دانشگاه خوارزمی
دانشکده مهندسی
گروه هوش مصنوعی و رباتیکز

نام و نام خانوادگی :

یاسین امینی

شماره دانشجویی :

۹۹۳۰۸۲۵۰۲

عنوان :

طراحی یک سیستم پشتیبان تصمیم یکپارچه مبتنی بر شبکه‌های عصبی مصنوعی و فرآیند تحلیل سلسله مراتبی برای پیش‌بینی خطر نارسایی قلبی

استاد :

دکتر عزیزاله معماریانی

1 چکیده

نارسایی قلبی[1] (HF) یکی از کشنده ترین و بحرانی‌ترین بیماری‌های انسانی، در سراسر جهان است و اثرات خیلی بدی روی کمیت و کیفیت زندگی انسان‌ها دارد. از همین رو پیش‌بینی کارا و به موقع از خطر مبتلا شدن به این بیماری می‌تواند برای پیشگیری و درمان آن حیاتی باشد. تشخیص بیماری قلبی از طریق تاریخچه پزشکی سنتی از بسیاری جهات قابل اعتماد نمی‌باشد و دارای مشکلات عدیدی است. برای پیش‌بینی خطر ابتلا به HF و همچنین دسته‌بندی به افراد سالم و دارای بیماری قلبی ، سیستم‌های پشتیبان تصمیم مبتنی بر شبکه‌های عصبی مصنوعی[2] (ANN) و الگوریتم‌های یادگیری ماشین، که از آن‌ها به عنوان روش‌های غیرتهاجمی هم یاد می‌شود، به طور گسترده ای ارائه و مورد استفاده قرار گرفته‌اند و کارایی و عملکرد قابل اعتماد خود را نشان داده‌اند. ضعف اساسی که در این سیستم‌ها موجود این است که معمولاً فرض می‌کنند که ویژگی‌های مختلف در پیش‌بینی و تشخیص خط نارسایی قلبی از ارزش و اثرگذاری برابری برخوردار هستند. با این حال، چندین تحقیق قبلی نشان داده است که سهم ویژگی‌ها در پیش‌بینی خطر ابتلا متفاوت است. بنابراین، برابری سهم ویژگی‌ها نمی‌تواند به درستی وضعیت تشخیص بیماری را منعکس کند. در این مطالعه، ۱۳ ویژگی HF که معمولاً مورد استفاده قرار می‌گیرد در نظر گرفته شد و سهم هر یک از آنها توسط یک پزشک متخصص قلب مشخص شد. و از روش فرآیند تحلیل سلسله مراتبی فازی (Fuzzy_AHP) برای محاسبه وزن ویژگی‌ها بر اساس سهم فردی آنها استفاده گردید. سپس وزن سراسری که نشان دهنده سهم ویژگی‌ها است ، برای آموزش یک طبقه بند برای پیش بینی خطر ابتلاء به HF در بیماران استفاده گردید.

---

[1] Heart Failure

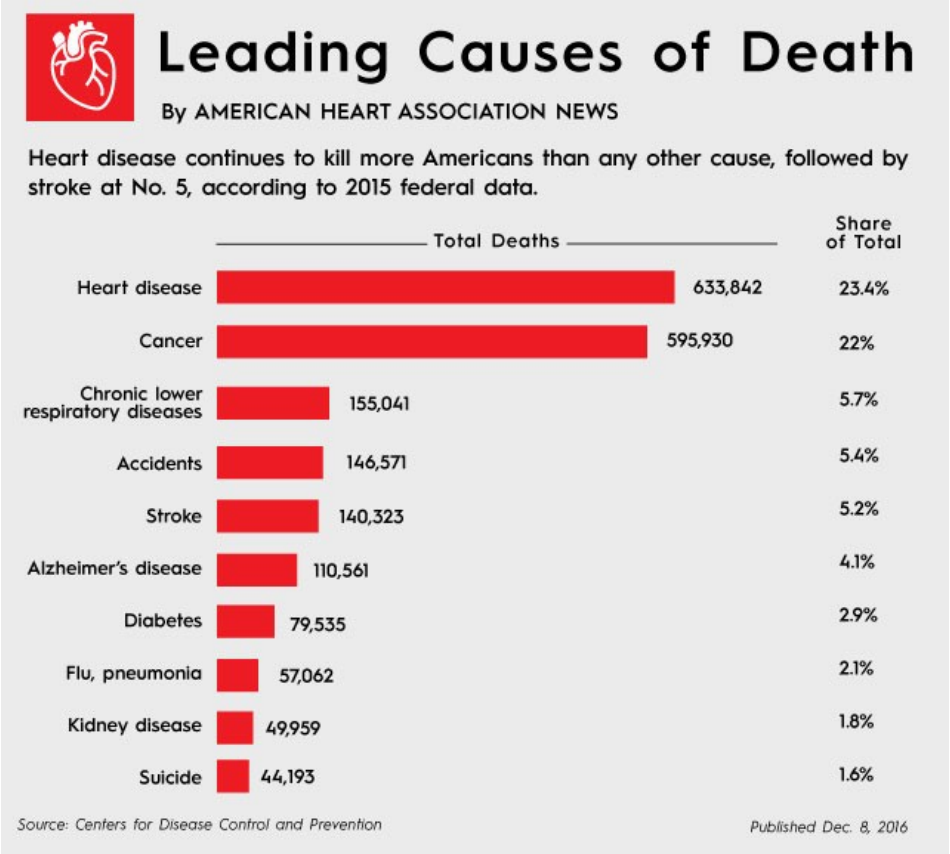
[2] Artificial Neural Network

عملکـرد سیسـتم پشـتیبانی تصـمیم اخـیراً پیشـنهادی مبتـنی بـر تلفیـق روش هـای ANN و Fuzzy_AHP با استفاده از مجموعه داده های بـالینی آنلاین ۲۹۷ بیمـار HF مـورد ارزیـابی قـرار گرفتـه و بـا روش ANN معمـولی مقایسـه شـده است. نتیجـه مـا نشـان می‌دهـد کـه روش پیشنهادی می تواند دقتی معادل ۹۱ درصد بدست آورد (بهترین نتیجه را گزارش داده‌اند) کـه در مقایسه با روش ANN معمولی چند درصد بهبود داشـته اسـت. بهبـود پیش بیـنی خطـر HF در مطالعه حاضر ممکن است به دلیل هر دو سهم مختلف از ویژگی های HF و روش ترکیـبی پیشنهاد شده باشد. این یافته ها نشان می‌دهد که می تـوان از روش پیشـنهادی بـرای پیش بیـنی دقیـق خطـرات HF در کلینیـک اسـتفاده کـرد. در روش ارائـه شـده بـا گرادیـان بهبودیافتـه (بوستینگ) به دقت ۹۱,۸ رسیدیم.

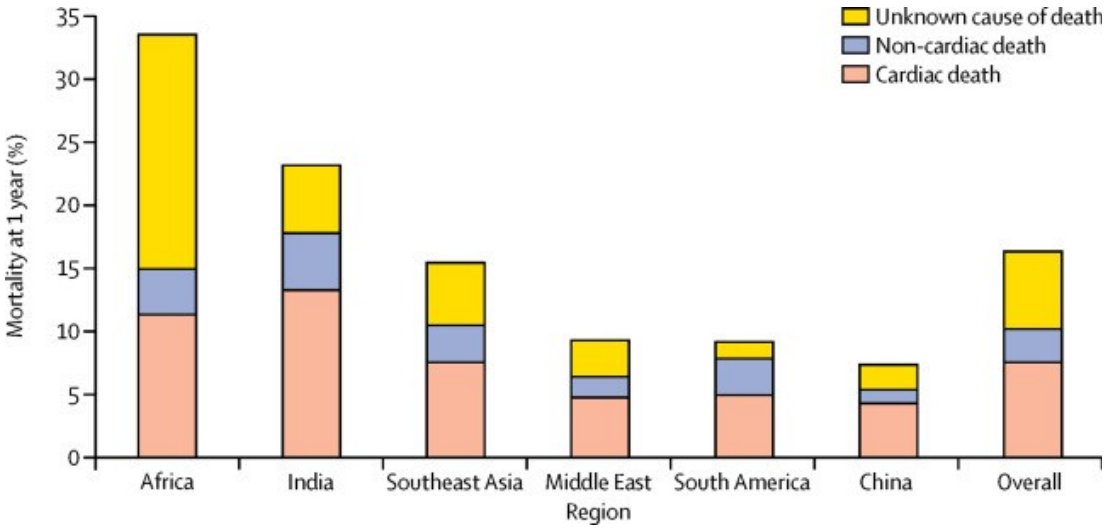
2 مقدمه

نارسایی‌های قلبی که از آن به عنوان نارسایی قلبی مزمن[3] هم یاد می‌شـود، یکی از رایج‌تـرین و چـالش برانگـیزترین مسـائل در حـوزه سـلامت اسـت کـه در تمـام کشـورهای دنیـا اعم از توسـعه‌یافته، در حـال توسـعه و توسـعه‌نیافته وجـود دارد. آمـار بـالا مـرگ و مـیر ناشـی از بیماری‌های قلبی در آمریکا را در تصویر ۱ و در قاره‌های مختلف دنیا را در تصویر ۲ مشاهده می‌نمایید.

[3] Chronic heart failure



تصویر ۱: آمار علت‌های مختلف مرگ و میر در آمریکا (بیماری‌های قلبی بالاترین درصد را دارند)

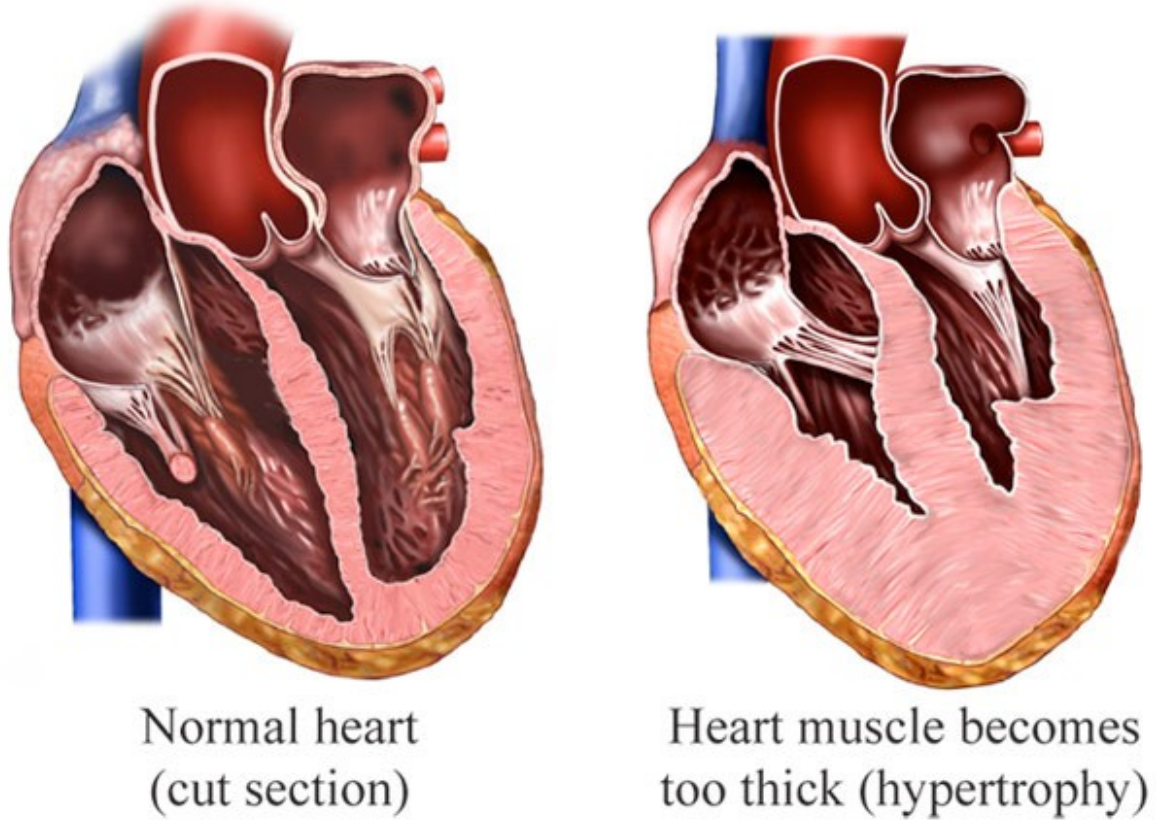


تصویر ۲: علل مرگ‌ومیر در قاره‌های مختلف (بیماری‌های قلبی در تمام قاره‌های غیر از آفریقا بالاترین آمار را دارد)

نارسایی قلبی وقتی اتفاق می‌افتد که قلب نتواند جریان خون کافی را برای اعضای بدن پمپاژ کند تا بدن توانایی انجام عملکرد طبیعی خود را داشته باشد و در این صورت بدن از عملکرد طبیعی خود خارج شده و نارسایی قلبی رخ می‌دهد. دلایل معمول بیماری نارسایی قلبی می‌تواند مواردی همچون بیماری عروق کرونر (سرخرگ کرونر تنگ و باریک می‌شوند (استنوزیس) و عضلات قلب از رسیدن خون و اکسیژن کافی محروم می‌گردند) از قبیل سکته قلبی (حمله قلبی)، فشار خون بالا، فیبریلاسیون دهلیزی، نارسایی دریچه قلب، سوءمصرف الکل و کاردیومیوپاتی است. این موارد با تغییر ساختار یا عملکرد قلب باعث نارسایی قلبی می‌گردند. تفاوت یک قلب سالم و قلب دارای نارسایی را در تصویر ۳ مشاهده می‌نمایید. در نارسایی قلب ماهیچه‌های قلب ضعیف شده و توان خود برای پمپاژ خون کافی را از دست می‌دهند.

انجمن قلب و عروق اروپا (ESC[4]) گزارش داد که ۲۶ میلیون بزرگسال در سراسر جهان مبتلا به بیماری قلبی و ۳,۶ میلیون نفر هر ساله به این تعداد افزوده می‌شود. تقریباً ۵۰ درصد بیماران قلبی طی ۱ تا ۲ سال اولیه می‌میرند و این در حالی است که هزینه‌های مربوط به مدیریت بیماری‌های قلبی تنها ۳ درصد بودجه مالی بهداشت و درمان را به خود اختصاص می‌دهد.

[4] European Society of Cardiology



تصویر ۳: تفاوت قلب سالم و قلبی که دچار نارسایی شده است

از علائم و نشانه‌های این بیماری می‌توان به مواردی از قبیل تنگی نفس، خستگی مفرط و ورم پاها اشاره کرد. تنگی نفس افرادی که دچار بیماری نارسایی قلبی هستند با ورزش کردن، دراز کشیدن و در هنگام خواب بدتر می‌شود و سبب ایجاد محدودیت‌هایی در فعالیت این افراد و مختل شدن زندگی آن‌ها می‌شود. نتایج تحقیقات حاکی از این است که نارسایی قلبی یک از اصلی‌ترین دلایل کاهش کیفیت زندگی افراد و مرگ و میر در جوامع پیشرفته است. شدت بیماری معمولاً با توجه به مقدار کاهش توان شخص برای ورزش کردن سنجیده می‌شود. مدیریت این بیماری در کشورهای در حال توسعه که دارای محدودیت در ابزارهای تشخیصی کافی و متخصصان پزشکی هستند، بیش از پیش خود را نشان می‌دهد. بنابراین ،

پیش‌بینی کارآمد خطر ابتلا به نارسایی قلبی در افراد، برای کاهش خطرات مرتبط با مشکلات شدید قلبی و افزایش ایمنی و بهره‌وری آنها در طول فعالیت‌های روزمره ضروری است.

روش رایج برای تشخیص این بیماری، روش‌های تهاجمی هستند. این روش‌های تهاجمی برای تشخیص نارسایی قلبی عمدتاً مبتنی بر تجزیه و تحلیل تاریخچه پزشکی بیماران، معاینه‌های فیزیکی و بررسی علائم مربوطه، توسط یک پزشک صورت می‌گیرد، که اغلب منجر به تشخیص نادرست ناشی از خطاهای انسانی و همچنین تأخیر در ارائه نتایج به دلیل کمبود متخصصین می‌شود. علاوه بر این، این روش‌ها گران تر و از نظر محاسباتی پیچیده‌تر و ارزیابی آن‌ها زمان‌بر است و باعث ایجاد تأخیر در تشخیص می‌شود و همین تشخیص دیرهنگام جان بیمار را به خطر می‌اندازد. برای حل این چالش، سیستم‌های پشتیبان تصمیم پزشکی[5] مبتنی بر روش‌های محاسباتی مانند ماشین بردار پشتیبان[6]، K نزدیک‌ترین همسایه[7]، درخت تصمیم[8]، منطق فازی[9] و شبکه‌های عصبی مصنوعی ارائه شدند که روش‌های غیرتهاجمی و بی‌خطری بوده و دارای کارایی و عملکرد مناسبی هستند و با پیش‌بینی مناسب می‌توانند در جهت تشخیص زود هنگام و پیشگیری و درمان بهتر عمل کنند و در نهایت منجر به کاهش آمار مرگ و میر ناشی از این بیماری و افزایش کیفیت زندگی افراد درگیری بیماری شود. در میان روش‌های ذکر شده، شبکه‌های عصبی مصنوعی به دلیل انعطاف بالایی که برای حل مسائل خطی و غیرخطی دارند به طور گسترده‌تری مورد استفاده قرار گرفتند. با وجود کارایی خوب این روش‌ها در پیش‌بینی خطر ابتلا به نارسایی قلبی، این روش‌ها دارای یک چالش کلیدی نیز هستند. در روال عادی شبکه‌های عصبی مصنوعی فرض بر این است

---

[5] Medical Decision Support Systems (MDSS)

[6] Support Vector Machine (SVM)

[7] K-Nearest Neighbor (KNN)

[8] Decision Tree

[9] Fuzzy Logic

که تمامی ویژگی‌های مسئله دارای وزن و اثرگذاری یکسانی بروی نتیجه پیش‌بینی هستند اما در بررسی‌های صورت گرفته، آشکار شده است که این ویژگی‌های دارای اثرگذاری یکسانی نیستند و برخی از ویژگی‌ها اثر و وزن بالاتری دارند. این تفاوت در اثرگذاری ویژگی‌ها در نتیجه‌گیری‌هایی که متخصصین این حوزه هم انجام می‌دهند صورت می‌گیرد. یعنی آن‌ها نیز در تصمیم‌گیری خود برای اینکه یک فرد مبتلا به نارسایی قلبی است یا در آینده احتمال دارد به این بیماری مبتلا شود، به تمامی ویژگی‌های ذکر شده در معاینات و آزمایشات وزن برابر نمی‌دهند و برخی از ویژگی‌ها اثرگذاری بیشتری دارند. به عنوان مثال احتمال اینکه یک فرد با سن بالای ۶۰ سال به نارسایی قلبی دچار شود از اینکه یک فرد ۲۰ ساله دچار شود بالاتر است یا افراد چاق به احتمال بیشتری به این بیماری مبتلا می‌شوند. در حالی که قد فرد احتمالاً اثری بروی این پیش‌بینی ندارد، پس باید سن و وزن نسبت به قد دارای ضریب و اثرگذاری بالاتری باشند.

3 داده‌ها

در این تحقیق از مجموعه داده Cleveland که از University of California, Irvine (UCI) قابل دانلود است، استفاده شده است. این مجموعه داده شامل ۳۰۳ نمونه است که هر نمونه ۱۳ ویژگی دارد. در اصل این داده‌ها دارای ۷۶ ویژگی هستند که با توجه به بی‌ربط بودن و همچین حریم خصوصی خیلی از این ویژگی‌ها حذف و ۱۳ از آن‌ها باقی‌مانده است. در این داده‌ها ۶ مقدار گم شده[10] وجود دارد که به دلیل ماهیت حساس حوزه سلامت آن‌ها را حذف می‌کنیم و از پر کردن آن‌ها با مقادیر میانگین یا مد، خودداری می‌کنیم. هدف نهایی مسأله این است که با توجه به ۱۳ ویژگی ارائه شده بگوییم که فرد به نارسایی قلبی دچار است یا در آینده خطر ابتلاء به این بیماری را دارد یا خیر. در اصل داده‌ها دارای چهار کلاس

[10] Missing Value

هستند که یکی از آن‌ها خیر و بقیه سطوح مختلف بیماری را نشان می‌دهند که برای ما اهمیتی ندارند و ما همه آن‌ها را به برچسب بله قلمداد می‌کنیم. پس داده‌های ما دارای دو برچسب absence و presence می‌باشند. اسامی و مقادیری که هر یک از ویژگی‌های مختلف می‌توانند داشته باشند را در تصویر ۴ مشاهده می‌نمایید.

