دانشکده مهندسي برق

گزارش کار پروژه

نام درس

شناسایی آماری الگو

نام دانشجو

پویا احمدپور

استاد:

دکتر محمدرضا دلیری

بهمن ماه 1400


بسم الله الرحمن الرحيم

چکیده

در ایـن مطلـب مبنـای تحقیـق، بررسـی و یـافتن روش‌هـای بهینـه و مناسـب جهـت ایجـاد یـک سیسـتم بـا قابلیـت تفکیـک بیمـاران هپاتیـت بـا احتمـال فـوت بـالا و احتمـال ادامـه‌ی حیـات آن‌هـا می‌باشـد. در ایـن گـزارش بـه شـرح مختصـری از بیمـاری هپاتیـت و اطلاعـات دیتاسـت مربـوط بـه آن، رویکردهـای انتخـاب شـده بـرای پیش‌پـردازش دیتـا، آمـوزش مـدل و کلاسبندی و دلایل انتخاب آن‌ها خواهیم پرداخت.

واژه‌هاي كليدي: دیتاست، پیش‌پردازش، کلاسبندی

فهرست مطالب

فهرست اشکال

فهرست جداول

فهرست علائم اختصاري

فصل 1:

مقدمه

1-1- مقدمه

هپاتیت یا Hepatitis یک بیماری شایع در میان مردم جهان است، که به التهاب در بافت کبد اطلاق می‌شود. سازمان جهانی بهداشت(WHO)، تخمین می‌زند که حدود 354 میلیون نفر در جهان، با نوع مزمن این بیماری زندگی می‌کنند.

یکی از چشم‌اندازهای مهم در علم پزشکی، تخشیص میزان خطر بیماری و احتمال تهدید شدن جان شخص به علت بیماری، با توجه به علائم حیاتی و نتایج آزمایشات وی می‌باشد. چرا که می‌تواند به انتخاب شیوه‌ای متفاوت برای درمان و تغییر روند رسیدگی به بیمار منجر شود. لذا مطلب پیش رو، با هدف یافتن راهی مناسب، با دقتی مقبول، به بررسی روش‌های شناسایی آماری الگو، برای دستیابی به این مهم می‌پردازد.

اطلاعات و دیتاست موجود در این مطلب، از سایت UCI Machine Learning دریافت شده و مبنای انتخاب رویکردهای موجود در این گزارش، این دیتا می‌باشد. همچنین برای ایجاد و تنظیم یک برنامه‌ی مناسب، جهت تفکیک بیماران مذکور، از نرم‌افزار متلب استفاده شده، که در ادامه، به تفصیل درباره‌ی نحوه‌ی عملکرد برنامه در آن و نتایج آن خواهیم پرداخت.

فصل 2:

مروری بر منابع

2-1- کارهای پیشین

یادگیری ماشین از موضوعات رو به رشد در دنیای امروز می‌باشد. تحقیق و رویکرد کلاسبندی در این پروژه نیز بر مبنای همین موضوع می‌باشد. در سال‌های اخیر مطالعات و کارهای ارزشمندی در این حوزه انجام شده است که اشاره به همه‌ی این موارد از حوصله‌ی مطلب خارج است. اگرچه می‌توان به تعدادی از کارهای مشابه که در این حوزه، بر روی دیتاست هپاتیت مورد نظر ما انجام شده، اشاره کرد.

برای مثال Michael L. Raymer، Travis E Doom و دیگران، سال 2003 در مقاله‌ای با رویکرد استفاده از Hybrid Bayes Classifier و انتخاب ویژگی‌های مناسب برای کلاسبندی، به دقت 79.4٪ رسیدند.[1]

همچنین Xiaoli Z.Fern و Carla Brodley در همان سال با ارائه‌ی یک الگوریتم boosting-style موفق به دریافت، 85.4٪ دقت، با استفاده از متد Bossting Lazy decision Trees در کلاسبندی شدند.[2]

اطلاعات برچسب‌گذاری شده در این پروژه از سایت UCI machine learning بدست آمده که توسط G.Gong از دانشگاه Carnegie-Mellon اهدا و قرار داده شده است.

فصل 3:

مواد و روش ها

3-1- مقدمه

برای رسیدن به تفکیک مناسبی از وضعیت بیماران، و یک دقت مناسب در کلاس‌بندی، از دسته‌بندی دیتاست، و موارد پیش‌پردازش برای رسیدن به یک دیتاست مناسب برای کلاسیفایر شروع می‌کنیم. سپس از روش‌هایی برای ارزش‌گزاری . انتخاب فیچرهای مناسب استفاده کرده و با یک دسته‌بندی مجدد و به هم ریختن مناسب داده، عملکرد مدل را که به روش‌های مختلف آموزش داده‌ایم، ارزیابی خواهیم کرد.

3-2- درباره‌ی داده‌ها

داده‌های مورد استفاده در این پروژه شامل 19 ستون از ویژگی‌های مختلف است که از 155 بیمار بدست آمده‌اند. برخی از این داده‌ها دارای نقص می‌باشند و مواردی از آن‌ها در دیتاست قید نشده که در صورت حذف شدن این موارد، دیتاست را به 66 مورد کاهش می‌دهند. لذا از حذف این موارد پرهیز کرده و مقادیر میانگین کل ستون را جایگزین موارد مجهول می‌کنیم تا بدون تاثیر بر مدل، بتوانیم از موارد دیتای موجود آن‌ها استفاده کنیم. می‌توان در نظر داشت که وجود این موارد قطعاً عملکرد کلاسیفایر را بهبود خواهد بخشید.

3-2-1- مروری بر جزئیات ویژگی داده‌ها

- Age

سن بیماران مقادیر متفاوتی را شامل میشود که از 7 تا حدود 80 سال را در بر می‌گیرد.

- Sex

جنسیت بیماران به دو گروه مرد و زن تقسیم شده است. عدد 1 به عنوان مرد و 2 به عنوان زن در نظر گرفته شده است.

- Steroid

مصرف داروی استروئیدی برای درمان هپاتیت.

- Antivirals

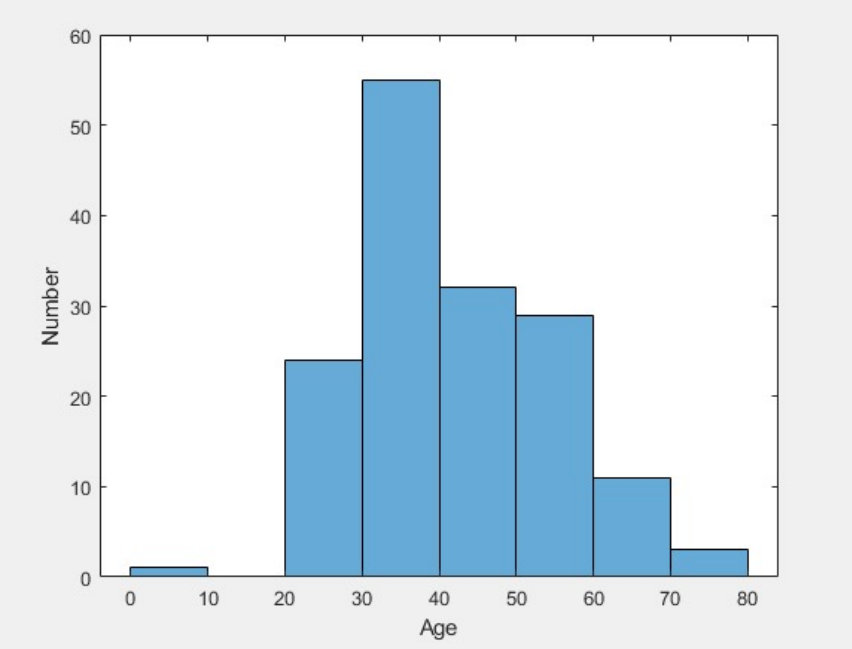
استفاده یا عدم استفاده از داروهای Antiviral که یک کلاس دارو برای درمان عفونت‌های ویروسی

هستند.

