گزارش پروژه دوم داده کاوی(بخش 1)

محمدرضا صیدگر-97222055

1) برای این بخش مجموعه مشخصات موبایل های مختلف بود. شروع به مرتب سازی داده ها کردیم ولی در کل مجموعه داده ی بسیار تمیزی بود و داده null نداشت و حتی outlier نداشتیم و همه داده ها هم داده های عددی بودند که این هم خودش راحتی کار با این مجموعه داده بود.
انتخاب ویژگی forward selection رو بدون استفاده از پکیج پیاده سازی کردم و با انتخاب 8 ویژگی بر اساس معیار auc که مساحت زیر نمودار roc curve است.
2) معیار های score-f1 ، recall ، precision را بررسی کرده و نتایج زیر را داشت :

score: 0.9675
precision: [0.98130841 ,0.95652174 ,0.97647059 ,0.95689655]
recall: [1. , 0.96703297 ,0.90217391 ,0.99107143]
f1 score: [0.99056604 ,0.96174863 ,0.93785311 ,0.97368421]
که نتایج بالا نشان می دهد بسیار عملکرد خوبی داشتیم

3و4) در این بخش کاهش فیچر داشتیم با الگوریتم pca با همان تعداد فیچر 8 تایی که در سوال قبلی رگرسیون رو روی این داده های اعمال کردیم که این مدل خوبی نشد و ارزیابی مدل به این صورت شد که precision حدود 0.28 و recall حدود 0.28 و f1-score حدود 0.27 داشتیم .

5) ما در svm از کرنل استفاده میکنیم چون کرنل مجموعه ای از توابع ریاضی مورد استفاده در svm ، پنجره ای را برای دستکاری داده ها به کار می برد.

بنابراین ، کرنل به طور کلی مجموعه آموزش داده ها را به گونه ای تغییر می دهد که سطح تصمیم گیری غیر خطی قادر به تبدیل شدن به یک معادله خطی در تعداد بیشتری از فضاهای بعد باشد.
درباره توابع اصلی و مهم کرنل به Gaussian Kernel میتوان اشاره کرد که هنگامی که هیچ ایده ای در مورد داده ها وجود ندارد ، برای transformation استفاده می شود.کرنل بعدی rbf است که همان شبیه کرنل فوق است با افزودن متد پایه شعاعی برای بهبود عملکرد آن.کرنل بعدی sigmoid است که این تابع معادل یک مدل پرسپترون دو لایه از شبکه عصبی است که به عنوان تابع فعال سازی برای نورونهای مصنوعی استفاده می شود.

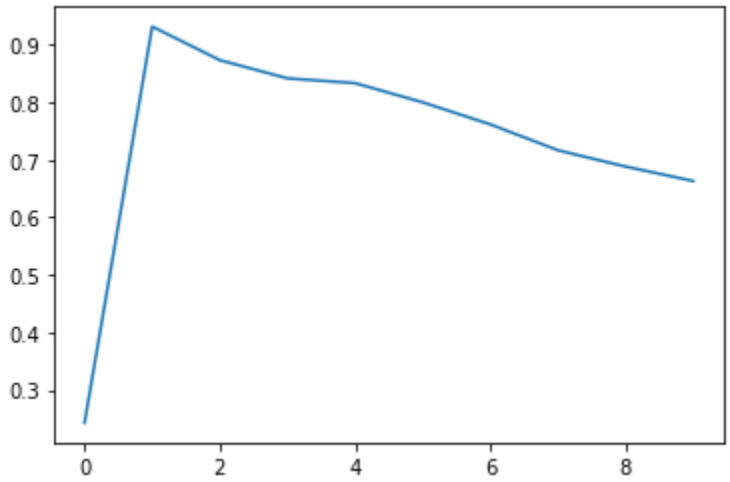
6) برای این سوال x , y را جدا کردیم تا به مدل هایمان بدهیم تا نتایج را بدست بیاوریم.
با استفاده از پکیج های sklearn مدل ساده svm ساختیم تا روی داده ها آموزش ببیند و سپس روی داده های تست ارزیابی شود که مدل ما حدودا 85 درصد درستی داشت.