S/N	Attribute Description	Attribute Code	Alternatives	Alternative Code	Range
1	Age (Years)	AGE	Young	YNG	< 33
			Medium	MED	34 – 40
			Old	OLD	41 – 52
			Very Old	VOLD	>52
2	Sex	SEX	Male	M	1
			Female	F	0
3	Chest Pain Type	CPT	Typical Angina	TA	1
			Atypical Angina	ATA	2
			Non-angina Pain	NAP	3
			Asymptomatic	ASY	4
4	Resting Blood Pressure	RBP	Low	LOW	<128
			Medium	MED	128 - 142
			High	HIGH	143 - 154
			Very High	VHIGH	>154
5	Serum Cholesterol	SCH	Low	LOW	<188
			Medium	MED	189 - 217
			High	HIGH	218 - 281
			Very High	VHIGH	>281
6	Fasting Blood Sugar	FBS	True	YES	1
			False	NO	0
7	Resting Electrocardiographic results	RES	Normal	NOR	0
			ST-T abnormal	ST-AB	1
			Hypertrophy	HYPER	2
8	Maximum Heart Rate Achieved	MHR	Low	LOW	<112
			Medium	MED	112 – 152
			High	HIGH	>152
9	Exercise Induced Angina	EIA	True	YES	1
			False	NO	0
10	Old peak	OPK	Low	LOW	<1.5
			Risk	RSK	1.5 - 2.55
			Terrible	TER	>2.55
11	Peak Exercise Slope	PES	Upsloping	UPS	1
			Flat	FLT	2
			Downsloping	DWS	3
12	Number of major Vessels Colored by fluoroscopy	VCA	Fluoroscopy-0	FL-0	0
			Fluoroscopy-1	FL-1	1
			Fluoroscopy-2	FL-2	2
			Fluoroscopy-3	FL-3	3
13	Thallium Scan	THA	Normal	NOR	3
			Fixed Defect	FDE	6
			Reversible Defect	RDE	7

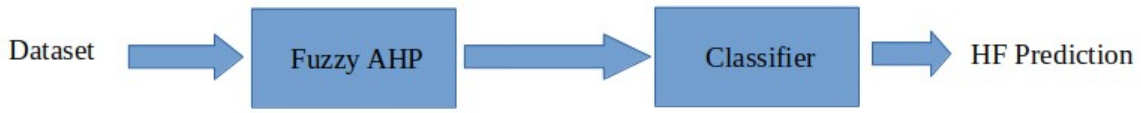
تصویر ۴: اسامی و مقدار هر یک از ویژگی‌های استفاده شده در مجموعه داده

ACCEPTED MANUSCRIPT

۴ روش انجام کار

روش ارائه شده از دو مرحله تشکیل شده است (تصویر ۵):

- در مرحله اول میزان اهمیت و اثرگذاری هر یک از ویژگی‌ها توسط فرآیند تحلیل سلسله مراتبی فازی محاسبه می‌شود.
- در مرحله دوم با استفاده از یک طبقه‌بند یا کلاس‌بند مثل شبکه‌های عصبی مصنوعی یا ماشین بردار پشتیبان کار طبقه‌بندی انجام می‌شود.



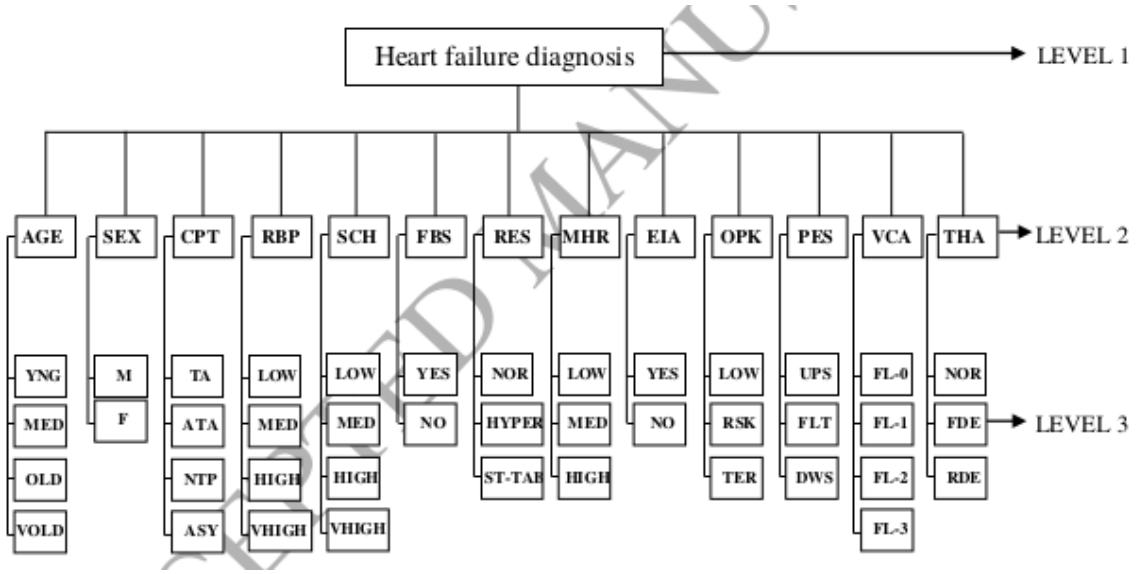
تصویر ۵: مراحل طبقه‌بندی

فرآیند تحلیل سلسله مراتبی فازی از سه مرحله تشکیل شده است:

✓ **گام اول**

در این گام، برای مسئله یک ساختار سلسله مراتبی ایجاد می‌کنیم. برای ایجاد این ساختار باید هدف، ویژگی‌های و انتخاب‌های ممکن برای هر ویژگی مشخص باشد. مثلاً در مسئله ما هدف (تشخیص نارسایی قلبی)، ویژگی‌ها (۱۳ ویژگی که برای هر نمونه وجود دارد) و انتخاب‌ها که به ازای هر ویژگی داریم هم در تصویر ۶ آمده بود. در این ساختار سلسله مراتبی هدف در ریشه یا سطح اول، ویژگی‌ها در سطح دوم و

مقادیر ویژگی‌ها در سطح سوم نمایش داده می‌شوند و ساختاری مشابه با تصویر ۶ پیدا می‌کنند.



تصویر ۶: ساختار سلسله مراتبی مسئله تشخیص نارسایی قلبی که در بالاترین سطح هدف و در سطح دوم ویژگی‌ها و در سطح سوم مقادیری که هر یک از ویژگی‌های می‌توانند داشته باشند، آمده است.

- ✓ **گام دوم**

بعد از شکستن مسئله به سه سطح هدف، ویژگی و مقادیر ویژگی‌ها، حال باید یک مقایسه دو به دو بین ویژگی‌ها انجام می‌دهیم تا میزان اهمیت هر ویژگی نسبت به بقیه ویژگی‌ها سنجیده شود. این سنجش و اهمیت ویژگی‌ها نسبت به هم براساس تجربیات و قضاوت پزشکان انجام می‌گردد.

ضرایبی که میزان اهمیت هر دو ویژگی را نسبت به هم نشان می‌دهد، در AHP به صورت دقیق[11] انتساب می‌شود. اما واقعیت این است که متغیرهای زبانی تعریف شده به صورت ذاتی دارای خصیصه فازی هستند و در آن‌ها یک عدم قطعیت وجود دارد

[11] Crisp

که نمی‌توان آن را توسط AHP مدل کرد. پس نیاز به Fuzzy AHP وجود دارد. میزان اهمیت ویژگی‌های مختلف نسبت به هم با استفاده از جدول ارائه شده در تصویر ۷، محاسبه می‌گردد.

Scalar Value	Reciprocal scalar value	Definition (criterion a in comparison to b)
1	1	Equally important
2	1/2	Weakly or slightly more important
3	1/3	Moderately more important
4	1/4	Moderately plus more important
5	1/5	Strongly more important
6	1/6	Strongly plus more important
7	1/7	Very strongly more important
8	1/8	Very, very strongly more important
9	1/9	Extremely more important

تصویر ۷: جدول محاسبه میزان اهمیت متقابل ویژگی‌های مختلف یک تصمیم

همانگونه که در تصویر ۷ هم مشاهده می‌گردد، ضرایب این مقداردهی‌ها کاملاً ماهیت فازی دارد. در اولین گام ماتریس مقایسه دوتایی ویژگی‌های ایجاد می‌شود، که یک ماتریس مربعی با ابعاد n×n است (n نشان‌دهنده تعداد ویژگی‌های موجود است). مثلاً برای این مسئله ما یک ماتریس ۱۳×۱۳ مطابق تصویر ۸ ایجاد می‌گردد. در این ماتریس سطر i ام و ستون j ام بیانگر میزان اهمیت ویژگی i ام نسبت به ویژگی j ام است و متقابلاً سطر j ام و ستون i ام هم بیانگر اهمیت ویژگی j ام نسبت به ویژگی i ام است، که عکس مقدار ارائه شده در سطر i ام و ستون j ام است.

$$if X_{i,j} = C then X_{j,i} = \frac{1}{C}$$

$$if F_i, F_j consideredequallyrelevant then X_{i,j} = X_{j,i} = 1$$

$$X_{i,i} = 1$$

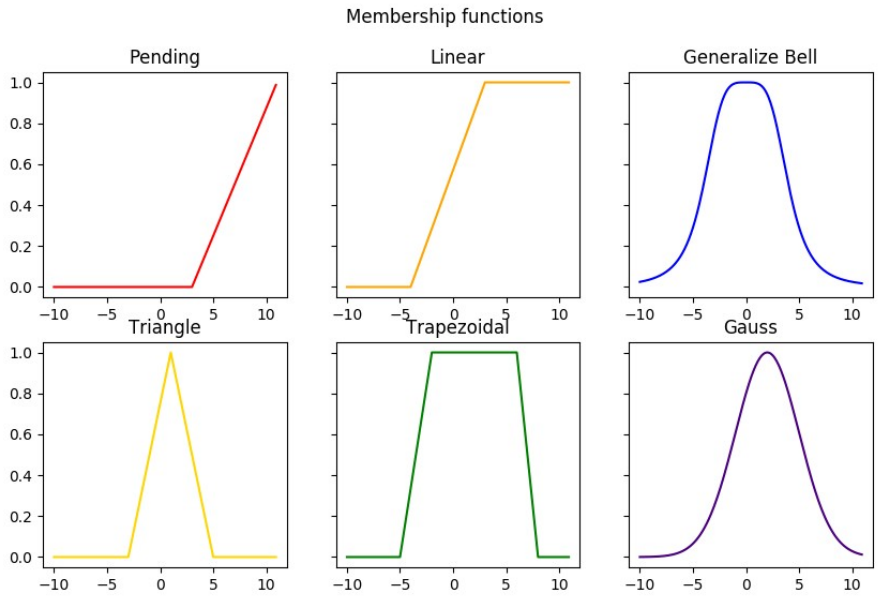
	1	2	3	4	5	6	7	8	9	10	11	12
1												
2	2											
3	3	2										
4	4	2	4									
5	5	5	5	5								
6	6	2	3	4	5							
7	7	2	3	4	5	6						
8	8	2	3	4	5	6	8					
9	9	2	9	4	5	9	9	8				
10	10	2	10	10	5	10	10	10	10			
11	11	2	3	4	5	11	11	8	11	10		
12	12	2	3	4	5	6	7	8	9	10	11	
13												

تصویر ۸: ماتریس مقایسه دوتایی یا Pair-Wise Comprison Matrix

سپس میزان اهمیت هر دو ویژگی موجود در ماتریس نسبت به هم سنجیده شده و با استفاده از جدول تصویر ۷ پر می‌شود. ماتریس ایجاد شده با استفاده از این روش یک ماتریس دقیق است که ما با توجه به ماهیت این تصمیم آن را به فضای اعداد فازی می‌بریم. تبدیل یک عدد دقیق به یک عدد فازی توسط تابع عضویت[۱۲] صورت می‌پذیرد و به این عملیات، عمل فازی‌سازی[۱۳] گفته می‌شود. هر عدد فازی به صورت یک سه تایی (حد پایین، حد متوسط، حد بالا) تعریف می‌گردد (low, mid, high). برای این تبدیل انواع مختلفی از توابع عضویت مانند تابع عضویت مثلثی، ذوزنقه‌ای، گوسی، زنگوله‌ای و ... وجود دارد (تصویر ۹)، که در این تحقیق از تابع عضویت مثلثی استفاده شده است.

[12] Membership Function

[13] Fuzzification



تصویر ۹: انواع مختلف توابع عضویت

با انجام فازی‌سازی جدول ارائه شده در تصویر ۷ با مقادیر موجود در جدول تصویر ۱۰ جایگزین می‌گردد.

Saaty scale	Definition	Fuzzy Triangular Scale
1	Equally important (Eq. Imp.)	(1, 1, 1)
3	Weakly important (W. Imp.)	(2, 3, 4)
5	Fairly important (F. Imp.)	(4, 5, 6)
7	Strongly important (S. Imp.)	(6, 7, 8)
9	Absolutely important (A. Imp.)	(9, 9, 9)
2	The intermittent values between two adjacent scales	(1, 2, 3)
4		(3, 4, 5)
6		(5, 6, 7)
8		(7, 8, 9)

تصویر ۱۰: جدول فازی اهمیت متقابل ویژگی‌های یک تصمیم

✓ **گام سوم**

تا این مرحله مـاتریس مقایسه دوبه‌دو ویژگی‌ها نسـبت به هم محاسـبه گردیـد. پس از این باید میزان اهمیت یا ضریب هر یک از ویژگی‌ها را در حالت کلی محاسـبه کـنیم. بـرای این منظـور باید مقدار میانگین هندسی فـازی[۱۴]، مـاتریس مقایسه را به دست آوریم. پس داده‌هـای هر سـطر را در هم ضرب می‌کنیم و از آن‌ها یک جذر مرتبه n می‌گیریم (مثل رابطه زیر).

$$W_i = \left[\prod_{i=1}^{n} X_{ij} \right]^{\frac{1}{n}}$$

W_i وزن متناسب با i امین ویژگی و n متناظر با تعداد کل ویژگی‌های موجود است.

برای ضرب دو عدد فازی نیز از رابطه زیر بهره استفاده می‌کنیم:

$$\mu(A) * \mu(B) = (l_1, m_1, u_1) * (l_1, m_2, u_2) = (l_1 * l_2, m_1 * m_2, u_1 * u_2)$$

سپس این میانگین‌های هندسی فازی به دست آمده را نرمال می‌کنیم (یعنی مجموع آن‌هـا را بـه دست آورده و هر یک از میانگین‌های هندسی را بر مجموع، تقسیم می‌کنیم) مثل رابطه زیر:

$$Y = \frac{W_i}{\sum_{i=0}^{n} W_i}$$

برای جمع فازی نیز از رابطه زیر بهره می‌گیریم.

$$\mu(A) + \mu(B) = (l_1, m_1, u_1) + (l_1, m_2, u_2) = (l_1 + l_2, m_1 + m_2, u_1 + u_2)$$

با استفاده از رابطه‌های ذکر شده ماتریس Y ماتریس نهایی ضرایب اسـت و در آن W_i متناسـب با ضریب i امین ویژگی می‌باشد. برای بررسی صـحت و درسـتی ضـرایب بـه دسـت آمـده، بایـد نسـبت سـازگاری[۱۵] از ۱۰ درصـد کمـتر باشـد ($CR < 0.0001$). تنهـا در این صـورت کـه

[14] Fuzzy Geometric Mean Value

[15] Consistent Ration

می‌توان درستی نتایج را فرض و از ضرایب موجود برای تصمیم‌گیری استفاده نمود. نسبت سازگاری معیاری است که میزان ناسازگاری را اندازه‌گیری می‌کند و هر چه بیشتر باشد یعنی ناسازگاری بیشتری وجود دارد. برای محاسبه این نسبت از رابطه زیر استفاده می‌شود:

$$ConsistencyRation(C.R.)=\frac{ConsistencyIndex(C.I.)}{RandomIndex(R.I.)}$$

$$ConsistencyIndex(C.I.)=\frac{\lambda_{max}-n}{n-1}$$

$$\lambda_{max}=\frac{1}{n}\sum_{i=1}^{n}\frac{\text{Weighted Sum Value}}{\text{Criteria Weights}}$$

در رابطه بالا برای محاسبه شاخص تصادفی، ماتریس مقایسه دودویی با مقدار تصادفی پر شده و شاخص سازگاری برای آن محاسبه می‌شود (برای مقادیر مشخص n این ماتریس محاسبه شده و در مقالات و منابع وجود دارد تصویر ۱۱).

Number of elements (n)	R.I.
3	0.52
4	0.89
5	1.11
6	1.25
7	1.35
8	1.40
9	1.45
10	1.49
11	1.51
12	1.54
13	1.56
14	1.57
15	1.58

تصویر ۱۱: مقدار شاخص تصادفی برای ماتریس‌های با تعداد ویژگی‌های مختلف

برای محاسبه شاخص سازگاری نیز در ماتریس مقایسه اولیه، هر یک از مقادیر ستون‌ها در ضریب آن ویژگی ضرب شده و در ماتریس به دست آمده، جمع مقادیر سطرها را به دست می‌آوریم. سپس این مقادیر به دست آمده را بر وزن هر یک از ویژگی‌ها تقسیم و از آن‌ها یک

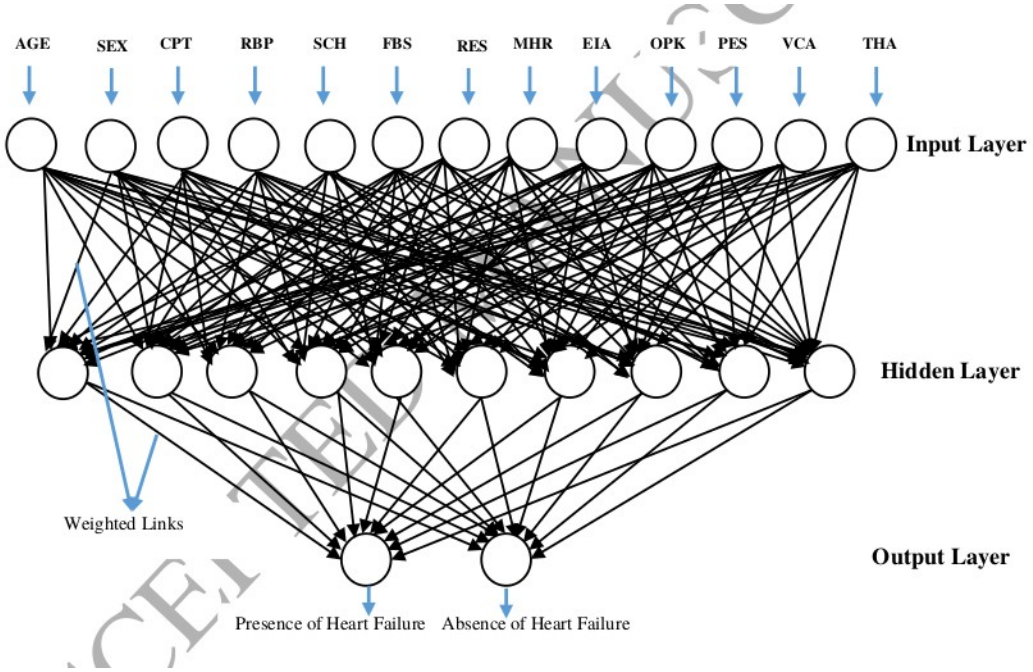
میانگین می‌گیریم که مقدار لاندا بیشینه محاسبه شده و از این طریق شاخص سازگاری نیز محاسبه می‌گردد (لاندا بیشینه معادل مقادیر ویژه اصلی[16] ماتریس است).

برای مرحله دوم که مرحله طبقه‌بندی است، ما از طبقه‌بندهای مختلفی استفاده کرده و نتایج آن‌ها را با هم مقایسه می‌کنیم و بین آن‌ها بهترین طبقه‌بند را برمی‌گزینیم و برای مدل نهایی سیستم تصمیم پشتیبان بهترین مدل را استفاده می‌کنیم. در ادامه مدل‌ها استفاده شده را به شکل مختصری معرفی می‌نماییم.

4.1 شبکه‌های عصبی مصنوعی

شبکه‌های عصبی مصنوعی از سیستم عصبی انسان الهام گرفته شده است. این یادگیرها بسیار منعطف و قدرتمند هستند و توانایی مدل‌سازی و حل انواع مسائل خطی و غیرخطی را دارند. معماری این شبکه‌ها از سه جزء کلیدی ورودی، لایه‌های مخفی و خروجی تشکیل شده است (تصویر ۱۲). با توجه به پیچیدگی مسئله لایه‌های مخفی می‌توانند از ۱ تا چندین لایه باشند. هر چه تعداد این لایه‌ها بیشتر باشد تعداد پارامترها مدل و در نتیجه پیچیدگی مدل بالاتر است. از مهم‌ترین پارامترهای شبکه‌های عصبی تعداد لایه‌های مخفی و همچنین تعداد واحدهای موجود در لایه‌های مخفی است که باید به شکل مناسبی تنظیم شوند و با تعداد واحدهای مختلف مثلا ۱۰ واحد یا ۲۰ و ... مورد ارزیابی قرار گرفته و بهترین تعداد انتخاب گردد.

[16] Principal Eigenvalue



تصویر ۱۲: مدل شبکه عصبی دارای سه لایه ورودی، لایه مخفی و خروجی است.

این الگوریتم با یک مجموعه از وزن‌های تصادفی کار خود را آغاز و سپس برای اینکه شبکه بتواند دسته‌بندی داده‌ها را به خوبی انجام دهد و دقت بالایی داشته باشد، باید مکانیزمی برای بروزرسانی این وزن‌ها ارائه شود. برای نیل به این هدف از الگوریتم پس‌انتشار خطا (Backpropagation) استفاده می‌گردد. این الگوریتم درواقع یک الگوریتم بهینه‌سازی است که هدف آن کمینه کردن اختلاف بین خروجی مطلوب و خروجی مدل یادگیری است. قاعده استفاده شده در این روش همان گرادیان کاهشی (Gradient Descent) است، که در هر لحظه در جهت خلاف گرادیان حرکت می‌کند تا بتواند به یک بهینه محلی یا سراسری برسد. در آموزش این شبکه‌ها دو گام اساسی وجود دارد که عبارت‌اند از:

- **جلورو (Feedforward)**

- در این گام یک ورودی به شبکه داده شده و شبکه این ورودی را لایه به لایه پیش برده و در پارامترها شبکه ضرب و حاصل را با هم جمع می‌کند تا به خروجی شبکه برسد و جواب این ورودی محاسبه شود.