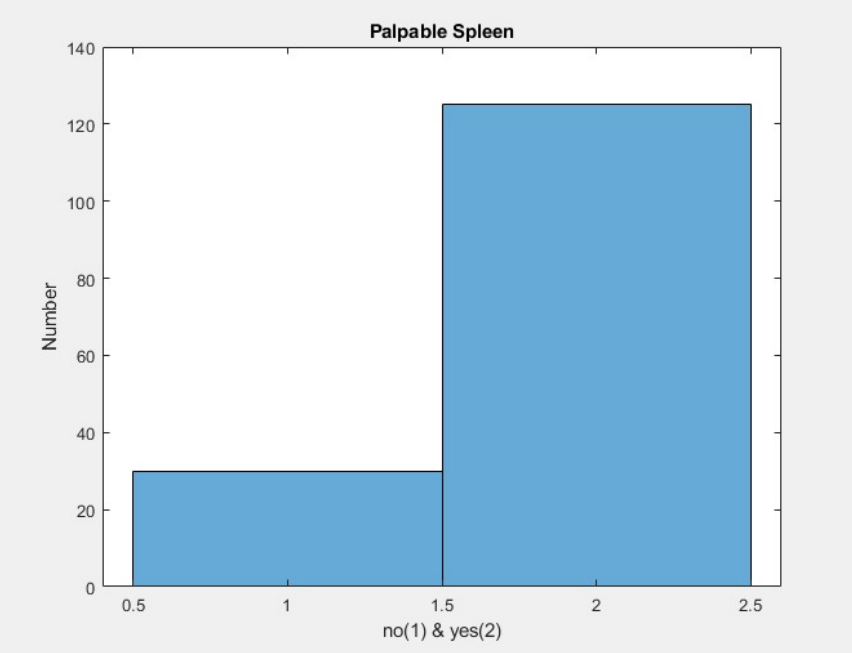
- Fatigue
احساس خستگی مفرط در بیماران.
- Malaise
احساس بیماری و ناخوشآیند در بیمار.
- Anorexia
داشتن بی‌اشتهایی.
- Big Liver
بزرگ‌تر شدن کبد نسبت به حالت نرمال.
- Firm Liver
سفت شدن بافت کبد به دلیل بیماری.
- Palpable Spleen
قابل لمس شدن طحال که در حالت عادی قابل دسترسی و لمس نیست.
- Spiders
بوجود آمدن خال‌ها یا نمایان شدن رگ‌های عنکبوتی که نشانه‌ای از بیماری کبدی هستند.
- Ascites
تجمع مایع در شکم که موجب برآمدگی ان می‌شود.
- Varices
واریس یا تورم و التهاب در رگ‌ها.
- Bilirubin
میزان بیلی روبین که از آزمایش خون بدست می‌آید، با مقادیر عددی مشخص می‌شود. لازم به ذکر است که این ماده تولید کسیه‌ی صفرا در مجاورت کبد می‌باشد.
- ALK phosphate
مقدار آلکین فسفاتاز که با آزمایش خون بدست آمده است و شامل مقادیر عددی است.
- Albumin
میزان آلبومین نیز عددی بوده و از آزمایش خون بدست آمده است.
- Protime
تست مدت زمان لخته شدن خون می‌باشد که طبیعتاً مقادیر متفاوت عددی را شامل می‌شود.

موارد غیرعددی در ویژگی‌های فوق‌الذکر شامل دو مقدار 1 به معنی منفی بودن نتیجه و 2 به معنی مثبت بودن نتیجه می‌باشند.

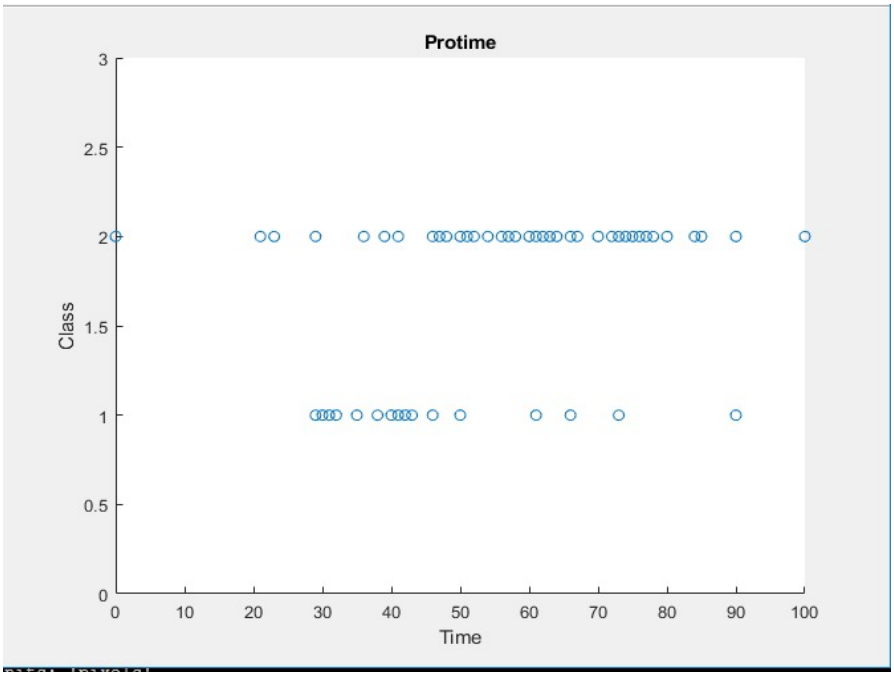
برای درک بهتر از نوع و توزیع داده‌ها، برخی از آن‌ها را در چند شکل ترسیم می‌کنیم.



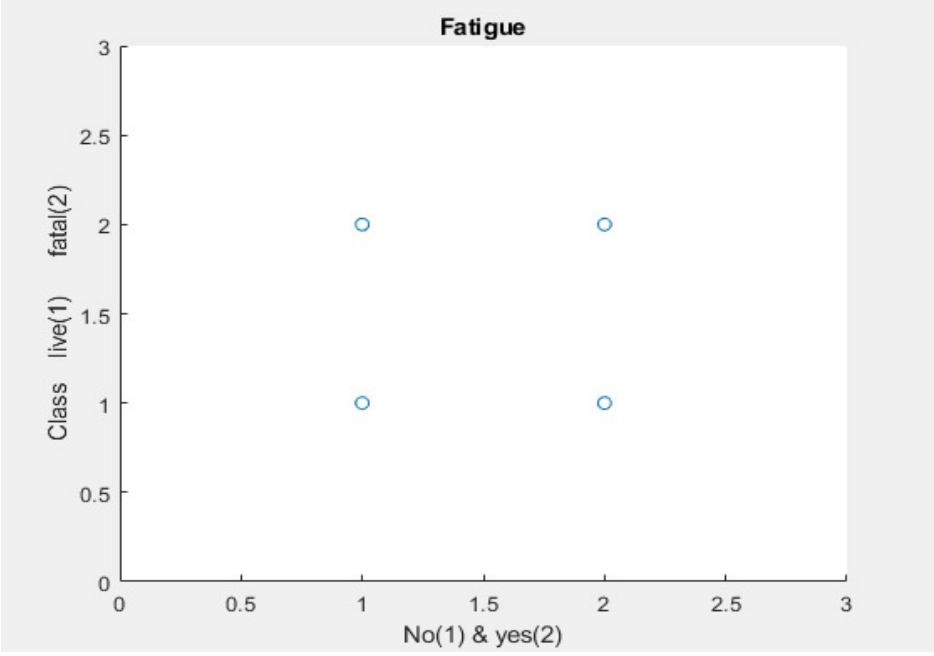
شکل 3-1. توزیع سن اشخاص بر حسب تعداد بیماران



شکل 3-2. تعداد بیمارانی که طحال قابل لمس داشتند



شکل 3-3 . نمودار پراکندگی مدت زمان مورد نیاز برای لخته شدن خون در بیمارانی که فوت کردند(کلاس2) و یا زنده ماندند(کلاس1).



