نظر قرار داد اول این که برای مدل سازی و محاسبه auc باید دو کتابخانه sklearn.linear\_model و sklearn.metrics را اضافه کنیم البته یک کتابخانه هم برای تقسیم به train و test اضافه کرده ایم. نکته دوم این است که در این تابع ما سه پارامتر داریم که پارامتر اول داده های ماست که فیچر ها هم به عنوان column قرار دارند پارامتر دوم قسمتی است که ما باید تخمین بزنیم و پارامتر سوم هم دقتی است که ما انتظار داریم پس از انتخاب این الگوریتم به آن دست پیدا کنیم. همچنین نکته ای که در ابتدای اجرای این الگوریتم وجود داشت این بود که در حالت اول ما هر فیچر را به صورت تنها در نظر گرفته ایم و برای فیت کردن مدل خود نیاز به کد متفاوتی با مراحل دیگر داریم برای همین در مرحله اول این کد را جدا نوشته ایم و در مراحل دیگر تکرار نمی شود. بنابراین ما در هر مرحله یک فیچر با بهترین شرایط را انتخاب کرده و آن را به لیست selected\_features اضافه می کنیم و در نهایت این لیست را باز می گردانیم.

2- همان طور که در ادامه می توان دید مدل لجستیک بر روی فیچر های انتخاب شده اعمال شده است و precision برای صفر و یک 0.99 و recall برای صفر 0.98 و برای یک 0.99 است همچنین f1-score هم برای هر دو برابر 0.99 است.

3 و 4- در ادامه الگوریتم pca را بر روی مدل خود اجرا کرده ایم که بدلیل این که الگوریتم ما پنج مورد خروجی داده ما تعداد خروجی ها را برابر 5 گرفته ایم. معیار precision برای صفر برابر 0.99 و برای یک برابر 0.98 است. همچنین معیار recall برای صفر 0.97 و برای یک 0.99 است و معیار f1-score هم برای صفر و یک 0.98 است.

5- در حالت کلی ما از کرنل ها برای تغییر ابعاد داده ها استفاده می کنیم مثلا وقتی که می خواهیم یک مسِیله غیرخطی را بوسیله یک classifier خطی حل کنیم از کرنل استفاده می کنیم. انواع مختلفی از کرنل ها وجود دارد که از جمله آن ها می توان به linear, nonlinear, polynomial, radial basis function و sigmoid اشاره کرد. در حالت کلی کرنل چند جمله ای برای پردازش تصویر کاربرد دارد و معادله آن بصورت زیر است که d در آن درجه چند جمله ای است:



کرنل گاوسی برای اهداف عمومی است. و هنگامی که هیچ دانش پیشینی در مورد داده ها وجود ندارد استفاده می شود. معادله آن بصورت زیر است:



کرنل rbf برای داده های غیر خطی که اطلاعات زیادی درباره شان نداریم استفاده می شود و معادله آن بصورت زیر است:



کرنل های سیگموید و تانژانت هایپربولیک در شبکه های عصب استفاده می شوند و معادلات آن ها بصورت زیر است:



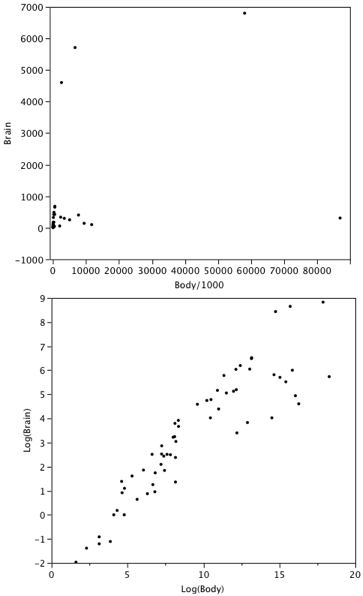


([منبع](https://shahaab-co.com/mag/edu/ml/svm-kernel-functions/))

7- با اجرای حالت های مختلف svm با کرنل های مختلف و مقادیر مختلف هایپر پارامترها مشاهده می شود که کرنل سیگموید بدترین عملکرد را روی داده های ما دارد و همچنین با دقت در میزان امتیاز های دریافتی سافت مارجین در این جا بهترین عملکرد را دارد البته با کم کردن c و زیاد کردن گاما مدل پیچیده ای با دقت نسبتا بالا پیدا کرد که حالتی نزدیک هارد مارجین دارد.

