



الْخَلَقُ



دانشکده مهندسی برق و کامپیو تر

سامانه پیش‌بینی ابتلا به دیابت  
(مبتنی بر الگوریتم‌های یادگیری ماشین)

نام و نام خانوادگی دانشجو: قوام الدین سلیمانی

استاد راهنما: سرکار خانم دکتر فرشته دهقانی

تابستان ۱۴۰۲

## تقدیر و تشکر

تقدیم به همه کسانی که با پاکدلی، پرتوی دانایی و آگاهی را بر تاریکی‌های کاناپی می‌افشانند.

در این زمان که به لطف پروردگار مهربان، بار دیگر فرصت آموختن به من عطا شد، و موفق به اتمام این دوره شدم؛ بر خود لازم می‌دانم از تمام بزرگوارانی که مرا در این دوره یاری کردند، قدردانی کنم.

از استاد راهنمای محترم، سرکار خانم دکتر دهقانی که راهنمایی و بزرگواری ایشان در شرایط مختلف، همواره شامل حال من بوده و انجام این پروژه بدون همیاری ایشان ممکن نبود، بی‌نهایت سپاسگزارم و همواره برای ایشان، آرزوی موفقیت و سلامتی دارم.

از خانواده عزیزم که در این دوره، بزرگترین پشتیبان و یاری‌گر من بودند، تشکر می‌کنم و امیدوارم چون همیشه، در تندرنستی و خوشبختی، مایه دلگرمی ام باشند.

همچنین از سایر اساتید محترم گروه مهندسی کامپیوتر دانشگاه کاشان و دوستانم که توجهات و محبت ایشان در این دوره مایه پیشرفت من بوده نیز کمال تشکر را دارم و برای ایشان بهترین‌ها را از خداوند منان خواستارم.

# فهرست مطالب

۱	تقدیر و تشکر
۲	فهرست تصاویر
۳	خلاصه
۴	۱ مقدمه
۵	۱-۱ تاریخچه . . . . .
۶	۱-۲ علل ابتلا . . . . .
۷	۱-۲-۱ فاکتورهای محیطی . . . . .
۸	۱-۲-۲ عوامل ژنتیکی و غیرمحیطی . . . . .
۹	۱-۲-۳ راهکارهای پیشگیری و استعدادسنجی . . . . .
۱۰	۱-۳-۱ مبتلایان به پیش‌دیابت . . . . .
۱۱	۱-۳-۲ سایر افراد جامعه . . . . .
۱۲	۲-۱ انگیزه و اهداف انجام این پژوهش . . . . .
۱۳	۲-۲ موارد انجام شده . . . . .
۱۴	۳-۱ ادبیات پژوهش
۱۵	۳-۲ مقدمه . . . . .
۱۶	۳-۳ روش‌های داده‌کاوی . . . . .
۱۷	۳-۴ طبقه‌بندی و خوشبندی . . . . .
۱۸	۳-۵ بیش برآش و کم برآش، افزایش داده‌ها . . . . .
۱۹	۳-۶ معرفی الگوریتم‌های طبقه‌بندی . . . . .
۲۰	۳-۷ ۱-۱ رگرسیون لجستیک . . . . .
۲۱	۳-۷ ۲-۱ درخت تصمیم . . . . .
۲۲	۳-۷ ۳-۱ جنگل درختان تصادفی . . . . .
۲۳	۳-۷ ۴-۱ AdaBoost . . . . .
۲۴	۳-۷ ۵-۱ Naive Bayes . . . . .
۲۵	۴-۱ دادگان، پیش‌پردازش و مصورسازی داده‌ها . . . . .
۲۶	۴-۲ ۱-۱ دادگان . . . . .
۲۷	۴-۲ ۲-۱ پیش‌پردازش داده‌ها . . . . .
۲۸	۴-۲ ۳-۱ مصورسازی . . . . .

۲۳	۴-۴-۲	انواع نمودارها و داده‌های اجمالی
۲۴	۵-۴-۲	اندازه گیری میزان خطای دقت
۲۶	۶-۴-۲	ROC و AUC
۲۷	۵-۲	نتیجه گیری
۲۸	۳	<b>کارهای پیشین</b>
۲۸	۱-۳	مقدمه
۲۸	۲-۳	مقاله ۱
۳۰	۳-۳	مقاله ۲
۳۱	۴-۳	نتیجه گیری
۳۲	۴	<b>روش‌ها و نتایج</b>
۳۲	۱-۴	مقدمه
۳۲	۲-۴	روش پیشنهادی
۳۲	۱-۲-۴	پیاده سازی
۴۵	۲-۲-۴	Arduino
۴۹	۳-۲-۴	ابزار Node Red
۵۱	۴-۲-۴	Flask
۵۲	۳-۴	نتیجه گیری
۵۴	۵	<b>جمع‌بندی و کارهای آتی</b>
۵۴	۱-۵	جمع‌بندی
۵۴	۲-۵	کارهای آتی
۵۵	۱-۲-۵	راهاندازی سامانه ثبت گزارشات دیابت و پیش‌بینی
۵۵	۲-۲-۵	تولید کیت‌های ثبت نتایج دیابت جدید
۵۶	پیوست	
۵۶	۱-	کد استاندار دسازی متون فارسی آمیخته به عبارات انگلیسی
۵۷	واژه‌نامه	
۵۹	مراجع	

# فهرست تصاویر

۱۱	۱-۲ مقایسه خوشه بندی و طبقه بندی
۱۱	۲-۲ بیش برازش و کم برازش
۱۳	۳-۲ روش K-Fold
۱۳	۴-۲ روش K-Fold تکرار شونده
۱۴	۵-۲ نمودار پراکندگی
۱۵	۶-۲ نمودار رگرسیون خطی چندگانه در فضا
۱۶	۷-۲ مقایسه رگرسیون لجستیک و خطی
۱۶	۸-۲ نمودار تابع لجستیک
۱۷	۹-۲ مدل Logist Regression
۱۹	۱۰-۲ الگوریتم AdaBoost: در اینجا مراحل ذکر شده به صورت متناوب تکرار می‌شود.[۲۴]
۲۰	۱۱-۲ نمونه‌ای از داده‌های گم شده
۲۰	۱۲-۲ تعداد داده‌های گم شده در این پروژه به ازای هر ستون
۲۱	۱۳-۲ متغیرهای دودویی
۲۱	۱۴-۲ نمونه‌ای از متغیرهای دودویی
۲۳	۱۵-۲ Pairplot
۲۴	۱۶-۲ نقشه حرارتی
۲۵	۱۷-۲ ساختار ماتریس آشفتگی: پارامترهای مذکور در این بخش در این ماتریس قرار گرفته اند.
۲۶	۱۸-۲ ساختار ماتریس آشفتگی برای الگوریتم جنگل درختان تصادفی
۲۷	۱۹-۲ : نمای کلی نمودار ROC
۲۹	۱-۳ مدل SVM
۲۹	۲-۳ دقت‌های اندازه گیری در مقاله اول
۳۰	۳-۳ منحنی ROC در مقاله اول
۳۰	۴-۳ دقت‌های اندازه گیری در مقاله دوم
۳۱	۵-۳ منحنی ROC در مقاله دوم
۳۳	۱-۴ جایگزینی مقادیر نامشخص با متوسط‌های آماری
۳۴	۲-۴ نمودار مقایسه ابتلا به دیابت در نژادهای مختلف
۳۵	۳-۴ ماتریس Scatter
۳۵	۴-۴ نمودار عوامل موثر در ابتلا دیابت (۱)
۳۶	۵-۴ نمودار عوامل موثر در ابتلا دیابت (۲)

۳۷	نقشه حرارتی	۶-۴
۴۱	: نمودار ACC	۷-۴
۴۱	: نمودار AUC	۸-۴
۴۲	: نمودار ROC برای الگوریتم رگرسیون لجستیک	۹-۴
۴۲	: نمودار ROC برای الگوریتم AdaBoost	۱۰-۴
۴۲	: نمودار ROC برای الگوریتم جنگل درختان تصادفی	۱۱-۴
۴۳	: نمودار ROC برای الگوریتم Naive Bayes	۱۲-۴
۴۳	: نمودار ROC برای الگوریتم درخت تصمیم	۱۳-۴
۴۵	: ساختار ماتریس آشتفتگی برای الگوریتم رگرسیون لجستیک	۱۴-۴
۴۹	: محیط توسعه Arduino در حال دریافت اعداد تولید شده توسط بورد	۱۵-۴
۵۰	: صفحه داشبورد پیش بینی کاربر و به روزرسانی دادگان	۱۶-۴
۵۱	: طراحی بخش به روزرسانی در NodeRed	۱۷-۴
۵۱	: طراحی بخش پیش بینی در NodeRed	۱۸-۴
۵۲	: بخشی از کد Flask	۱۹-۴

## خلاصه (چکیده)

**بیان موضوع:** دیابت یک بیماری مزمن است که فرد مبتلا، قند خون بالاتر از حد مجاز را دارا است و این مسئله موجب عوارض و مشکلات جدی در سلامت وی (از جمله برخی نارسایی‌ها، سکته‌ها، آسیب و از کار افتادن اندام‌ها) می‌شود. لازمه ابتلا و بروز این بیماری، عوامل ژنتیکی و محیطی می‌باشد. لذا در صورت وجود احتمال ابتلا به این بیماری در افراد، می‌توان با تغییر سبک زندگی و کنترل‌های پزشکی، تا حدی از ابتلای به این بیماری در افراد مستعد و محتمل، جلوگیری کرد.

**روش تحقیق:** روش‌هایی که بتواند به ما کمک کند تا با دقت مناسبی بتوانیم ابتلای افراد مختلف به بیماری را در آینده را پیش‌بینی کنیم، بسیار حائز اهمیت هستند. یادگیری ماشین و داده‌کاوی با استفاده از داده‌های مختلف که در گذشته جمع آوری شده اند می‌توانند کمک شایانی به ما در این امر داشته باشند. پس از الگوریتم‌های مختلف یادگیری ماشین از جمله رگرسیون لجستیک، جنگل درختان تصادفی و... استفاده کردیم تا دقت هر یک را اندازه‌گیری کنیم.

داده‌هایی که برای آموزش مدل‌هایمان استفاده کرده ایم، از مجموعه دادگان بیماران آمریکایی<sup>۱</sup> است که در سال ۲۰۰۹ تا ۲۰۱۲ جمع آوری شده بودند. این داده‌ها را با استفاده از روش‌های داده‌کاوی گوناگون، پردازش و سپس مصورسازی کردیم و سپس با الگوریتم‌های مذکور و معیارهای ارزیابی مربوط به آن‌ها، سعی در شناسایی بهترین مدل پیش‌بینی کننده کردیم.

**یافته‌ها و نتایج:** مدل جنگل درختان تصادفی حاوی بهترین نتایج نسبت به سایر مدل‌ها بود.

**کلید واژه‌ها:** داده کاوی، دیابت، درخت تصمیم، الگوریتم درخت تصادفی، قندخون، رگرسیون لجستیک، طبقه‌بندی، پیش‌بینی، یادگیری ماشین

---

<sup>۱</sup>NHANES

# فصل ۱

## مقدمه

### ۱ - ۱ تاریخچه

دیابت چیست؟

دیابت، یک بیماری مزمن است و زمانی رخ می‌دهد که بدن انسولین کافی تولید نمی‌کند یا نمی‌تواند به طور موثر از انسولین تولید شده استفاده کند.<sup>[۵]</sup> انسولین هورمونی است که به تنظیم سطح قند خون کمک می‌کند. هنگامی که دیابت به درستی مدیریت نشود، می‌تواند منجر به عوارض جدی سلامتی مانند بیماری‌های قلبی، انواع سکته مغزی و قلبی، نارسایی کلیه، کوری و آسیب عصبی شود.<sup>[۱۷]</sup> آمار ابتلا و مرگ و میر بسیار بالایی از این بیماری در جهان وجود دارد و متاسفانه روز به روز این آمار افزایش می‌یابد.

طبق آمارها، از هر ۱۰ نفر که به دیابت مبتلا هستند، بیش از ۸ نفر آن‌ها از این مسئله آگاهی ندارند و عدهٔ زیادی از افراد هم به پیش‌دیابت مبتلا هستند.<sup>[۶]</sup> در پیش‌دیابت، سطح قند خون بالاتر از حد طبیعی است، اما به اندازه کافی برای تشخیص دیابت بالا نیست. پیش‌دیابت خطر ابتلا به دیابت، بیماری قلبی و سکته را افزایش می‌دهد.<sup>[۵]</sup> اگر پیش‌دیابت در افراد وجود داشته باشد، یک برنامه برای تغییر سبک زندگی، می‌تواند به افراد در جلوگیری از این بیماری کمک کند.<sup>[۱۷]</sup>

این بیماری سه نوع دارد:<sup>[۵]</sup>

۱. دیابت نوع اول که معروف به دیابت جوانی است چون افراد با سن کمتر از ۳۰ سال معمولاً مبتلا می‌شوند. در این نوع به طور ساده می‌توانیم بگوییم میزان انسولین مورد نیاز که توسط پانکراس باقیستی ساخته شود و در خون وجود داشته باشد کافی نیست.

۲. دیابت نوع دوم که به بزرگسالی معروف است و در افراد میانسال و مسن رایج‌تر است در اثر عدم جذب انسولین موجود در خون توسط سلول‌ها می‌باشد.

۳. دیابت نوع سوم دیابت بارداری است که در خانم‌های باردار به طور موقت اتفاق می‌افتد.