139	12	0	0
12	120	14	0
0	23	116	9
0	0	20	135

این هم confusion matrix این مدل ما بود که اعداد روی قطر نسبت به اعداد دیگر بزرگتر است یعنی درستی خوبی داشتیم.

7)

برای این بخش پارامتر ها را روی مدل سوال قبلی اجرا کردیم تا ببینیم چه تاثیری در خروجی یعنی درصد درستی ما خواهد داشت
در مدل جدید اول ما از کرنل خطی استفاده کردیم در واقع از کرنل خطی زمانی استفاده می شود که داده ها بصورت خطی قابل تفکیک باشند ، یعنی می توان آنها را با استفاده از یک خط جدا کرد. آموزش svm با کرنل خطی سریعتر از کرنل های دیگر است. در این مدل درستی ما حدود 93 درصد بود.
در مدل دوم سعی کردیم 10 مدل مختلف بگنجانیم ، از کرنل poly استفاده کردیم از درجه 0 تا 10 که درستی به شکل زیر شده است:



که x درجه poly را و y میزان درستی را نشان میدهد.
در مدل سوم از تابه تصمیمگیری one versus one استفاده کردیم تا ببینم درستی چقدر خواهیم داشت که درستی 85 درصد داشتیم.
در مدل چهارم متغیر gamma = auto قرار دادیم که این متغیر میزان تاثیر داده آموزشی را مشخص میکند که در این حالت درستی 87 درصد داشتیم که درستی بهتری شد.

9) در بخش اول این سوال داده ها را در bin های مختلف ریختیم. در بازه های زیر بر حسب ستون battery power
(500,875.25),(875.25,1249.5),(1249.5,1623.75),(1623.75,1999)

و در مرحله بعدی به هر کدوم از این بازه ها برچسب های bad و medium و good و nice دادیم و این ها را به عنوان یک ستون جدید به داده ها اضافه کردیم.

در بخش دوم این سوال چون طبق بخش قبلی داده هایی با 4 برچسب مختلف به داده ها اضافه شد پس باید این داده ها را one hot encoding کنیم که از تابع get dummies در پکیج pandas استفاده کردیم و دلیل استفاده از one hot encoding این است که اجازه می دهد تا نمایش داده های categorical رساتر باشد. بسیاری از الگوریتم های یادگیری ماشین نمی توانند مستقیماً با داده های دسته ای کار کنند. این مورد هم برای متغیرهای ورودی و هم برای متغیرهای خروجی که categorical هستند لازم است.

در بخش سوم این سوال از گفته شد از log transform یا exp transform استفاده کنیم که من از تابع transform پکیج pandas استفاده کردم و داده ها را به log و exp بردم که اصلا نتایج خوبی را به دنبال نداشت چرا که طبق تصویر زیر که می بینید بعضی از داده ها برچسب -inf خورده اند به این معنی که داده های 0 که داشتیم در لگاریتم طبیعتا حالا به منفی بینهایت تبدیل شده اند و این اجازه نمیدهد که بتوانیم مدل های یادگیری ماشین مثل svm را روی آن ها پیاده سازی کنیم.

	battery_power	blue	clock_speed	dual_sim	fc	four_g
0	6.735780	-inf	0.788457	-inf	0.000000	-inf
1	6.928538	0.0	-0.693147	0.0	-inf	0.0
2	6.333280	0.0	-0.693147	0.0	0.693147	0.0
3	6.421622	0.0	0.916291	-inf	-inf	-inf
4	7.507141	0.0	0.182322	-inf	2.564949	0.0

اما در بخش چهارم این سوال خواسته شد که ویژگی جدیدی به نام مساحت به داده ها اضافه کنیم که من ستون های طول و عرض را در هم ضرب کردم و با عنوان area یک ستون به داده ها اضافه کردم.