8- همان طور که در بالا بررسی شد هارد مارجین در حالت کلی روی داده های ما به خوبی عمل نمی کند اما اگر مدل پیچیده ای درنظر بگیریم می توان با هارد مارجین هم به نتایج خوبی رسید.

9- معمولا برای داده هایی که بشدت به یک طرف تمایل دارند و حالت نرمال ندارند از log transform استفاده می کنیم تا آن ها را به یک توزیع نرمال تبدیل کنیم. مثلا در تصویر زیر یک دسته داده را قبل و بعد از تبدیل لگاریتمی نشان می دهد([منبع](https://onlinestatbook.com/2/transformations/log.html#:~:text=The%20log%20transformation%20can%20be,can%20make%20patterns%20more%20visible)):



همان طور که می توان دید الگویی در تصویر بالا قابل مشاهده نیست اما در تصویر پایین این الگو با چشم غیر مسلح هم قابل دیدن است. علاوه بر مطالب بالا در [این سایت](https://gdcoder.com/when-why-to-use-log-transformation-in-regression/) گفته شده که تبدیل لگاریتمی برای داده هایی که به سمت راست تمایل دارند عملکرد بهتری دارد.

تابع نمایی در حالت کلی بصورت وارون تابع لگاریتمی عمل می کند اما تبدیل نمایی هم برای داده های با تمایل به یک طرف استفاده می شود البته این هم باید در نظر داشته باشیم که این تبدیل برای داده های unimodal استفاده می شود و بدرد جایی که ممکن است بیش از یک ماکزیمم داشته باشیم نمی خورد.

10- با اجرای تغییرات گفته شده روی مدل امتیاز ما در همه حالات کمتر شده ولی این کم شدن امتیاز با one hot encoding کمتر از بقیه بوده و با ساخت مساحت گوشی از همه حالات بیش تر کم شده پس می توان گفت مهندسی ویژگی ما در قسمت د از بقیه بدتر بوده و در قسمت ب از بقیه قابل قبول تر بوده است.

11- الگوریتم های ساخت درخت برای ساخت یک درخت از روی یک دیتاست استفاده می شوند. الگوریتم های مختلفی برای این کار وجود دارد که می توان به ID3 - CHAID - CART و .. اشاره کرد. ID3 یکی از الگوریتم‌­های بسیار ساده درخت تصمیم که در سال 1986 توسط Quinlan مطرح شده است. در این الگوریتم اطلاعات به دست آمده به عنوان معیار تفکیک به کار می­‌رود. این الگوریتم هیچ فرایند هرس کردن را به کار نمی­‌برد و مقادیر اسمی و مفقوده را مورد توجه قرار نمی­‌دهد.

C4.5: این الگوریتم تکامل یافته ID3 است که در سال 1993 توسط Quinlan مطرح شده است. در این الگوریتم Gain Ratio به عنوان معیار تفکیک در نظر گرفته می­‌شود. عمل تفکیک زمانی که تمامی نمونه‌­ها پایین آستانه مشخصی واقع می­‌شوند، متوقف می­‌شود. پس از فاز رشد درخت عمل هرس کردن بر اساس خطا اعمال می­‌شود. این الگوریتم مشخصه­‌های اسمی را نیز در نظر می­‌گیرد.

CART برای برقراری درخت­‌های رگرسیون و دسته‌­بندی از این الگوریتم استفاده می­‌شود. در سال 1984توسط Breiman و همکارانش ارائه شده است. نکته حائز اهمیت این است که این الگوریتم درخت­‌های باینری ایجاد می­‌کند به طوری که از هر گره داخلی دو لبه از آن خارج می­‌شود و درخت­‌های بدست آمده توسط روش اثربخشی هزینه، هرس می­‌شوند. یکی از ویژگی‌­های این الگوریتم، توانایی در تولید درخت­‌های رگرسیون است. در این نوع از درخت­‌ها برگ‌ها به جای کلاس مقدار واقعی را پیش­بینی می‌­کنند. الگوریتم برای تفکیک کننده‌­ها، میزان مینیمم مربع خطا را جستجو می­‌کند. در هر برگ، مقدار پیش­بینی بر اساس میانگین خطای گره‌­ها می­‌باشد.