شکل 3-4. نمودار پراکندگی وجود خستگی مفرط در بیمارانی که زنده ماندند(کلاس1) و بیمارانی که فوت کردند(کلاس2).

هدف از نشان دادن نمودارهای فوق، ارائه‌ی یک دید کلی به خواننده از نوع توزیع و پراکندگی داده‌ها می‌باشد. مشخص است که برخی ویژگی‌ها دارای مقادیر پیوسته و عددی متفاوت بوده، درحالی که برخی دیگر فقط با فرمت 1 و 2، و به نوعی باینری می‌باشند. این موضوع استفاده از رویکردها و ابزاری را که صرفا با فرض توزیع خاصی از داده کار می‌کنند را مشکل‌ساز خواهد کرد.

3-3- پیش پردازش

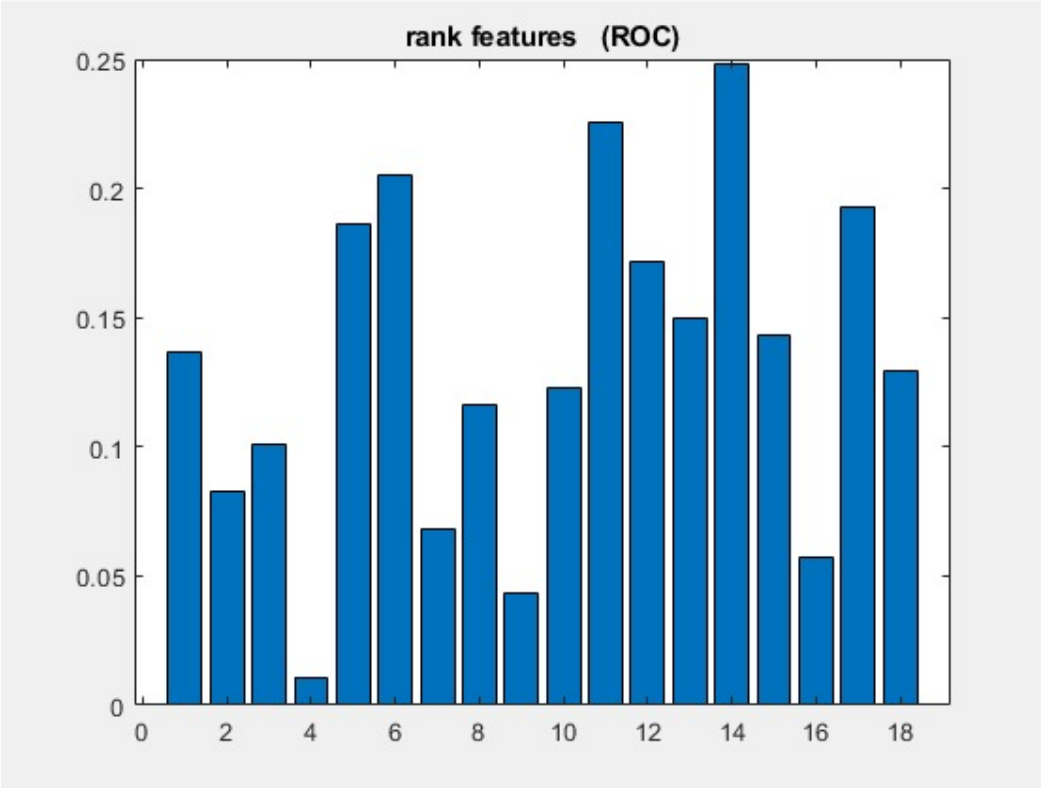
در ابتدای امر، داده‌های ورودی به صورت text یا string هستند که باید به مقادیر عددی تبدیل شوند. سپس داده‌ها را به دو دسته‌ی train و test تقسیم کرده و ستون ویژگی‌ها را از برچسب‌ها جدا می‌کنیم. یادآور می‌شویم که تعداد قابل توجهی از داده‌ها دارای مقادیر مجهول و به اصطلاح Nan، می‌باشند. برای اصلاح این مورد، حذف داده‌ها به دلیل محدوودیت تعداد نمونه‌ها ممکن نیست، لذا مقدار میانگین داده‌های training را محاسبه کرده و به جای موارد مجهول، هم در مورد داده‌های training و هم داده‌های test قرار می‌دهیم.
با توجه به تفاوت بسیاری که در میان داده‌های عددی، مقدار و توزیع آن‌ها وجود دارد، نیاز به نرمال‌سازی داده‌ها را برای ما مبرم می‌سازد. نتیجه‌ی این نرمالیزه کردن، به وضوح در نتایج کلاس‌بندی قابل مشاهده می‌باشد. اگرچه در میان دو روش نرمال سازی Minmax و Z-Score تفاوتی دیده نمی‌شود.
در بخش پیش‌پردازش داده‌ها، حذف Outliers یا داده‌های پرت نیز کاملاً مطرح است. زیرا می‌توانند، تخمین کلاسیفایر را بایاس کنند، اما در نتایج بدست آمده در 500 تکرار، مشاهده می‌شود که حذف یا جایگزینی این داده‌ها با مقدار میانگین، به شدت موجب کاهش، دقت می‌شود. دو دلیل این مسئله را می‌توان، وجود گوناگونی ویژگی‌های حیاتی و درمانی بیماران و عدم امکان حذف داده‌ی تست به عنوان داده‌ی پرت عنوان کرد. چرا که در موضوع مورد بررسی ما کلاسیفایر باید بتواند هر بیمار را با هر ویژگی که می‌تواند داده‌ی پرت لحاظ شود، به عنوان یک نمونه تفکیک کند.