## ۱-۲ علل ابتلا

در ابتلا به این بیماری بنا به نوع آن و همچنین شرایط ژنتیکی و محیطی افراد مختلف، فاکتورهای متنوعی مطرح است: [۵]

### ۱-۲-۱ فاکتورهای محیطی

مطابق تحقیقات و بررسی‌های انجام شده از سال‌ها پیش تا کنون، عوامل سبک زندگی چون رژیم غذایی نامناسب، عدم فعالیت بدنی و اضافه وزن (مخصوصاً میزان توده بدنی) می‌تواند خطر ابتلا به دیابت را افزایش دهد. [۱۷] همچنین وجود بیماری‌های زمینه‌ای مثلاً در پانکراس بین افراد می‌تواند در مبتلا شدن به این بیماری موثر باشد که بنا به تعریف دیابت نوع یک، این عامل مربوط به همین نوع می‌شود. [۵]

### ۱-۲-۲ عوامل ژنتیکی و غیرمحیطی

برخی از افراد استعداد ژنتیکی برای دیابت دارند، به این معنی که بدن آن‌ها بیشتر در معرض ابتلا به این بیماری است. عواملی مثل جنسیت، نژاد و شاخص‌هایی خونی مختلف که می‌توانند در اثر بیماری‌های خانوادگی و ارثی دیگری در افراد وجود داشته باشد. مثل برخی ویروس‌ها، وجود کلسترول، چربی و فشار خون و ... [۱۷] [۳] [۴]

## ۱-۳ راهکارهای پیش‌گیری و استعدادسنجی

مطابق توصیه متخصصین اگر بتوانیم افرادی را که استعداد ابتلا به این بیماری را دارند، شناسایی کنیم و این افراد سبک زندگی و روش‌هایی خاصی را در پیش بگیرند، می‌توانند از ابتلا به این بیماری پیش گیری کنند. [۳]

### ۱-۳-۱ مبتلایان به پیش‌دیابت

مطابق توصیه پزشکان، در افرادی که به پیش‌دیابت مبتلا باشند یا سابقه این بیماری در خانواده آن‌ها وجود داشته باشد، به طور پیش‌فرض باید بر یک سبک زندگی سالم، اهتمام ورزند. در این راستا می‌توان به موارد ذیل اشاره کرد: [۴]

- حفظ رژیم غذایی غنی از فیبر مثل انواع میوه‌ها و سبزیجات و کاهش مصرف غذاهای شور، چرب و شیرین
- ورزش منظم
- استفاده از برخی داروها مطابق تجویز پزشک

### ۱-۳-۲ سایر افراد جامعه

مطابق آمارها، سالانه بخش دیگری از افراد جامعه که از دسته قبلی سوا بوده اند، به بیماری دیابت مبتلا می‌شوند.<sup>[۳]</sup> در اینجا با تحلیل برخی فاکتورهای سلامتی می‌توان پیش‌بینی کرد که آیا این افراد ممکن است با ادامه سبک زندگی کنونی، در آینده به این بیماری دچار شوند و آیا بهتر است با تغییر سبک زندگی خود از ابتلا به این بیماری جلوگیری کنند یا نه؟

در این زمینه تحقیقات آماری و بررسی‌های مختلفی انجام شده تا بتوانیم با اندازه‌گیری برخی فاکتورهای کمی و کیفی در افراد، مسئله استعداد در ابتلا به این بیماری را در آنها بررسی کنیم. چالشی که در این زمینه وجود دارد این است که بسیاری از داده‌های غیرخطی و غیراستاندارد پزشکی با ارتباطات و ساختارهای پیچیده وجود دارند که این بررسی‌ها را دشوار می‌سازد.<sup>[۶]</sup>

### ۱-۴ انگیزه و اهداف انجام این پژوهش

در راستای همه موارد مطرح شده در بخش‌های قبلی، بر آن شدیم تا تحقیق کنیم با توجه به امکانات و امروزی و دسترسی به مقالات و منابع گوناگون و همچنین توسعه ابزارهای مبتنی بر یادگیری ماشین و هوش مصنوعی، در صدد یافتن بهترین راهکارها برای نجات جان انسان‌های بیشتر با درنظرگیری مناسب‌ترین الگوریتم‌ها باشیم.

اگر بتوانیم افرادی را که احتمال ابتدا به دیابت در آینده برای آنان زیاد است را شناسایی کنیم می‌توانیم با ارائه برنامه‌های پزشکی مناسب از ابتلای افراد به بیماری مذکور جلوگیری کنیم.

در نهایت نیز جهت ادامه پروژه، یک سیستم پیش‌بینی آنلاین با کمک اینترنت اشیا و دو چارچوب Flask و NodeRed طراحی و معرفی شد که افراد گوناگون بتوانند از آن بهره ببرند.

### ۱-۵ موارد انجام شده

در ابتدا مقالات مختلفی را مطالعه کردم و درمورد الگوریتم‌هایی که مورد بررسی قرار دادم از جمله رگرسیون لجستیک، درخت تصمیم، جنگل درختان تصادفی و بیز ساده، اطلاعات زیادی کسب کردم. پس از یافتن مجموعه دادگان نمونه که مربوط به اطلاعات بیماران آمریکایی در سال‌های ۲۰۰۹ تا ۲۰۱۲ بوده است، با به کار گیری کتابخانه‌های مختلف پایتون از جمله `seaborn`, `Pandas`, `scikit learn`, `numpy` و ... عملیات‌های گوناگونی بر روی داده‌ها انجام شد. از جمله: تمیز کردن دادگان، مقیاس بندی و عملیات توزیع مختلف، تصویر سازی و در نهایت مدلسازی و دستیابی به نتایج نهایی که در نهایت الگوریتم جنگل درختان تصادفی با دقت ۹۶٪ بهترین عملکرد را بین باقی الگوریتم‌ها در این پیش‌بینی، شناخته شد.

## فصل ۲

### ادبیات پژوهش

#### ۱-۲ مقدمه

روش‌های نظارت شده‌ای مانند طبقه‌بندی و تخمین تلاش می‌کنند تا رابطه‌ای میان صفات خاصه ورودی (که گاه متغیرهای مستقل نامیده می‌شوند) را با یک یا چند صفت خاصه هدف (که گاه متغیر وابسته نامیده می‌شود) کشف کنند. در نهایت این رابطه با یک ساختار به عنوان مدل نمایش داده می‌شود. [۱]

به بیان دیگر، در یادگیری تحت نظارت همانطور که از نام آن پیداست، یک ناظر به عنوان یاددهنده در این نوع الگوریتم یادگیری ماشین حضور خواهد داشت. ما مدل خود را با داده‌های برچسب گذاری شده مناسب آموزش می‌دهیم. الگوریتم‌های یادگیری نظارت شده سعی می‌کنند روابط و وابستگی‌هایی بین متغیرها ایجاد کنند که به ترتیب «ویژگی‌ها» و «برچسب‌ها» نامیده می‌شوند. سپس الگوریتم‌ها از داده‌ها، (با استفاده از روابط بین ویژگی‌ها) یاد می‌گیرند و خروجی را پیش‌بینی می‌کنند. [۱۴]

اما در یادگیری بدون نظارت هیچ ناظر یا یاد دهنده‌ای وجود نخواهد داشت، بنابراین هیچ آموزشی یا آموزشی به ماشین ارائه نخواهد شد. یادگیری بدون نظارت با داده‌های بدون برچسب سروکار دارد که در آن ما نمی‌توانیم روابط و وابستگی‌ها را در داده‌ها اندازه گیری کنیم. در این مدل، مدل‌های ما سعی می‌کنند داده‌های مرتب نشده را بر اساس الگوها و شباهت‌ها در داده‌ها به صورت خوش‌ای گروه‌بندی کنند. [۱۴]

#### ۲-۲ روش‌های داده‌کاوی

##### ۱-۲-۲ طبقه‌بندی و خوش‌بندی

در این بخش به طور واضح تر درمورد تفاوت الگوریتم‌های با نظارت (طبقه‌بندی کننده و پیش‌بینی) و بدون نظارت (خوش‌بندی) در مدل‌های رایج توضیح می‌دهیم.

###### • طبقه‌بندی

در این مدل، گیرنده پیشنهاد می‌تواند پاسخ دهد یا پاسخ ندهد. متقاضی وام می‌تواند به موقع

بازپرداخت کند، دیر بازپرداخت کند یا اعلام ورشکستگی کند. تراکنش کارت اعتباری می‌تواند عادی یا تقلیلی باشد. بسته‌ای از داده‌هایی که در یک شبکه حرکت می‌کنند می‌تواند عادی یا تهدیدکننده باشد. یک اتوبوس در سیستم حمل و نقل می‌تواند در دسترس باشد یا در دسترس نباشد. قربانی یک بیماری می‌تواند بهبود یابد، یا خیر. [۱۱]

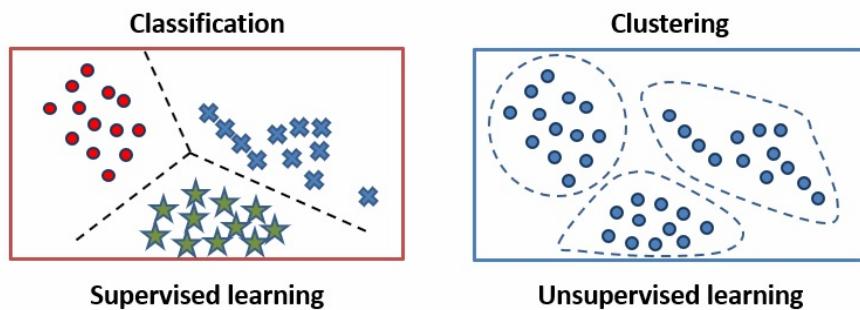
یک کار رایج در داده کاوی، بررسی داده‌هایی است که در آن طبقه‌بندی ناشناخته است یا در آینده رخ خواهد داد، با هدف پیش‌بینی اینکه طبقه‌بندی چیست یا چه خواهد بود. داده‌های مشابهی که طبقه‌بندی ناشناخته است، برای توسعه قوانین استفاده می‌شوند که بعداً روی داده‌های دارای طبقه‌بندی ناشناخته اعمال شوند. [۱۱]

به بیان دیگر، پیش‌بینی مشابه طبقه‌بندی است، با این تفاوت که ما سعی می‌کنیم ارزش یک متغیر عددی (مثلاً مقدار خرید) را به جای یک طبقه (مثلاً خریدار یا غیرخریدار) پیش‌بینی کنیم. البته در طبقه‌بندی سعی داریم یک طبقه را پیش‌بینی کنیم، اما گاهی در ادبیات داده کاوی، از اصطلاحات تخمین و رگرسیون برای اشاره به پیش‌بینی مقدار یک متغیر پیوسته استفاده می‌شود، و پیش‌بینی ممکن است هم برای داده‌های پیوسته و هم برای داده‌های طبقه‌ای استفاده شود. [۱۱] ضمناً در فرایند ساخت مدل طبقه‌بندی، داده‌های اولیه به دو بخش تقسیم می‌شوند که در ادامه توضیح داده خواهد شد؛ داده‌های آموزشی برای مدل را ساخته و داده‌های اعتبارسنجی که زیر مجموعه‌ای از داده‌های اصلی هستند و مستقل از داده‌های آموزشی اند، عملکرد مدل را برایمان ارزیابی می‌کنند. [۱۱]

• **خوشه‌بندی تجزیه و تحلیل خوشه‌ای**، برای تشکیل گروه‌ها یا خوشه‌هایی از رکوردهای مشابه بر اساس چندین اندازه گیری انجام شده بر روی آن‌ها انجام می‌شود. ایده کلیدی خوشه‌بندی به این صورت است که خوشه‌ها را به روش‌هایی توصیف کنیم که برای اهداف تحلیل مفید باشد. این ایده، در بسیاری از زمینه‌ها از جمله نجوم، باستان‌شناسی، پزشکی، شیمی، آموزش، روانشناسی، زبان‌شناسی و جامعه‌شناسی به کار گرفته شده است. برای مثال، زیست‌شناسان از طبقات اصلی و طبقات فرعی برای سازماندهی گونه‌ها استفاده گسترده‌ای کرده اند. [۱۱] بدیهی است که ممکن است ما ندانیم چه تعداد دسته یا خوشه در پایان این فرایند ایجاد می‌شود.

به زبان دیگر، می‌توانیم بگوییم زمانی که پیش‌فرض خاصی در مورد کلاس‌هایمان نداریم و صرفاً می‌خواهیم بینیم داده‌هایمان در چه گروه‌هایی قرار می‌گیرند از خوشه‌بندی استفاده می‌کنیم [۱۱] (مثلاً چندین متن داریم و می‌خواهیم متن‌های مشابه در یک دسته قرار گیرند) اما اگر بخواهیم هر رکورد از داده‌ها را الزاماً در یک طبقه یا کلاس خاص قرار دهیم از طبقه‌بندی استفاده می‌شود که روش با نظرات است [۱۱] (مانند همین پروژه).

از دید دیگر می‌توانیم بگوییم هرگاه حسایت ما برای تشخیص نمونه‌های گوناگون زیاد باشد، (یعنی لزوماً برچسبی بر داده ای بزنیم که اهمیت زیادی برای ما داشته باشد مثل تشخیص دیابت در افراد) در اینجا از مدل‌های طبقه‌بندی استفاده می‌کنیم و اگر تنها بخواهیم داده‌هایمان را در دسته‌های گوناگون قرار دهیم و نام کلاس‌های دسته‌بندی ما مشخص نباشد، می‌توانیم خوشه‌بندی را اعمال کنیم. (مثلاً یک فروشگاه اینترنتی که به گروه‌هایی از خریداران مختلف، پیشنهاد‌های گوناگونی در ازای خرید های قبلی آن‌ها می‌دهد).