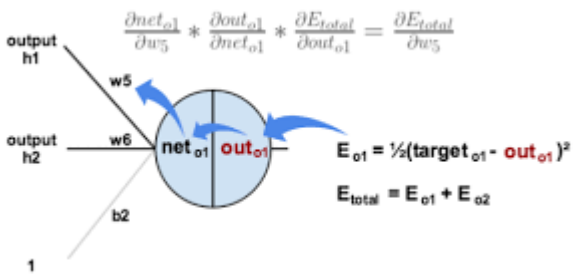
$$Net_j = \sum_i^m x_i * w_{ij} + \theta_j$$
$$x_i : Input$$
$$w_{ij} : NetworkWeights$$
$$\theta_j : LayerBias$$

در نهایت حاصل از یک تابع فعالساز مانند سیگموید عبور داده می‌شود.

$$f(Net_j) = \frac{1}{1+e^{-Net_j}}$$

- **عقب‌رو (Backpropagation)**

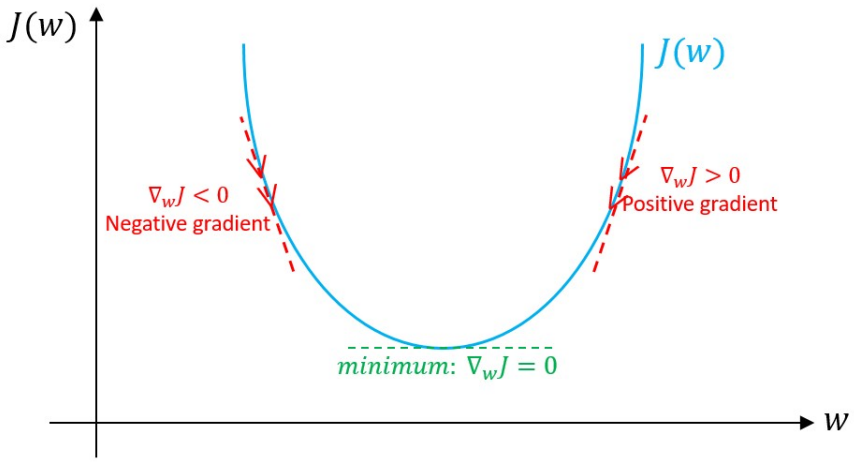
- در این گام اختلاف بین خروجی محاسبه شده توسط شبکه و خروجی واقعی متناظر این ورودی به دست آمده و به شکل عقب‌رو، بروزرسانی وزن‌ها به گونه‌ای صورت می‌پذیرد که خطای مدل رفع و در جهت کم کردن خطای مدل پیش برود. با تکرار این دو گام به صورت مکرر آموزش شبکه صورت می‌پذیرد.



تصویر ۱۳: الگوریتم پس‌انتشار خطا

همانگونه که در تصویر ۱۳ مشاهده می‌شود، هدف کمینه کردن اختلاف خروجی مطلوب و خروجی الگوریتم با وزن‌های موجود است. در هر مرحله الگوریتم با محاسبه این اختلاف و با توجه به بزرگی آن، وزن‌های شبکه را به صورت عقب‌گرد بروزرسانی می‌کند. این کار را تا زمانی انجام می‌دهد که مسأله به یک نقطه بهینه همگرا شود.

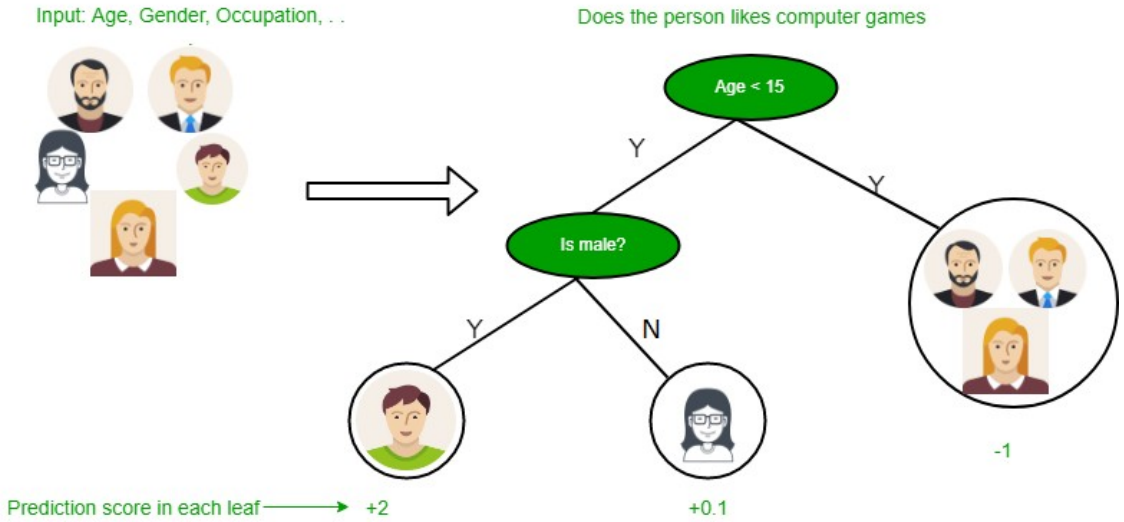
$$W_{i+1} = W_i + \Delta W_{ij}$$
$$b_{i+1} = b_i + \Delta b_{ij}$$
$$\Delta W_{ij} = -\eta \frac{\partial E}{\partial W_{ij}}$$
$$E : TotalError$$
$$\eta : Learningrate$$



تصویر ۱۴: گرادیان کاهشی

4.2 درخت تصمیم

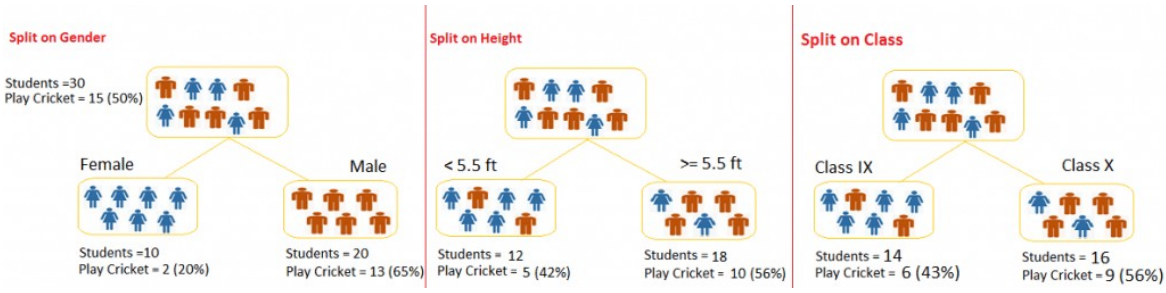
یکی از رایج‌ترین و ساده‌ترین الگوریتم‌های یادگیری ماشین و پشتیبانی از تصمیم است که از یک ساختار شبه درختی برای مدل کردن تصمیم استفاده می‌کند. به آن‌ها درخت‌های تصمیم می‌گویند زیرا می‌توانند یک تصمیمِ خاص (مثلاً اینکه به یک شخص وام بدهیم یا نه) را بر اساسِ اطلاعاتِ گذشته اتخاذ کنند. وظیفه اصلی یک درخت تصمیم، ساخت یک ساختار شبه درختی به صورت پویا (بدون کد نویسیِ صریح) است به گونه‌ای که خودِ درخت بتواند از روی داده‌های آموزشیِ موجود، شاخه‌ها و برگ‌های خود را پیدا کند. برگ‌های درخت همان برچسب هدف ما هستند. در تصویر ۱۵ یک الگوریتم درخت تصمیم برای بررسی اینکه آیا یک فرد به بازی‌های کامپیوتری علاقه‌مند است یا خیر؟ همانطور که می‌بینید در ریشه اولین جداساز این است که سن فرد از ۱۵ کمتر است یا خیر؟ اگر سن فرد بالای ۱۵ باشد احتمال اینکه به بازی‌های کامپیوتری علاقه‌مند باشد خیلی کم است.



تصویر ۱۵: الگوریتم درخت تصمیم برای بررسی اینکه یک فرد به بازی‌های رایانه‌ای علاقه‌مند است یا خیر؟

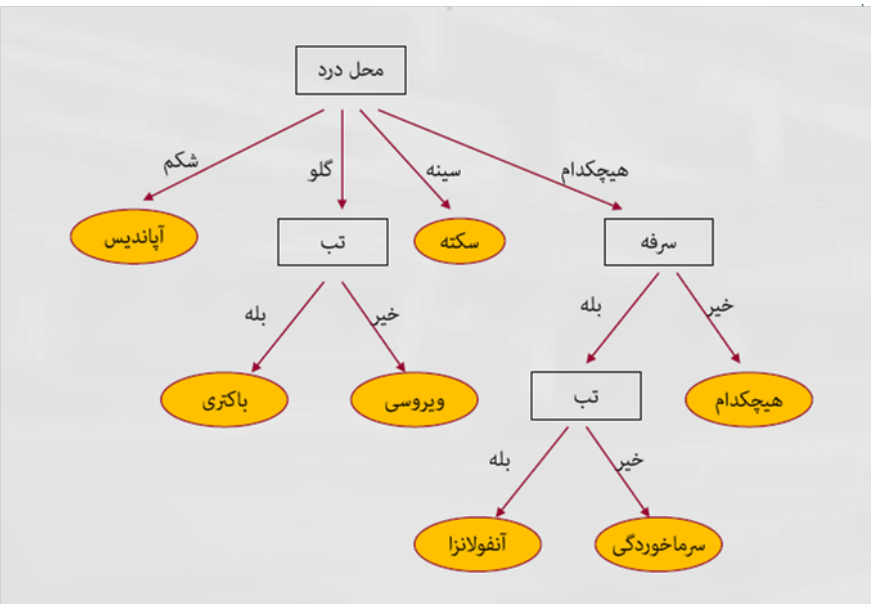
برای ساخت درخت تصمیم الگوریتم‌های مشهوری وجود دارد که می‌توان به ID3، C4.5، CART و CLS اشاره کرد. تفاوت این الگوریتم‌ها در روشی است که برای تعیین بهترین

راه برای تقسیم داده در هر شاخه استفاده می‌شود یعنی شروع و ادامه تقسیم داده‌ها براساس کدام ویژگی باشد که بهترین نتیجه حاصل شود (تصویر ۱۶). از روش‌های رایج برای این کار می‌توان به Gini Impurity و Information Gain نام برد.



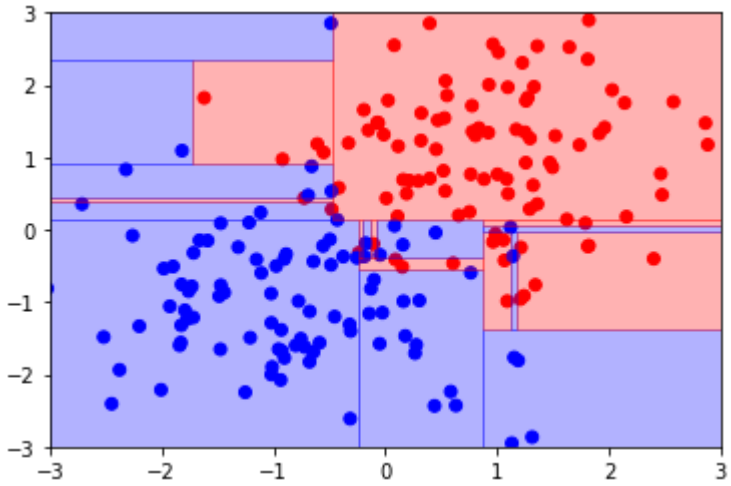
تصویر ۱۶: شاخه‌بندی براساس کدام ویژگی ما را به نتایج و تقسیم‌بندی بهتری می‌رساند؟

یکی از بهترین مزایای درخت تصمیم تفسیرپذیری آن است. یعنی تحلیل و آنالیز درخت تصمیم ما را به یک سری قواعد قابل تفسیر و معنادار می‌رساند که چرا این برچسب به دست آمده است؟ یا چرا این تصمیم اخذ شده است؟ و مشابه یک رابطه فلوچارت و دسته‌بندی انسان است. مثلا در تصویر ۱۷ یک درخت تصمیم را مشاهده می‌کنید که برای یافتن علت بیماری یک فرد با توجه به علائمی که دارد نشان داده شده است.



تصویر ۱۷: الگوریتم درخت تصمیم برای پیدا کردن یک بیماری با توجه به علائم موجود

یکی دیگر از مزایای درخت تصمیم این است که یک روش غیرپارامتریک است و نیاز به تنظیم خاصی برای افزایش دقت الگوریتم ندارد که به مدل امکان یادگیری و جداسازی مرزهای پیچیده و با درجه غیرخطی بالا را می‌دهد (تصویر ۱۸).

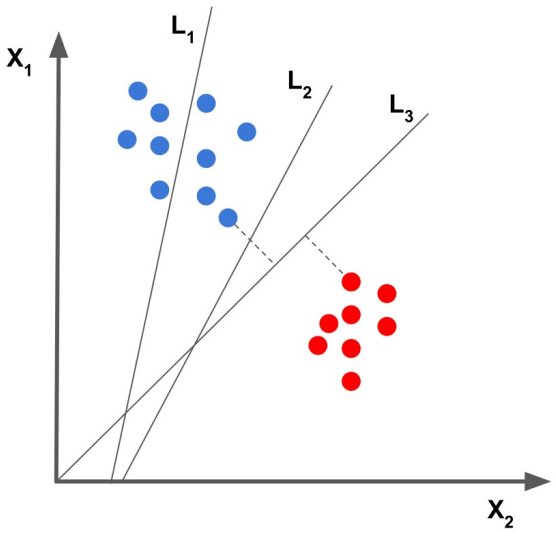


تصویر ۱۸: درخت تصمیم قابلیت یادگیری مرزهای بسیار پیچیده را دارد.

از معایب این روش هم می‌توان گفت به هزینه بالای ساخت درخت تصمیم و هرس کردن آن و پیچیدگی کار با داده‌های پیوسته اشاره نمود.

4.3 ماشین بردار پشتیبان

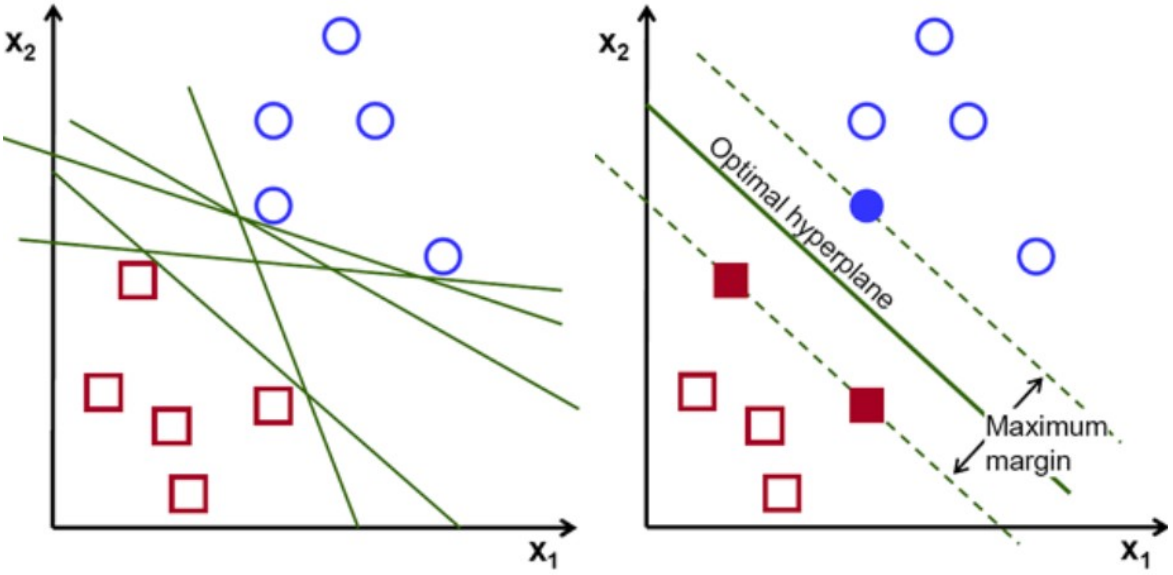
در الگوریتم‌های قبلی ما به دنبال پیدا کردن مرز جدایی کلاس‌ها بودیم و هر مرزی که کلاس‌ها را جدا می‌کرد، پذیرفته می‌شد و هیچ معیار و راهکاری برای انتخاب بهترین مرز جدایی‌ساز نداشتیم. به عنوان نمونه در تصویر ۱۹، کدام مرزبندی بین دو کلاس مرزبندی بهتری است؟



تصویر ۱۹: کدام‌یک از مرزبندی‌های بین دو کلاس زیر مرزبندی بهتری است؟

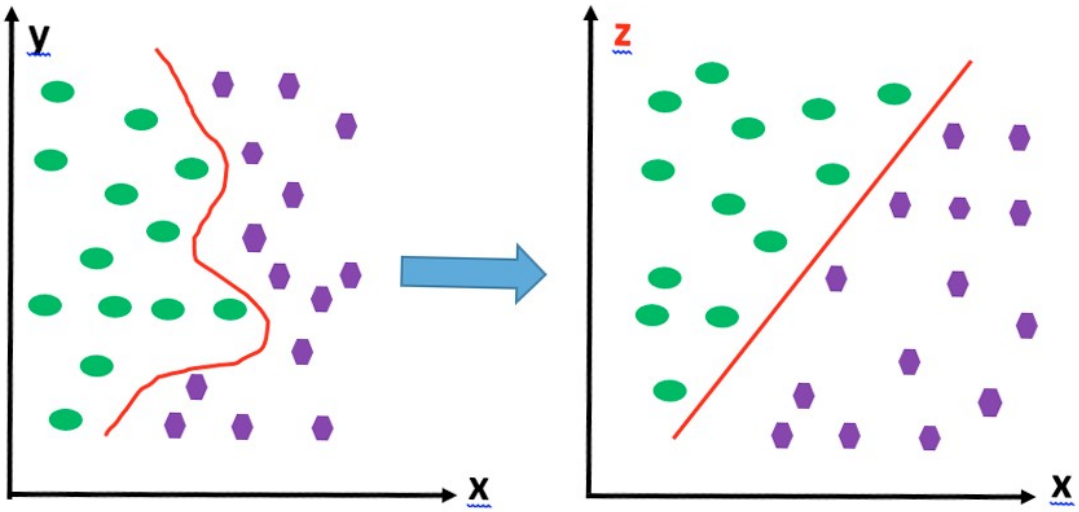
برای حل این چالش الگوریتم SVM ارائه گردید که در این الگوریتم ما به دنبال بیشینه‌سازی مرز بین کلاس‌های مختلف هستیم، یعنی از بین مرزهایی که کلاس‌های یک مسئله را جدا می‌کنند، مرز جدایی انتخاب می‌شود که از تمام کلاس‌ها بالاترین فاصله را داشته باشد. به همین دلیل از تابع هدفی استفاده می‌شود که به دنبال

بیشینه‌سازی فاصله بین بردارهای پشتیبان (نزدیک‌ترین نقاط هر کلاس به خط جداکننده) و خط جدا کننده است. به عنوان نمونه برای مثال تصویر ۲۰، SVM این مرزبندی را انتخاب می‌کند.



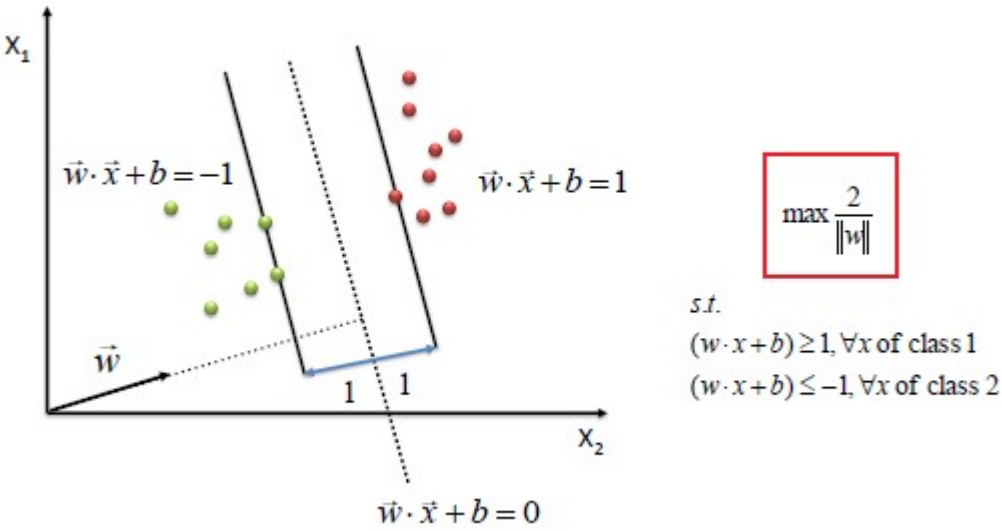
تصویر ۲۰: SVM به دنبال انتخاب بهترین مرز جدایی کننده دو کلاس است

SVM هم برای مسائل جدایی‌پذیر خطی و هم برای مسائل جدایی‌ناپذیر خطی کاربرد دارد. برای مسائلی که جدایی‌پذیر خطی نیستند از ترفند کرنل و نگاشت مسئله به ابعاد بالاتر به این امید که در بعدهای بالاتر جدایی‌پذیر خطی شود، استفاده می‌کند (تصویر ۲۱).



تصویر ۲۱: SVM برای مسائلی که به صورت خطی جدایی‌پذیر نیستند، ابتدا آنها را به ابعاد بالاتر نگاشت می‌کند، تا در ابعاد بالاتر جدایی‌پذیر خطی شوند.

در این الگوریتم با یک بهینه‌سازی غیرخطی سروکار داریم که با توجه به محدودیت‌های تعیین شده و ماهیت مسأله از روش لاگرانژ و دوگان برای بهینه‌سازی آن استفاده می‌گردد. ساختار مسأله در تصویر ۲۲ قابل مشاهده است.



تصویر ۲۲: تابع هدف ماشین بردار پشتیبان

بیشینه کردن تابع هدف ارائه شده در تصویر ۴، برابر با کمینه کردن مخرج یعنی W است. به این ترتیب مسأله به یک مسأله کمینه‌سازی تبدیل می‌شود (تصویر ۲۳).