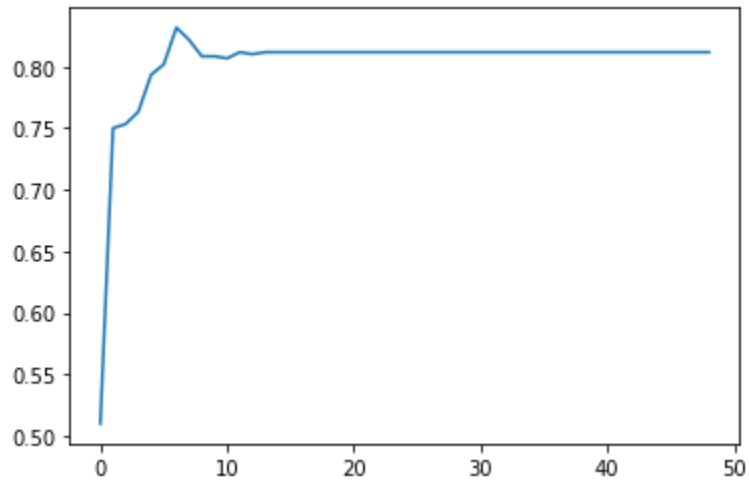
10) در این بخش پیاده سازی svm روی موارد بالا را بایستی ببینیم که روی مورد اول و دوم بهم وابسته اند از طرفی مدل سوم هم قابل اجرا نبود بخاطر داده های null و inf پس فقط به صورت جدا روی بخش 4 مدل svm زدیم که درستی 85 درصد داشتیم و به طور کل روی بخش 1 و 2 و 4 مدل svm درستی 81 درصد داشته است.

11) الگوریتم درخت تصمیم به گونه ای عمل میکند که سعی دارد گوناگونی و یا تنوع را در گره ها به حداقل ممکن برساند.این عدم یکنواختی در گره ها با استفاده از معیار های measure impurity قابل اندازه گیری است که مهمترین و پرکاربردترین ان شاخص جینی میباشد.
اغلب تفاوت انواع درخت های تصمیم در همین معیار اندازه گیری measure impurity، شیوه شاخه بندی یا splitting و هرس کردن گره های درخت میباشد.
الگوریتم اول ID3 است که یکی از الگوریتم‌های بسیار ساده درخت تصمیم است که در سال 1986 توسط Quinlan مطرح شده است. 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.
الگوریتم دوم CART است که متغیر های ورودی را برای یافتن بهترین تجزیه می آزماید تا شاخص ناخالصی حاصل از تجزیه کمترین مقدار باشد. در تجزیه دو زیرگروه تعیین میشود و هر کدام در مرحله بعد به دو زیگروه دیگر تقسیم خواهند شد و این روند ادامه می یابد تا زمانی که یکی از معیار های توقف برآورده شود .درخت CART بازگشتی دو دویی است که گره های والدین را دقیقا به دو گروه فرزند منشعب میکند و به طور بازگشتی منشعب کردن را تا زمانی که انشعاب دیگری نتواند ساخته شود ادامه میدهد.
الگوریتم بعدی C4.5 است .این الگوریتم درخت تصمیم، تکامل یافته ID3 است

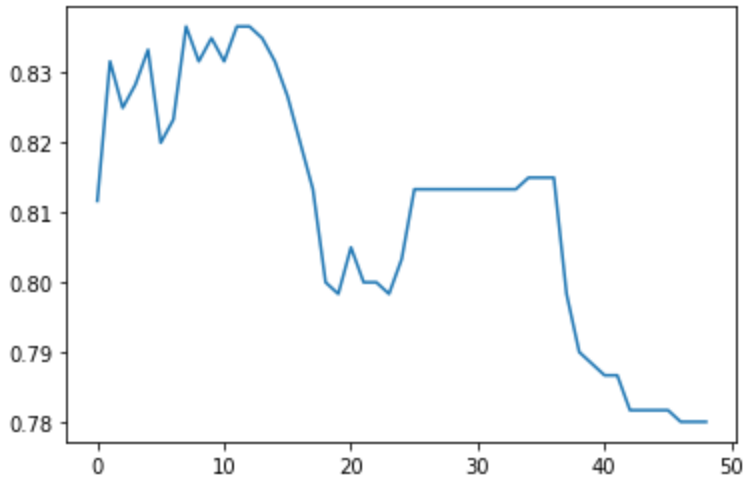
Gain Ratio به عنوان معیار تفکیک در نظر گرفته می‌شود. عمل تفکیک زمانی که تمامی نمونه‌ها پایین آستانه مشخصی واقع می‌شوند، متوقف می‌شود. پس از فاز رشد درخت عمل هرس کردن بر اساس خطا اعمال می‌شود. این الگوریتم مشخصه‌های اسمی را نیز در نظر می‌گیرد.