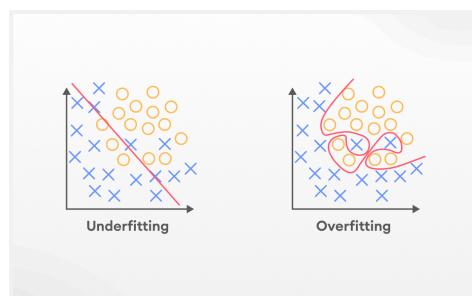
شکل ۱-۲: مقایسه خوشه بندی و طبقه بندی

## ۲-۲-۲ بیش برازش و کم برازش، افزایش داده ها

اگر خیلی ساده بخواهیم در مورد این دو مفهوم توضیح دهیم، می‌توان گفت در یک مدل یادگیری ماشین، کم برازش در خروجی زمانی اتفاق می‌افتد که مدل خیلی ساده است و نمی‌تواند رابطه و الگوی بین داده‌ها را با دقت مشخص کند و از این جهت در پیش‌بینی و سنجش ناتوان خواهد بود، در حالی که بیش‌برازش زمانی اتفاق می‌افتد که مدل بیش از حد پیچیده باشد و داده‌های آموزشی را به خاطر بسپارد. بنابراین در مقابل داده‌های جدید تعمیم و انطباق پیدا نمی‌کند. [۱۹] راه حل جلوگیری از این مشکل، تقسیم داده‌ها و اعتبارسنجی متقابل است [۱۹] که در ادامه توضیح داده می‌شود.

یک مثال خوب که می‌توان برای این مسئله بیان کرد، نمونه لیوان است. اگر فرض کنیم مدلی که ساخته ایم، داشتن دسته را برای یک شیء به نام "لیوان" را ضروری بداند، پس به نظر می‌رسد شرط اضافه‌ای برای تشخیص لیوان بودن یا نبودن اشیاء منظور کرده است؛ و در این حالت می‌گوییم مدل ما دچار بیش برازش شده است.

بر عکس در حالتی که فرض کنیم مدل ما اهمیتی برای داشتن ارتفاع نسبت به سطح را در نظر نگیرد، ممکن است یک بشقاب را به عنوان یک لیوان در نظر بگیرد و در این حالت می‌گوییم مدل ما دچار کم برازش شده است.



شکل ۲-۲: بیش برازش و کم برازش

بخش کردن داده‌ها یا افزایش<sup>۱</sup> برای ارزیابی مدل به طور مؤثر استفاده می‌شود. این تکنیک، برای ارزیابی عملکرد مدل یادگیری ماشین به این صورت عمل می‌کند شما یک مجموعه داده داده شده را می‌گیرید و آن را به دو یا سه زیر مجموعه تقسیم می‌کنید. [۲۰]

- داده‌های آموزشی یا (Train) :

مجموعه‌ای از داده‌های مورد استفاده برای ساخت و یادگیری جهت تناسب پارامترها با مدل یادگیری ماشین و ایجاد روابط در مدل مد نظرمان است.

- داده‌های اعتبارسنجی (استفاده از این مورد اختیاری است.) :

مجموعه‌ای از داده‌ها برای ارائه یک ارزیابی بی طرفانه از یک مدل برآشش شده در مجموعه داده آموزشی در حین تنظیم روابط درون مدل استفاده می‌شود.

- داده‌های سنجش (Test) :

مجموعه‌ای از داده‌ها برای ارائه یک ارزیابی بی طرفانه از یک مدل نهایی برآشش شده در مجموعه داده آموزشی استفاده می‌شود. در واقع این قسمت تضمین می‌کند که مدل بر اساس داده‌های دیده نشده ارزیابی می‌شود. در واقع هدف از استفاده از داده‌های آزمایشی، تعیین میزان تعمیم مدل به داده‌های جدید و دیده نشده است.

برای دستیابی به نتایج دقیق، بسیار مهم است که مطمئن شویم داده‌ها به صورت تصادفی انتخاب شده اند و شامل طیف متنوعی از نمونه‌ها هستند. این مسئله برای به حداقل رساندن سوگیری و بهبود قابلیت های تعمیم مدل کمک می‌کند. [۲۰]

با این حال، با تقسیم داده‌های موجود به سه مجموعه، تعداد نمونه‌هایی را که می‌توان برای یادگیری مدل استفاده کرد، به شدت کاهش می‌دهیم که راه حل آن استفاده از اعتبارسنجی متقابل است که در آن نیازی داده‌های اعتبارسنجی نیست [۲۲] و در ادامه توضیح داده می‌شود.

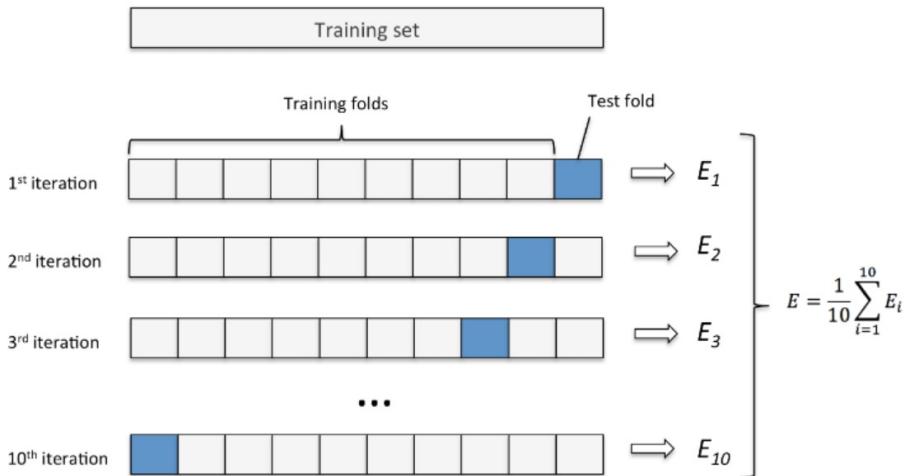
### اعتبارسنجی متقابل<sup>۲</sup>

برای جلوگیری از برآشش بیش از حد، می‌توانیم از تکنیک‌هایی استفاده کنیم که داده‌های آموزشی و سنجش ما را به قسمت‌های بیشتری تقسیم کنند و آن‌ها را به طرق مختلف در مقابل هم مورد ارزیابی قرار دهد تا مدل ما دقت بیشتری پیدا کند.

این روش، شامل تقسیم داده‌های آموزشی به k مجموعه‌های کوچکتر و آموزش یک مدل با استفاده از  $k-1$  عدد از این مجموعه‌ها است، در حالی که از مجموعه باقی مانده، به عنوان مجموعه آزمایشی، برای محاسبه میزان عملکرد مدل استفاده می‌شود. این فرآیند k بار تکرار می‌شود و هر مجموعه یک بار به عنوان مجموعه تست عمل می‌کند. سپس میانگین معیارهای عملکرد به دست آمده (که در فصل های آینده ذکر می‌شود) در هر تکرار، به عنوان معیار عملکرد کلی در نظر گرفته می‌شود. این روش از نظر محاسباتی گران است اما از هدر رفتن بیش از حد داده‌ها جلوگیری می‌کند. [۲۲]

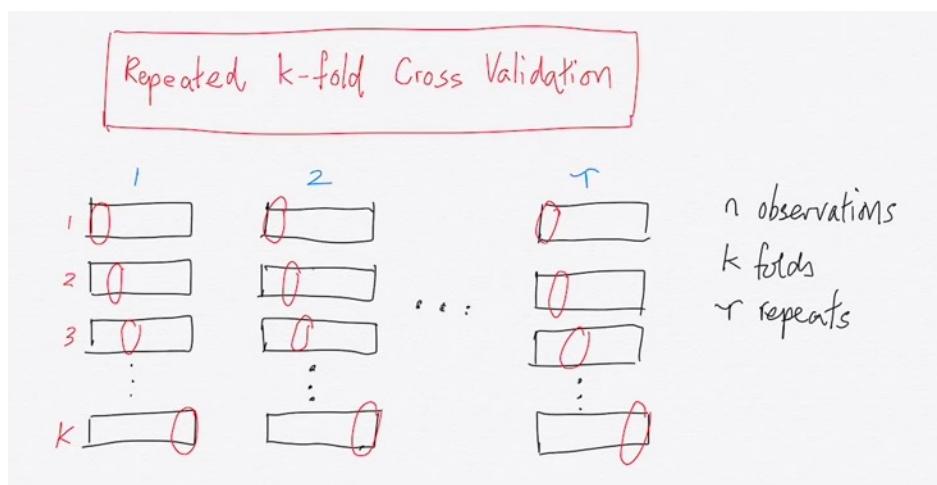
<sup>1</sup>Data Splitting

<sup>2</sup>Cross validation



شکل ۲-۳: روش K-Fold

روش دیگری که ذکر شد، اعتبارسنجی K-Fold مکرر است که فرآیند اعتبارسنجی K-Fold را  $n$  بار با تقسیم های مختلف هر بار تکرار می کند. این روش زمانی مفید است که چندین بار لازم است فرآیند اعتبارسنجی تکرار شود. [۲۲] درواقع این روش علاوه بر تقسیم داده ها به K-Fold، فرآیند را چندین بار تکرار می کند و قبل از هر تکرار به طور مستقل داده ها را به هم می زند. این روش با میانگین گرفتن مکرر اعتماد به مدل را افزایش می دهد. [۱۵]



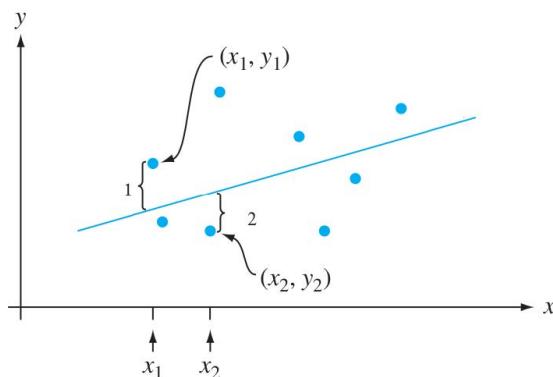
شکل ۴-۲: روش K-Fold تکرار شونده

## ۳-۲ معرفی الگوریتم‌های طبقه‌بندی

### ۳-۲-۱ رگرسیون لجستیک<sup>۳</sup>

خوب است در ابتدا مروری بر رگرسیون خطی<sup>۴</sup> داشته باشیم تا در ادامه بتوانیم رگرسیون لجستیک را بهتر درک کنیم.  
رگرسیون خطی:

در بسیاری از بررسی‌های آماری، لازم است یک متغیر وابسته را از روی یک یا چند متغیر مستقل پیش‌بینی کنیم که اصطلاحاً به آن رگرسیون یا برگشت می‌گوییم.<sup>[۲]</sup> برای مثلاً میزان ساعت مطالعه یک متغیر مستقل است و نمره اخذ شده در درسی متغیری وابسته است و بین این دو رابطه وجود دارد. سپس نمونه‌ای از جمعیت را در نظر گرفته و در آن مقدارهای  $X_1$  تا  $X_n$  در متغیر مستقل خود مقابله مقادیر نظیر در متغیر وابسته از  $Y_1$  تا  $Y_n$  قرار می‌دهیم.<sup>[۲]</sup> سپس آنها را مثل یک نمودار در صفحه مختصات به یکدیگر متصل کرده که به آن نمودار پراکندگی گوییم.<sup>[۲]</sup>



شکل ۲-۵: نمودار پراکندگی

حالا می‌توان خطی را در این صفحه مختصات در نظر گرفت که تا حد زیادی منطبق بر نقاط باشد که در واقع یک نمودار پیش‌بینی کننده  $Y$  بر مبنای  $X$  است که به آن معده رگرسیون  $Y$  بر روی  $X$  گویند.<sup>[۲]</sup> حالا رابطه این نقاط و منحنی را با  $\mu_{Y|x} = E(Y|x) = \alpha + \beta x$  مشخص می‌کنیم و  $\alpha$  و  $\beta$  پارامترهایی هستند که باید مقدار دهی شوند تا خط بر نقاط منطبق باشد.<sup>[۲]</sup> مسئله‌ای که در اینجا مطرح است این است که ممکن است خط ما بر نقاط مختلف منطبق نشود. لذا اینجا باید حالت بهینه‌ای را در نظر گرفت حداقل مقدار خط (یا بهتر بگوییم اختلاف) در مجموع داشته باشیم که روش حداقل مربعات برای یافتن میزان بهینه  $\alpha$  و  $\beta$  که دارای بیشترین انطباق و کمترین خطای باشند برای ما کمک کننده است.<sup>[۲]</sup> بر اساس همین روش، با معادلات زیر به مقادیر بهینه  $\alpha$  و  $\beta$  دست می‌یابیم.<sup>[۲]</sup>

$$\alpha = \bar{y} - \beta \bar{x} \quad (1-2)$$

$$\beta = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2} \quad (2-2)$$

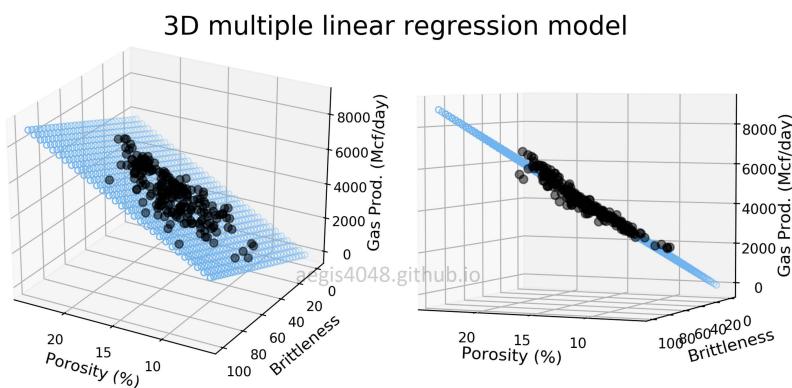
<sup>۳</sup>Logistic Regression

<sup>۴</sup>Linear Regression

که در روابط فوق  $\bar{x}$  و  $\bar{y}$  میانگین  $y$  و  $x$  هستند. با تعمیم این روابط و اصول بیان شده می‌توان حالتی را در نظر گرفت که چندین متغیر مستقل داریم. (مثلاً ۲ تا) که این مدل به رگرسیون خطی چندگانه معروف است و در آنجا نمودار ما حالت فضایی پیدا خواهد کرد و با رابطه زیر می‌توانیم آن را بیان کنیم.