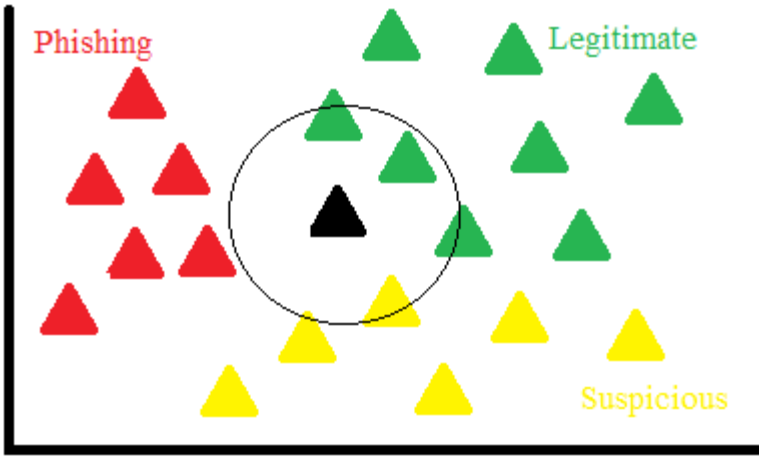
$$\min \frac{1}{2}\|w\|^2$$
$$s.t.\ y_i(w \cdot x_i + b) \geq 1,\ \forall x_i$$

تصویر ۲۳: تابع هدف نهایی ماشین بردار پشتیبان

۴.۴ K نزدیک‌ترین همسایه

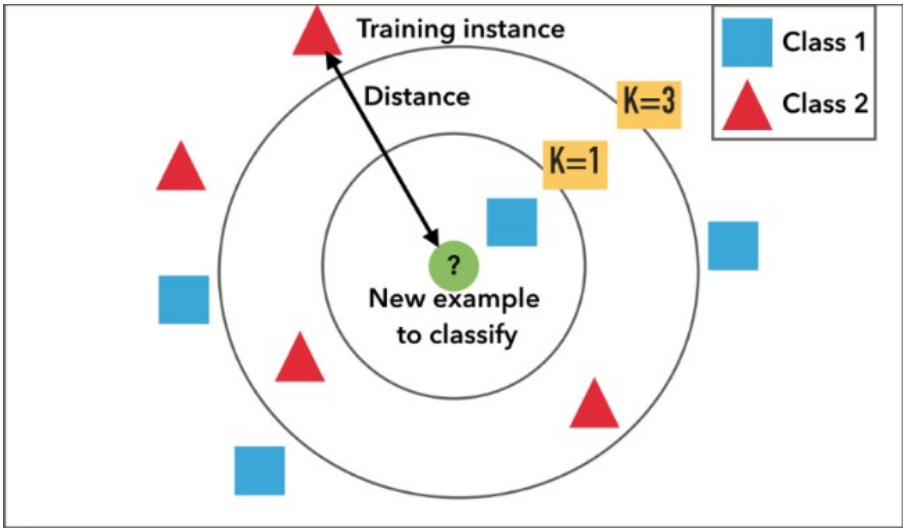
یکی از پرکاربردترین و ساده‌ترین الگوریتم‌های یادگیری ماشین است. یک مدل مبتنی بر نمونه[۱۷] است و غیرپارامتری است. یعنی در خلال فرآیند آموزش هیچ گونه پارامتری یادگرفته نمی‌شود و مدل هیچ پارامتری ندارد. در این مدل فاز یادگیری یا آموزش وجود ندارد و در فاز آموزش فقط داده‌ها ذخیره می‌شوند و تمام فرآیند تصمیم‌گیری در فاز ارزیابی اتفاق می‌افتد. نحوه کار الگوریتم به این صورت است که با وارد شدن یک نمونه که می‌خواهیم بدانیم عضو چه کلاسی است، با تک تک نمونه‌های موجود در پایگاه مقایسه و K تا از نزدیک‌ترین نمونه‌ها را انتخاب می‌شود و بین این K نمونه یک رأی‌گیری انجام می‌دهد و هر برچسبی که تعداد آرا بیشتری به دست آورد به عنوان برچسب داده انتخاب می‌گردد (تصویر ۲۴).

[17] Instance Based Model



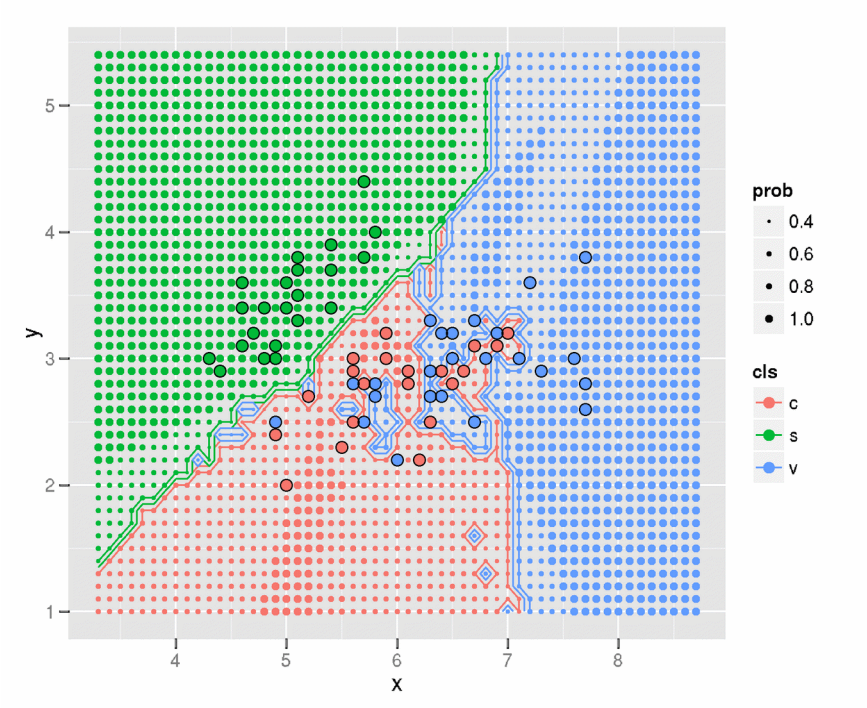
تصویر ۲۴: الگوریتم KNN با K=3، در اینجا چون بیشتر داده‌ها به Legitimate رای داده‌اند، این برچسب را به آن می‌دهیم.

تنها پارامتر این روش K است که می‌تواند تصمیم‌گیری نهایی را به شدت تحت تاثیر قرار دهد. همانگونه که در تصویر ۲۵ مشاهده می‌کنید تغییر K از ۱ به ۳ سبب عوض شدن برچسب داده از مربع به مثلث می‌شود.



تصویر ۲۵: الگوریتم KNN و تاثیر مقدار K بروی برچسب داده تحت کویری

از معایب این روش می‌توان به پیچیدگی فضایی و زمانی آن اشاره کرد، چون به ازای هر نمونه باید تمامی داده‌های آموزشی ذخیره و مقایسه شوند. برای حل این چالش از یافتارهایی مانند Approximate KNN و ... استفاده می‌گردد. از مزایا این روش هم می‌توان به سادگی آن اشاره کرد. یکی از بهترین مزایای KNN که در تصویر ۲۶ نیز قابل مشاهده است، این است که توانایی جداسازی داده‌هایی با مرزهای بسیار پیچیده غیرخطی را دارد.



تصویر ۲۶: مرز تصمیم ایجاد شده توسط الگوریتم KNN

4.5 رگرسیون لجستیک

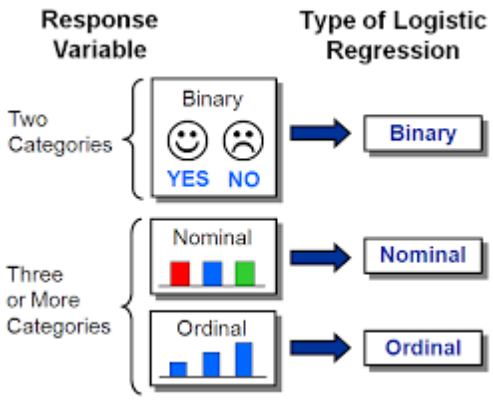
یکی از روش‌های تحلیل رگرسیون است که برای مسائل کلاس‌بندی مورد استفاده قرار می‌گیرد. رگرسیون یک نوع مدل آماری‌ست که برای پیش‌بینی یک متغیر مستقل از روی یک

یا چند متغیر وابسته دیگر مورد استفاده قرار می‌گیرد. رگرسیون لجستیک زمانی استفاده می‌شود که متغیر وابسته به صورت گروه‌بندی یا طبقه‌بندی شده باشد. مثلاً

- آیا یک ایمیل اسپم است یا غیراسپم
- آیا یک تومور بدخیم است یا خوش‌خیم

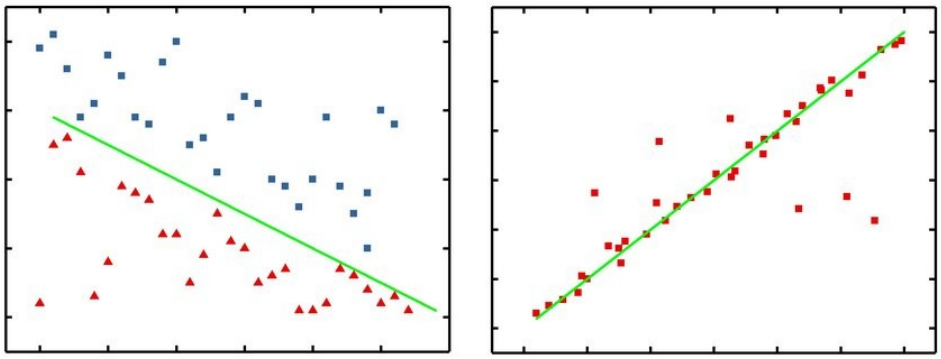
به طور ساده می‌توان گفت که رگرسیون لجستیک یک تابع ریاضی هست که با استفاده از آن می‌توانیم به داده‌ها برچسب خاصی رو نسبت دهیم.

رگرسیون لجستیک می‌تواند باینری یا دو کلاسه یا دو سویه، چند کلاسه یا چند سویه و ترتیبی باشد (تصویر ۲۷). منظور از دو سویی بودن، رخ داد یک واقعه تصادفی در دو موقعیت ممکن است. به عنوان مثال خرید یا عدم خرید، ثبت‌نام یا عدم ثبت‌نام، ورشکسته شدن یا ورشکسته نشدن و ... متغیرهایی هستند که فقط دارای دو موقعیت هستند و مجموع احتمال هر یک آن‌ها در نهایت یک خواهد شد. کاربرد این روش عمدتاً در ابتدای ظهور در مورد کاربردهای پزشکی برای احتمال وقوع یک بیماری مورد استفاده قرار می‌گرفت. لیکن امروزه در تمام زمینه‌های علمی کاربرد وسیعی یافته‌است. که در مسأله ما نوع آن رگرسیون لجستیک باینری است، چون ما فقط دو کلاس بیمار و سالم داریم.



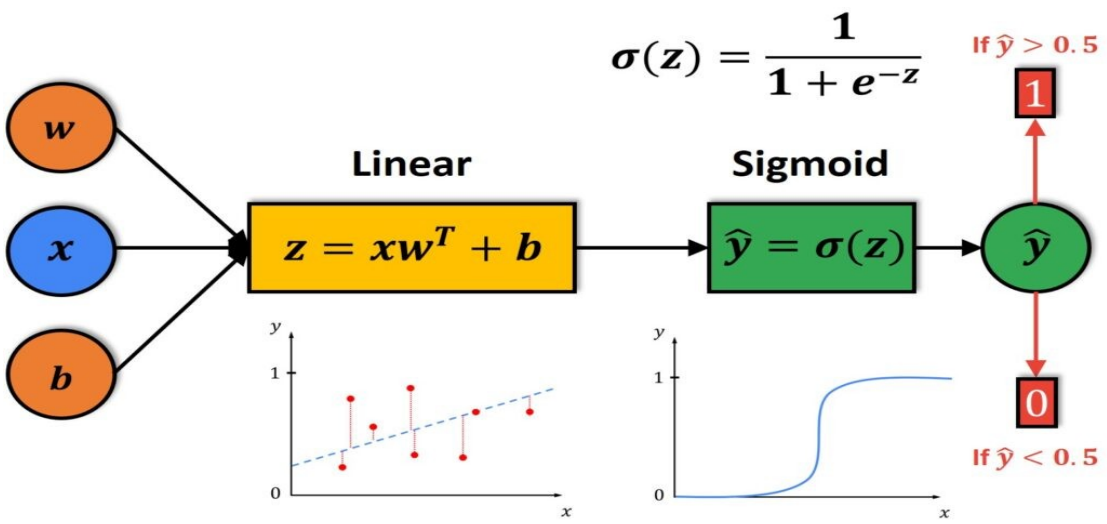
تصویر ۲۷: انواع حالات رگرسیون لجستیک

رگرسیون لجستیک می‌تواند یک مورد خاص از مدل خطی عمومی و رگرسیون خطی فرض شود. مدل رگرسیون لجستیک، بر اساس فرض‌های کاملاً متفاوتی (دربارهٔ رابطه متغیرهای وابسته و مستقل) از رگرسیون خطی است. تفاوت مهم این دو مدل در دو ویژگی رگرسیون لجستیک است. اول توزیع شرطی به جای توزیع گوسی یک توزیع برنولی است، چون متغیر وابسته دودویی است. دوم مقادیر پیش‌بینی احتمالاتی است و محدود بین بازه صفر و یک و به کمک تابع توزیع لجستیک بدست می‌آید. یعنی در نهایت رگرسیون لجستیک احتمال خروجی را پیش‌بینی می‌کند. یکی از اصلی‌ترین تفاوت‌های رگرسیون خطی و لجستیک در ذات این دو است. رگرسیون خطی برای کاربردهای پیش‌بینی کمیت پیوسته و رگرسیون لجستیک برای کاربردهای پیش‌بینی کمیت گسسته و رسته‌ای استفاده می‌گردد (تصویر ۲۸).



تصویر ۲۸: تفاوت میان رگرسیون خطی و رگرسیون لجستیک

از نظر رفتاری یک شباهت بین رگرسیون لجستیک و شبکه‌های عصبی مصنوعی وجود دارد (تصویر ۲۹). همانگونه که مشاهده می‌کنید رگرسیون لجستیک، همان رگرسیون خطی است که از یک تابع Logit یا سیگموید عبور داده شده است تا یک بتوان از آن یک تعبیر احتمالاتی داشت و احتمال انتساب برچسب هر کلاس را محاسبه نمود.



تصویر ۲۹: شباهت ساختاری بین رگرسیون لجستیک و شبکه‌های عصبی مصنوعی

فرض کنید که یک مجموعه داده با دو ویژگی x_1 و x_2 و یک خروجی باینری Y داریم و می‌خواهیم احتمال اینکه یک داده ورودی، برچسب کلاس یک را بگیرد، محاسبه کنیم. یعنی $p=P(Y=1)$ را به دست آوریم. فرض می‌کنیم که یک رابطه خطی بین متغیر مورد پیش‌بینی و تابع *logit* رخداد $Y=1$ وجود دارد. رابطه خطی مذکور به صورت زیر نوشته می‌شود.

$$l=\log_b \frac{p}{1-p}=\beta_0+\beta_1 x_1+\beta_2 x_2$$

رابطه ۱

در رابطه ۱، l بیانگر تابع logit و b پایه لگاریتم و بتاها پارامترهای مدل هستند. با حذف لگاریتم از رابطه ۱ رابطه ۲ حاصل می‌شود.

$$\frac{p}{1-p}=b^{\beta_0+\beta_1 x_1+\beta_2 x_2}$$

رابطه ۲

از رابطه ۲ می‌توان مقدار p را محاسبه نمود.

$$p=\frac{b^{\beta_0+\beta_1 x_1+\beta_2 x_2}}{b^{\beta_0+\beta_1 x_1+\beta_2 x_2}+1}=\frac{1}{1+b^{-(\beta_0+\beta_1 x_1+\beta_2 x_2)}}=S_b(\beta_0+\beta_1 x_1+\beta_2 x_2)$$

رابطه ۳

در رابطه ۳، Sb نمایانگر تابع سیگموید است و به این می‌توانیم با عبور دادن پارامترهای به دست آمده توسط رگرسیون خطی از یک تابع سیگموید احتمال یک کلاس خاص را محاسبه نماییم.

4.6 بیزین ساده[18]

دسته‌بندی بر اساس روش‌های احتمالی، خانواده بزرگی از الگوریتم‌های یادگیری ماشین را شامل می‌شوند و در مسائل زیادی از قبیل دسته‌بندی متون، تشخیص هرزنامه و کاربردهای پزشکی و بیوانفورماتیکی استفاده می‌شوند. ایده پایه این روش استفاده از قاعده بیز است که در رابطه ۴ مشخص شده است.

$$P(A|B)=\frac{P(B|A)P(A)}{P(B)}$$

رابطه ۴

ما این فرمول را به نحوی تغییر می‌دهیم که مناسب مسأله کلاس‌بندی شود.

$$P(\omega_i|x)=\frac{P(x|\omega_i)P(\omega_i)}{P(x)}$$

رابطه ۵

[18] Naive Bayes

در رابطه ۵، ما می‌خواهیم احتمال داشتن برچسب w_i را برای داده ورودی داده شده محاسبه کنیم که به $P(w_i|x)$ آن احتمال پسین (Posterior Probability) می‌گویند و به $P(x|w_i)$، درست‌نمایی (Likelihood) و $P(w_i)$ احتمال پیشین (Prior Probability) گفته می‌شود. $P(x)$ یا احتمال حاشیه‌ای در محاسبات اثری ندارد و معمولاً نادیده گرفته می‌شود. در نهایت ما احتمال پسین بیشینه را به عنوان برچسب کلاس انتخاب می‌نماییم.

$$MaximumAPosteriori = argMaxP\left(x|\omega_i\right)P\left(\omega_i\right)$$

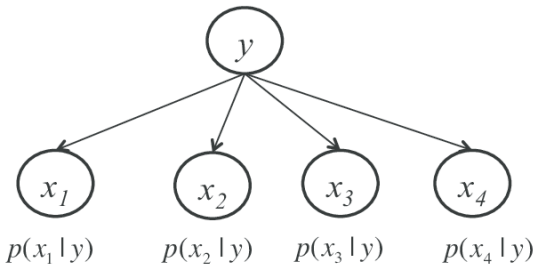
رابطه ۶

اگر کلاس‌ها دارای توزیع یکنواخت باشند یعنی احتمال پیشین آن‌ها با هم برابر باشید مسأله به مسأله Maximum Likelihood تبدیل می‌گردد.

$$MaximumLikelihood = argMaxP\left(x|\omega_i\right)$$

رابطه ۷

در روش بیزین ساده فرض بر این است که ابعاد فضای ویژگی از هم مستقل هستند و در اینصورت به جای برآورد تابع x که یک تابع n بعدی است، n تابع یک بعدی را برآورد و حاصل را در هم ضرب می‌کنیم. مثلاً در تصویر ۳۰، به جای برآورد یک توزیع توأمان چهار بعد، چهار توزیع یک بعدی را برآورد و آن‌ها را در هم ضرب می‌کنیم و حاصل برابر احتمال رخداد کلاس y خواهد بود.



تصویر ۳۰: مدل بیزین ساده

$$P(Y_1|x)=P(Y)P(x_1|Y)P(x_2|Y)P(x_3|Y)P(x_4|Y)$$
$$P(Y_2|x)=P(Y)P(x_1|Y)P(x_2|Y)P(x_3|Y)P(x_4|Y)$$
$$if P(Y_1|x)>P(Y_2|x)\Rightarrow Label=Y1$$
$$else\Rightarrow Label=Y2$$

رابطه ۸

4.7 یادگیرهای ترکیبی

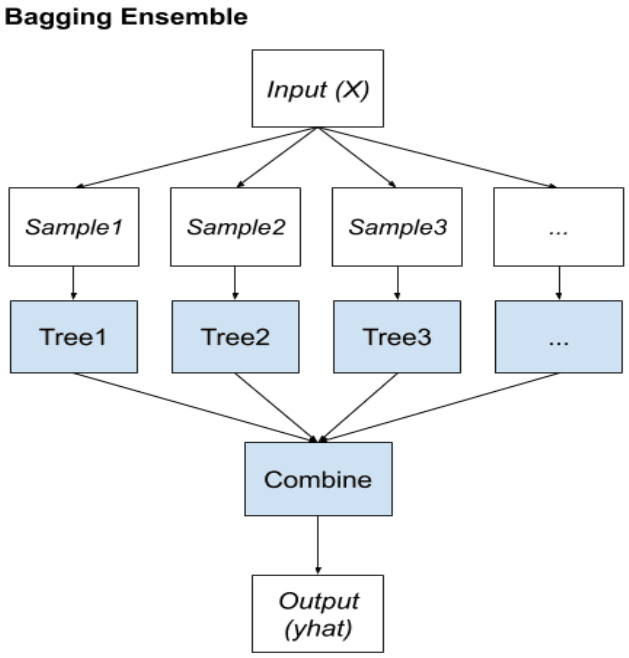
طبقه‌بندهای ترکیبی از ترکیبِ چندین طبقه‌بند استفاده می‌کنند. در واقع این طبقه‌بندها، هر کدام مدلِ خود را بر روی داده‌ها ساخته و این مدل را ذخیره می‌کنند. در نهایت برای طبقه‌بندیِ نهایی یک رای‌گیری در بین این طبقه‌بندها انجام می‌شود و آن طبقه‌ای که بیشترین میزانِ رای را بیاورد، طبقه‌ی نهایی محسوب می‌شود. روش‌های مختلفی برای ترکیب این طبقه‌بندها وجود دارد:

- ✓ **Bagging**

در این روش مدل‌های مختلف به صورت موازی آموزش می‌بینند و مستقل از هم هستند. این روش از الگوریتم Bootstrapping برای ساخت مجموعه داده‌های مختلف استفاده می‌کند و سپس بروی هر یک از این مجموعه داده‌های ایجاد شده یک الگوریتم ناپایدار (یعنی کوچک‌ترین تغییر در داده آموزشی باعث ایجاد تغییر در مرز تصمیم‌گیری می‌شود) آموزش می‌دهد و در آخر نتایج الگوریتم‌های مختلف را باهم ترکیب می‌نماید. برای کلاس‌بندی معمولاً از Majority voting یا رأی اکثریت و برای رگرسیون از میانگین‌گیری و دیگر روش‌های موجود استفاده می‌کند.