CHID: این الگوریتم درخت تصمیم به جهت در نظرگرفتن مشخصه­‌های اسمی در سال 1981 توسط Kass طراحی شده است. الگوریتم برای هر مشخصه ورودی یک جفت مقدار که حداقل تفاوت را با مشخصه هدف داشته باشد، پیدا می­‌کند.

هرکدام از این الگوریتم ها تفاوت ها و شباهت هایی دارند مثلا ID3, C4.5 and C5.0 از Information gain به عنوان معیار سنجش استفاده می کنند و الگوریتم CART از Gini impurity استفاده می کند. همچنین بعضی الگوریتم ها مثل CART از نوآوری هایی مثل Variance reduction استفاده می کنند.([منبع](https://pmpiran.com/decision-tree/))

13- بله همان طور که می توان دید درختی که عمق حداکثر 5 دارد از درختی که عمق حداکثر 10 دارد امتیاز کمتری دارد. همچنین بنظر می رسد میزان نمونه های موجود در هر گره هم بر کیفیت درخت تاثیرگذار است اما این تاثیر مثل حالت قبلی خطی نیست.

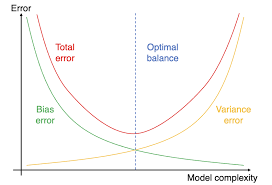
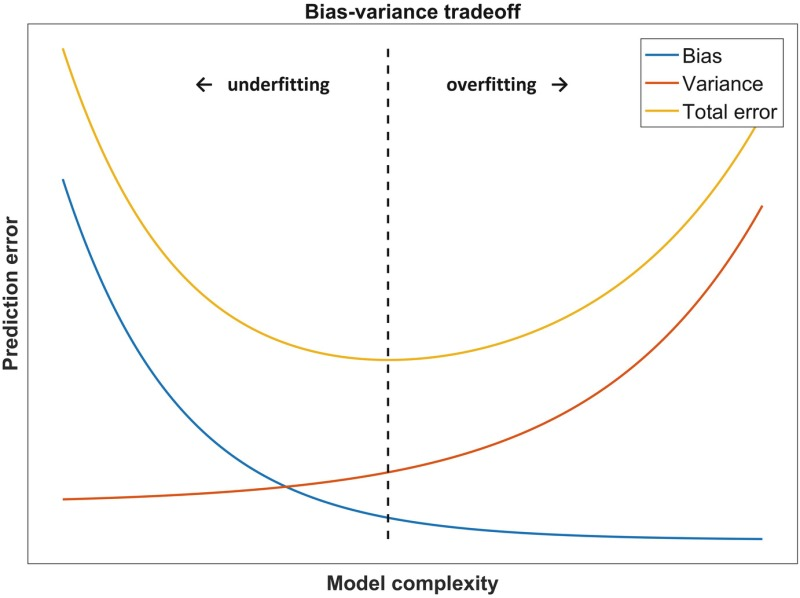
14- هرس کردن درخت به ما کمک می کند که از overfitting جلوگیری کنیم همچنین باعث می شود تصمیم های ضعیف تر و با تاثیر کم تر را حذف کرده و درختی کاربردی تر(efficient) و کوچک تر و در عین حال قابل استفاده داشته باشیم.