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} + \epsilon \quad i = 1, \dots, n \quad (3-2)$$

که در این رابطه،  $p$  ابعاد ما می‌باشد. [۲۷]



شکل ۲-۶: نمودار رگرسیون خطی چندگانه در فضا

در کل می‌توانیم بگوییم روش‌های رگرسیون زمانی مناسب است که مقادیر مستقل در مجموعه داده‌ها به کلاس‌هایمان (به بیان دیگر طبقه‌بندی‌ها) وابستگی داشته باشند. ضریب همبستگی خطی که با رابطه ۴-۲ می‌شود میزان این وابستگی را برای ما نشان می‌دهد.

$$p(X, Y) = \text{corr}(X, Y) = \frac{\text{Cov}(x, y)}{(\text{Var}(x)\text{Var}(y))^{\frac{1}{2}}} \quad (4-2)$$

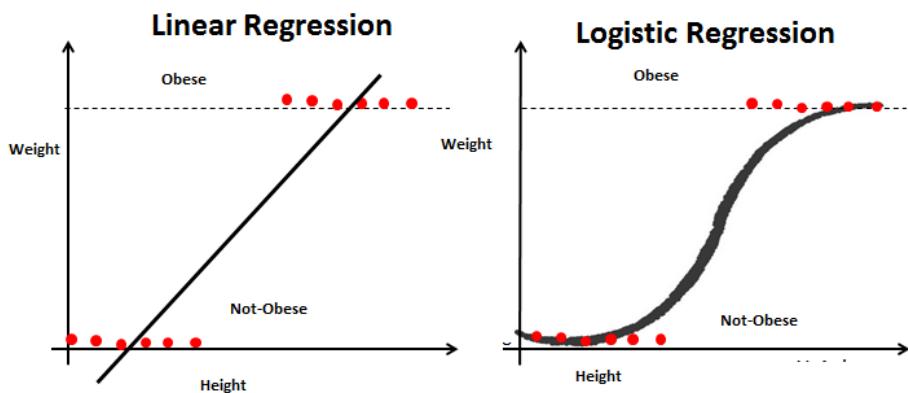
صورت کسر کواریانس  $X$  و  $y$  است و مخرج واریانس  $X$  و واریانس  $y$  می‌باشد. حاصل مقداری است از ۱ - تا ۱ که میزان وابستگی مستقیم یا معکوس را نشان می‌دهد [۱] و در صورت نبود میزان وابستگی مقدار ۰ است. [۲]

### رگرسیون لجستیک:

حالا که با مفاهیم ابتدایی رگرسیون آشنایی پیدا کردیم به بیان نوع دیگری از آن به نام رگرسیون لجستیک می‌پردازیم.

در این مدل به جای اینکه مقدار عددی برای متغیر وابسته تعریف شود، بر اساس احتمال متغیرهای دودویی را برای پیشگویی در نظر داریم [۱] که در ادامه این مسئله را بیشتر توضیح می‌دهیم. مثلاً کار ماء، فاکتور BMI در افراد یک متغیر مستقل است و بر ابتلا به دیابت در افراد موثر بوده؛ همچنین در اینجا متغیر وابسته ماء، همان دیابت گرفتن یا نگرفتن فرد می‌باشد که با ۰ و ۱ آن را در نظر می‌گیریم. پس این مدل برای مواردی استفاده می‌شود که حالت کلاس بندی برای متغیرهایمان داریم. [۸] [۹] ضمناً در حالت کلاس بندی، نمی‌توانیم حالت ۰ یا ۱ را به عنوان اعدادی در حالت رگرسیون خطی منظور کنیم؛

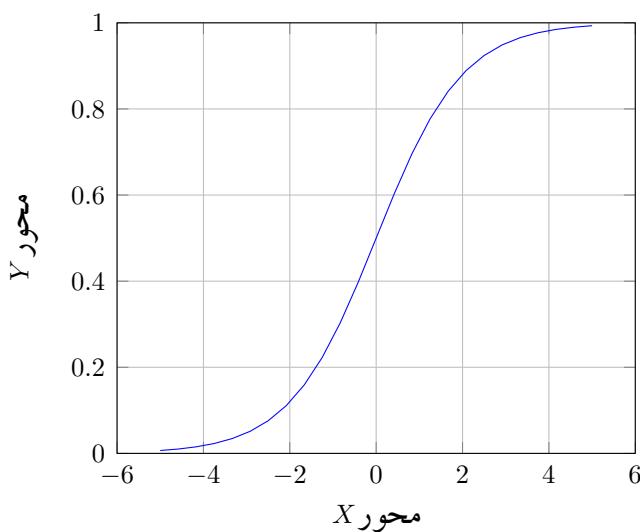
چرا که  $\circ$  و  $\circ$  را به عنوان مقادیر عدد در نظر گرفته می‌شود و امکان نمایش کلاس‌های گوناگون در در یک نمودار خطی مشخص وجود ندارد.<sup>[۹]</sup> اگر به شکل ۷-۲ دقت کنید، متوجه می‌شوید در حالت رگرسیون خطی برخی از نمونه‌ها در کلاس مربوطه قرار نگرفته‌اند.



شکل ۷-۲: مقایسه رگرسیون لجستیک و خطی

علت این امر مشخص است. زیرا زمانی که کلاس‌های گوناگون داریم، ساختار استدلالی که در الگوریتم رگرسیون خطی مطرح است نمی‌تواند طبقه بندی صحیحی برای ما انجام دهد. نهایتاً چاره این است که از یک نمودار منحنی شکل برای فشرده کردن نتایج بین  $\circ$  تا  $\circ$  استفاده کنیم [۹]: رابطه ۵-۲ و شکل ۸-۲

$$\text{logistic}(x) = \frac{1}{1 + \exp^{-x}} \quad (5-2)$$



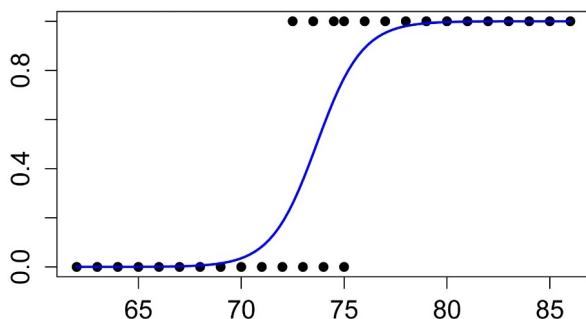
شکل ۸-۲: نمودار تابع لجستیک

سپس اگر سمت راست رابطه ۳-۲ را در جای X در معادله ۵-۲ (که آن را تابع سیگماوار<sup>۵</sup> می‌نامند) قرار دهیم، فقط مقادیر ۰ و ۱ را به ما می‌دهد.

حال در این مدل، چون قرار است هر نمونه به یک کلاس تعلق یابد، بر اساس روابط مطرح شده در این بخش و تعمیم خواص آماری (واریانس)، می‌دانیم وزن ویژگی هایمان عاملی اثر گذار تعیین مرز جداسازی در نمودار ما خواهد بود. لذا نهایتاً به رابطه زیر می‌رسیم که بر اساس احتمال قرار گیری هر نمونه در هر کلاس برای ما کلاس‌بندی را انجام می‌دهد [۸][۹]:

$$g(x) = \ln\left(\frac{p(x)}{1-P(x)}\right) = \frac{e^{b_0+b_1x}}{1+e^{b_0+b_1x}} = \ln(e^{b_0+b_1x}) = b_0 + b_1x \quad (6-2)$$

در رابطه فوق، کاری که انجام می‌شود به این صورت است که احتمال قرار گیری یک نمونه در یک کلاس نسبت به حالت دیگر سنجیده می‌شود و در صورتی که مثلاً با احتمال بالای ۵۰ درصد در طبقه ۱ قرار می‌گیرد، این کار صورت می‌پذیرد و در غیر این صورت به کلاس ۰ متعلق است. (می‌دانیم که احتمال دقیقاً ۵۰ درصد روی نقطه (۰،۰) قرار دارد). [۸][۹] به بیان دیگر می‌توانیم بگوییم، Logit پاسخ ۷، ترکیب خطی پیش‌بینی‌کننده‌ها (یا همان X) است. [۶]



شکل ۹-۲: مدل Logistic Regression

### ۲-۳-۲ درخت تصمیم

#### انتخاب بهترین ویژگی

### ۳-۳-۲ جنگل درختان تصادفی

این الگوریتم از ساختار درختان تصمیم استفاده می‌کند. ساختار درختان تصمیم ...

حال می‌توانیم بگوییم این الگوریتم، با ایجاد چندین درخت تصمیم مختلف از داده‌هایمان برای ما تصمیم گیری را انجام می‌دهد. ممکن است بپرسید که خب، آیا مثلاً ۴ درخت تصمیم مختلف نتایج یکسانی دارند؟ با توجه به ساختار درختان تصمیم جواب قطعاً خیر است. پس در اینجا از جواب‌ها رای گیری می‌شود. یعنی در نمونه‌ای ۳ درخت حکم می‌کنند که فرد به دیابت مبتلا خواهد شد و ۱ درخت حکم می‌کند که این فرد به دیابت مبتلا نخواهد شد. لذا اکثریت آراء بر مبتلا شدن فرد اتفاق نظر دارند. پس نتیجه آن پیش‌بینی، ابتلا شدن فرد است.

<sup>۵</sup>Sigmoid/ Logistic Function

ادامه دارد ...

### AdaBoost ۴-۳-۲

این الگوریتم یک فرض ساده در نظر دارد و آن یادگیری گروهی است.<sup>[۱۳]</sup> فرض کنید کسانی که در مجموعه داده‌های آموزشی دچار اشتباه در تشخیص ابتلا به بیماری دیابت شدند، از مجموعه بقیه داده‌های آموزشی جدا می‌شوند و در یک طبقه بندی جدید مجدداً مورد ارزیابی قرار می‌گیرند؛ همانطور که در واقعیت ممکن است هرپزشکی معیارهای مختلفی را برای تشخیص بیماری مراجعه کننده اش داشته باشد و هر پزشکی نمی‌تواند همه افراد را درست تشخیص دهد پس اگر یک تیم پزشکی داشته باشیم نظرات جمعی می‌توانند بیماران بیشتری را به درستی شناسایی کنند.<sup>[۱۱]</sup>

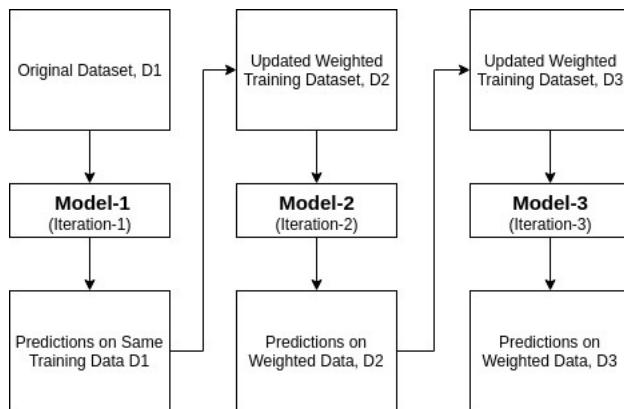
این روند جداسازی داده‌های آموزشی و امتیاز دهنده در تشخیص توسط طبقه بندی کننده‌های مختلف درون درختان تصمیم ضعیف، مکرراً تکرار می‌شود تا درنهایت طبقه بندی‌های متعددی داشته باشیم که هر کدام بر اساس میزان سرآمدی در تست‌های مختلف، امتیازات مختلفی دریافت کنند.<sup>[۱۳]</sup> بعد از ساخته شدن مدل، حالا می‌توانیم به نسبت امتیاز هر طبقه بندی کننده داده‌های دریافتی را به صورت شناسی بین هر کدام تقسیم کنیم تا پیش‌بینی صورت گیرد و این مسئله موجب تقویت سنجش می‌گردد. به همین دلیل به آن الگوریتم Adaptive Boosting یعنی تقویت کننده تطبیقی گویند.<sup>[۱۱]</sup> [۱۳]

اگر از دید ریاضی الگوریتم را بررسی کنیم، برای محاسبه میزان خطای مدل،  $M_i$  وزن هر یک از تاپل‌های  $D_i$  را که  $M_i$  اشتباه طبقه بندی کرد، جمع می‌کنیم.

$$\text{error}(M_i) = \sum_{j=1}^d w_i \times \text{err}(X_j) \quad (7-2)$$

حالا اگر طبقه بندی کننده  $X_j$  اشتباه کند، میزان اror برای آن ۱ اندازه گیری می‌شود و در غیر این صورت ۰ است. اگر یک طبقه بندی کننده آن قدر ضعیف باشد که خطایش از ۰.۵ بیشتر شود آن را دیگر در نظر نمی‌گیریم.<sup>[۱۱]</sup> سپس یک مجموعه داده جدید به نام  $D_i$  تولید می‌کنیم و از آن یک  $M_i$  جدید استخراج می‌کنیم و دوباره این روند را ادامه می‌دهیم. هر چه میزان خطای طبقه بندی کننده کمتر باشد، دقیق‌تر است و بنابراین، وزن آن برای انتخاب شدن باید بیشتر باشد. از رابطه ۸-۲ برای محاسبه وزن هر طبقه بندی استفاده می‌شود.<sup>[۱۱]</sup>

$$\log\left(\frac{1 - \text{error}(M_i)}{\text{error}(M_i)}\right) \quad (8-2)$$



شکل ۲-۱۰: الگوریتم AdaBoost: در اینجا مراحل ذکر شده به صورت متناسب تکرار می‌شود.<sup>[۲۴]</sup>

## Naive Bayes ۵-۳-۲

### ۴-۲ دادگان، پیش‌پردازش و مصورسازی داده‌ها

#### ۱-۴-۲ دادگان

دادگانی<sup>۶</sup> که از آن استفاده کردیم، برگرفته از یک برنامه مطالعاتی به نام NHANES بوده که جهت بررسی سلامتی کودکان و بزرگسالان از سوی CDC<sup>۷</sup> تدوین شده است.