در این روش برای ساخت مجموعه داده‌های مختلف از روش Bootstrapping استفاده می‌شود. یعنی نمونه برداری با جایگزینی صورت

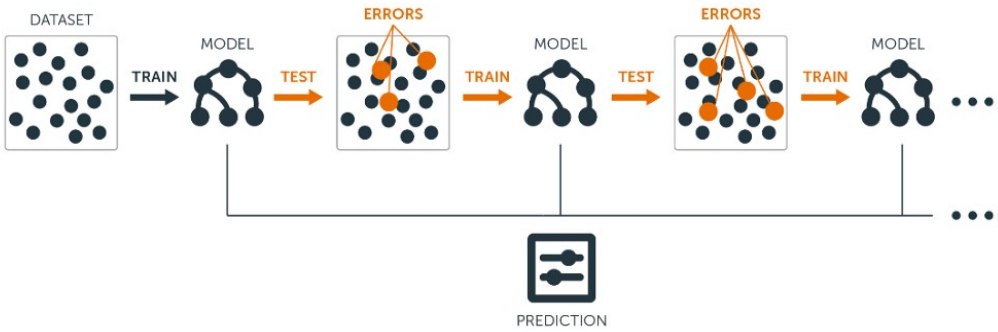
می‌پذیرد و این امکان وجود دارد که در یکی از مجموعه داده‌های تولید شده یک نمونه چندین بار تکرار شود و یک نمونه اصلاً ظاهر نشود. طبق مقاله Breiman در دیتاست‌های ایجاد شده ۶۳ درصد داده‌ها را دارد و ۳۷ درصد را ندارد. یعنی هر یک از مدل‌های ایجاد شده بروی یک حوزه خاص از داده‌ها تخصص دارند و این سبب می‌شود که دقت کلی مدل بالاتر برود. در تصویر ۳۱، روش Bagging را مشاهده می‌نمایید.



تصویر ۳۱: ترکیب یادگیرها به روش Bagging

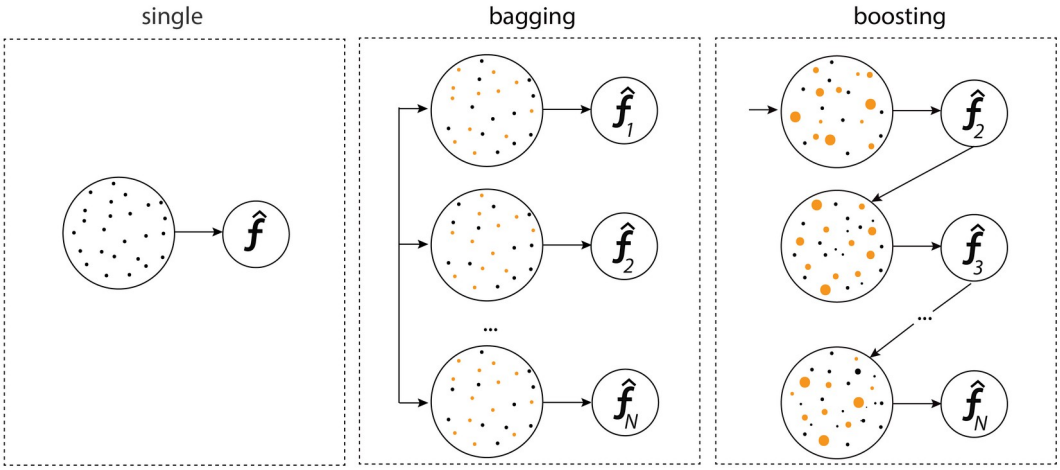
- ✓ **Boosting**

در این روش یادگیرهای ضعیف (Weak Learner)- صورت سریال در کنار هم قرار می‌گیرند. یعنی دیتاست (در شروع کار همه داده‌ها دارای وزن یکسان هستند) ابتدا به یک مدل داده می‌شود و مدل بروی آن آموزش دیده و دقتی را حاصل می‌کند. سپس به داده‌هایی که مدل جاری نتوانسته است که آن‌ها را درست پیش‌بینی یا کلاس‌بندی کند، وزن بیشتری اختصاص می‌دهد تا مدل بعدی بیشتر بروی آن‌ها متمرکز شود و این کار را تا مرحله آخر ادامه می‌دهد. در این روش با توجه به دقت حاصل شده توسط هر یک از مدل‌های مذکور به هر مدل یک وزن نسبت می‌دهد و در آخر کلاس‌بندی یا پیش‌بینی را به کمک ترکیب این وزن‌ها با یکدیگر انجام خواهد داد. تصویر ۳۲ یک نمونه از ترکیب یادگیرها به روش Boosting را نشان می‌دهد.



تصویر ۳۲: ترکیب یادگیرها به روش Boosting

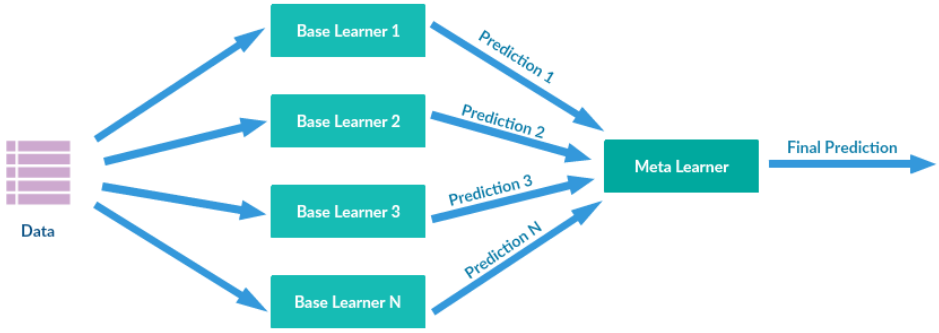
تفاوت دو روش Bagging و Boosting در تصویر ۳۳، به خوبی نشان داده شده است.



تصویر ۳۳: تفاوت روش‌های ترکیبی Bagging و Boosting

✓ Stacking

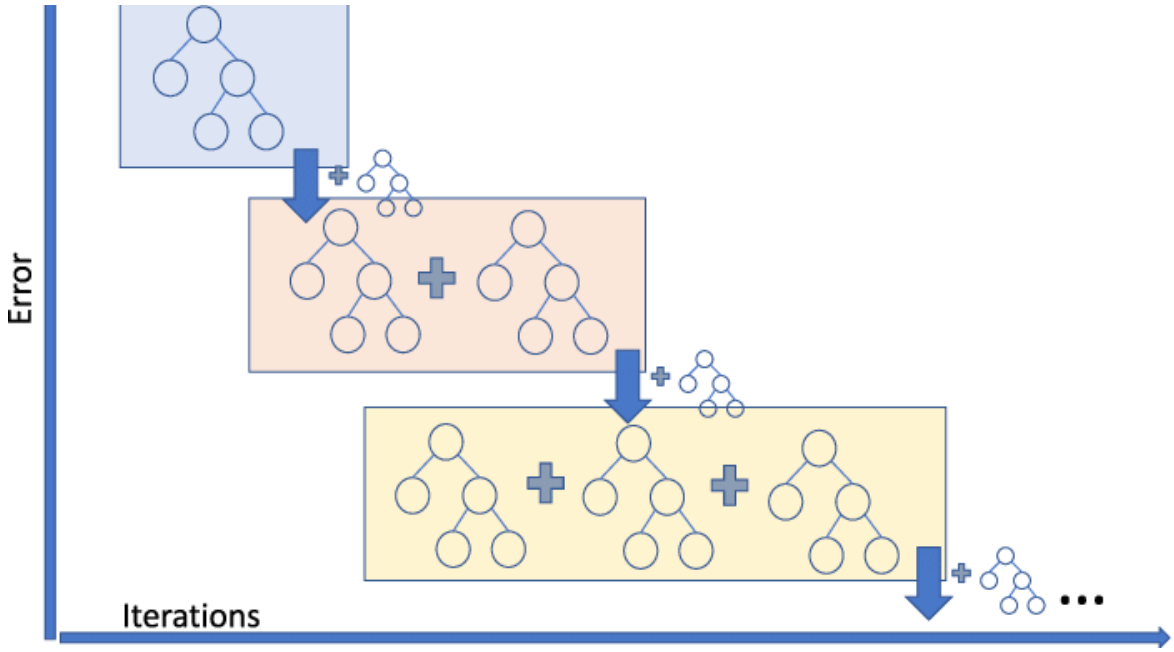
این روش درواقع از ایـده Meta-Learning اسـتفاده می‌کند و هـدف آن تـرکیب یادگیرها به صـورت لایه‌ای است. یادگیرهـای مختلف را به صـورت پشـته‌ای روی هم قـرار می‌دهـد، یعـنی داده‌هـای ورودی به چنـدین یـادگیر داده شـده و خـروجی این یادگیرهـای لایه اول به عنـوان ورودی مـدل لایه بعـدی اسـتفاده می‌شـود. این روش معمولاً نسبت به دو روش قبلی نتایج بهـتری را برمی‌گرداند. به یادگیرهـایی که در لایه اول قـرار دارنـد، یادگیرهـای پایه (Base Learner)ـ و به یـادگیر لایه دوم Meta Learner گفته می‌شود. در تصویر ۳۴ نمـونه‌ای از معمـاری Stacking را مشـاهده می‌کنید.



تصویر ۳۴: ترکیب یادگیرها به روش **Stacking**

4.7.1 گرادیان بوستینگ

یکی از الگوریتم‌های ترکیبی یادگیری ماشین است که برای ترکیب یادگیرها از معماری Boosting استفاده می‌کند. این الگوریتم به دلیل عملکرد موفقی که در مسابقات مختلف داشته است به عنوان یکی از قوی‌ترین الگوریتم‌های یادگیری ماشین شناخته می‌شود. در این الگوریتم معمولاً از درخت تصمیم به عنوان یادگیر پایه استفاده می‌شود و تعداد زیادی از این یادگیرها به صورت سریال کنار هم قرار گرفته و یک الگوریتم ترکیبی بسیار قوی را ایجاد می‌کنند. این الگوریتم دارای عملکرد قوی‌تری نسبت به درخت تصمیم و جنگل تصادفی است. همانگونه که در تصویر ۳۵ مشاهده می‌نمایید، هر چه تعداد یادگیرهای پایه‌ای که به صورت سریال کنار هم قرار می‌دهیم افزایش پیدا کند، میزان خطای مدل و متناسب با آن میزان دقت آن افزایش پیدا می‌کند.



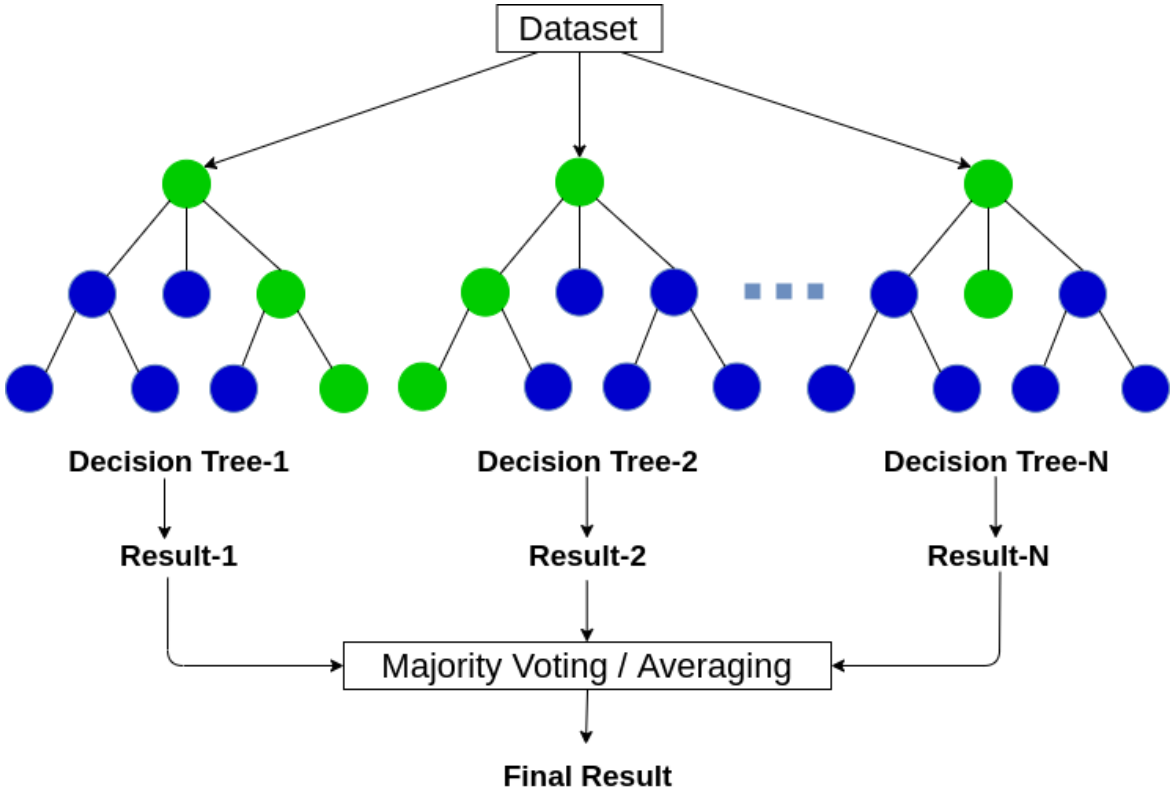
تصویر ۳۵: کاهش خطای مدل با افزایش تعداد یادگیرهای پایه در Gradient Boosting

یکی دیگر از الگوریتم‌های مشهوری که از معماری Boosting برای ترکیب یادگیرها استفاده می‌کند AdaBoost (Adaptive Boosting) است که نسبت به XGBoost دارای عملکرد ضعیف‌تری است. آدابوست یک متا الگوریتم است که به منظور ارتقاء عملکرد، و رفع مشکل رده‌های نامتوازن همراه دیگر الگوریتم‌های یادگیری استفاده می‌شود. در این الگوریتم، طبقه‌بند هر مرحله جدید به نفع نمونه‌های غلط طبقه‌بندی شده در مراحل قبل تنظیم می‌گردد. آدابوست نسبت به داده‌های نویزی و پرت حساس است؛ ولی نسبت به مشکل بیش برازش از بیشتر الگوریتم‌های یادگیری برتری دارد. طبقه‌بند پایه که در اینجا استفاده می‌شود فقط کافیست از طبقه‌بند تصادفی (۵۰٪) بهتر باشد و به این ترتیب بهبود عملکرد الگوریتم با تکرارهای بیشتر بهبود می‌یابد. حتی طبقه‌بندهای با خطای بالاتر از تصادفی با گرفتن ضریب منفی عملکرد کلی را بهبود می‌بخشند. این روش معمولاً از decision stump ها به عنوان یادگیر پایه استفاده می‌کند که همان درخت تصمیم است که فقط یک نود ریشه و دو برگ

دارد. این روش نسبت به XGBoost حساسیت بالاتری نسبت به نویز نشان می‌دهد و دقت آن هم معمولا پایین‌تر است.

4.7.2 جنگل تصادفی

یکی از مشهورترین و کارآمدترین الگوریتم‌های یادگیری ماشین است. جنگل تصادفی یک یادگیر ترکیبی است که ترکیب یادگیرها را با استفاده از معماری Bagging انجام می‌دهد. الگوریتم پایه‌ای که در جنگل تصادفی مورد استفاده قرار می‌گیرد، درخت تصمیم است. یعنی یک جنگل تصادفی از کنار هم قرار دادن تعدادی درخت تصمیم ساده (دارای دقت و عملکرد پایین) حاصل می‌شود (تصویر ۳۶) و این جنگل می‌تواند تصمیمات بهتری را نسبت به یک درخت به تنهایی اخذ کند.



تصویر ۳۶: جنگل تصادفی از کنار هم قرار دادن چندین درخت تصمیم به صورت موازی حاصل شده است.

فرآیند آموزش و ارزیابی جنگل تصادفی به این ترتیب است که به هر کدام از درخت‌های تصمیم یک زیر مجموعه از داده‌ها تزریق می‌شود. هر کدام از الگوریتم‌ها عملیات یادگیری را انجام می‌دهند. در هنگام پیش‌بینی، یعنی وقتی که یک سری داده‌ی جدید به الگوریتم، جهت پیش‌بینی داده می‌شود، هر کدام از این الگوریتم‌های یادگرفته شده، یک نتیجه را پیش‌بینی می‌کنند. الگوریتمِ جنگلِ تصادفی در نهایت، می‌تواند با استفاده از رای‌گیری، آن طبقه‌ای را که بیشترین رای را آورده است انتخاب کرده و به عنوان طبقه‌ی نهایی جهت انجام عملیات طبقه‌بندی قرار دهد.

5 پیاده‌سازی و محاسبه برازش نکویی روش ارائه شده

برای پیاده‌سازی روش از زبان برنامه‌نویسی پایتون استفاده شده است. در این پیاده‌سازی داده‌ها را به سه دسته آموزش[19]، ارزیابی[20] و اعتبارسنجی[21] تقسیم شده است. نحوه تقسیم داده‌ها هم صورت زیر است:

- **آموزش**
  - ۶۵٪
  - ۱۹۳ نمونه
- **ارزیابی**

[19] Training

[20] Testing

[21] Validating

- ۲۰٪
- ۵۹ نمونه

- **اعتبارسنجی**
  - ۱۵٪
  - ۴۵ نمونه

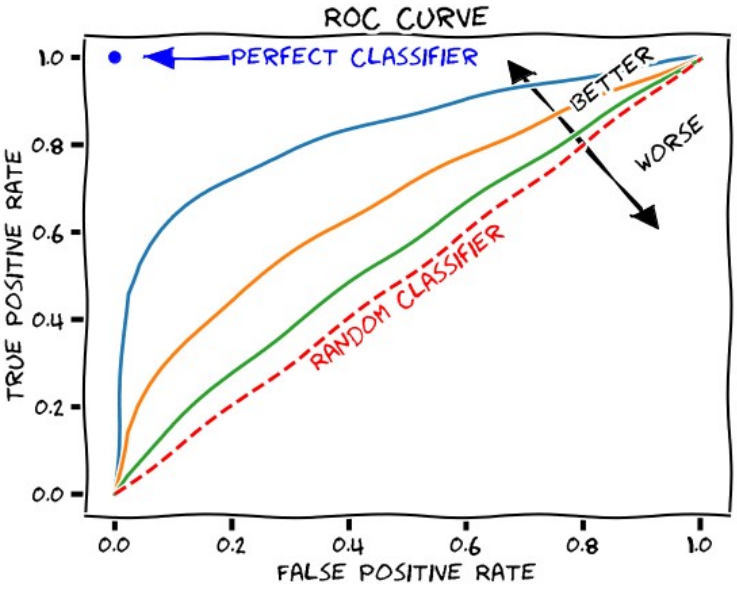
برای ارزیابی عملکرد مدل یا برازش نکویی[۲۲] از معیارهای زیر استفاده شده است:

به عنوان مثال قصد داریم طی یک روند یادگیری نظارت‌شده مدلی برای پیش‌بینی بیماری سرطان ایجاد کنیم. برای آموزش مدل یک جامعه آماری تهیه می‌کنیم که تعدادی بیمار واقعاً سرطان دارند و تعدادی هم ندارند و مدل را به کمک بخش آموزش، ایجاد کرده و بر روی بخش آزمون یا ارزیابی آن را اجرا می‌کنیم تا میزان خطا یا دقت مدل را بررسی کنیم.

- **صحت (Accuracy)**
  - منعکس کننده کارایی و قابل اعتماد بودن مدل.
  - یعنی مقدار اندازه‌گیری شده چقدر به مقدار واقعی نزدیک است برای accuracy باید precision بالا باشد ولی برعکسش لزوماً برقرار نیست. بالا بودن بایاس و واریانس به معنای accuracy کم است.

- **نمودار ROC (Receiving Operating Characteristic)**
  - تجزیه و تحلیل ROC می تواند برای تعیین مستقل یک مدل شبکه بهینه از نظر هزینه یا توزیع کلاس استفاده گردد. نمونه‌ای از این نمودار را در تصویر ۳۷ مشاهده می‌کنید.

[22] Goodness of fit



تصویر ۳۷: نمودار ROC

- **نمودار عملکرد**
  - بررسی قابلیت تعمیم مدل و اینکه مدل دچار بیش‌برازش شده است یا خیر؟
- **دقت (Precision)**
  - برای اندازه‌گیری‌های متوالی از یک مقدار میزان نزدیک بودن مقدارهای اندازه‌گیری را نشان می‌دهد. مثلاً اگر یک ساعت هر روز فقط ۲ ساعت جلو رود مقدار accuracy یا صحتش پایین است ولی مقدار precision آن بالاست.
- **پوشش یا بازیابی (Recall or Sensitivity)**
  - کسری از جوابهای مثبت که درست تشخیص داده شده‌اند درصد افرادی که طبق پیش‌بینی مدل سرطان دارند و در دنیای واقعی هم سرطان دارند.
- **Specificity**

- کسری از جواب‌های منفی که به درستی تشخیص داده شده‌است مثلاً درصد افرادی که طبق پیش‌بینی مدل سرطان ندارند و در دنیای واقعی هم سرطان ندارند.

- **F1-Measure**
  - میانگین هندسی دقت و پوشش.

برای درک هر چه بهتر این معیارها و ارتباط آن‌ها با ماتریس درهم[23] و فرمول هر یک از معیارها تصویر ۳۸ را بررسی نمایید.