15- تفاوت Bootstrapping و Cross validation در این است که ما در Bootstrapping یا داده کمی داریم و یا نیاز به کم کردن واریانس داده های خود داریم در این شرایط می توانیم از Bootstrapping استفاده کرده و تعداد بیش تری داده بسازیم. اما در Cross validation ما به دنبال یافتن خطای دقیق تری هستیم و با مشخص کردن عددی به عنوان k می گوییم که روی k-1/k ازداده ما training را انجام بده و روی بقیه test را انجام بده و در نهایت این کار را روی همه بخش های مختلف ما اجرا کن مثلا اگر k برابر با 5 باشد باید 5 بار روی ⅘ داده های خود Train انجام دهیم و 5 بار روی مقدار باقیمانده Test انجام دهیم و سپس میانگین بگیریم. این کار باعث می شود خطای واقعی تری پیدا کنیم.

16- در cross validation پنج در 2 ما باید پنج بار ازtwo fold-CV استفاده کنیم و به زبان ساده تر باید پنج بار داده های خود را به صورت تصادفی به دو قسمت مساوی تقسیم کرده و هربار میزان خطای تست خود را اندازه می گیریم و در نهایت میانگین می گیریم.

این روش زمانی استفاده می شود که ما بدانیم الگوریتم یادگیری ما به اندازه کارا است که بتوان آن را ده بار تکرار کرد همچنین عدد 2 برای این انتخاب شده است که مطمئن شویم هر مشاهده تنها در دیتاست تست یا ترین در هر تقریب از مدل ما استفاده می شود.

17- به نظر می رسد در این نمودار آرنج نمودار روی حالتی که بایاس و واریانس در کمترین اندازه خود هستند همچنین با توجه به ترید آف بایاس و واریانس انتظار می رود همواره جمع این دو در نقطه ای متعادل تر از بقیه نقاط باشد بنابراین روش elbow می تواند جواب قابل قبولی بدهد البته باید در نظر داشت که روش elbow یک روش هیوریستیک است و ممکن است همیشه بهترین جواب را به ما ندهد اما با توجه به نمودار های مختلف به نظر می رسد معمولا جواب بهینه ای بدست بدهد:



تسک های امتیازی:

1- با تغییر تابع forwards آن را به تابعی برای محاسبه backward selection تبدیل می کنیم. برای این کار ورودی های دیتا فریم و تارگت را تغییر نمی دهیم و به جای معیار دقت آلفا را جایگزین کرده و این بار در هر بار محاسبه pvalue را بوسیله کتابخانه statsmodels.api محاسبه می کنیم و مقداری که از همه بیش تر بود را حذف می کنیم تا وقتی که چیزی کمتر از آلفای انتخاب شده وجود نداشته باشد این تابع را forwards می نامیم.

precision برای صفر 0.96 و برای یک 0.97 و recall برای صفر 0.96 و برای یک 0.97 است همچنین f1-score هم برای صفر 0.96 و برای یک 0.97 است. با توجه به این مقادیر می توان گفت که با وجود تعداد فیچر های کمتر الگوریتم forward selection در این جا بهتر از backward selection عمل می کند و همه مقادیر در الگوریتم اجرا شده forward selection بهتر هستند.