برنامه NHANES در اوایل دهه ۱۹۶۰ آغاز شد و به صورت مجموعه‌ای از نظرسنجی‌ها با تمرکز بر گروه‌های مختلف جمعیتی یا موضوعات بهداشتی انجام شده است. در سال ۱۹۹۹، این نظرسنجی به یک برنامه مستمر تبدیل شد که تمرکز در حال تغییری بر روی انواع اندازه گیری‌های سلامت و تعذیب برای رفع نیازهای نوظهور دارد.<sup>[۱۸]</sup>

مصاحبه NHANES شامل سوالات جمعیت شناختی، اجتماعی-اقتصادی، رژیم غذایی و سلامتی است. جزء معاينه شامل اندازه گیری‌های پزشکی، دندانی و فیزیولوژیکی و همچنین تست‌های آزمایشگاهی است که توسط پرسنل پزشکی بسیار آموزش دیده انجام می‌شود.<sup>[۱۸]</sup>

#### ۲-۴-۲ پیش‌پردازش داده‌ها

##### پیش‌پردازش داده‌ها چیست؟

داده‌هایی که جمع آوری می‌کنیم، انواع مختلفی دارند. مانند رشته‌ها، انواع اعداد، مقادیر نامشخص و ... . همچنین منابع داده‌های ما نیز ممکن است طبقه‌بندی‌های مختلفی از داده‌ها را با فرمت‌های گوناگون ارائه کنند که کار را برای سیستم یادگیری ماشین ما دشوار می‌سازد.

##### داده‌های گم‌شد چیست؟

وقتی دادگانمان را مورد بررسی قرار می‌دهیم، در برخی از سلول‌ها، جای برخی مقادیر خالی هستند (در

<sup>6</sup>Dataset

<sup>7</sup>US Centers for Disease Control and Prevention

پرسه جمع آوری داده ها این اطلاعات به هر دلیلی ثبت نشده اند) یا در اثر عملیات های مربوط به شناسایی داده های پرت، برخی از داده ها را حذف می کنیم و جای آن ها خالی می ماند. در این موقع باید با استفاده از تدابیری، داده های گم شده را با مقادیری جایگزین کنیم یا کلا آن رکوردها را حذف کنیم تا مدل های دقیق تری داشته باشیم. (شکل ۱۱-۲)

	Gender	Age	Race1	Education	MaritalStatus	Work	Weight	Height	BMI	BPSysAve	BPDiaAve	DirectChol
0	male	34	White	High School	Married	NotWorking	87.4	164.7	32.22	113.0	85.0	1.29
1	male	34	White	High School	Married	NotWorking	87.4	164.7	32.22	113.0	85.0	1.29
2	male	34	White	High School	Married	NotWorking	87.4	164.7	32.22	113.0	85.0	1.29
3	male	4	Other	Nan	Nan	Nan	17.0	105.4	15.30	Nan	Nan	Nan
4	female	49	White	Some College	LivePartner	NotWorking	86.7	168.4	30.57	112.0	75.0	1.16

شکل ۱۱-۲: نمونه ای از داده های گم شده

dataset.isnull().sum()	
Gender	0
Age	0
Race1	0
Education	1256
MaritalStatus	1247
Work	979
Weight	40
Height	180
BMI	186
BPSysAve	672
BPDiaAve	672
DirectChol	695
TotChol	695
PhysActive	763
Diabetes	74
dtype:	int64

شکل ۱۲-۲: تعداد داده های گم شده در این پروژه به ازای هر ستون

## راه حل های داده های گم شده

### • حذف ردیف های حامل داده های گم شده

یکی از راهکارهایی که در هنگام کار با داده های گم شده انجام می شود، حذف کل رکورد هایی است که دارای این مقادیر هستند. این مسئله یک بدی دارد و بدی آن این است که داده های کم تری برای آموزش مدلمنان در اختیار خواهیم داشت و مدل ما ضعیف تر خواهد بود. اما اگر به هر دلیلی نتوانیم این مقادیر را با مقادیری دارای تقریب خوب پر کنیم ، چاره دیگری نداریم.

### • جایگزینی با متوجه های آماری

یکی از راهکارهای دیگر برای مدیریت داده های گم شده، جایگزین کردن مقادیر گم شده با جایگزینی با متوجه های آماری است. برخی از ستون ها را که زیاد اهمیت نداشته باشند می توان با مقادیر میانگین پر کنیم. مثلا اگر یک دادگان داشته باشیم که حاوی اطلاعات مشتریان یک بانک باشد که بخواهیم از آن برای وام دادن به آن ها استفاده کنیم، میتوانیم در ستونی که مربوط

به میزان حقوق ماهیانه هر فرد می باشد، در صورت مشاهده مقادیر NaN، آنها را با میانگین حقوق مشتریان جایگزین کرد. این راهکار، صرفا به ما امکان می دهد تا تحلیل را ادامه دهیم و اطلاعات موجود در این رکورد را برای متغیرهای دیگر از دست ندهیم. [۱۲]

### استاندارد سازی داده‌ها

برای ادامه مراحل، لازم است کارهای بیشتری را روی داده‌های خود انجام دهیم. از جمله ایجاد متغیرهای ساختگی (دودویی) و مقایس بندی.

### متغیرهای ساختگی (دودویی)

در مدل‌های مختلف یادگیری ماشین به عنوان مثال در همین رگرسیون لجستیک، لازم است تا تنها متغیرهای عددی را به عنوان ورودی جهت مدل سازی ارائه کنیم و این مدل نمی‌تواند متغیرهای رشته‌ای را تشخیص دهد. پس از یک راهکار استفاده می‌کنیم. ستون‌هایی که حاوی مقادیر گسسته هستند را به چندین ستون دیگری تقسیم می‌کنیم (شکل ۲-۱۳) و مقادیر ستون‌های جدید را با بله/خیر (۰ و ۱) پر می‌کنیم. [۱۲] به عنوان مثال در ستون مربوط به وضعیت تا هل فرد، وضعیت‌های گوناگون را به ستون‌های معجزا تقسیم کردیم و برای هر کدام مقادیری در نظر گرفته شد. (شکل ۲-۱۴)

```
df3 = dataset_new.copy()
# These columns must be converted
df3 = pd.get_dummies(df3,columns = ['Gender','Race1','Education','MaritalStatus','Work','PhysActive','Diabetes'],
print(df3.columns)
dataset_new=df3.copy()

Python

Index(['Age', 'Weight', 'Height', 'BMI', 'BPSysAve', 'BPDiaAve', 'DirectChol',
       'TotChol', 'Gender_male', 'Race1_Hispanic', 'Race1_Mexican',
       'Race1_Other', 'Race1_White', 'Education_9 - 11th Grade',
       'Education_College Grad', 'Education_High School',
       'Education_Some College', 'MaritalStatus_LivePartner',
       'MaritalStatus_Married', 'MaritalStatus_NeverMarried',
       'MaritalStatus_Separated', 'MaritalStatus_Widowed', 'Work_NotWorking',
       'Work_Working', 'PhysActive_Yes', 'Diabetes_Yes'],
      dtype='object')
```

شکل ۲-۱۳: متغیرهای دودویی

MaritalStatus_LivePartner	MaritalStatus_Married	MaritalStatus_NeverMarried	MaritalStatus_Separated	MaritalStatus_Widowed
0	1	0	0	0
0	1	0	0	0
0	1	0	0	0
0	1	0	0	0
1	0	0	0	0

شکل ۲-۱۴: نمونه‌ای از متغیرهای دودویی

### مقایس بندی در نرمال‌سازی

برخی از الگوریتم‌ها نیاز دارند که داده‌ها قبل از پیاده سازی مؤثر الگوریتم نرمال‌سازی شوند. برای نرمال‌سازی یک متغیر، میانگین را از هر مقدار کم می‌کنیم و سپس بر انحراف استاندارد تقسیم می‌کنیم. این عملیات گاهی اوقات استانداردسازی نیز نامیده می‌شود. [۱۲]

علت این امر این است که واحد اندازه گیری مورد استفاده می‌تواند بر تجزیه و تحلیل داده‌ها تأثیر بگذارد. به عنوان مثال، تغییر واحدهای اندازه گیری از متر به اینچ برای قد، یا از کیلوگرم به پوند برای

وزن، ممکن است به نتایج بسیار متفاوتی منجر شود. [۱۲] به طور کلی، درنظرگرفتن یک ویژگی در واحدهای کوچک‌تر، به محدوده بزرگتری برای آن منجر می‌شود و بر وزن آن ویژگی اثرگذار است. برای جلوگیری از به وجود آمدن این مشکل، داده‌ها باید نرمال یا استاندارد شوند. یعنی داده‌ها در محدوده‌های کوچک‌تر یا نرمال‌تر قرار بگیرند. این مسئله باعث می‌شود تا به همه ویژگی‌ها وزن یکسانی داده شود. [۱۲] لذا قبل از انجام این عملیات، بررسی داده‌ها و ساختارشان دارای اهمیت است. [۲۳]

برای روش‌های مبتنی بر فاصله، نرمال‌سازی به جلوگیری از برتری ویژگی‌هایی با دامنه‌های اولیه بزرگ (مانند درآمد) کمک می‌کند تا ویژگی‌هایی با دامنه‌های اولیه کوچک‌تر (مانند ویژگی‌های باينری) سبقت بگیرد. [۱۲]

روش مورد استفاده در این پروژه از بین روش‌های مختلف StandardScaler است.

از رابطه زیر مقدار جدید ویژگی مقیاس بندی شده به دست می‌آید [۲۳]:

$$x_{scaled} = (x - mean(x)) \div StandardDeviation(x) \quad (۹-۲)$$

در واقع استانداردسازی فرآیندی است که متغیر را روی ° (میانگین صفر) قرار می‌دهد و واریانس را به ۱ (واریانس واحد) استاندارد می‌کند. پس از مقیاس بندی استاندارد، انحراف معیار نیز ۱ (انحراف معیار استاندارد) خواهد بود. در واقع برای استاندارد کردن ویژگی‌ها، میانگین را از هر مشاهده کم کردیم و سپس نتیجه را بر انحراف استاندارد تقسیم کردیم. خاصیت دیگر این کار، سرعت بخشیدن به روند ساخته شدن مدل‌های یادگیری ماست. [۲۳]

نتیجه این تبدیل، z-score نامیده می‌شود که نشان می‌دهد یک مشاهده معین از میانگین چند انحراف استاندارد انحراف دارد. از این رو، استانداردسازی، نرمال‌سازی امتیاز z نیز نامیده می‌شود. [۲۳]

در ایجاد بسیاری از مدل‌های یادگیری ماشین انجام این فرایند ضروری است. [۱۲]

### ۳-۴-۳ مصورسازی

**تعریف:** به طور ساده می‌توانیم بگوییم زمانی که داده‌هایمان را به صورت انواع نمودارها، نقشه‌ها و شکل‌های مختلف بصری دریاباوریم تا نتیجه‌گیری و تحلیل آنها توسط مغز جهت شناسایی الگوها و نقاط پرت در داده آسان‌تر شود، این کار انجام می‌گیرد. [۲۱]

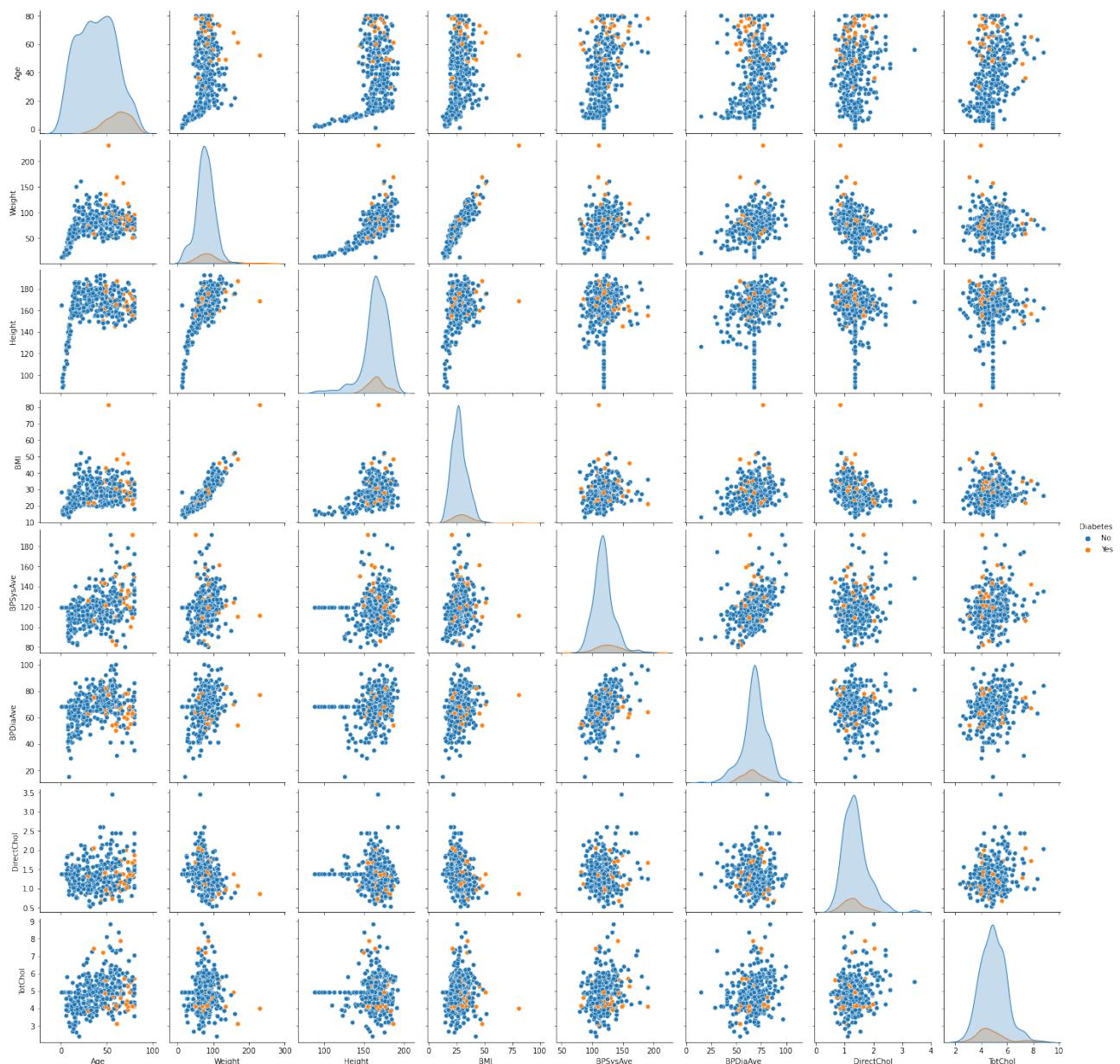
**اهمیت:** یک تصویر هزاران برابر بیشتر از کلمات ارزش دارد.