		Predicted Class: Positive	Predicted Class: Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

تصویر ۳۸: ماتریس درهم و معیارهای محاسبه عملکرد مدل

- **True Positive (TP)**
  - مواردی که مدل پیش‌بینی کرده که بیماراند و واقعا بیمار بوده‌اند.
- **False Positive (FP)**

[23] Confusion Matrix

- موارد‌ی که مدل پیش‌بینی کرده بیماراند، اما بیمار نبوده‌اند.

× **False Negative (FN)**

- موارد‌ی که مدل پیش‌بینی کرده که سالم‌اند، اما بیمار بوده‌اند.

× **True Negative (TN)**

- موارد‌ی که مدل پیش‌بینی کرده که سالم‌اند و در واقعیت هم سالم بوده‌اند.

6 گزارش نتایج

برای پیاده‌سازی این مقاله و مسئله از زبان برنامه‌نویسی پایتون و ابزارهایی که این زبان برای پردازش و کار با داده‌ها در اختیارمان می‌گذارد، استفاده کرده‌ایم. در ادامه لیستی از کتابخانه‌های استفاده شده و کاربرد آن در پروژه حاضر ارائه شده است.

- **خواندن و وارد کردن داده‌ها**
  - ✓ Pandas
- **پیش‌پردازش و انجام پردازش‌های اولیه بروی داده‌ها**
  - ✓ Pandas
  - ✓ Sklearn
  - ✓ Numpy
  - ✓ Scipy
- **بصری‌سازی و آنالیز داده اکتشافی**
  - ✓ Matplotlib

- ✓ Seaborn
- ✓ Plotly

- **ساخت و آموزش مدل و الگوریتم‌های استفاده شده**
  - ✓ Sklearn
  - ✓ xgboost
  - ✓ lightgbm
- **گزارش نتایج**
  - ✓ Sklearn
- **پیاده‌سازی و کار با AHP**
  - ✓ AHP
- **ذخیره‌سازی و سریالایز کردن نتایج**
  - ✓ Pickle
- **پیاده‌سازی و بردن مدل بر بستر وب**
  - ✓ Streamlit

6.1 فرآیند تحلیل سلسله مراتبی

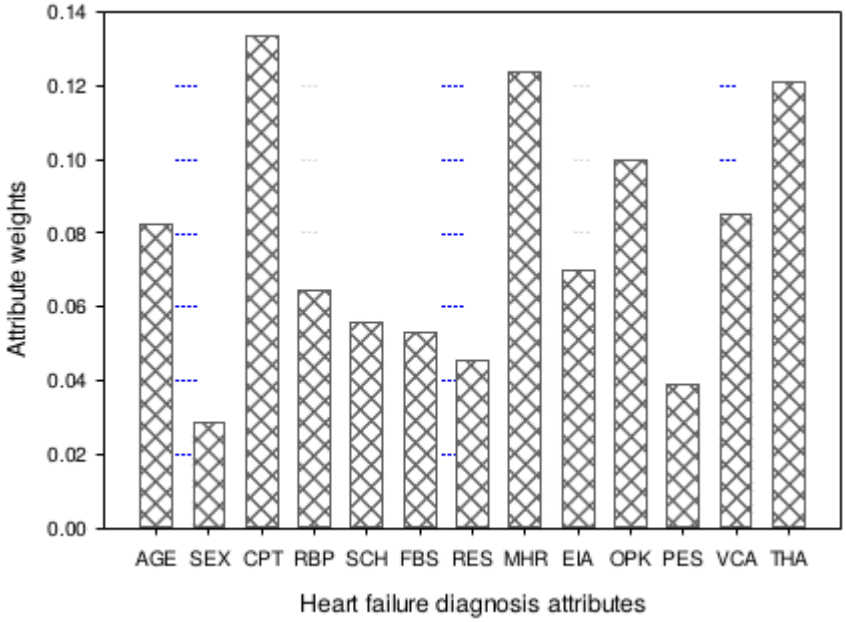
طبق گزارش ارائه شده در مقاله بعد از رتبه‌بندی ۱۳ ویژگی موجود در مجموعه داده و تشکیل ماتریس مقایسه دو به دو برای آن و حل مسأله با استفاده از روش AHP، مقدار وزن برای هر یک از ویژگی‌ها محاسبه شده و مطابق تصویر ۳۹ می‌باشد.

S/N	Attribute	Attribute Weight	Alternative weight (ranging from 1 – 4)			
			1	2	3	4
1	AGE	0.0822	0.0082	0.0164	0.0247	0.0329
2	SEX	0.0287	0.0096	0.0191	-	-
3	CPT	0.1333	0.0133	0.0267	0.0400	0.0533
4	RBP	0.0645	0.0065	0.0129	0.0194	0.0258
5	SCH	0.0559	0.0056	0.0112	0.0168	0.0224
6	FBS	0.0531	0.0177	0.0354	-	-
7	RES	0.0452	0.0075	0.0151	0.0226	-
8	MHR	0.1235	0.0206	0.0412	0.0618	-
9	EIA	0.0696	0.0232	0.0464	-	-
10	OPK	0.0997	0.0166	0.0332	0.0499	-
11	PES	0.0386	0.0064	0.0129	0.0193	-
12	VCA	0.0849	0.0085	0.0170	0.0255	0.0340
13	THA	0.1208	0.0201	0.0403	0.0604	-

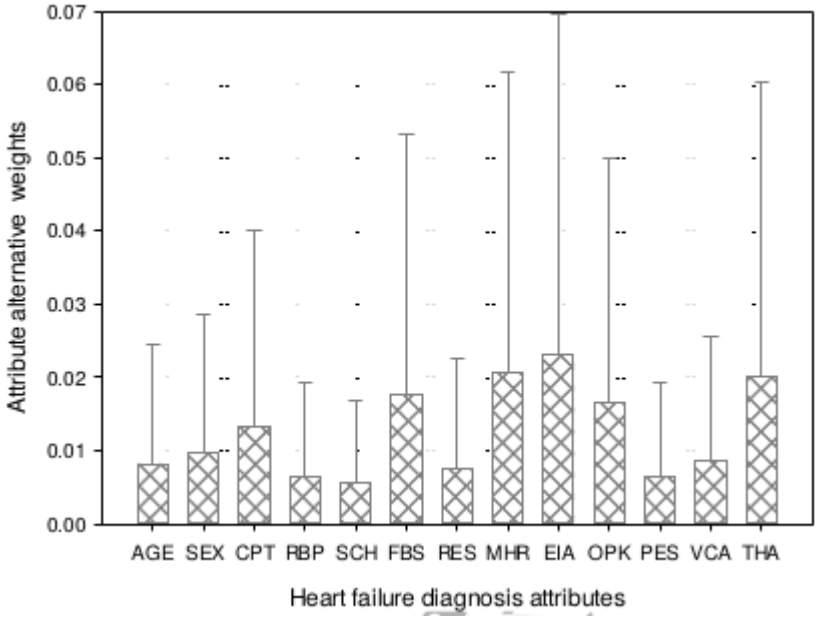
MANUSCR

تصویر ۳۹: وزن‌های به دست آمده توسط AHP برای هر یک از ویژگی‌ها و زیرویژگی‌های موجود در مجموعه داده

همانطور که در تصویر ۳۹ مشاهده می‌شود، هر ویژگی دارای دو وزن محلی (وزن هر ویژگی) و سراسری (وزن مقادیر هر ویژگی) است. مثلاً برای جنسیت وزن محلی برابر ۰٫۰۲۸۷ و وزن سراسری برای آن، برای مقدار مذکر، ۰٫۰۱۹۱ و برای مونث برابر ۰٫۰۰۹۶ است. پس مقدار وزن محلی برای هر ویژگی برابر با مجموع وزن‌های سراسری آن است و مجموع وزن محلی تمام ویژگی‌ها باید برابر یک شود. میزان اهمیت ویژگی‌ها و زیرویژگی‌ها در تصویر ۴۰ و ۴۱ آمده است.



تصویر ۴۰: مقایسه اهمیت ویژگی‌ها



تصویر ۴۱: حد بالا و پایین ضرایب هر مقدار از ویژگی‌ها

همانطور که از تصاویر ۴۰ و ۴۱ قابل استنباط است، CPT یا نوع درد در سینه و MHR یا بالاترین میزان ضربان قلب به ترتیب دارای بالاترین اهمیت و ضریب هستند و ویژگی‌های جنسیت و PES یا حداکثر شیب جذب کمترین اهمیت را دارند. ضرایب سراسری به دست آمده به جای مقادیر دیتاست قرار داده شده و مدل به کمک این ضرایب آموزش دیده است. برای تنظیم هایپرپارامترهای مسأله مثل تعداد لایه‌ها و نرخ یادگیری و توابع فعالساز و هزینه استفاده شده از GridSearchCV استفاده می‌کنیم. در این روش مقادیر مختلف پارامترها را در تعیین و مدل را به کمک این مقادیر و در حالات مختلف آموزش می‌دهیم و حالتی که در آن مدل به بهترین دقت روی داده‌های آموزش یا ارزیابی می‌رسد را به عنوان معیاری برای انتخاب بهترین پارامترها در نظر می‌گیریم.

برای مقایسه و بررسی اینکه عمل‌کرد مدل مورد بررسی با بهبود همراه است یا خیر، همزمان از دو مدل ترکیب AHP و شبکه عصبی مصنوعی و شبکه عصبی به تنهایی استفاده کرده‌ایم.

6.2 گزارش نتایج

پیاده‌سازی در فایلی تحت عنوان Cleveland Heart Disease Dataset Analysis قرار دارد. بعد از وارد کردن داده‌ها، در بخش اول پیاده‌سازی، آنالیز اکتشافی داده‌ها[۲۴] انجام شده است، که به سؤالات زیر پاسخ می‌دهیم:

- تعداد نمونه‌ها و ویژگی‌ها چقدر است؟
- توزیع مقدار هر یک از ویژگی‌ها چگونه است (برای ویژگی‌های رسته‌ای[۲۵])؟
- هر ویژگی در چه فضایی پخش شده است (برای ویژگی‌های پیوسته)؟
- مقدار حداکثر و حداقل داده‌ها چقدر است؟

[24] Exploratory Data Analysis

[25] Categorical

- میانگین و واریانس و دیگر ویژگی‌های آماری داده‌ها چقدر است؟
- آیا مقادیر خالی یا گم شده[26] در داده‌ها وجود دارد؟
- آیا در دیتاست، داده پرت[27] وجود دارد؟
- آیا بین ویژگی‌های موجود همبستگی[28] وجود دارد؟
- آیا ارتباط معناداری بین یک ویژگی خاص و برچسب نهایی وجود دارد؟

پاسخ به سؤالات مطرح شده می‌تواند ما را در شناخت هر چه بیشتر و بهتر داده‌ها یاری کند و دید و بینش عمیق‌تری از داده‌ها در اختیار ما بگذارد.

مرحله بعدی پیش‌پردازش داده‌ها است که وابستگی بالایی به کیفیت کار در مرحله قبل دارد، یعنی هر چه ما داده‌ها را بهتر بشناسیم، می‌توانیم از روش‌های بهتر و متناسب‌تری برای پیش‌پردازش و آماده‌سازی داده‌ها برای تزریق به مدل استفاده کنیم. کارهای انجام شده در این مرحله را می‌توان به کارهای زیر دسته‌بندی کرد:

- **مدیریت مقادیر مفقود یا گم شده**
  - اولین و ساده‌ترین استراتژی حذف داده‌ای است که در آن یکی از ویژگی‌ها مقدار ندارد. اما اگر در تعداد داده‌ها دچار محدودیت هستیم، می‌توانیم از استراتژی‌های مختلفی مثل قرار دادن مقدار میانگین یا مُد ویژگی‌ها به جای مقدار مفقود شده استفاده کنیم.

- **باینری کردن یا دسته‌ای کردن ویژگی‌های رسته‌ای**

---

[26] Imputer, Null Values

[27] Outlier

[28] Correlation

- الگوریتم‌های یادگیری ماشین با داده‌های عددی کار می‌کنند و در دیتاست ممکن مقادیر رسته‌ای مثل (کم، متوسط، زیاد) یا باینری مثل جنسیت (مرد، زن) داشته باشیم. برای مدیریت کردن این مقادیر می‌توانیم از یک نگاشت ساده عددی مثلاً مرد متناسب با مقدار ۱ و زن متناسب با مقدار ۰ یا روش بهتری مثل One Hot Encoding استفاده کنیم.

- **نرمال‌سازی یا استانداردسازی داده‌ها**

- ویژگی‌های مختلف در مجموعه داده، دارای مقادیر مختلفی از نظر نوع داده (کمی یا کیفی)، مقدار داده (پیوسته یا گسسته)، بازه داده (نمره بین ۰ تا ۲۰ و قد بین ۰ تا ۲۵۰) و یا در نوع کمیت اندازه‌گیری است (مثلاً کمیت دما درجه سلسیوس است و کمیت وزن کلیوگرم یا قد سانتی‌متر). این تفاوت‌های محاسباتی می‌تواند منجر به تفسیر و نتیجه‌گیری اشتباه مدل از داده‌ها شود (تفکیکی بین داده‌ها ایجاد کند که منعکس کننده تفاوت ذاتی داده‌ها نیست، بلکه به واسطه متفاوت بودن کمیت یا ابعاد و بازه تغییرات داده‌ها بوجود آمده است). برای حال این مشکل از نرمال‌سازی یا نگاشت داده به بازه بین صفر و یک (با استراتژی‌های مختلفی مثل تقسیم بر بزرگترین داده یا روش MinMax و ...) یا استانداردسازی یا نگاشت داده‌ها به بازی بین منفی یک تا مثبت یک (ایجاد یک توزیع نرمال یا گوسی از داده‌ها با میانگین صفر و انحراف معیار یک).

- **جدا کردن داده‌ها**

- برای ارزیابی و میزان کارآیی مدل باید داده‌ها را به دسته‌های آموزش، اعتبارسنجی و ارزیابی تقسیم کنیم تا دقت به دست آمده دقت خوب و قابل اعتمادی باشد.

یعنی کارایی مدل را با داده‌هایی بررسی می‌کنیم که مدل تا الان آن‌ها را مشاهده نکرده است.

بعد از پیش‌پردازش و جداسازی داده‌ها، حال باید مدل را آموزش دهیم. یکی از مهم‌ترین مراحل آموزش یک مدل، تنظیم هایپرپارامترهاست. یعنی اون پارامترهایی به صورت خارجی و توسط ما تعیین می‌شوند را به گونه‌ای تعیین کنیم که مدل بروی داده‌های ما به عمل‌کرد خوبی برسد. در شبکه‌های عصبی هایپرپارامترها می‌توانند تعداد لایه‌ها، تعداد واحدهای نورونی موجود در هر لایه، نوع توابع فعالیت یا فعال‌ساز، نوع توابع زیان[۲۹]، مقدار نرخ یادگیری[۳۰]، نوع نرخ یادگیری (ثابت یا انطباقی[۳۱]). بعد از تعیین این هاپیرپارامترها، مدل را به کمک داده‌های آموزشی، آموزش داده و مدل یاد گرفته شده را ذخیره می‌کنیم (تا بتوانیم آن را به سیستم‌ها و برنامه‌های دیگر انتقال دهیم).

بعد از فاز آموزش، دقت مدل را توسط داده‌های تست یا ارزیابی مورد بررسی قرار می‌دهیم و با مقایسه دقت به دست آمده و اختلاف بین دقت تست و آموزش می‌توانیم تحلیلی بروی عمل‌کرد مدل داشته باشیم. مثلاً اینکه:

- آیا دچار بیش‌برازش[۳۲] شده است یا کم برازش[۳۳]؟
- آیا مدل دارای قدرت تعمیم[۳۴] خوبی است یا خیر؟

---

[29] Loss function

[30] Learning rate

[31] Constant or Adaptive

[32] Overfitiing

[33] Underfitting

[34] Generalizability

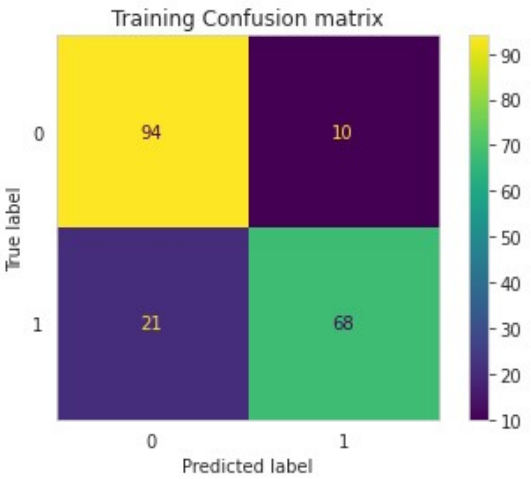
اگر مدل مورد بررسی دارای دقت و قدرت تعمیم خوبی بود، نتایج به دست آمده را با معیارهای مختلفی که معرفی کردیم، ارائه داده و آن را ذخیره می‌کنیم و در غیر اینصورت فرآیند ساخت و ارزیابی مدل مجدداً انجام می‌شود.

آخرین مرحله کار هم ارائه و بارگذاری مدل بروی بستری است که ذی نفع یا کاربران نهایی از آن استفاده کنند و بتوانند با مدل ارائه شده تعامل داشته باشند یعنی ورودی‌های مورد نظر خود را به مدل داده و مدل به آن بگوید که در خطر دچار شدن به عارضه نارسایی قلبی قرار دارند یا خیر؟

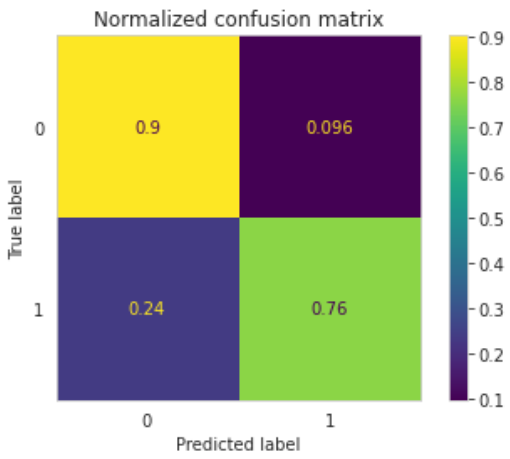
برای این مرحله از بستر وب استفاده کرده‌ام و پروژه ارائه شده را می‌توانید در این لینک مشاهده نموده و با مدل ارائه شده تعامل داشته باشید.

6.3 ارائه نتایج

در ادامه ماتریس درهم و نمودارهای مقایسه عمل‌کرد بین مدل ترکیبی شبکه عصبی و AHP و مدل شبکه عصبی تنها آمده است. نتایج به دست آمده توسط ما با نتایج مقاله تفاوت کمی دارد ولی در کار ما نیز استفاده از مدل ترکیبی سبب بهبود نتیجه شده است. در تصویر ۴۲ و ۴۳ ماتریس درهم مدل ترکیبی بروی داده‌های آموزشی را مشاهده می‌نمایید.

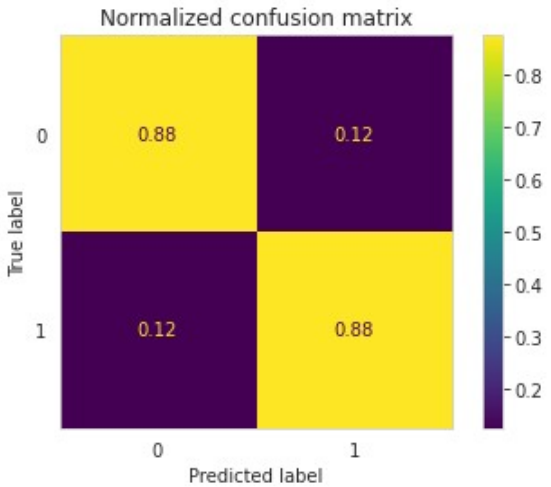


تصویر ۴۲: ماتریس درهم مدل ترکیبی شبکه عصبی و AHP روی داده‌های آموزشی

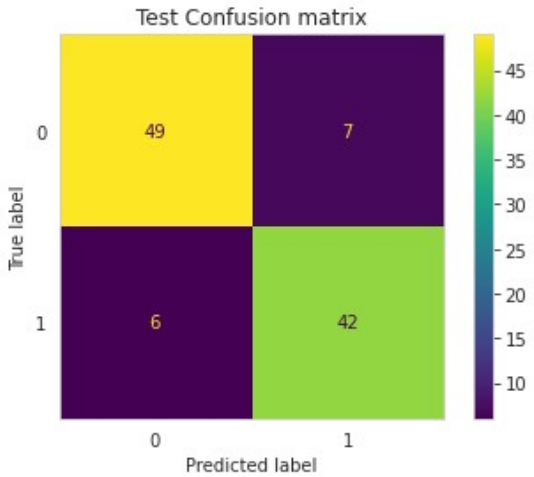


تصویر ۴۳: ماتریس درهم مدل ترکیبی شبکه عصبی و AHP روی داده‌های آموزشی

و در تصویر ۴۴ و ۴۵ نتایج مدل بروی داده‌های تست را مشاهده می‌نمایید.