3-4- استخراج ویژگی

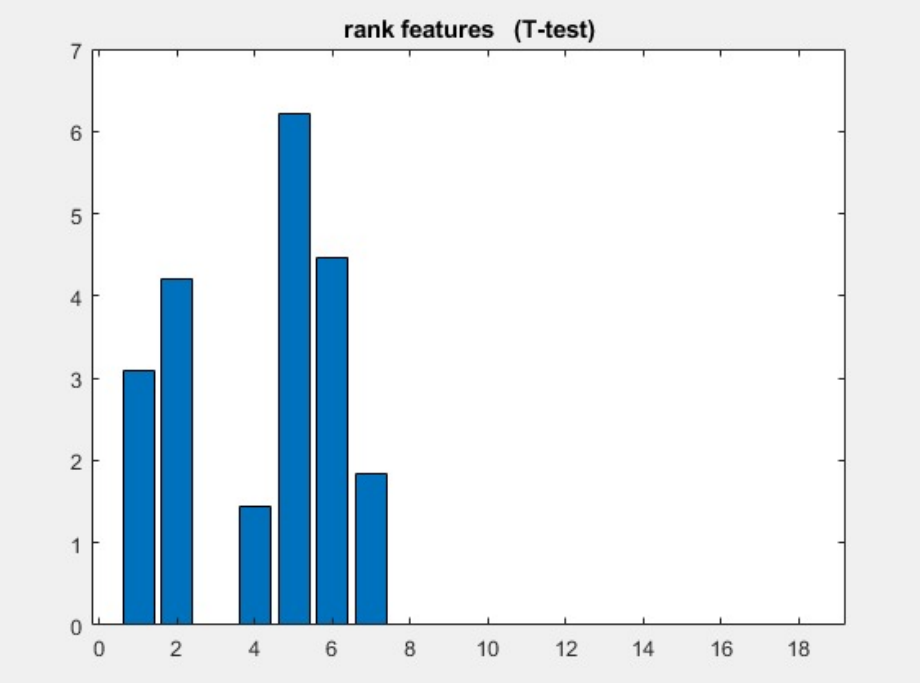
همانطور که پیشتر نیز اشاره شد، داده‌های ما برچسب گذاری شده و حاوی نتایج آزمایشات و اطلاعات مرتبط با بیماری هپاتیت، در بیماران می‌باشد. لذا داده‌ها خام نبوده و نیازی به استخراج ویژگی از آن‌ها وجود ندارد.

3-5- انتخاب ویژگی

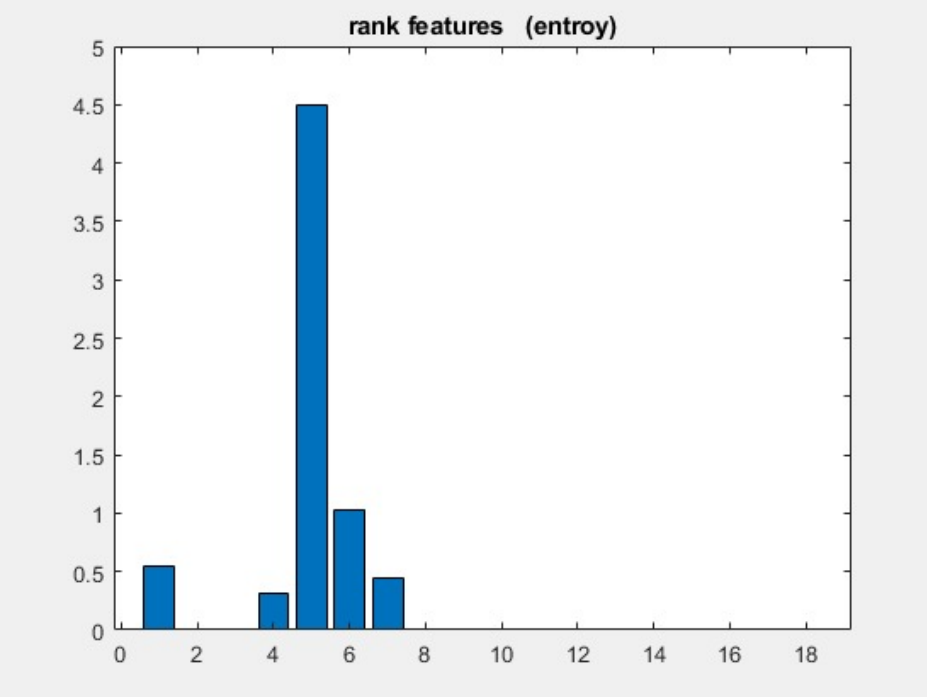
پیش‌تر به محدودیت تعداد ویژگی‌ها اشاره شـد و مـی‌دانیم کـه حـذف ویژگی‌هـا می‌توانـد تـاثیر منفـی در کلاسبندی داشته باشد. اگرچه این امر توسط روش Rank features تست شده، و مشاهده می‌شود که حذف فقط تا دو ویژگی بر اساس رویکرد اسکالر، بعضاً می‌تواند تاثیر مثبت اندکی بر نتـایج کلاسـبندی مـا داشـته باشد.



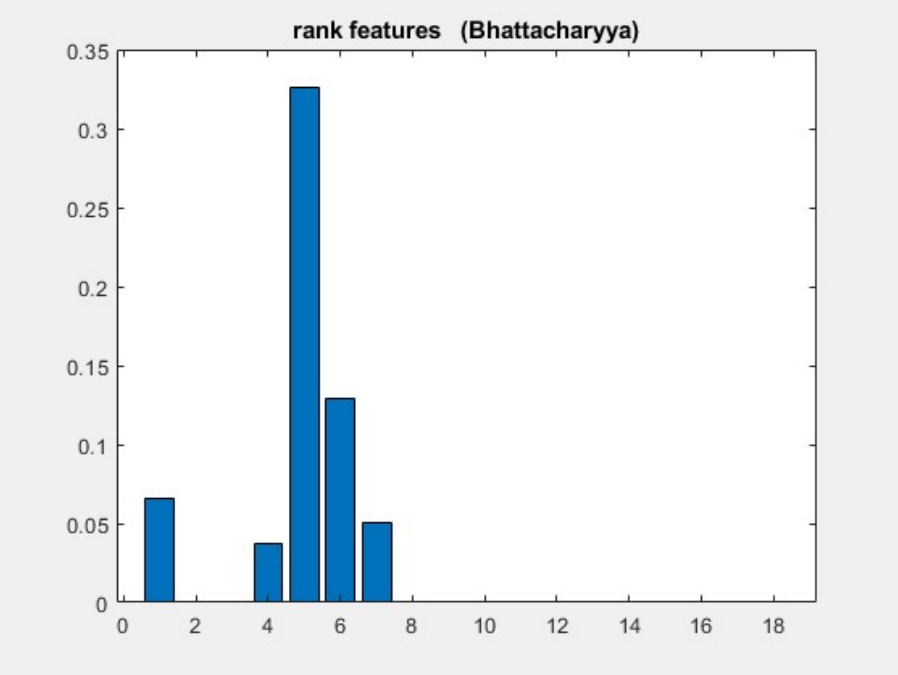
شکل 3-5. رتبه‌بندی ویژگی‌ها با معیار ROC



شکل 3-6. رتبه‌بندی ویژگی‌ها براساس معیار T-test
(مشاهده می‌شود، که بسیاری از داده‌ها درهم رفتگی زیاد و میانگین کمی دارند)



شکل 3-7. رتبه‌بندی ویژگی‌ها بر اساس معیار آنتروپی یا دیورژانس



شکل 8-3. رتبه‌بندی ویژگی‌ها براساس کمترین میزان خطای تفکیک در کلاسبندی

در شکل‌های فوق رتبه‌بندی ویژگی‌ها اطلاعات مفیدی در اختیار ما قـرار می‌دهنـد، بـا ایـن حـال، انتخـاب ویژگی را بر عهده‌ی ابزار PCA می‌گذاریم تا از معیار چند جانبه‌ی آن برای انتخاب ویژگی بهره‌منـد شـویم. می‌دانیم که حذف ویژگی‌ها برای مورد مزبور به دلیل محدودیت تعداد چندان مد نظر نیست، لـذا در تسـت برنامه مشاهده می‌شود PCA نیز نهایتاً یک تا دو ویژگـی را از فضـای ویژگـی را بـا Threshold برابـر 99٪ حـــذف مـــی‌کنیم کـــه بهتـــرین بـــازخورد را در نتـــایج کلاســـبندی بدســـت می‌دهـــد. اگرچه Uncorrelated بودن داده‌ها از یکدیگر نیز تا حدودی مشهود می‌باشد. Uncorrelated بـودن ویژگـی داده‌ها توسط توابع آماده در Matlab مشاهده شده و از کوچک بودن مقادیر Correlation بـین آنهـا مطلـع هستیم.