جمله فوق، به نوعی اهمیت مصورسازی را برای ما نمایان می‌کند. درواقع ما با انجام این کار، در ک بهتر و سریع‌تری نسبت به داده‌های عظیم خود خواهیم داشت. قابل ذکر است که Dataset مورد استفاده ما در اینجا، حدود ۱۰۰۰۰ رکورد را در خود جای داده است.

ضمیما در کنار این موارد، با مصورسازی داده‌ها دیگر نیازی به توضیحات اضافه نخواهیم داشت و بسیاری از افراد عادی قادر به درک موضوع مطرح شده خواهد بود. [۱۰]

## ۴-۴-۲ انواع نمودارها و داده‌های اجمالی

نمودار Scatter یا Pairplot



شکل ۱۵-۲

در شکل ۱۵-۲ نمودار Pairplot را مشاهده می‌کنیم. این نمودار برای تعیین رابطه بین دو متغیر استفاده می‌شود. هر نقطه داده با یک نقطه در نمودار نشان داده می‌شود که محور x و y متغیرهای مختلف را نشان می‌دهد. نمودارهای پراکندگی، به شناسایی خوشها، الگوها یا همبستگی‌های بین متغیرها کمک می‌کند و محققان را قادر می‌سازد تا در انتخاب شاخصه‌های کلیدی، تصمیمات آگاهانه بگیرند. [۱۰]

در این نمودار، نقاط نارنجی رنگ نشان دهنده افراد مبتلا به دیابت است و نقاط آبی رنگ نشان دهنده افرادی بدون ابتلا به این بیماری است و این نمودار از دادگان همین پروژه استخراج شده است. به

راحتی می توان تشخیص داد که افرادی که به دیابت مبتلا هستند در هر ویژگی فیزیولوژی در چه سمتی تمکن دارند.

### نقشه های حرارتی

نقشه های حرارتی در تجسم مقادیر زیادی از داده ها در قالب ماتریس مانند موثر هستند. آنها از گردایان رنگ برای نشان دادن مقدار مقادیر در دو متغیر استفاده می کنند. نقشه های حرارتی به ویژه در شناسایی همبستگی ها و کشف الگوهای پنهان در مجموعه داده های چند بعدی مفید هستند، و به انتخاب ویژگی و تشخیص نقاط پرت کمک می کنند. سایه های تیره تر با همبستگی قوی تر (مثبت یا منفی) مطابقت دارد. [۱۲]

نمونه ای از این نمودار که در پروژه خودمان استفاده کرده ایم در شکل ۱۶-۲ نمایش داده شده است.

	Age	Weight	Height	BMI	BPSysAve	BPDiaAve	DirectChol	TotChol
Age	1.000000	0.485982	0.448773	0.396693	0.436330	0.192314	0.087136	0.279955
Weight	0.485982	1.000000	0.722345	0.870981	0.211238	0.239157	-0.256111	0.109253
Height	0.448773	0.722345	1.000000	0.434615	0.099839	0.156478	-0.091657	0.056475
BMI	0.396693	0.870981	0.434615	1.000000	0.231158	0.213611	-0.268621	0.130599
BPSysAve	0.436330	0.211238	0.099839	0.231158	1.000000	0.426362	0.004474	0.202014
BPDiaAve	0.192314	0.239157	0.156478	0.213611	0.426362	1.000000	-0.019679	0.250050
DirectChol	0.087136	-0.256111	-0.091657	-0.268621	0.004474	-0.019679	1.000000	0.221467
TotChol	0.279955	0.109253	0.056475	0.130599	0.202014	0.250050	0.221467	1.000000

شکل ۱۶-۲: نقشه حرارتی

همانطور که ملاحظه می شود، در نقاط گرم تر، ارتباط بین دو متغیر قوی تر است.

نمودار های هیستوگرام (میله ای) برای نمایش داده های دسته بندی یا عددی در قالب گرافیکی استفاده می شود. نمودارهای میله ای داده ها را از طریق میله های عمودی نشان می دهند، در حالی که هیستوگرام ها توزیع متغیرهای پیوسته را نشان می دهند. این تجسم ها برای درک توزیع هر ویژگی و شناسایی عدم تعادل داده ها یا نقاط پرت مفید هستند. [۱۲] نمونه ای از آن ها در شکل ۲-۴ و ۴-۴ در بخش پیاده سازی نمایش داده شده اند.

### ۵-۴-۲ اندازه گیری میزان خطای دقت

به دلیل پدید آمدن مشکلاتی از قبیل بیش برازش و کم برازش داده ها که در بخش های قبلی به آن اشاره شد مدل های ما دچار استباها تی در پیش بینی ها می شود. در این بخش به بیان انواع خطاهای و روابطی که برای سنجش آن ها به کار گرفته ایم می پردازیم. [۱۱]

ابتدا به معروفی ۴ تاپل مختلف می پردازیم که بعدا از آن ها در محاسبه میزان خطای استفاده خواهیم کرد :

- **TP<sup>۸</sup>**: این تاپل موارد مثبت واقعی را علامت گذاری می کند. مثلا در این پروژه مواردی که احتمال ابتلا به دیابت در آن ها مثبت پیش بینی شده و در دادگان هم مثبت بوده در این دسته قرار می گیرد.

<sup>۸</sup>True positives

- **TN<sup>۹</sup>**: این تاپل موارد منفی واقعی را علامت گذاری می‌کند. مثلاً در این پروژه مواردی که احتمال ابتلا به دیابت در آن‌ها منفی پیش‌بینی شده و در دادگان هم منفی بوده در این دسته قرار می‌گیرد.
- **FP<sup>۱۰</sup>**: این تاپل موارد مثبت کاذب را علامت گذاری می‌کند. مثلاً در این پروژه مواردی که احتمال ابتلا به دیابت در آن‌ها مثبت پیش‌بینی شده اما در دادگان هم منفی بوده در این دسته قرار می‌گیرد.
- **FN<sup>۱۱</sup>**: این تاپل موارد منفی کاذب را علامت گذاری می‌کند. مثلاً در این پروژه مواردی که احتمال ابتلا به دیابت در آن‌ها منفی پیش‌بینی شده اما در دادگان هم مثبت بوده در این دسته قرار می‌گیرد.

• رابطه ۱۰-۲ نشان دهنده نسبت افراد سالم شناسایی شده واقعی به کل افراد سالم (افراد اشتباهه بیمار تشخیص داده شده به علاوه افراد سالم شناسایی شده واقعی) است.

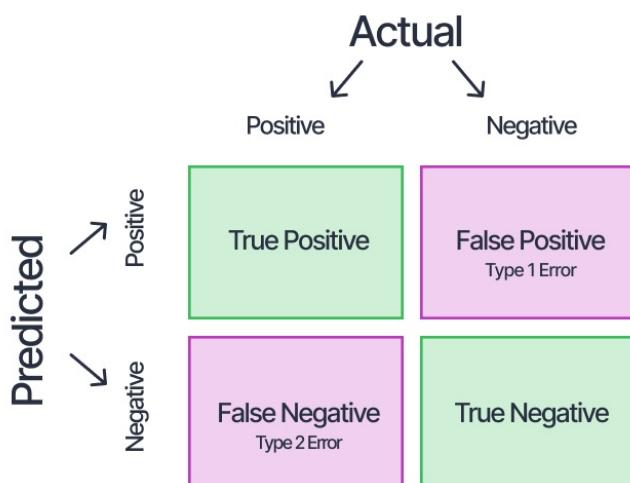
$$1 - Specificity = FPR = \frac{FP}{FP + TN} \quad (10-2)$$

• Sensitivity یا **TPR**

• رابطه ۱۱-۲ نشان دهنده نسبت بیماران شناسایی شده واقعی به کل بیماران (افراد اشتباهه سالم تشخیص داده شده به علاوه افراد بیمار شناسایی شده واقعی) است.

$$Sensitivity(Recall) = TPR = \frac{TP}{TP + FN} \quad (11-2)$$

تعدادی از این مقادیر را در ماتریس آشفتگی<sup>۱۲</sup> نشان می‌دهند [۲۶] و ساختار آن در شکل ۱۷-۲ است. همچنین این ماتریس را برای مدل جنگل درختان تصادفی رسم کردیم (شکل ۱۸-۲) :



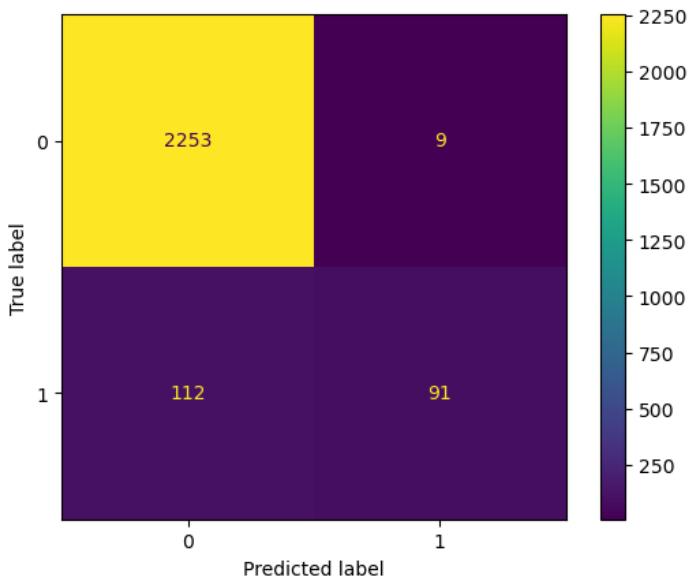
شکل ۱۷-۲: ساختار ماتریس آشفتگی: پارامترهای مذکور در این بخش در این ماتریس قرار گرفته اند.

<sup>۹</sup>True negatives

<sup>۱۰</sup>False positives

<sup>۱۱</sup>False negatives

<sup>۱۲</sup>Confusion



شکل ۲-۱۸: ساختار ماتریس آشفتگی برای الگوریتم جنگل درختان تصادفی

به طور کلی این ماتریس به ما خلاصه‌ای از درستی نتایج پیش‌بینی هایمان را نشان می‌دهد. [۲۶] حال میزان میزان دقیقت را با رابطه ۲-۱۲ محاسبه می‌کنیم.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (12-2)$$

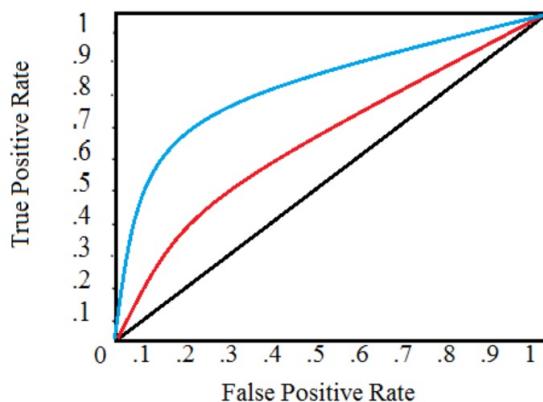
## ROC و AUC ۶-۴-۲

برای اینکه مقادیر پیش‌بینی‌های صحیح و غلط را به صورت یک نمودار نمایش دهیم، از دو شاخصه TPR و FPR که در بخش قبلی آنها را معرفی کردیم استفاده می‌کنیم. [۱۱] ما در اینجا مقادیر مختلف برای هر مرز (اشاره به تابع Sigmoid که در بخش‌های قبلی مطرح شد) پیش‌بینی را از ۰ تا ۱ اندازه گیری و در نمودار علامت می‌زنیم. [۱۱] اگر مقدار TPR برای یک مرز برابر با ۱ باشد یعنی در آن مرز مدل خیلی خوب عمل کرده و اگر برابر ۵٪ باشد یعنی تصادفی عمل شده است. [۱۱] مثلاً در یک مرز در نظر می‌گیریم اگر مقدار تابع Sigmoid از ۷٪ بیش تر بود، آن مورد را یک مورد مثبت شناسایی کن. در نمودار ROC <sup>۱۳</sup> مولفه‌ی طول‌ها برابر با مقدار FPR است و مولفه‌ی عرض‌ها برابر TPR در آستانه‌های گوناگون است. [۱۱]

سطح زیر این نمودار را با AUC <sup>۱۴</sup> نمایش می‌دهیم که بالا بودن این مساحت بیانگر این است که پیش‌بینی‌های بیشتری در این مدل درست بوده است. [۱۱]

<sup>۱۳</sup>Receiving Operating Characteristic

<sup>۱۴</sup>Area Under the Curve



شکل ۲-۱۹: نمای کلی نمودار ROC

## ۵-۲ نتیجه‌گیری

در این فصل سعی شد تا ادبیات و مفاهیم مهمی که در انجام این پژوهه با آن‌ها درگیر هستیم بیان شود از جمله بیان مفاهیم ابتدایی روش‌های داده کاوی شامل خوشه بندی و طبقه بندی (یادگیری بی ناظارت و بانظارت)، معرفی و بیان روش عملکرد الگوریتم‌های مورد استفاده از جمله رگرسیون لجستیک، درخت تصمیم، جنگل درختان تصادفی، AdaBoost و بیز ساده.