12) برای این سوال از پکیج sklearn استفاده کردم از بخش decision tree classifier که یک مدل درخت تصمیم ساختیم که درستی 81 درصد داشته است روی داده های تست.

13)
در این سوال در بخش اول ماکزیمم عمق درخت رو تغییر دادیم از 1 تا 50 که نمودار تغییرات به شکل زیر شد

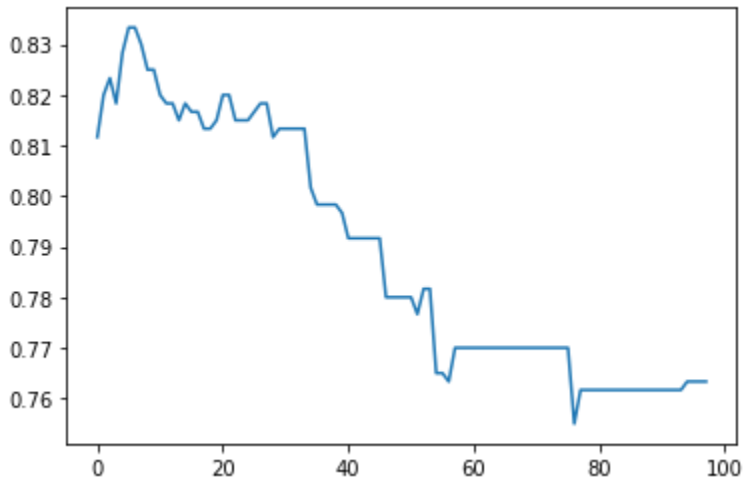


که کمترین مقدار 51 درصد در عمق 1 بوده و بیشترین 83 درصد در عمق 7 بوده است و از جایی به بعد درستی ثابت شده است.
در بخش بعدی بررسی کردیم تغییرات تعداد نمونه های موجود در گره های برگ را که شکل زیر شده است:



که کمترین مقدار 78 درصد در تعداد نمونه های 49 بوده و بیشترین 84 درصد در تعداد نمونه های 8 بوده است که به طور کلی از جایی به بعد نزولی شده است.

در بخش بعدی بررسی کردیم تغییرات تعداد نمونه های موجود در گره های داخلی را که شکل زیر شده است:



که کمترین مقدار 76 درصد در تعداد نمونه های 77 بوده و بیشترین 83 درصد در تعداد نمونه های 2 بوده است که به طور کلی از جایی به بعد نزولی شده است.

14) هرس روش حذف شاخه های بلا استفاده از درخت تصمیم است. برخی از شاخه های درخت تصمیم ممکن است نمایانگر داده های نویز یا outlier باشد.

هرس درخت روشی است که شاخه های ناخواسته درخت را کاهش می دهد. این امر از پیچیدگی درخت می کاهد و به تحلیل پیش بینی موثر کمک می کند. به دلیل از بین بردن شاخه های غیر مهم درختان ، از overfitting آن می کاهد.
دو رویکرد رایج برای هرس درخت وجود دارد:

A)Pre pruning:

در این رویکرد یک درخت به وسیله توقف‌های مکرر در مراحل اولیه ساخت درخت، هرس می‌شود. به محض ایجاد یک توقف گره به برگ تبدیل می‌شود.

B)Post pruning:

رویکرد هرس پسین درخت تصمیم رایج‌تر است به این صورت که زیر درخت‌ها از یک درخت رشد یافته کامل را حذف می‌کند. یک زیر درخت در یک گره به وسیله حذف کردن شاخه‌ها و جایگزینی آن‌ها با یک برگ، هرس می‌شود.