	1	2	3	4	5	6
1	1	-0.0837	-0.2178	-0.0190	-0.2459	-0.1630
2	-0.0837	1	-0.1170	-0.0904	-0.0796	-0.0336
3	-0.2178	-0.1170	1	0.0787	0.2923	0.2856
4	-0.0190	-0.0904	0.0787	1	-0.0582	-0.0777
5	-0.2459	-0.0796	0.2923	-0.0582	1	0.5886
6	-0.1630	-0.0336	0.2856	-0.0777	0.5886	1
7	0.0506	-0.0110	0.1206	-0.1239	0.3776	0.5979
8	-0.0237	-0.2089	0.2603	0.0411	0.0988	0.0581
9	-0.0603	-0.0389	0.0897	0.0922	0.2807	0.1069
10	-0.0319	-0.1063	0.0775	-0.1117	0.2197	0.0661
11	-0.1658	-0.0841	0.0090	-0.1913	0.3401	0.2690
12	-0.1642	0.1074	-0.0274	-0.1373	0.2739	0.3567
13	-0.0947	0.0226	-0.0225	-0.1482	0.1912	0.1946
14	0.1013	-0.0611	-0.1445	0.1278	-0.2627	-0.3539
15	0.0269	0.1011	-0.1335	0.0954	-0.1163	-0.2247
16	0.1066	0.0026	-0.0240	0.0451	-0.2050	-0.1947
17	-0.3110	-0.0309	0.2641	-0.0990	0.3246	0.3713
18	-0.1810	-0.0232	0.0748	-0.0696	0.2385	0.2732

شکل 3-9. قسمتی از ماتریس Correlation از ویژگی‌های بین داده‌های Training

3-6- ارزیابی و طبقه‌بندی

مرحله‌ی پایانی، پروژه کلاسبندی و تفکیک داده‌ها می‌باشد. برای کلاس‌بندی داده‌ها از چهار روش KNN، SVM، Logistic Regression و Decision Tree می‌باشد.

پیش‌تر به تفاوت در توزیع داده‌ها، اشاره کردیم. این امر استفاده از مدل‌هایی مانند Navie Bayes را که با پیش فرض نرمال بودن داده‌ها، کار کلاس‌بندی را انجام می دهند، دچار مشکل می‌سازد. این مسئله افزون بر بی‌اطلاعی ما از تابع توزیع احتمال، میتوانند دلایلی برای انتخاب روش‌های غیر پارامتری و مناسب دیگر باشند.

- **SVM**

این روش که با یافتن یک بردار مناسب به تفکیک داده‌ها در فضای ویژگی می‌پردازد، از رویکردهای مناسب برای کلاسبندی روی داده‌های ما می‌باشد.

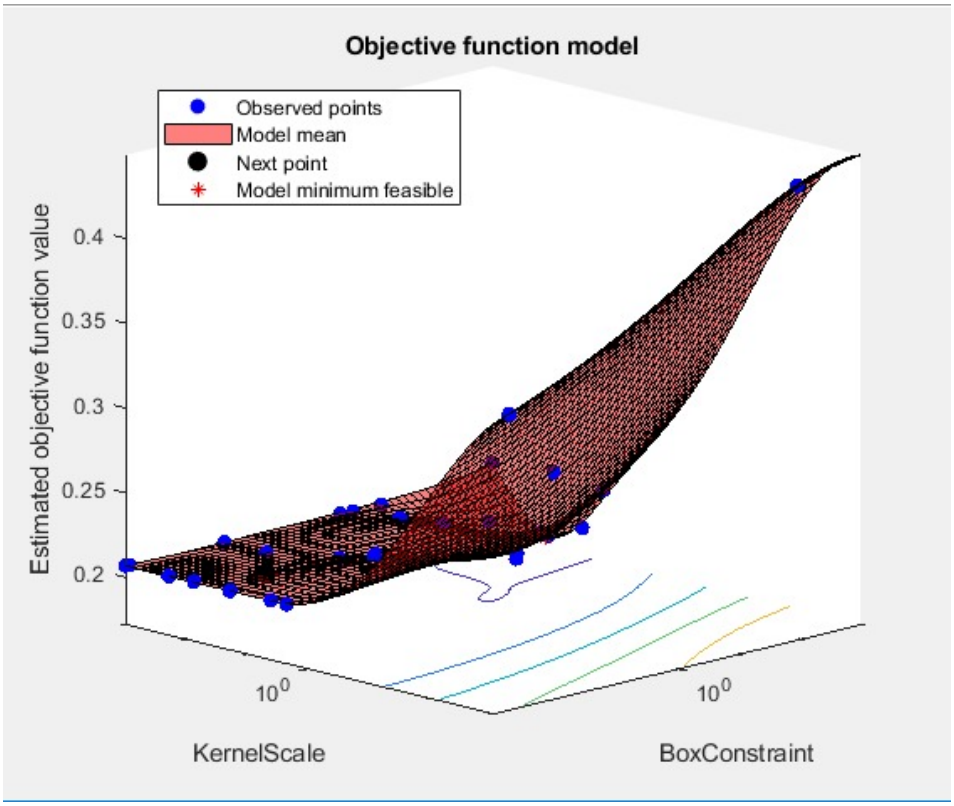
در ابتدای امر باید برای مدل SVM یک کرنل مناسب بیابیم که بتوان توسط آن به بهترین دقت از تفکیک

داده‌ها رسید. با آزمودن، سه کرنل مختلف Gaussian، Linear و Polynomial و مقایسه‌ی آن‌ها در می‌یابیم که بهترین نتایج از کرنل Gaussian حاصل می‌شود. مقایسه‌ی عملکرد این سه کرنل را می‌توان در جدول زیر مشاهده نمود.

Linear	Polynomial	Gaussian
81.0%	81.2%	**85.5%**
80.4%	81.1%	84.7%

جدول 3-1. مقایسه‌ی دقت SVM به ازای سه کرنل متفاوت

الگوریتم مورد نظر ما برای کلاس‌بندی دارای پارامترهایی است، بعد از انتخاب کرنل می‌توان آنها را به روش اتوماتیک با ابزار Optimize Hyper Parameters یافت.



شکل 3-10. بهینه‌سازی پارامترهای مدل SVM توسط متلب

در 5 تکرار، مقادیر متفاوتی را برای الگوریتم، توسط ابزار فوق‌الذکر، بدست می‌آوریم. سپس مقادیر بدست

آمده را به صورت دستی در الگوریتم وارد کرده و میانگین دقت آنها را به ازای 400 تکرار برای هر کدام، توسط برنامه محاسبه می‌کنیم. نتایج حاصل در جدول زیر نوشته شده‌اند.