همچنین درمورد داده‌ها و عملیات آنها از جمله روش‌های پیش پردازش داده و انواع مصور سازی‌ها و درنهایت روش‌هایی برای سنجش و مقایسه کارایی الگوریتم‌هایمان نیز مفاهیمی مطرح شد.

در فصل آینده در مورد پیش زمینه‌های انجام این تحقیق مطالبی بیان می‌شود.

## فصل ۳

### کارهای پیشین

#### ۱-۳ مقدمه

پیرو مباحث قبلی، در انجام پژوهش‌های گوناگون بررسی و مقایسه روش‌های مختلف در روند توسعه و تحقیق مجدد، نکته بسیار مهمی است و می‌تواند اشکالات کارهای پیشین را برطرف نمود و در زمینه موارد به روز آن‌ها را به کار گرفت. همچنین ایده‌ها و نکاتی در هر مقاله ذکر شده است که می‌تواند رهنماوهای مفید برای کارهای آتی طلقی شوند. از جمله الگوریتم‌های مختلف و بیان روش‌های داده کاوی مربوط به آن

#### ۲-۳ مقاله ۱

مقاله [۷] در سال ۲۰۱۸ با استفاده از داده‌های بیماران هندی<sup>۱</sup> روند بررسی را بر روی سه الگوریتم درخت تصمیم، SVM و بیز ساده انجام داده و نتایج بیانگر این بوده که الگوریتم بیز ساده ۷۶٪ به عنوان موثر ترین الگوریتم در این بررسی درنظر گرفته شده است.

نکته قابل ملاحظه در این مقاله این است که از الگوریتم SVM هم برای پیش‌بینی استفاده شده است که معمولاً برای دادگان با مقادیر زیاد روش مناسبی است. [۲۵]

به طور خلاصه می‌توان گفت الگوریتم SVM با تقسیم داده‌ها به دسته‌های مختلف بر اساس داده‌های آموزشی داده شده کار می‌کند. سعی می‌کند خطی را ترسیم کند که به عنوان ابر صفحه (یک فضای  $n$  بعدی زیرمجموعه ابعاد  $1-n$  یا معادل آن که بعده‌مبند در  $V$  می‌باشد). شناخته می‌شود که به بهترین وجه دسته‌های مختلف داده‌ها را از هم جدا می‌کند. این کار را با یافتن ابر صفحه بهینه که فاصله بین نزدیکترین نقاط داده‌ها را به حداقل می‌رساند، انجام می‌دهد. [۲۵]

به عبارت ساده‌تر، داده‌ها را به صورت نقطه‌هایی روی یک تکه کاغذ در نظر بگیرید و هر نقطه به دسته خاصی تعلق دارد. الگوریتم SVM سعی می‌کند خطی را بر روی کاغذ بکشد که نقاط را به

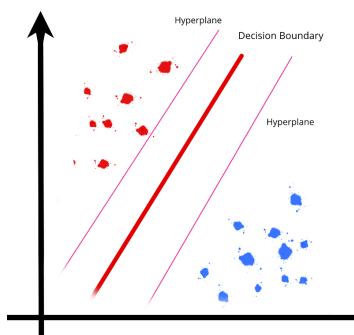
<sup>۱</sup>PIDD

بهترین نحو از هم جدا کند، به طوری که نقاط متعلق به یک دسته در یک طرف خط و نقاط متعلق به دسته دیگر در سمت دیگر قرار گیرند.

با این حال، گاهی اوقات نقاط داده را نمی توان به طور کامل با یک خط از هم جدا کرد. در چنین مواردی، الگوریتم از تکنیکی به نام "ترفند هسته" برای تبدیل داده ها به فضایی با ابعاد بالاتر استفاده می کند، جایی که یافتن یک خط جدا کننده آسان تر می شود. [۲۵]

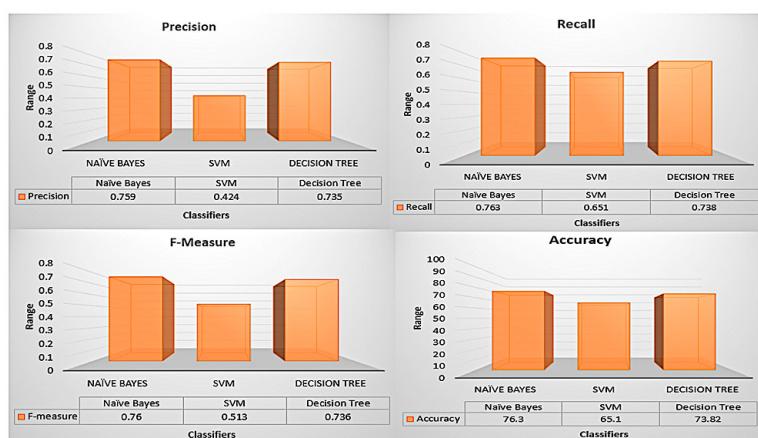
هنگامی که خط رسم شد، نقاط داده جدید و نادیده را می توان با تعیین اینکه در کدام سمت خط قرار می گیرند، به یک دسته اختصاص داد.

هدف SVM یافتن بهترین ابر صفحه است که حاشیه بین نقاط داده دسته های مختلف را به حداقل رساند، که به پیش بینی دقیق داده های جدید کمک می کند. [۲۵]

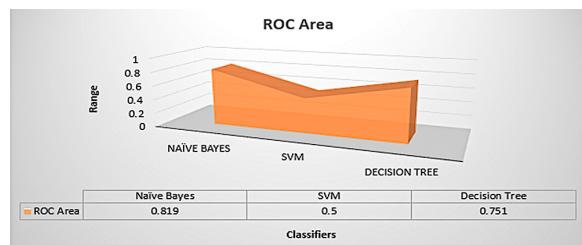


شکل ۱-۳: مدل SVM

خلاصه معیار های دقت طبقه بندی در این مقاله به شرح زیر است:



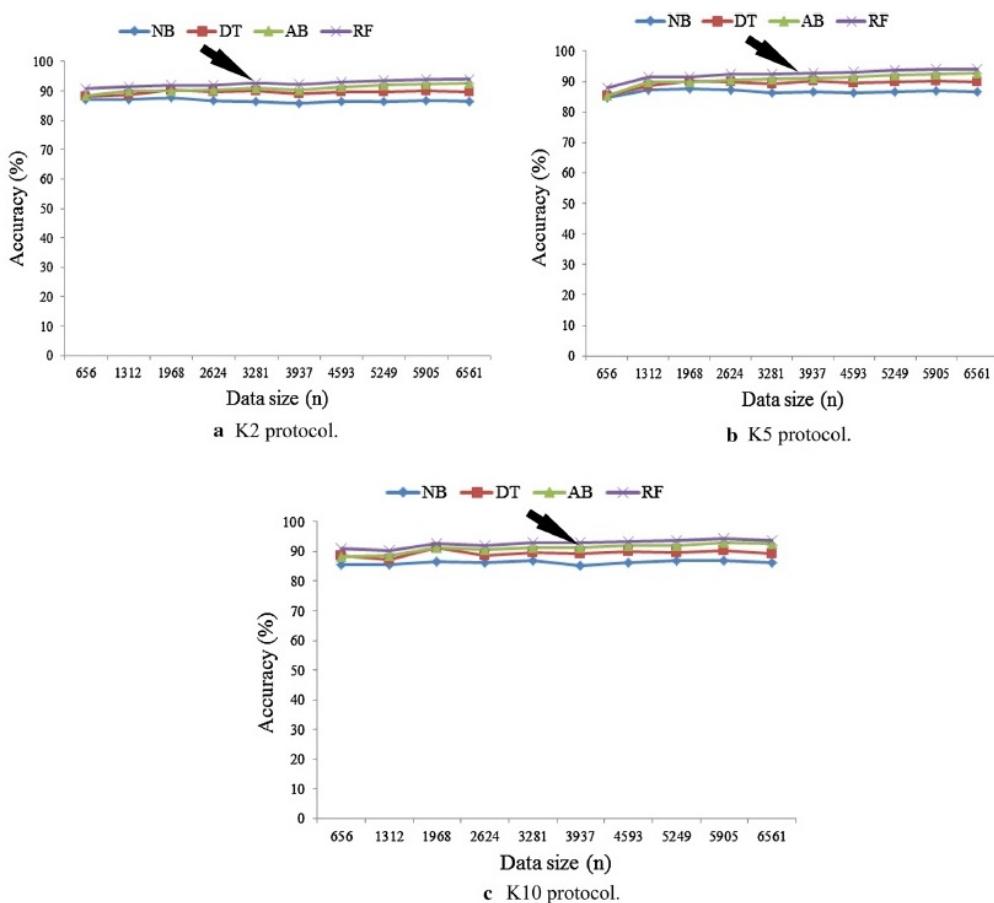
شکل ۲-۳: دقت های اندازه گیری در مقاله اول



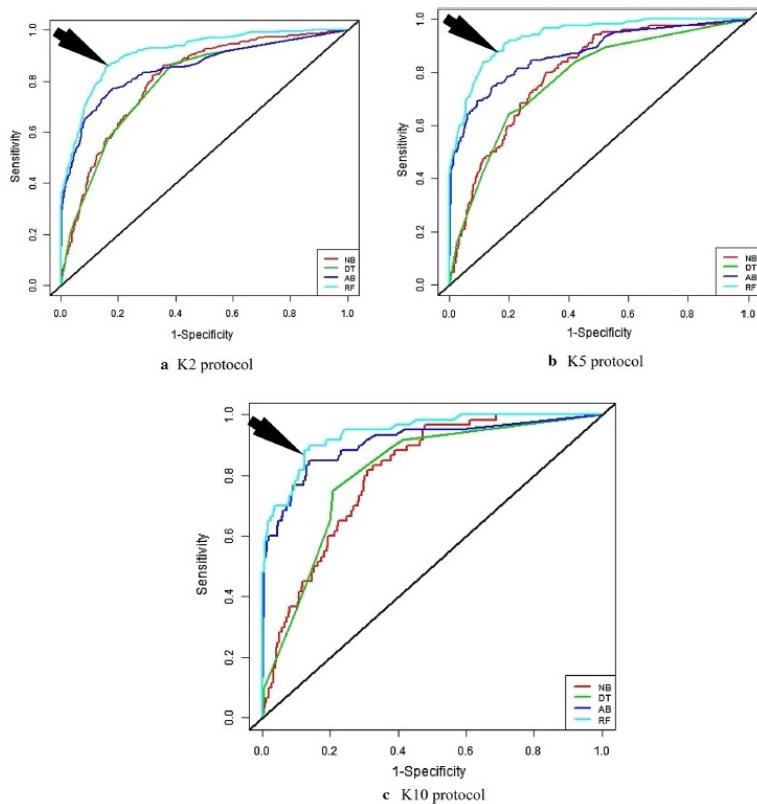
شکل ۳-۳: منحنی ROC در مقاله اول

### ۳-۳ مقاله ۲

در مقاله اشاره شده [۶] در سال ۲۰۲۰ روش‌های مختلفی برای پیش‌بینی ابتلا به دیابت انجام شده و ترکیب جنگل درختان تصادفی و رگرسیون لجستیک در اعتباری سنجی متقابل K-Fold "مکرر" با مقدار  $K=10$  بهترین نتیجه طبقه بندی را با دقت ۹۴٪ حاصل کرده است. در این مقاله روش‌های بیز ساده و AdaBoost هم مورد استفاده قرار گرفته که نتایج آنها داری دقت مناسبی نبوده است.



شکل ۴-۳: دقت های اندازه گیری در مقاله دوم



شکل ۳-۵: منحنی ROC در مقاله دوم

ضمیرا در این مقاله، از همان مجموعه دادگانی استفاده شده که در همین پایان نامه مورد استفاده قرار گرفته است.

### ۴-۳ نتیجه‌گیری

در این بخش، کوشیدیم تا برخی از مقالاتی که این موضوع را مورد بحث قرار دادند را بررسی کنیم و متوجه تفاوت‌ها در زمینه یافتن بهترین الگوریتم برای تشخیص ابتلا به دیابت، شویم.

در نتیجه، حاصل مطالعه مقالات اساسی که در گذشته تالیف شده بودند، این شد که الگوریتم‌های مختلفی در روند توسعه استفاده پروژه مان مورد بررسی واقع شوند که از جمله می‌توان به الگوریتم‌های بیز ساده، جنگل درختان تصادفی و الگوریتم درخت تصمیم و رگرسیون لجستیک اشاره کرد.

## فصل ۴

### روش‌ها و نتایج

#### ۱ - ۴ مقدمه

در این بخش بررسی می‌کنیم که الگوریتم‌های گوناگونی که برای ساخت مدل‌های گوناگون استفاده کردیم تا چه حد می‌تواند قابل اعتماد واقع شود و میزان خطا در هر کدام چقدر است. چرا که ما در این پژوهه، کار پیش‌بینی انجام دادیم و داده‌های آزمون به ما نشان داده که برخی از ردیف‌های به صورت اشتباه پیش‌بینی شده‌اند. برای انتخاب بهترین مدل روشی که انجام می‌شود این است که میزان خطاهای را با روش‌های گوناگون به دست آوریم و سپس الگوریتمی که بیشترین صحت را ارائه داده به عنوان بهترین مدل برگزینیم. [۱۱]

همچنین برای بخش دیگری از این سنجش می‌توانیم نمودارهای گوناگونی را رسم کنیم از جمله ROC که می‌تواند این میزان دقت را به صورت بصری به ما نمایش دهد.

و در نهایت به بررسی روش طراحی یک مدل ساده برای سامانه آنلاین پیش‌بینی دیابت با ابزارهای اینترنت اشیاء و دو چارچوب NodeRed و Flask پرداختیم.