15)Bootstrapping روشی است شامل نمونه گیری تکراری با جایگزینی، از داده های اصلی به منظور تخمین پارامتر جمعیت.
از Bootstrapping استفاده میکنیم برای این که زمانی که داده کمی داریم؛ انتخاب تصادفی داده ها نیز این اطمینان را نمی دهد چرا که 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 چون می توان تعداد نمونه های زیادی ایجاد کرد.

در Bootstrapping چند نمونه از داده های اولیه ساخته می شود و در ادامه مدل روی هر یک از آنها آموزش می بیند و روی تفاضل هر یک از جامعه اصلی ارزیابی می شود و در اخر برای ارزیابی مدل از نتایج میانگین گیری می شود.

16) در این روش ما 5 بار 2fold cross validation رو انجام میدیم.

گزارش پروژه دوم داده کاوی(بخش 2)

محمدرضا صیدگر-97222055

1- طبقه بندی کننده بیز با استفاده از قضیه بیزی به خانواده طبقه بندی کننده های احتمال تعلق دارد. دلیل نامیده شدن آن به دلیل نیاز به فرض مستقل بودن بین متغیرهای ورودی است.

هدف طبقه بندی کننده بیز محاسبه احتمال شرطی زیر است:

$$p(C_k|x_1, x_2, \ldots, x_n)$$

برای هر یک از K نتایج ممکن یا کلاس های Ck. در نظر می گیریم x=(x1,x2,…,xn). با استفاده از قضیه بیزی، می توانیم به دست آوریم:

$$p(C_k|x) = \frac{p(C_k)p(x|C_k)}{p(x)} \propto p(C_k)p(x|C_k) = p(C_k, x_1, x_2, ..., x_n)$$

احتمال مشترک را می توان به صورت زیر نوشت:

$$\begin{aligned} & p(C_k, x_1, x_2, ...x_n) \\ &= p(x_1|x_2, ..., x_n, C_k) \cdot p(x_2, ..., x_n, C_k) \\ &= p(x_1|x_2, ..., x_n, C_k) \cdot p(x_2|x_3, ..., x_n, C_k) \cdot p(x_3, ..., x_n, C_k) \\ &= p(x_1|x_2, ..., x_n, C_k) \cdot p(x_2|x_3, ..., x_n, C_k) ... \cdot p(x_n|C_k) \cdot C_k \end{aligned}$$

فرض کنید تمام ویژگی های x از یکدیگر مستقل هستند، می توانیم به دست آوریم:

$$p(x_1|x_2, ..., x_n, C_k) = p(x_1|C_k)$$

بنابراین، فرمول را می توان به صورت زیر نوشت:

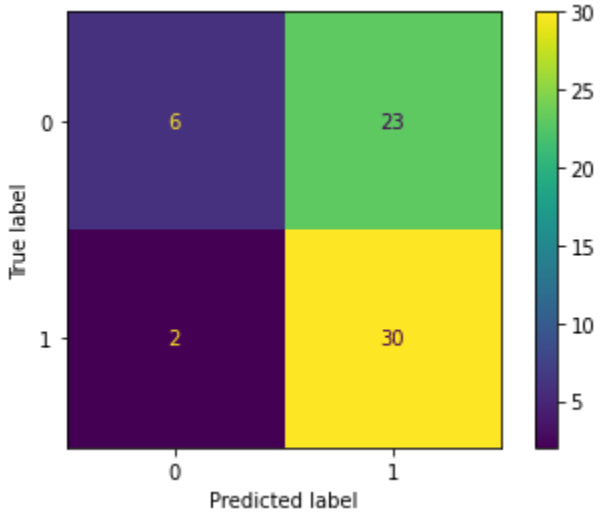
$$
\begin{aligned}
& p(C_k|x_1, x_2, ..., x_n) \\
\propto\ & p(C_k, x_1, x_2, ...x_n) \\
=\ & p(x_1|C_k) \cdot p(x_2|C_k) ... \cdot p(x_n|C_k) \cdot p(C_k) \\
=\ & p(C_k) \prod_{i=1}^{n} p(x_i|C_k)
\end{aligned}
$$

بنابراین، این فرمول نهایی برای طبقه بندی کننده بیز است.