2- در حالت کلی برای مقایسه دو مدل مختلف می توان از معیار های مختلفی استفاده کرد که ساده ترین آن ها معیار دقت است که ما می توانیم با استفاده از آن تصمیم بگیریم که یک مدل از دیگری بهتر است. اما در عمل برای تصمیم گرفتن نیاز به درنظر گرفتن نکات بیش تری داریم. یک راه برای انجام این مقایسه استفاده از آزمون های فرض است. مثلا ما در نظر می گیریم که فرض صفر ما این است که هیچ تفاوت خاصی در مدل های ما نیست و سپس باید یک آزمون فرض درنظر بگیریم. این مرحله کمی حساس تر از قبل است و آزمون های فرض زیادی وجود دارند که ما می توانیم استفاده کنیم و هر کدام یک تعداد شرایط دارند که باید داده های ما در این شرایط صدق کنند. آزمون های فرض به دو دسته تقسیم می شوند آزمون های فرض پارامتری و غیر پارامتری. تفاوت اصلی این دو نوع در این است که آزمون های پارامتری درباره نوع توزیع داده ها فرض هایی دارند. مثلا آزمون پارامتری t-test فرض می کند که داده های ما توزیع نرمال دارند. با وجود این اگر ما بخواهیم از آزمون t-test استفاده کنیم و داده های ما نرمال نباشند می توان از یک جایگزین غیر پارامتری مثل Wilcoxon signed rank test استفاده کرد. در نهایت اگر p-value ما از آلفا کمتر باشد می توان فرض صفر را رد کرد. اما نکته دیگری که باید در نظر گرفت این است که آیا اگر فرض صفر ما رد شود و در واقع اگر بین داده های ما اهمیت آماری وجود داشته باشد به این معناست که تفاوت مهمی بین این دو مدل وجود دارد؟ واقعیت این است که لزوما وجود تفاوت معنادار آماری به معنای اهمیت واقعی این تفاوت نیست و اگر داده های ما به اندازه کافی بزرگ باشند با وجود این که ممکن است تفاوت بنظر ناچیز بیاید اما ممکن است باز هم از لحاظ آماری با معنی باشد به همین دلیل در این جا باید به مفهوم practical significance هم توجه کرد. مفهوم practical significance میزان بزرگی تاثیر یا Effect size هم درنظر می گیرد و می توان با استفاده از این مفهوم به نتیجه بهتری دست پیدا کرد.([منبع](https://www.kdnuggets.com/2019/01/comparing-machine-learning-models-statistical-vs-practical-significance.html))

3- mcc یا Matthew’s correlation coefficient معیاری برای خلاصه کردن confusion matrix یا error matrix است و یک confusion matrix دارای 4 موجودیت است:

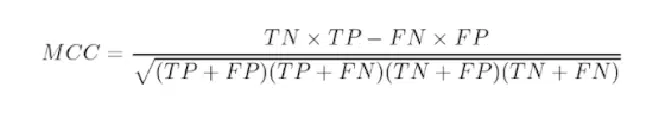
True positives (TP)

True negatives (TN)

False positives (FP)

False negatives (FN)

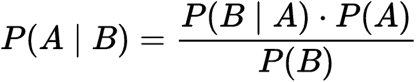
معیار MCC با استفاده از چهار موجودیت بالا بصورت زیر بدست می آید:



پس از محاسبه MCC توسط فرمول بالا باید آن را تحلیل کنیم. مقدار MCC بین -1 و 1 است و هر چه به یک نزدیک تر باشد به این معناست که مدل بهتری داریم و اگر به صفر نزدیک تر باشد یعنی مدل ما به یک دسته بند تصادفی نزدیک تر است.

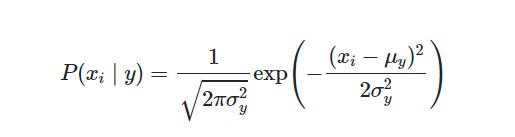
**دیتاست شماره 2:**

1- قضیه بیز روشی برای دسته بندی داده ها بر پایه وقوع یا عدم وقوع آن هاست. با این قضیه می توان احتمال وقوع یک پیشامد را نسبت به وقوع یا عدم وقوع پیشامد دیگر محاسبه کرد. از آن جایی که در بسیاری از حالت ها محاسبه وقوع یا عدم وقوع یک پیشامد به طور مستقیم کار ساده ای نیست با استفاده از این قضیه و مشروط کردن پیشامد مورد نظر به پیشامد دیگر می توان احتمال مورد نظر را محاسبه کرد. فرمول قضیه بیز بصورت زیر می باشد:



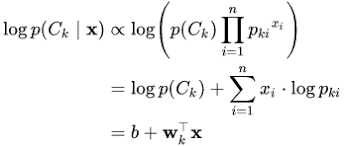
که در آن A و B رویداد های ما هستند و P آن ها احتمالات آن ها است همچنین B|A و A|B احتمال وقوع B به شرط صحیح بودن A و احتمال وقوع A به شرط صحیح بودن B را نشان می دهد.

اگر مشاهدات و داده ها پیوسته باشند از مدل احتمالی با توزیع گاوسی برای متغیر ها استفاده می کنیم. در این حالت همه دسته ها دارای توزیع گاوسی هستند بنابراین اگر k دسته داشته باشیم برای هر دسته می توان میانگین و واریانس و بقیه متغیر ها را محاسبه کرده و پارامترهای توزیع نرمال را برآورده کنیم.