Box Constraint	Kernel Scale	Accuracy
0.78147	2.4823	76%
67.034	23.4823	82.5%
0.0010543	0.18737	76.9%
0.42749	4.2557	**84.8%**
0.0010419	0.095452	81.5%
Default	Auto	82.2%

جدول 2-3. پارامترهای الگوریتم SVM بدست آمده در متلب و دقت عملکرد آن به ازای مقادیر مختلف در 400 تکرار

لازم به ذکر است که پارامتر BoxConstraint نوعی معیار برای لحاظ کردن ارزش نمونه‌های Training می‌باشد که توسط الگوریتم misclassify خواهند شد، تا الگوریتم به بهترین تفکیک از دو کلاس دست یابد. به مطابق انتظار پارامترهای سطر چهارم، به عنوان مقادیر بهینه به الگوریتم SVM داده می‌شود که حامل دقت مناسبی از آن می‌باشد.

- **Tree decision**

یک انتخاب مناسب برای کلاس‌بندی داده‌های ما با توجه به خروجی ابزار App در متلب، می‌تواند، درخت تصمیم باشد. این کلاسیفایر در حالت پیش فرض میانگین دقتی برابر با 77٪ دارد، که مقدار مناسب و قابل توجهی برای رقابت با SVM نیست. لذا از ابزار مشابهی که برای بهینه‌سازی پارامترهای SVM استفاده کردیم، برای درخت تصمیم نیز استفاده می‌کنیم. پارامتر بدست آمده از این ابزار تحت عنوان MinleafSize، در 5 تکرار محاسبه شده است. پس از آن مقادیر مذکور را به صورت دستی به الگوریتم داده و میانگین دقت را در 400 تکرار، محاسبه کرده‌ایم که در جدول تحریر شده است.

Min Leaf Size	Accuracy
25	81.2%
11	81.4%
13	**82.9%**
6	80.6%
24	81%

جدول 3-3. پارامترهای بهینه برای برای درخت تصمیم و دقت حاصل از کلاسبندی با آنها

از مقادیر جدول بالا 13 را به عنوان پارامتر بهینه برای درخت تصمیم استفاده می‌کنیم. که نتیجه‌ای به مراتب بهتر از حالت پیش فرض آن دارد.

- **Logistic Regression**

این کلاسیفایر، عملکرد مناسب و مقبولی را تفکیک در بخش Apps نرم‌افزار ارائه می‌دهد. ابزاری که در متلب برای کلاس‌بندی استفاده کردیم مدل Linear می‌باشد که خود حامل دو مدل SVM و Logistic Regression می‌باشد. انتخاب یکی از از این دو الگوریتم، تحت عنوان Learner برای کاربر، به صورت دستی امکان پذیر می‌باشد. برای انتخاب کلاسیفایر بهینه و بدست آوردن پارامتر مناسب آن، از ابزار Optimize Hyper Parameters که پیش‌تر اشاره شد استفاده می‌کنیم. در 5 تکرار، اجازه می‌دهیم تا نرم‌افزار پارامتر و الگوریتم بهینه را انتخاب کند. سپس نتایج را به صورت دستی به مدل می‌دهیم و در 400 تکرار میانگین دقت را محاسبه می‌کنیم. نتایج در جدول زیر آمده است.

Lambda	Learner	Accuracy
0.070583	**Logistic**	**85.6%**
0.029303	SVM	81.1%
0.25185	SVM	83.6%
0.089269	Logistic	85.4%
0.086219	SVM	82.2%

جدول 4-3. نتایج بهینه‌سازی مدل Linear شامل دقت و تعیین پارامتر

با در نظر گرفتن نتایج فوق، الگوریتم Logistic Regression با پارامتر تعیین شده، از سطر نخست جدول 4، انتخاب و به صورت دستی برای مدل تعیین می‌شود.

- **KNN**

آخرین مدل پیشنهادی برای تفکیک داده‌های هپاتیت، که به عنوان یک روش غیرپارامتری برای ویژگی‌های داده‌های ما مناسب به نظر می‌رسد، K-Nearest neighbor می‌باشد. این کلاسیفایر، شامل دو پارامتر مهم فاصله و تعداد همسایه می‌باشد که باید به طور بهینه انتخاب شوند. با استفاده از ابزار Optimization متلب، در 16 تکرار، فواصل و تعداد همسایه‌ی بهینه برای تفکیک را محاسبه می‌کنیم. سپس هر مقدار را به صورت دستی وارد الگوریتم مدل می‌کنیم تا در 100 تکرار، میانگین دقت کلاسیفایر را برای ما محاسبه کنند. نتایج مزبور در جدول زیر آمده‌اند.

Number of Neighbors	Distance	Accuracy
23	**Correlation**	**84.6%**
20	Correlation	83.6%
15	Correlation	83.5%
7	Chebychev	82%
5	Chebychev	82%
1	Cosine	82%
22	Cosine	83%
19	Cosine	82%
19	Cosine	82.1%
5	Minkowski	83.9%
6	Euclidian	83.9%
9	CityBlock	79%

جدول 5-3. نتایج بهینه‌ی بدست آمده توسط متلب به ازای فواصل و تعداد مختلف همسایگی و دقت عملکرد آن‌ها در 100 تکرار

از میان مقادیر فوق، ردیف اول جدول 5 را به عنوان بهترین انتخاب، در نظر می‌گیریم. علی‌رغم اینکه کلاسیفایرهایی که پیش‌تر به آن‌ها اشاره شد، در روش‌های Cross-Validation برای آموزش مدل و ارزیابی آن، عملکرد بهتری از خود نشان نمی‌دادند، اما KNN نسبت به این رویکرد، پاسخ بهتر و دقت مناسب‌تری را ارائه می‌دهد. از آنجا که داده‌ها در برنامه، دو بار به هم ریخته شده‌اند، نیازی به انتخاب رندوم داده‌های Validation نیست. لذا مستقیماً سراغ روش K-Fold Cross validation می‌رویم، و به ازای تعداد مختلف Foldها، عملکرد سیستم را ارزیابی می‌کنیم تا به یک مقدار بهینه برسیم. نتایج مذکور و دقت حاصل از

100 تکرار هر کدام، در جدول زیر قابل مشاهده هستند.

Folds Number	Accuracy
20	86.2%
10	85%
30	86%
80	**86.46%**

جدول 6-3. دقت حاصل از روش Cross-Validation به ازای تعداد مختلف Foldها

علی‌رغم اینکه حالت 80-Folds دقت بیشتری را ارائه می‌دهد، اما تفاوت آن با حالت 20-Folds چندان متفاوت نیست. در عین حال زمان مورد نیاز برای حالت 80-Folds چندین برابر حالت دوم است، که شاید چندان مطلوب نباشد. لذا در همین نقطه از ارزیابی Leave-one-Out نیز صرفه نظر کرده و به ارزیابی 20-Fold برای مدل KNN بسنده می‌کنیم.

فصل 4:

نتایج و تفسیر آنها

4-1- بحث و نتیجه‌گیری

در این بخش به تفسیر و نتیجه‌گیری از نتایج بدست آمده در قسمت های قبل می‌پردازیم. یادآور می‌شویم که برای پیش پردازش داده‌های Training در برنامه از نرمال‌سازی داده‌ها استفاده کردیم. از آنجا که داده‌ها از نظر Range و کمیت با هم کاملاً متفاوت هستند، تاثیر نرمال کردن داده‌ها کاملاً مطابق انتظار می‌باشد و آن را از ملزومات پردازش داده‌های هپاتیت می‌دانیم.

در جدول زیر نتایج حاصل از کلاس‌بندی همراه با نرمال‌سازی و بدون آن را در 100 تکرار، بدون هیچ پردازش دیگری می‌کنید.