در Gaussian Naive Bayes، مقادیر پیوسته مرتبط با هر ویژگی فرض می‌شود که بر اساس توزیع گاوسی توزیع می‌شوند. توزیع گاوسی را توزیع نرمال نیز می گویند.
Multinomial Naive Bayes: بردارهای مشخصه فرکانس هایی را نشان می دهند که با آن رویدادهای خاصی توسط یک توزیع multinomial ایجاد شده اند. این مدل رویدادی است که معمولاً برای طبقه بندی اسناد استفاده می شود.
Bernoulli Naive Bayes: در مدل رویداد چند متغیره برنولی، ویژگی‌ها بولین‌های مستقل (متغیرهای باینری) هستند که ورودی‌ها را توصیف می‌کنند. مانند مدل چندجمله‌ای، این مدل برای کارهای طبقه‌بندی اسناد محبوب است، که در آن از ویژگی‌های وقوع اصطلاح دودویی (یعنی کلمه در یک سند رخ می‌دهد یا نه) به جای فرکانس‌های اصطلاحی (یعنی فراوانی یک کلمه در سند) استفاده می‌شود.

2 و3 و4 و5 -
برای این دیتاست اول بررسی کردم که داده ی null ی وجود دارد یا نه که اصلا وجود نداشت. در ادامه روی داده ها minmax scale زدم که کار مدل با داده ها راحت تر باشه و بعد 3 ویژگی گفته شده در سوال را در نظر گرفتیم به عنوان داده های اصلی مدل. 80 درصد داده ها را برای آموزش و 20 درصد دیگه برای تست در نظر گرفتم.

در ادامه مدل Gaussian naive bayes را بدون استفاده از پکیج های آماده پیاده سازی کرده و بعد روی داده های آموزشی fit کردم و درصد درستی مدل ما روی داده های تست 59% بود و شکل زیر ماتریس confusion را برای این مدل نشان میدهد:



که می بینیم مدل ما اکثرا داده ها را 1 پیش بینی میکند که این اصلا خوب نیست و خود مدل هم فقط یکم بهتر از شانس عملکرد داشته و در زیر هم نتیجه معیار های دیگه را که بررسی کردم می بینیم:

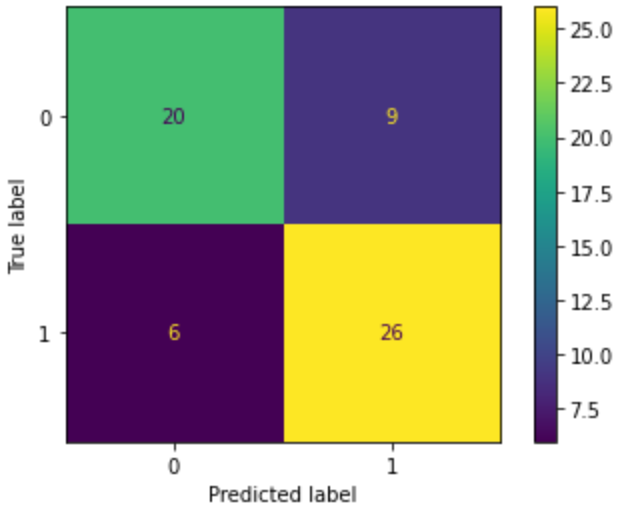
F1 = 0.71

Precision = 0.57

Recall = 0.94

می بینیم که Precision مدل ما بسیار پایین است.

در ادامه مدل Gaussian naive bayes را با استفاده از پکیج های آماده پیاده سازی کرده و بعد روی داده های آموزشی fit کردم و درصد درستی مدل ما روی داده های تست 75% بود و شکل زیر ماتریس confusion را برای این مدل نشان میدهد:



که می بینیم مدل ما با درصد خوبی هر دو کلاس را پیش بینی میکند .

در زیر هم نتیجه معیار های دیگه را که بررسی کردم می بینیم:

F1 = 0.78

Precision = 0.74

Recall = 0.81

می بینیم که تمام معیار ها بجز recall پیشرفت بسیار خوبی داشتند.