تصویر ۴۴: ماتریس درهم مدل ترکیبی شبکه عصبی و AHP روی داده‌های تست

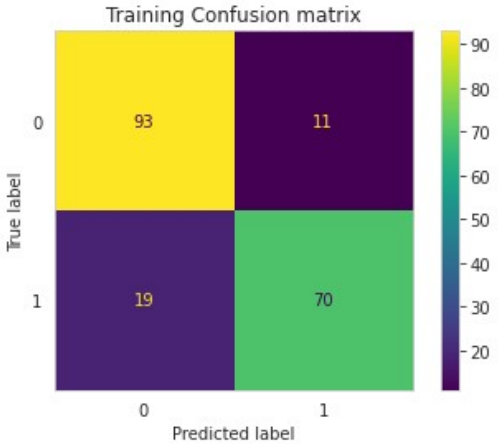


تصویر ۴۵: ماتریس درهم مدل ترکیبی شبکه عصبی و AHP روی داده‌های تست

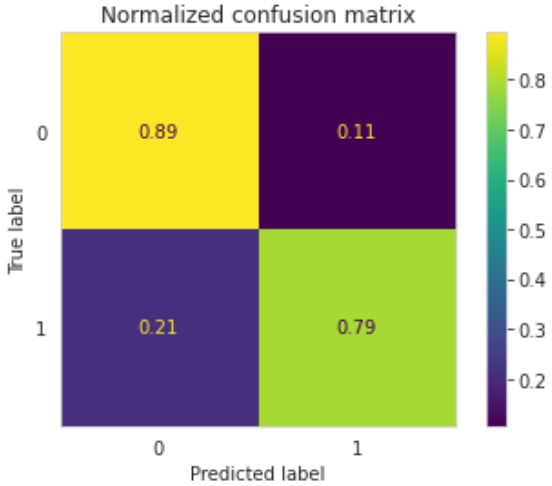
طبق نتایج حاصل شده دقت مدل برای داده‌های آموزشی ۸۴,۴۵ و برای داده‌های تست نتایج معیارهای مختلف عملکردی مدل در جدول زیر قابل مشاهده است.

معیار	مقدار
Accuracy	0.875
Precision	0.875
Recall	0.857
F1-measure	0.865
Specificity	0.890

در ادامه همین نتـایج بـرای مـدل شـبکه عصـبی تنها ارائه شـده اسـت. در تصـویر ۴۶ و ۴۷، ماتریس درهم مدل شبکه عصبی بروی داده‌های آموزشی نشان داده شده است.

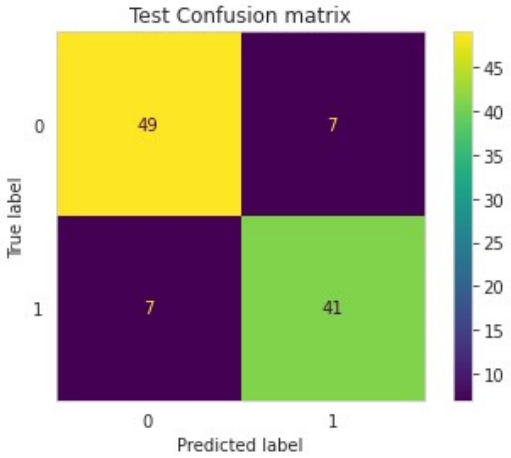


تصویر ۴۷: ماتریس درهم مدل شبکه عصبی روی داده‌های آموزشی

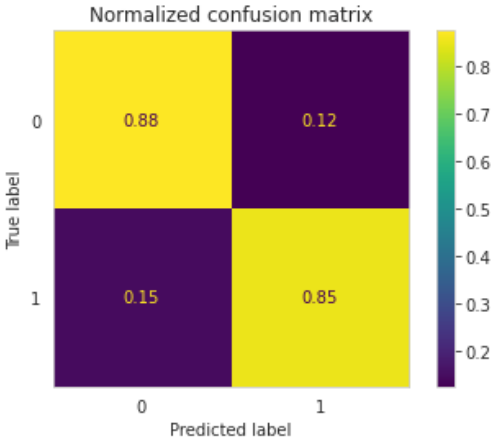


تصویر ۴۶: ماتریس درهم مدل شبکه عصبی روی داده‌های آموزشی نرمال شده

و ماتریس درهم مدل روی داده‌های تست مطابق تصویر۴۸ و ۴۹ خواهد بود.



تصویر ۴۸: ماتریس درهم مدل شبکه عصبی روی داده‌های تست

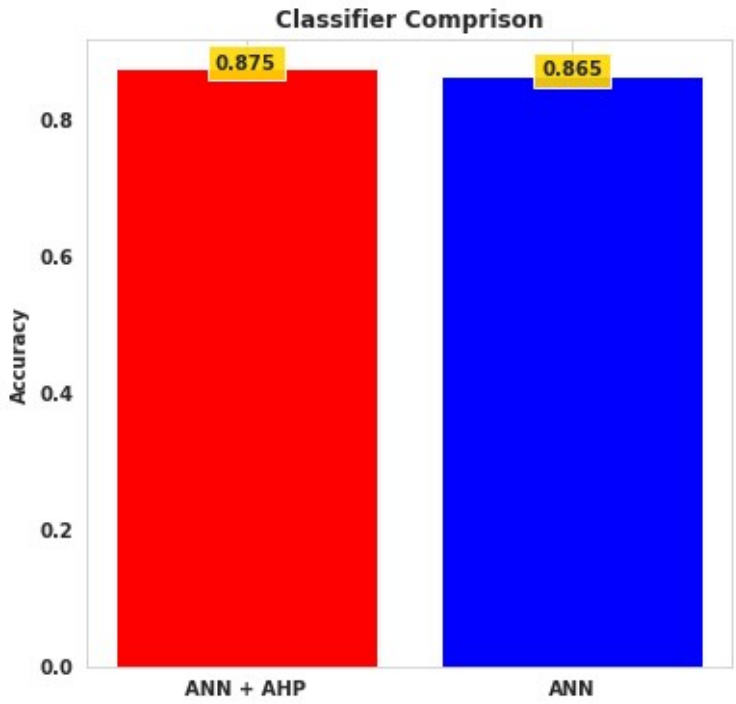


تصویر ۴۹: ماتریس درهم مدل شبکه عصبی روی داده‌های تست نرمال شده

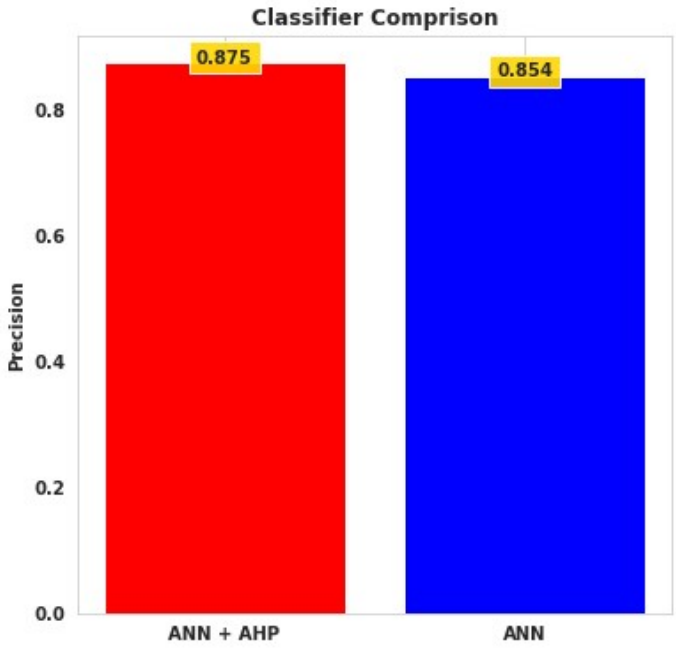
و جدول گزارش معیارها برای شبکه عصبی مطابق زیر می‌باشد:

معیار	مقدار
Accuracy	0.865
Precision	0.854
Recall	0.854
F1-measure	0.854
Specificity	0.875

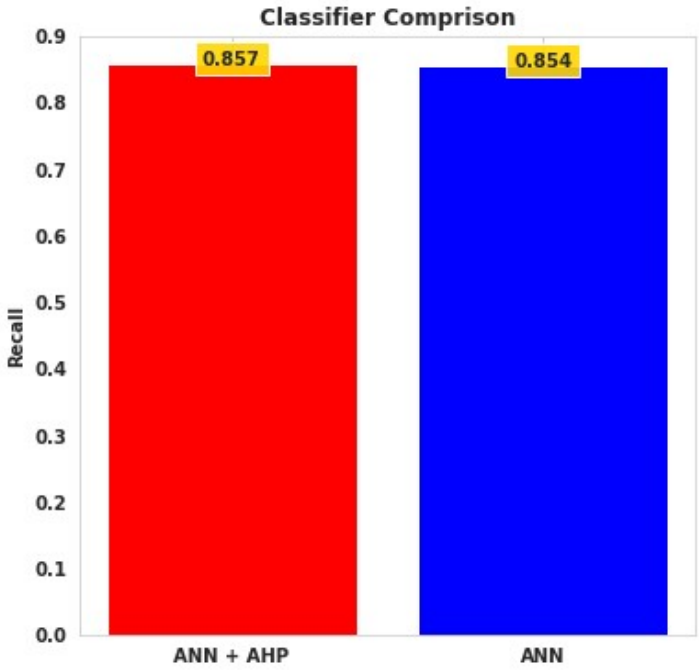
در ادامه مقایسه عملکردهای ارائه شده را نمایش داده‌ایم.



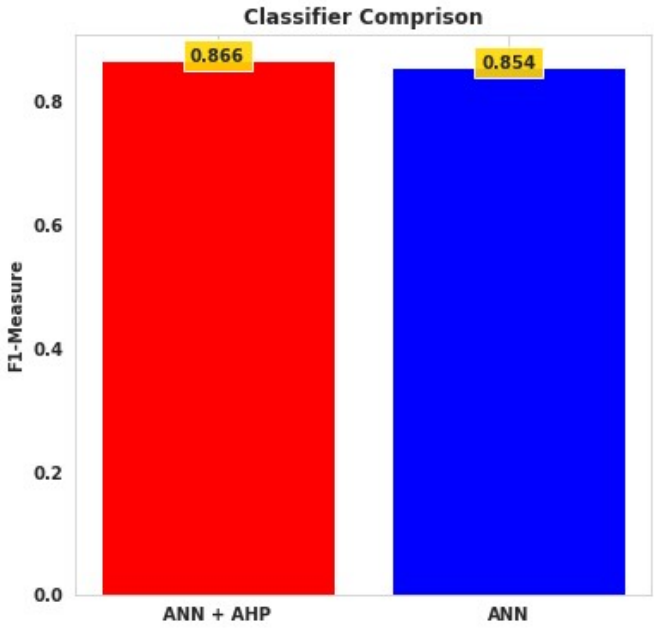
تصویر ۵۰: مقایسه Accuracy مدل ترکیبی و شبکه عصبی تنها



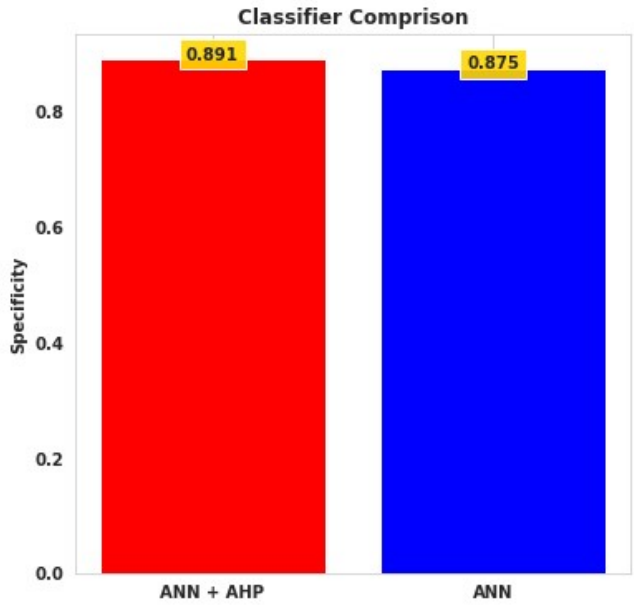
تصویر ۵۱: مقایسه Precision مدل ترکیبی و شبکه عصبی تنها



تصویر ۵۲: مقایسه Recall مدل ترکیبی و شبکه عصبی تنها

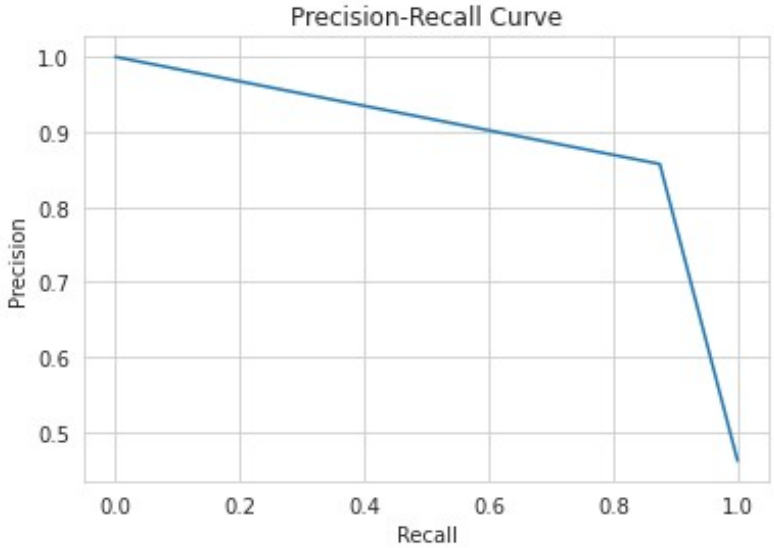


تصویر ۵۳: مقایسه F1-measure مدل ترکیبی و شبکه عصبی تنها

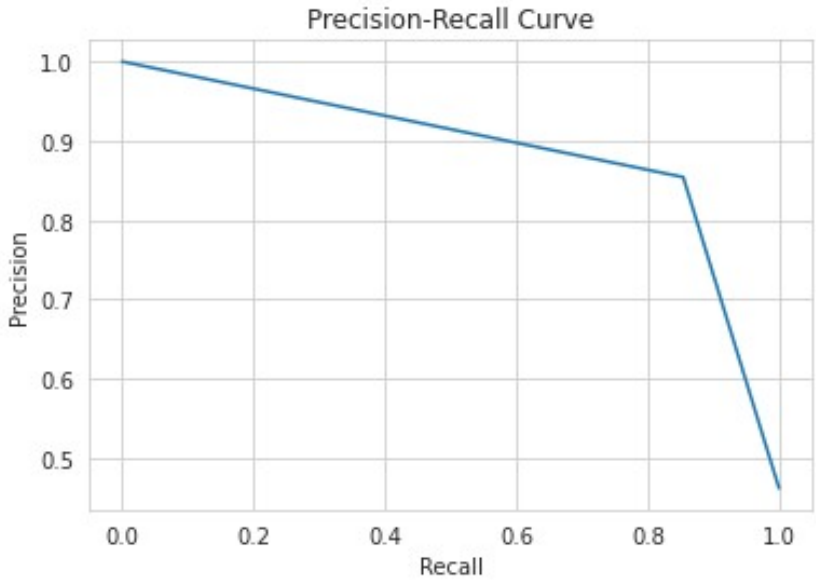


تصویر ۵۴: مقایسه Specificity مدل ترکیبی و شبکه عصبی تنها

و در تصـــویر ۵۵ و ۵۶، precision recall curve این دو روش نشـــان داده شـــده اســـت. مساحت زیر این نمودار تحت عنـوان AUC شـناخته می‌شـود و می‌توانـد بـه عنـوان یـک معیـار کارایی استفاده گردد.

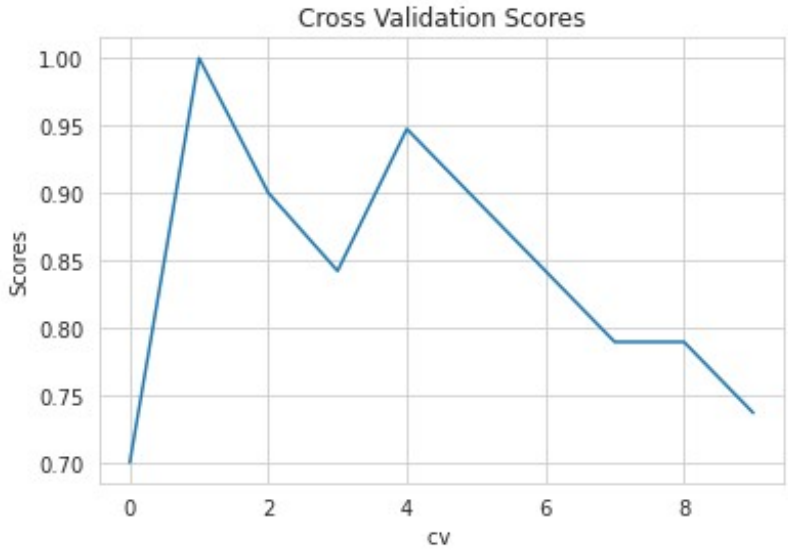


تصویر ۵۵: نمودار PRC برای روش ترکیبی شبکه عصبی و AHP

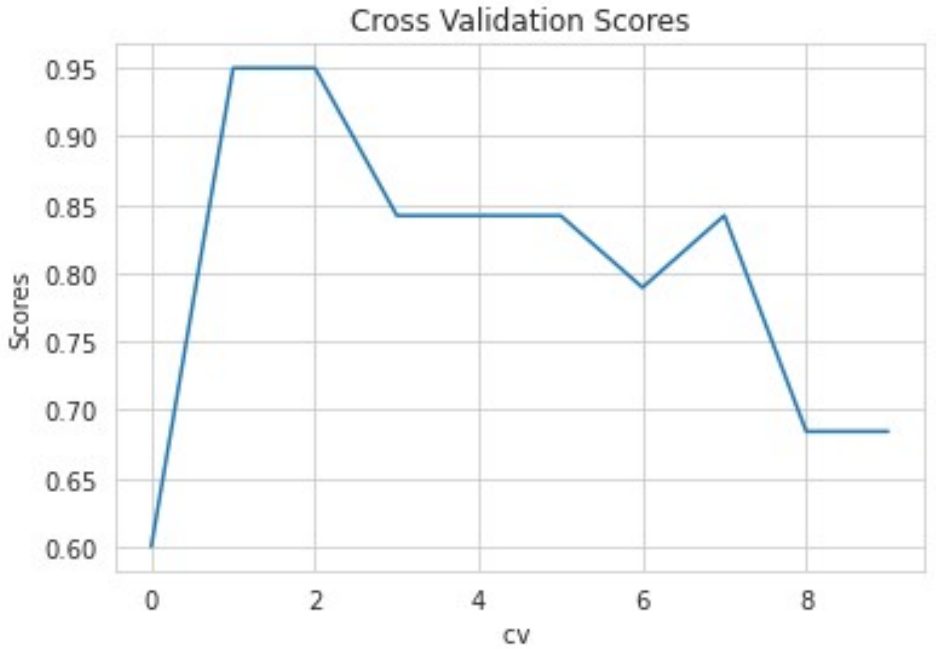


تصویر ۵۶: نمودار PRC شبکه عصبی

برای مقایسه بهتر و کاراتر نتایج به کمک k-fold cross validation چندین مرتبه الگوریتم را اجرا کرده و نتایج حاصل را میانگین می‌گیریم و این دقت میانگین و انحراف معیار هم می‌توانند یک مقایسه عالی و قابل اتکا در اختیارمون می‌گذارد، که در تصاویر ۵۷ و ۵۸ نتایج حاصل از این اجراهای مختلف و در ادامه نیز مقایسه آن‌ها آورده شده است.



تصویر ۵۷: نتایج مدل ترکیبی حاصل از ۱۰ fold



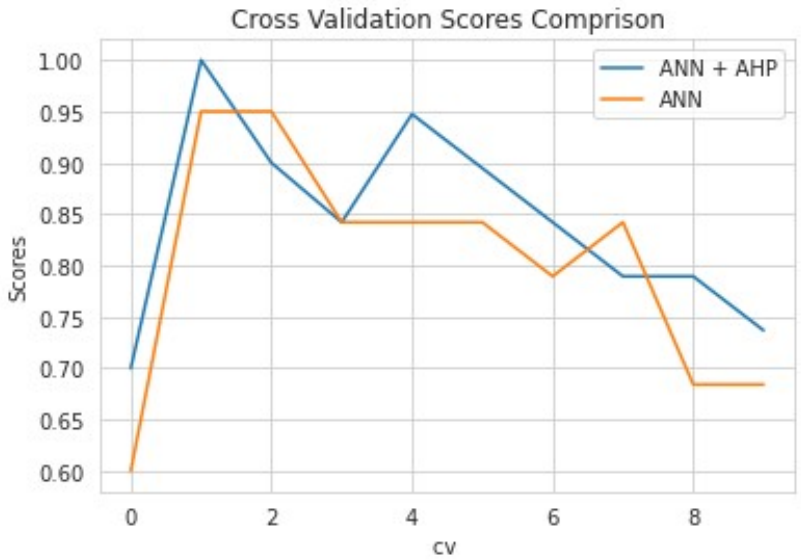
تصویر ۵۸: نتایج مدل شبکه عصبی حاصل از ۱۰ fold

و مقایسه این دو با هم در تصویر ۵۹ آمده است و مشاهده می‌شود که روش ترکیبی در اجراهای مختلف نیز معمولاً نتایج بهتری برمی‌گرداند. یعنی می‌توان گفت که نتیجه به دست آمده قابل اعتماد و اتکا می‌باشد و استفاده از مدل ترکیبی سبب بهبود کلی تصمیم‌گیری شده است. و در جدول زیر مقایسه دقت میانگین و انحراف معیار روش آمده است.