Classifier	With Normalization	With Out Normalization
SVM	72.8٪	79.48٪
Logistic Regression	66.6٪	78.61٪
Decision Tree	74.6٪	77٪
KNN	79٪	83.1٪

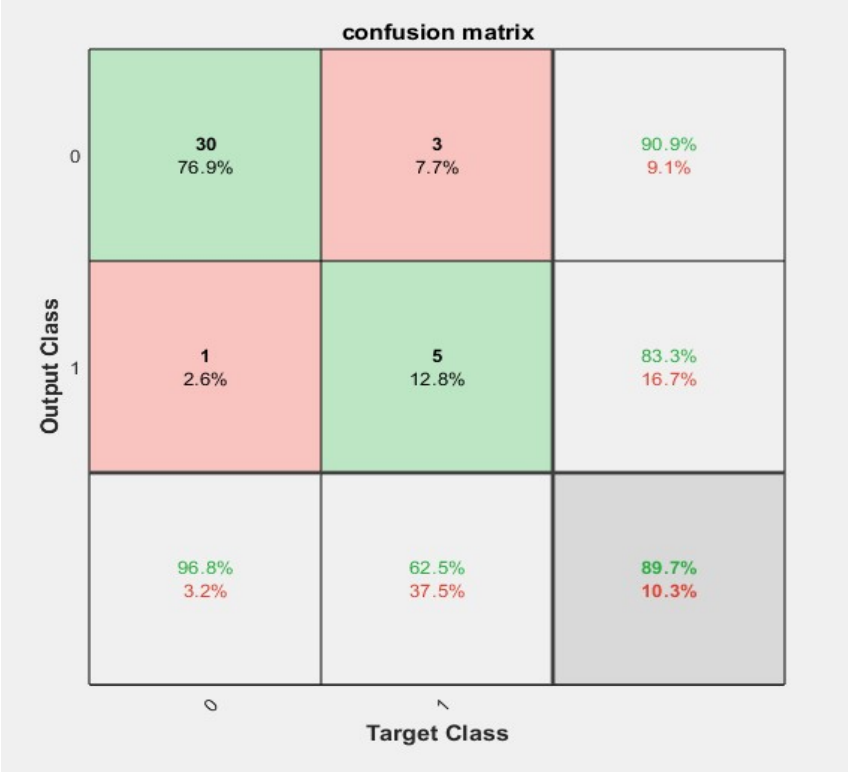
جدول 4-1. مقایسه‌ی نتایج حاصل از کلاسبندی همراه با نرمال‌سازی و بدون آن

برای کلاسبندی و تفکیک بیماران هپاتیت، از چهار کلاسیفایر استفاده کردیم، که با استفاده از ابزار متلب، بهینه شده‌اند و نتایج دقت نهایی آن‌ها در 500 تکرار جمع‌بندی و میانگین گرفته شده است، که در جدول زیر قابل مشاهده هستند. (تنها برای آموزش KNN از ارزیابی 20-Folds استفاده شده است).

Classifier	Final Obtained Accuracy
SVM	84.7%
Logistic Regression	85.4%
Decision Tree	82.7%
KNN	**86.1%**

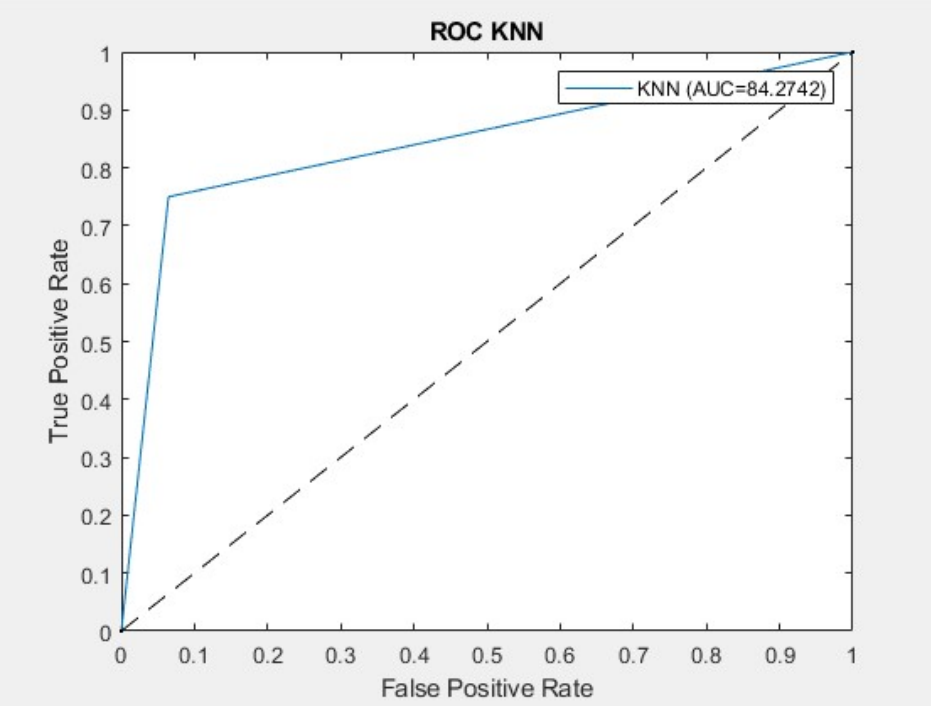
جدول 4-2 . مقایسه‌ی میانگین دقت کلاسیفایرها در 500 تکرار

دیده می‌شود که نتایج حاصل از KNN به عنوان یک روش غیرپارامتری، کاملاً مناسب برای داده‌های هپاتیت هستند. در شکل زیر Confusion Matrix مربوط به نتایج کلاس‌بندی توسط الگوریتم KNN را در یک اجرا مشاهده می‌کنید.



شکل 1-4. Confusion matrix برای کلاسیفایر KNN

قابل درک است که کلاسیفایر خروجی قابل قبولی دارد، که با Specificity برابر 62٪ و حساسیت 96٪ به دقت نهایی 89٪ رسیده است. این مسئله را می‌توان با نمایش منحنی ROC و محاسبه‌ی AUC نیز نمایش داد، که در شکل زیر ترسیم شده است.



شکل 2-4. منحنی ROC و مقدار AUC برای KNN

اگرچه آنچه گفته شد، کیفیت بالای KNN را برای تفکیک داده‌های ما نشان میدهد، اما بقیه‌ی کلاسیفایرها نیز دقتی نزدیک به KNN ارائه می‌دهند و می‌توان هنوز آن‌ها را به عنوان یک متد پیشنهادی برای تفکیک داده‌های بیماران هپاتیت استفاده کرد. ضمن اینکه شافل شدن دیتاست در هر بار اجرا موجب دقت کلاسیفایرها در هر اجرا می‌شود. و با توجه به این که در یک اجرا دقت میانگین مطرح نیست، ممکن است یکی دیگر از کلاسیفایرها بیشترین دقت را ارائه دهد.