در فرمول بالا μ میانگین σ انحراف از معیار و exp عدد نپر می باشد.

Multinomial Naive Bayes یا بیز ساده چند جمله ای برای دسته بندی متنی بسیار کار آمد است. برای این کار بر حسب مدل احتمالی یا توزیع چند جمله ای برداری از n ویژگی با n احتمال در نظر گرفته می شود و مشخص است در این حالت بردار مشاهدات برابر مشاهداتی است که ویژگی خاصی دارند بنابراین تابع آن بصورت زیر است:



و همان طور که می توان دید با گرفتن لگاریتم از تابع بصورت یک دسته بند خطی در می آید.

Bernoulli Naive Bayes هم مانند بیز چند جمله در دسته بندی متن کاربرد دارد اما با این تفاوت که بیشترین کاربرد را در دسته بندی متن های کوتاه دارد. این مدل در حالت چند متغیره وجود یا عدم وجود یک ویژگی را در نظر می گیرد. مثلا با توجه به یک لغتنامه دلخواه متن مورد نظر ما مورد تجزیه و تحلیل قرار می گیرد که آیا کلمات مربوط به لغتنامه در متن وجود دارد یا خیر. به این ترتیب مدل ما بر اساس کلاس های مختلف به شکل زیر نوشته می شود:



2- الگوریتم خود را با نام bayes تعریف کرده ایم و این الگوریتم چهار ورودی دارد که ورودی اول و چهارم داده train و test می باشد و ورودی دوم و سوم داده ستون هدف و ستون های انتخاب شده است (البته نیازی به قرار داده این ستون نبود اما از آن جایی که در صورت سوال گفته شده بود از چه ستون هایی استفاده کنیم این ستون هم اضافه شد.) در ادامه به تعداد مقادیر موجود متغیر ساخته و بعد به ازای مقادیر موجود در ورودی 3 محاسبات را انجام می دهیم و ضرب در مقدار کسری هر کدام می کنیم در نهایت نتایج را در یک دیتافریم ذخیره کرده و برای هر سطر بیشترین مقدار را به عنوان خروجی بر می گردانیم.

3- پس از اجرای داده های روی دسته بند ساخته شده نتیجه precision برای داده های صفر برابر 0.72 و برای داده های یک برابر 0.89 - نتیجه recall برای داده های صفر برابر 0.94 و برای داده های یک برابر 0.59 و نتیجه f1-score برای داده های صفر برابر 0.82 و برای داده های یک برابر 0.71 می باشد.

4- پس از ساخت مدل با GaussianNB نتیجه precision برای داده های صفر برابر 0.74 و برای داده های یک برابر 0.78 - نتیجه recall برای داده های صفر برابر 0.84 و برای داده های یک برابر 0.67 است و نتیجه f1-score برای داده های صفر برابر 0.79 و برای داده های یک برابر 0.72 می باشد.

5- بررسی اولیه معیار ها در مدل ساخته شده توسط ما و مدل ساخته شده توسط GaussianNB نشان می دهد که مدل ما در حالت میانگین کمی بهتر عمل کرده است اما مدل ساخته شده با پکیج کمی متعادل تر عمل کرده و هر دو مقدار را با precision مشابهی بدست آورده است. نکته دیگری که با بررسی و مقایسه مقادیر بدست آمده می توان گفت این است که هر دو مدل در پیش بینی مقادیر 0 بهتر عمل کرده اند و به سمت مقدار صفر متمایل اند اما تمایل مدل ساخته شده توسط ما کمی بیش تر است. همچنین دلیل این تفاوت ها می تواند مقادیر استفاده شده در عدد و نپر و عدد پی و همچنین میزان اندازه اعداد برای تقریب باشد که ما در این جا برای این اعداد از پکیج numpy استفاده کردیم ولی ممکن است در GaussianNB از تقریب های دیگری استفاده شده باشد.