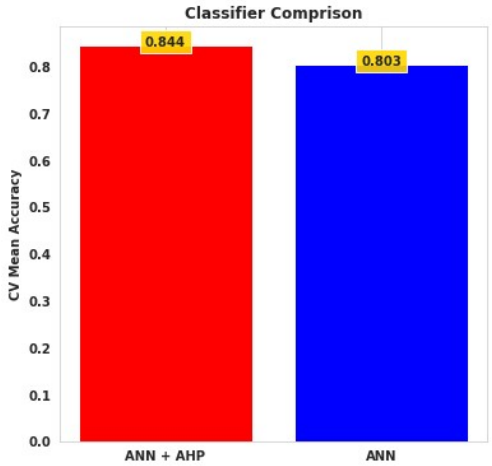
مدل	میانگین دقت	انحراف معیار
مدل ترکیبی	۰.۸۴۴۲	۰.۰۸۸
مدل شبکه عصبی تنها	۰.۸۰۲۶	۰.۱۰۹

همانگونه که مشاهده می‌گردد، از نظر میانگین عملکرد نیز مدل ارائه شده دارای دقت بهتری است و از نظر انحراف معیار هم مدل ترکیبی انحراف معیار کمتری دارد. میانگین دقت و

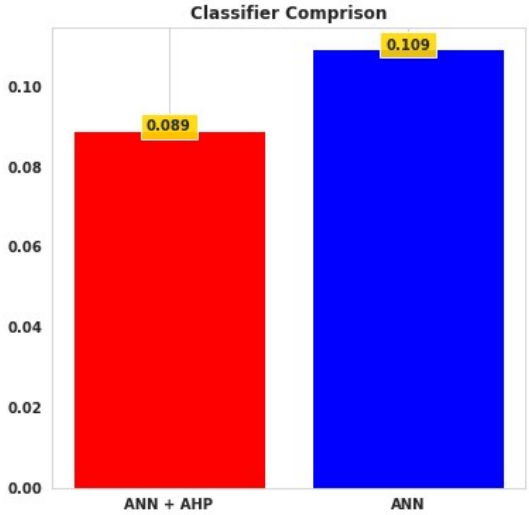
انحراف معیار یعنی مدل دقت مـدل ارائه شـده بین ۰,۰۸۸ – ۰,۸۴۴۲ و ۰,۰۸۸ + ۰,۸۴۴۲ خواهد بود.



تصویر ۵۹: مقایسه عملکرد مدل ترکیبی و شبکه عصبی تنها در اجراهای مختلف به کمک k fold cross validation



تصویر ۶۰: مقایسه دقت میانگین روش ترکیبی و شبکه عصبی تنها (مدل ترکیبی به طور میانگین دقت بهتری ارائه می‌دهد)



تصویر ۶۱: مقایسه انحراف معیار روش ترکیبی و شبکه عصبی تنها (مدل ترکیبی دارای انحراف معیار پایین‌تری هست، یعنی نتایج آن حول میانگین پایداری بالاتری دارند)

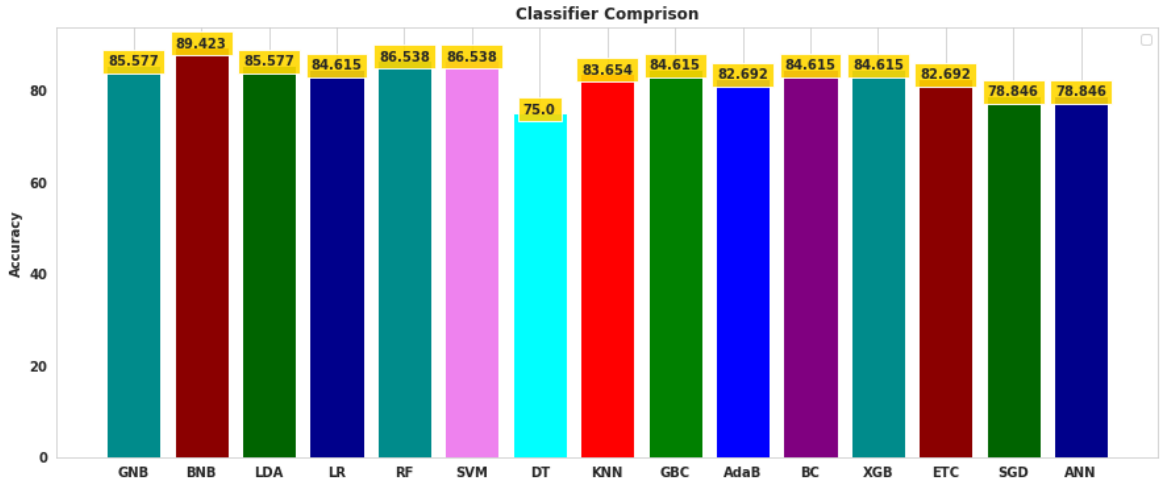
در ادامه نیز از چندین الگوریتم یادگیری ماشین، برای حل همین مسئله استفاده شده است و نتایج الگوریم‌های مختلف در جدول زیر و تصویر ۶۲ نمایش داده شده است. فقط توجه داشته باشید که تمام الگوریتم‌های ارائه شده در حالت پایه و با پارامترهای پیش‌فرض تعریف شده‌اند و هیچ گونه تنظیم پارامتری[۳۵] بروی آن‌ها انجام نشده است.

Algorithm	Accuracy	Precision	Recall	F1-measure	Specificity
Gaussian Naive Bayes	0.856	0.875	0.823	0.848	0.886
Bernouli Naive Bayes	0.894	0.875	0.893	0.884	0.894
Linear Discriminant Analysis	0.856	0.833	0.851	0.842	0.859
Logistic Regression	0.846	0.854	0.82	0.836	0.870
Random Forest	0.865	0.833	0.869	0.851	0.862
Support Vector Machine	0.865	0.833	0.869	0.851	0.862
Decision Tree	0.75	0.708	0.739	0.723	0.758
K Nearest Neighbors	0.836	0.77	0.86	0.813	0.819
Gradient Boosting	0.846	0.833	0.833	0.833	0.857

[35] Hyperparameter Tunning

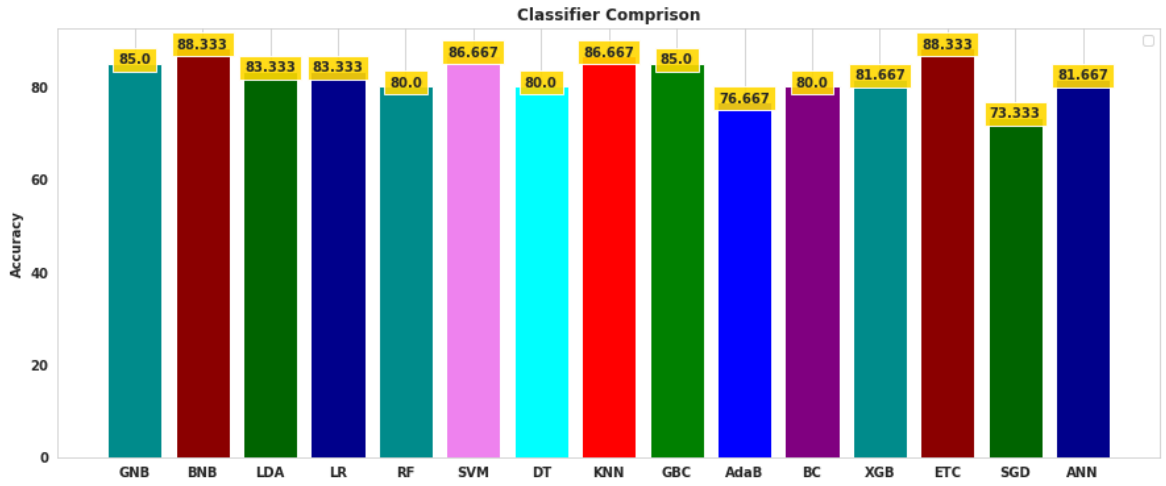
Ada Boost	0.826	0.812	0.812	0.812	0.839
Bagging Classifier	0.846	0.791	0.863	0.826	0.833
Extra Trees	0.846	0.812	0.847	0.829	0.844
XGB Classifier	0.826	0.75	0.857	0.799	0.806
Stochastic Gradient Descent	0.788	0.708	0.809	0.755	0.774
Neural Nets	0.788	0.791	0.76	0.775	0.814

و نمودار مقایسه دقت الگوریتم‌ها در تصویر ۶۲ نشان داده شده است (در این حالت از ۶۵ درصد داده‌ها برای آموزش مدل و از ۳۵ برای تست آن استفاده شده است).



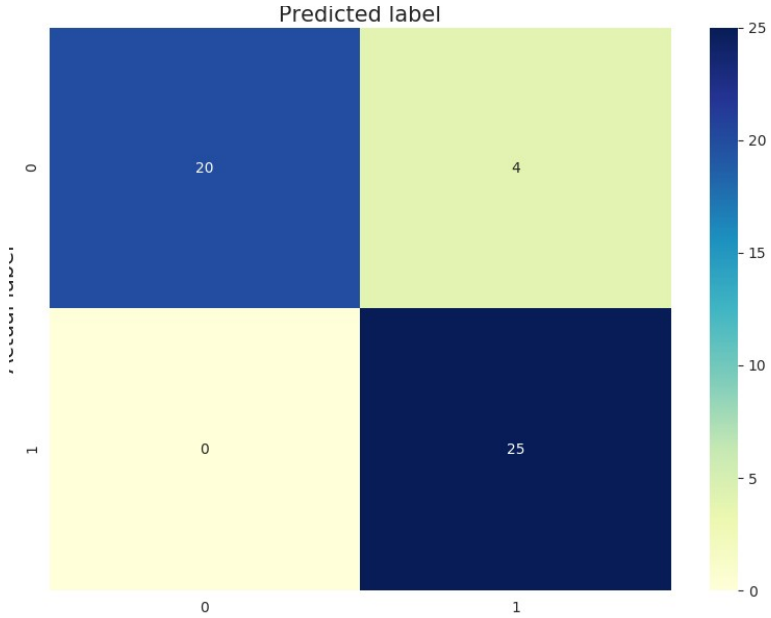
تصویر ۶۲: مقایسه دقت الگوریتم‌های مختلف

اگر به جای ۶۵ درصد از ۸۰ درصد داده‌ها برای آموزش استفاده کنیم، این درصدها با تصویر ۶۳ تغییر می‌کند. این تغییرات می‌تواند معیاری از قابلیت تعمیم‌دهی مدل در اختیار ما قرار دهد.



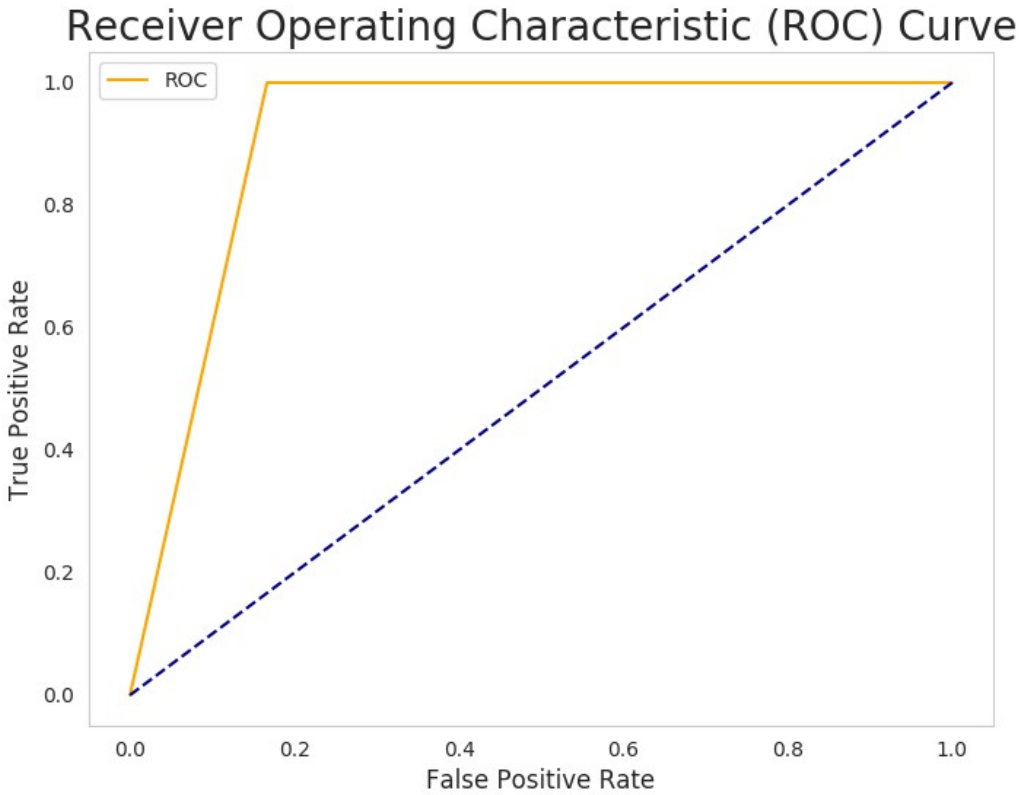
تصویر ۶۳: مقایسه الگوریتم‌های مختلف (از ۲۰ درصد داده‌ها برای تست استفاده شده است).

همانگونه که در نمودار و مقایسه آن‌ها مشاهده می‌شود، الگوریتم بیزین ساده برنولی دارای بالاترین دقت بوده است. به همین جهت است که بسیار در کاربردهای پزشکی هم مورد استفاده قرار می‌گیرد. اما ما در اینجا با توجه به ظرفیت بالا و کارایی خوبی که الگوریتم‌های ترکیبی از خود نشان داده‌اند از الگوریتم گرادیان بوستینگ برای توسعه نهایی مدل استفاده کردیم. اما برای اینکه این الگوریتم به دقت بالایی دست پیدا کند یک تنظیم پارامتر دقیق بروی آن انجام دادیم و دقت حاصل شده، دقت میانگین ۹۱٫۸ درصد بود که دقت قابل اعتمادی است و می‌تواند در کنار یک پزشک در تشخیص بالینی این بیماری کمک کند. ماتریس درهم این الگوریتم در تصویر ۶۴ ارائه شده است.



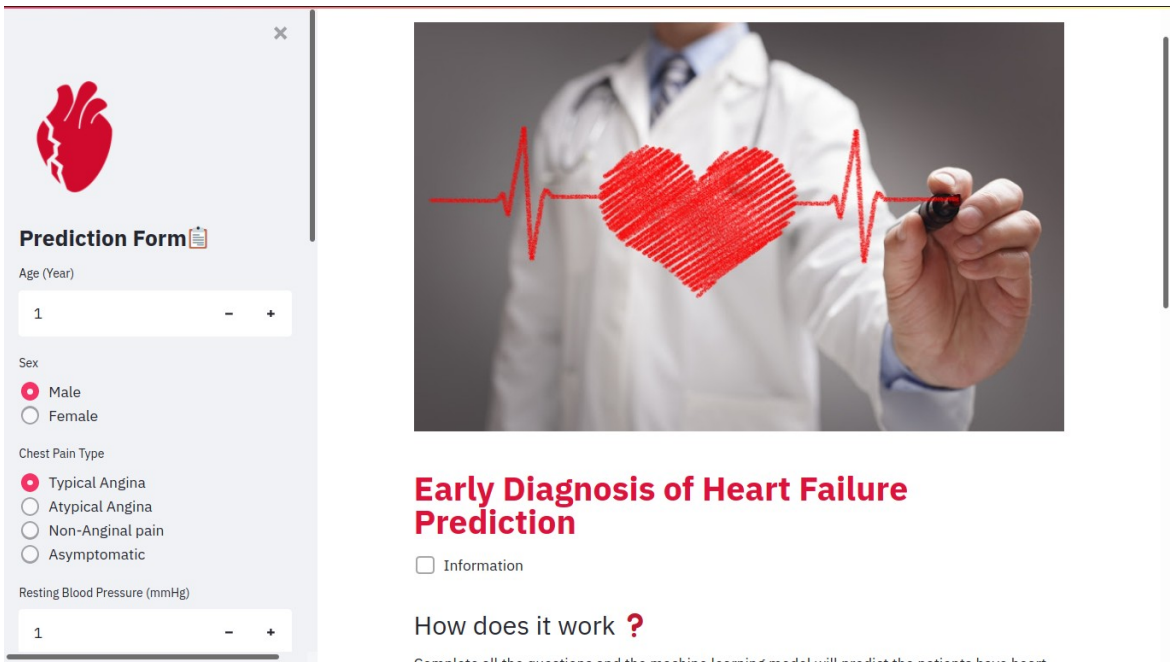
تصویر ۶۴: ماتریس درهم مدل ترکیبی گرادیان بوستینگ

و نمودار ROC آن مطابق تصویر ۶۵ خواهد بود و auc آن هم برابر ۹۱,۷ است.

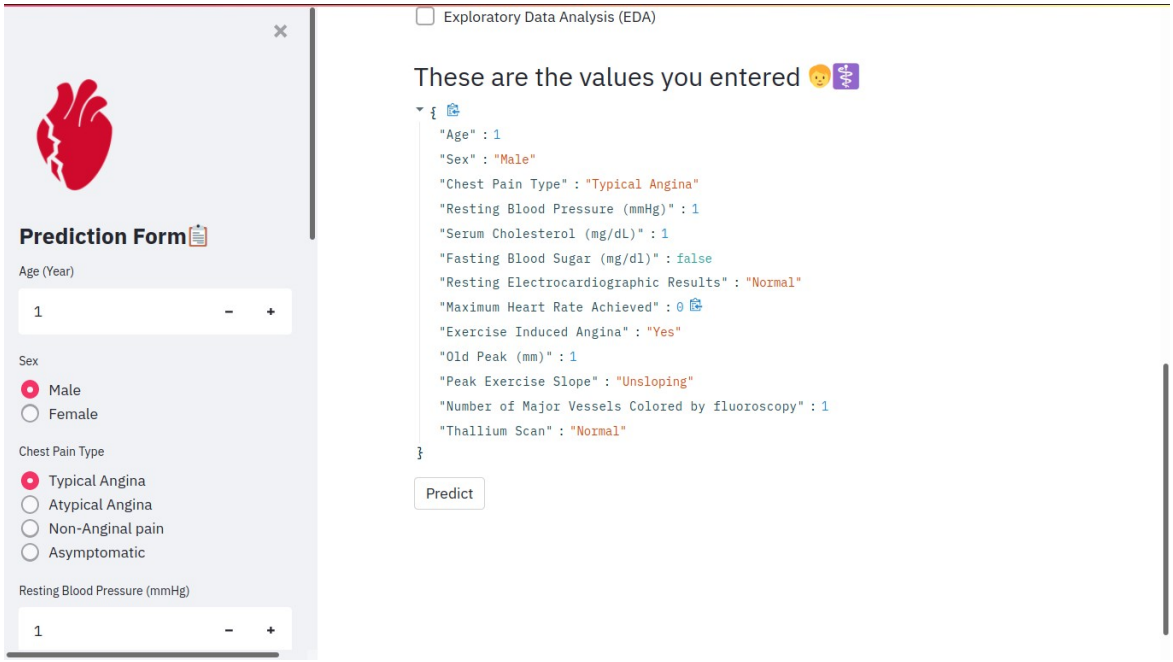


تصویر ۶۵: نمودار ROC مدل گردایان بوست ارائه شده

در نهایت ما یک سامانه تحت وب برای کاربر نهایی ارائه کردیم و کاربر با این سامانه در ارتباط خواهد بود و ورودی خود را به سامانه داده و سامانه پیش‌بینی اینکه فرد سالم است یا بیمار را برمی‌گرداند. تصویری از سامانه توسعه داده شده در تصویر ۶۷ و ۶۶ مشاهده می‌گردد.

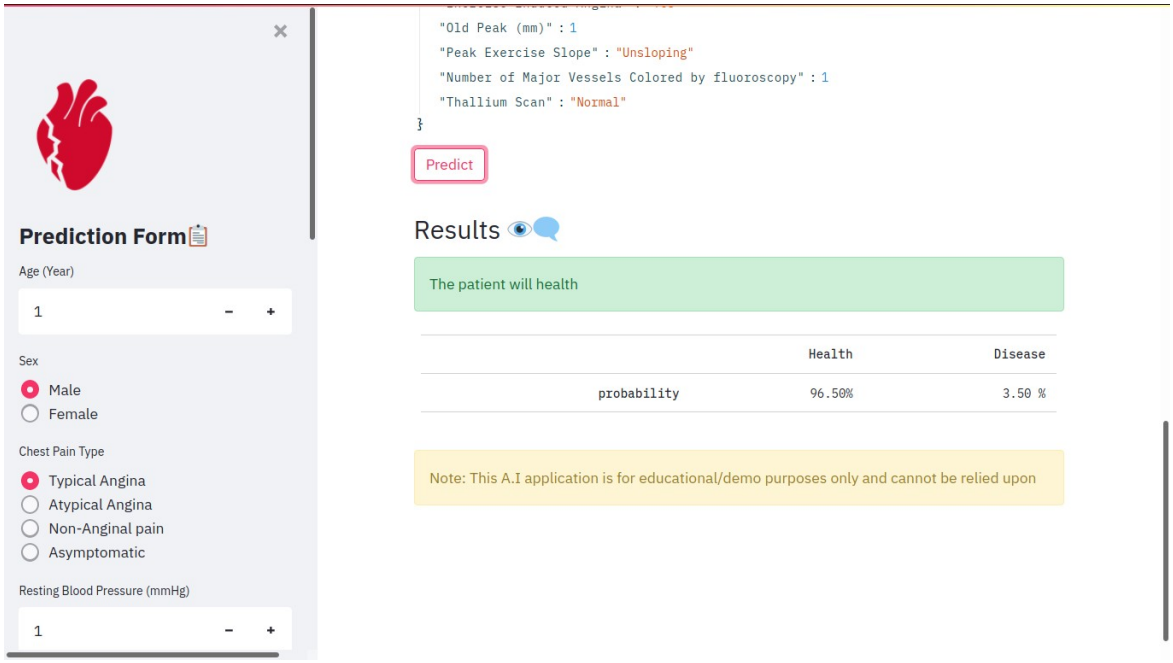


تصویر ۶۶: سامانه نهایی توسعه داده شده برای سیستم تصمیم پشتیبان



تصویر ۶۷: سامانه نهایی توسعه داده شده برای سیستم تصمیم پشتیبان

برای کار با این سیستم کافی است که اطلاعات موجود در نوار بار سمت چپ پر شده و predict را بزنید، و با این کار نتیجه نهایی در قالبی مشابه با تصویر ۶۸ به شما نمایش داده می‌شود.



تصویر ۶۸: نمایش تصمیم نهایی سیستم تصمیم پشتیبان توسعه داده شده

با تشکر.