فصل 5:

جمع‌بندي و پيشنهادها

5-1- مقدمه

در این مطلب، به پردازش داده‌های بیماران هپاتیت پرداختیم تا بتوانیم به یک تفکیک مناسب از بیماران با احتمال فوت بالا و بیمارانی که زنده می‌مانند برسیم. ابتدا مقادیر مجهول دیتاست را با مقادیر میانگین جایگزین کردیم و با توجه پراکندگی و تفاوت کمیت داده‌ها، آن‌ها را نرمال کردیم تا به تفکیک بهتری برسیم. در برنامه برای حذف و جایگزینی Outliers و یا داده‌های پرت اقدام شد، که با نتیجه‌ی مثبتی روبه‌رو نبود، لذا از اعمال آن صرف نظر شده است. پراکندگی و فواصل میانگین ویژگی داده‌ها از یکدیگر رتبه‌بندی و نمایش داده شد و در نهایت انتخاب ویژگی‌های مناسب برای کلاسبندی، بر عهده‌ی ابزار PCA با مرز تصمیم بیش از 99٪ واریانس، گذاشته شده است، که می‌تواند اثر مثبت اندکی بر دقت نتایج داشته باشد.
برای آموزش چهار مدل از تمام داده‌های Training استفاده شده، و با استفاده از ابزار متلب، پارامترهای بهینه برای کلاس‌بندی را انتخاب و به صورت دستی وارد الگوریتم کردیم. لازم به ذکر است که که تنها برای آموزش کلاسیفایر KNN از روش K-Fold Cross Validation استفاده، زیرا تاثیر بسزایی بر عملکرد بقیه‌ی کلاسیفایرها نداشته و صرفاً موجب کند شدن عملکرد برنامه می‌گردد.
در نهایت دقت کلاسیفایرها مقایسه و مشاهده می‌شود که روش غیرپارامتری KNN بهترین عملکرد را به طور میانگین با توجه رویکردی که ما پیش گرفتیم، ارائه می‌کند. اما، پیشنهاد می‌شود که پاسخ دیگر کلاسیفایرها نیز از نظر خارج نشوند، چرا که به ریختن داده‌ها در هر اجرا موجب تغییر دقت‌ها به صورت لحظه‌ای می‌شود و بعضاً دیگر کلاسیفایرها می‌توانند دقت بهتری را ارائه دهند.

5-2- محتوا

5-2-1- جمع‌بندي

پیش‌تر به برخی کارهای کلاس‌بندی که بر روی داده‌های هپاتیت انجام شد، اشاره کردیم. علی‌رغم نوآوری و ارائه‌ی متدهای جدید در برخی از این مقالات، پاسخ کلاسیفایر KNN در هر کدام از موارد مذکور، دقتی پایین‌تر از کلاسیفایر KNN در این پروژه داشته است. و از دلایل احتمالی آن نیز، بهینه نبودن پارامتر انتخابی و کم بودن تعداد نمونه‌های یک کلاس نسبت به کلاس دیگر، در کلاسیفایرها در مقالات گذشته

بوده که موجب عملکرد ضعیف‌تر و بایاس شدن تفکیک می‌شود. این مشکلات تا حد امکان در برنامه‌ی ما برطرف شده و به دقتی مقبول، برای هر چهار کلاسیفایر رسیدیمف تا بتوانند، به حد امکان تفکیک مناسبی را از دیتاست هپاتیت ارائه دهند.

5-2-2- نوآوري

اگرچه تمامی روش‌های موجود در این مطلب، در مقاله‌های پیشین و توسط اشخاص دیگر، احتمالاً استفاده شده‌اند، اما موردی که به نظر در مقالات مشاهده شده، در نظر گرفته نشده بود، کم بودن تعداد نمونه‌های کلاس 2، (اشخاصی که بر اثر بیماری جان خود را از دست دادند)، به نمونه‌های کلاس 1، (افرادی که زنده ماندند) بود. در صورتی که بخواهیم دقت درستی از کلاسیفایر بدست آوریم بدیهی است که باید دیتاست به هم ریخته شود، نتیجتاً در برخی موارد، تعداد کمی از نمونه‌های کلاس (2) برای داده‌های Training قرار داده می‌شد و تعداد زیادی از آن‌ها برای داده‌های تست قرار می‌گرفت (یا برعکس)، که بعضاً موجب بایاس تخمین و کاهش چشمگیر دقت لحظه‌ای یا افزایش واریانس تخمین، و تاثیر منفی بر روی میانگین دقت می‌شد، که این موضوع در این مطلب برطرف شده است. در برنامه، ما ابتدا داده‌ها را مرتب کرده و تمام پرونده‌های دو کلاس را از یکدیگر جدا کردیم، سپس داده‌های موجود در هر دو بخش را به هم ریخته و با نسبت 70-30 برای Test Set و Training Set جدا کردیم. سپس داده‌های هردو کلاس را که برای Training Set قرار داده شده در یک ماتریس قرار داده، همین کار را برای Test Set نیز انجام داده و داده‌ها را مجدداً به هم ریختیم.

این عمل موجب می‌شود، داده‌هایی که برای آموزش مدل استفاده می‌شوند، علی‌رغم رندوم بودن، شامل تعداد متناسبی از کلاس (2) و کلاس (1) باشند. در نتیجه کلاسیفایرها نسبت به داده‌های تست، عملکرد بهتر و با بایاس تخمین کمتری ارائه خواهند داد.

5-2-3- پیشنهادها

برای تحقیقات آینده، از مواردی که می‌توان در نظر داشت که به آن به حد کافی در مورد این دیتاست پرداخته نشده، استفاده از وزن دهی به ویژگی‌ها برای کلاس‌بندی می‌باشد. نمودارهای پراکندگی کلاس‌ها بر حسب ویژگی‌ها نشان می‌دهد برخی از ویژگی‌ها اثر بسیار کمتری از دیگر ویژگی‌ها در امر تفکیک دارند، که

در این پروژه لحاظ نشده‌اند. لذا پیشنهاد می‌شود که در مقالات و تحقیقات بعدی این موضوع لحاظ شود. همچنین استفاده از کلاسیفایرهای بهینه شده و آزمودن عملکرد دیگر کلاسیفایرها برای امر تفکیک توصیه می‌شود که میتواند، به نتایج قابل توجهی بی‌انجامد.

مراجع

مراجع

[1] Knowledge Discovery in Medical and Biological Datasets Using a Hybrid Bayes Classifier/Evolutionary Algorithm Michael L. Raymer, Member, IEEE, Travis E. Doom, Member, IEEE, Leslie A. Kuhn, and William F. Punch

[2] Boosting Lazy Decision Trees, Xiaoli Zhang Fern xz@ecn.purdue.edu Carla E. Brodley brodley@ecn.purdue.edu School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907

[3] Pattern Classification, Richard O. Duda, Peter E Hart , David G. Stark , Publisher: Wiley

پیوست‌ها

Abstract:

Hepatitis is being considered as a common condition between people in the world, referring to the inflammation in Liver's tissue. World Health's Organization (WHO), estimates that an enormous number of people around 354 million, live with chronic kind of this condition, One of important perspectives in medical science, is diagnosing the risk and danger, threating patient's life, despite diagnosing the condition itself. Which can be achieved, using some vital signs and prior knowledge about the patient. This knowledge can help the Physician to choose a better approach on treatment process, and can possibly help saving patient's life. Therefore, this paper is looking to find a suitable procedure in order to separate cases which are likely to be fatal and the ones who have a higher chance of recovery from eachother, using pattern recognition techniques.

The information and dataset used in this paper is downloaded from UCI Machine learning website, and the basis of the procedures, chosen in this project, are built upon the information extracted from the Hepatitis Dataset. The program, produced to separate mentioned classes in the dataset is coded using Matlab 2020b, which is explained in the paper step by step, and the final results are also evaluated at the end.

Keywords: Hepatitis Dataset , Machine Learning , Class separation , Coding

Iran University of Science and Technology
Electrical Engineering Department

Title

A Classification Approach on Hepatitis Dataset

By:
Pouya Ahmadpour

Supervisor:
Dr. Mohammad Reza Daliri

Advisor:
Saeed Sang Sefidi

January of 2022