#### ۲ - ۴ روش پیشنهادی

##### ۱ - ۲ - ۴ پیاده‌سازی

پس از وارد کردن کتابخانه‌های مورد نیاز، به عملیات وارد کردن داده‌ها و پیش‌پردازش آن‌ها پرداختیم:

```
dataset_new = dataset_new.dropna(subset=['Diabetes'])
```

در کد بالا، عملیات حذف داده‌های گم شده در ستون متغیر وابسته انجام شد. در واقع ردیف‌هایی که ستون آخرشان (یعنی Diabetes) دارای مقادیر نامشخص بودند را حذف کردیم. زیرا تمام مدل سازی‌های ما وابسته به آخرین ستون است و اگر این ستون مقادیر نادرستی داشته باشد، مدل‌هایی که می‌سازیم دقت پایینی خواهند داشت.

پس از شمارش تعداد ردیف‌هایی با مقادیر گم شده، عملیات مربوط به جایگزینی آن‌ها را با روش زیر انجام دادیم. یعنی مقادیر ستون‌هایی که مقدار پیوسته داشتند را با میانگین ستون جایگزین کردیم و در مقادیر گسسته، از مُد (نمونه‌ای با بیشترین تکرار) استفاده کردیم.

```
dataset_new["Weight"].fillna(dataset_new["Weight"].mean(), inplace = True)
dataset_new["Height"].fillna(dataset_new["Height"].mean(), inplace = True)
dataset_new["BMI"].fillna(dataset_new["BMI"].mean(), inplace = True)
dataset_new["BPSysAve"].fillna(dataset_new["BPSysAve"].mean(), inplace = True)
dataset_new["BPDiaAve"].fillna(dataset_new["BPDiaAve"].mean(), inplace = True)
dataset_new["DirectChol"].fillna(dataset_new["DirectChol"].mean(), inplace = True)
dataset_new["TotChol"].fillna(dataset_new["TotChol"].mean(), inplace = True)

dataset_new.isnull().sum()
dataset_new["PhysActive"].fillna(dataset_new["PhysActive"].mode()[0], inplace = True)
dataset_new["Education"].fillna(dataset_new["Education"].mode()[0], inplace = True)
dataset_new["MaritalStatus"].fillna(dataset_new["MaritalStatus"].mode()[0], inplace = True)
dataset_new["Work"].fillna(dataset_new["Work"].mode()[0], inplace = True)

dataset_new.isnull().sum()
```

شکل ۴-۱: جایگزینی مقادیر نامشخص با متوسط‌های آماری

یکی از کارهای دیگری که انجام شد، جایگزینی مقادیر  $\circ$  با NaN در ستون‌هایی با مقادیر پیوسته بود مثل سن. زیرا به خوبی مشخص است این موارد از جمله داده‌های پرت محاسبه می‌شوند که دقت مدل‌های مارا کاهش می‌دهند.

```
dataset_new[['Education','MaritalStatus','Work','Weight','Height','BMI',
'BPSysAve','BPDiaAve','DirectChol','TotChol','PhysActive']] = dataset_new
[[['Education','MaritalStatus','Work','Weight','Height','BMI','BPSysAve',
'BPDiaAve','DirectChol','TotChol','PhysActive']]].replace(0, np.Nan)
```

سپس عملیات مربوط ایجاد مصورسازی داده‌ها صورت گرفت:

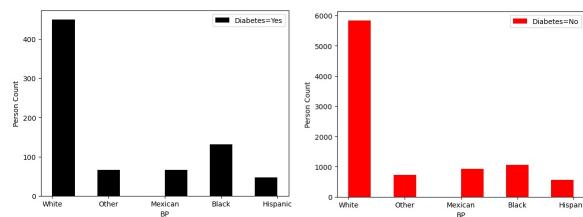
```
YesDia = dataset_new['Diabetes'].values == 'Yes'
NoDia = dataset_new['Diabetes'].values == 'No'
YesDia=dataset_new[YesDia]
NoDia=dataset_new[NoDia]
```

```
Race1 = YesDia['Race1'].tolist()
Race0 = NoDia['Race1'].tolist()
```

```
plt.hist([Race1], color=[ 'Black'], label=['Diabetes=Yes'])
plt.xlabel('BP')
plt.ylabel('Person Count')
plt.legend()
plt.show()
```

```
plt.hist([Race0], color=[  
    'Red'], label=['Diabetes=No'])  
plt.xlabel('BP')  
plt.ylabel('Person Count')  
plt.legend()  
plt.show()
```

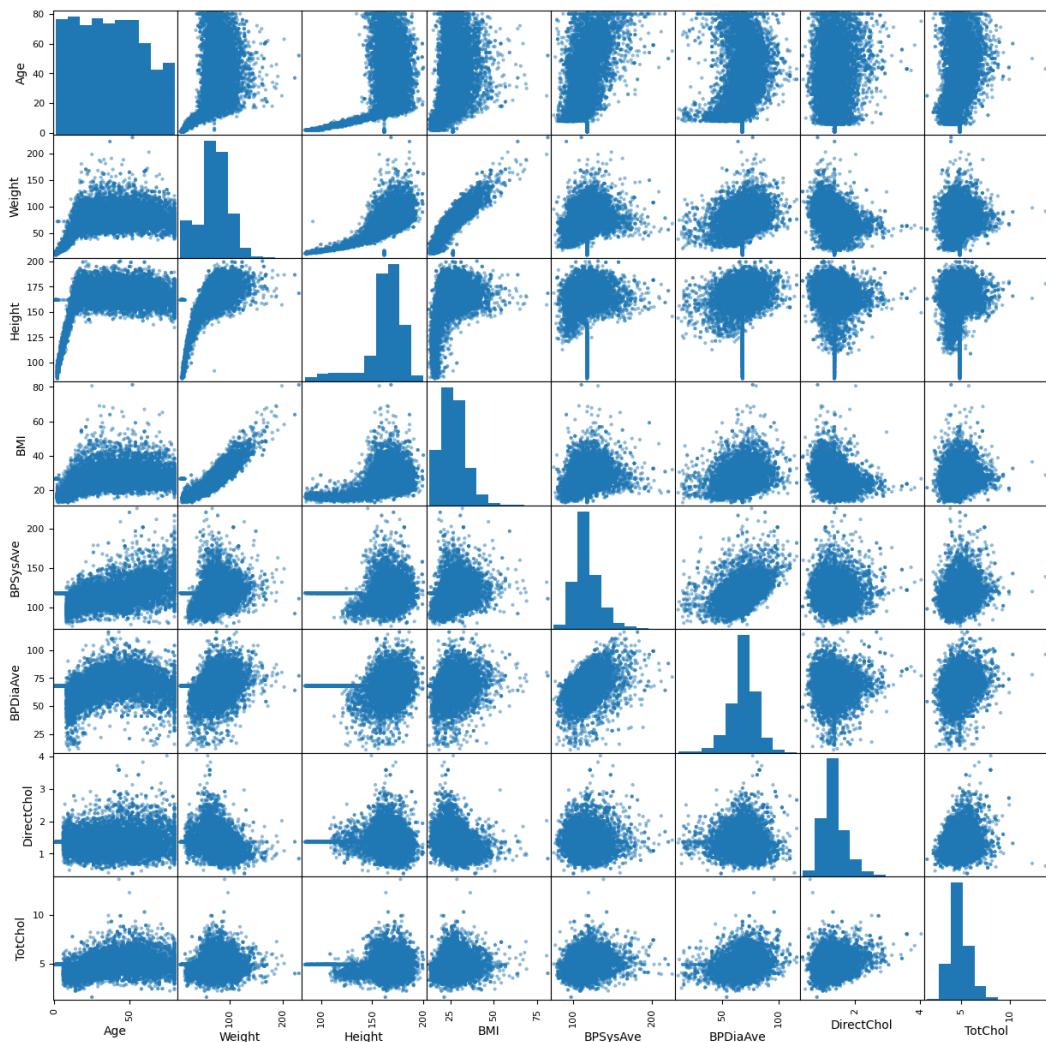
نتیجه عملیات فوق، نمودار های زیر می باشد (در این نمودار، تاثیر نژاد در ابتلا به دیابت مشخص است):



شکل ۲-۴: نمودار مقایسه ابتلا به دیابت در نژاد های مختلف

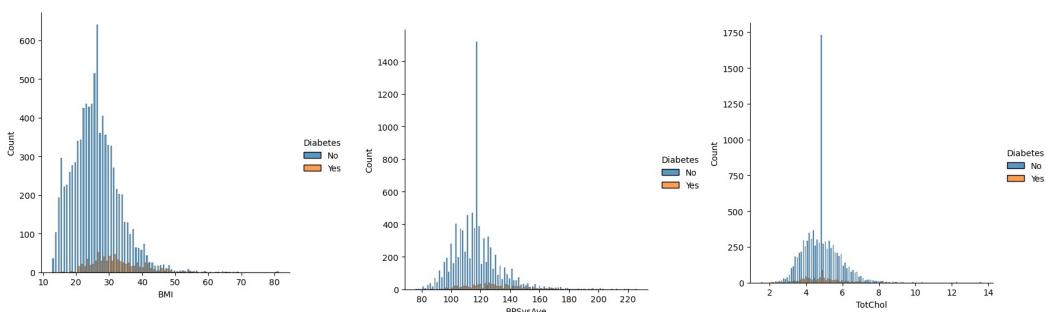
سپس با کد زیر، ماتریس scatter برای نمایش همبستگی میان داده ها رسم می کنیم:

```
scatter_matrix(dataset , figsize=(15, 15))  
plt.show()
```

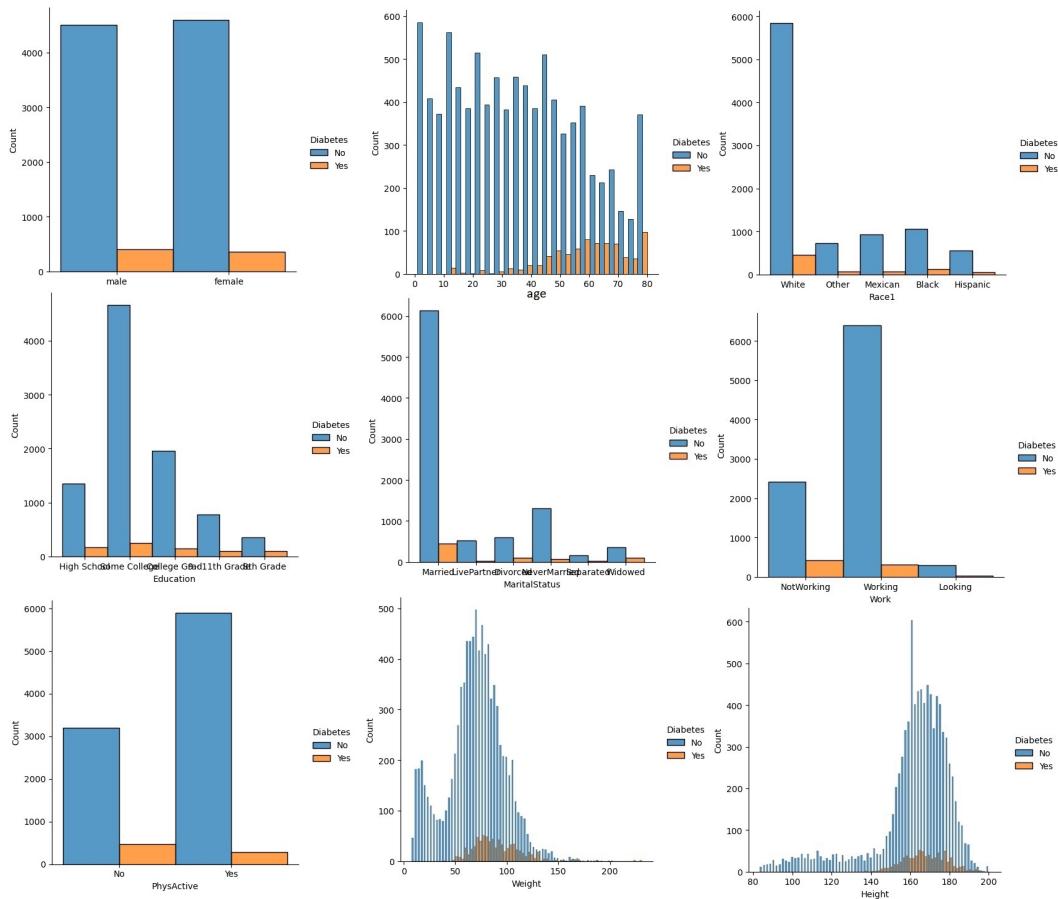


شکل ۳-۴: ماتریس Scatter

در شکل ۳-۴، برخی عوامل مرتبط با هم مشخص اند. (مثل رابطه کلسترول و وزن) در نمودارهایی که در ادامه خواهد دید، تاثیر پارامتر های گوناگون در ابتلای افراد به دیابت مورد بررسی است (شاخص)، (BMI فشارخون، کلسترول خون، جنسیت، سن، نژاد، وضعیت تأهل، تحصیلات، وضعیت شغلی، قد، وزن) :



شکل ۴-۴: نمودار عوامل موثر در ابتلا دیابت (۱)



شکل ۴-۵: نمودار عوامل موثر در ابتلا دیابت (۲)

همچنین از کد زیر برای رسم نمودارهای فوق استفاده شد.

```

sns.countplot(x = 'PhysActive', data = YesDia)
sns.countplot(x = 'PhysActive', data = NoDia)
sns.lineplot(x="Age", y="Diabetes", data=dataset_new)

for i in dataset_new.columns:
    sns.displot(dataset, x=i, multiple="dodge", hue="Diabetes")

```

نمودار نقشه حرارتی را نیز با این کد رسم کردیم:

```

df = dataset_new
corr = df.corr()
#Displaying dataframe of correlation values
corr.style.background_gradient(cmap = 'coolwarm')

```

پارامترهایی که با هم ارتباط دارند، با رنگ های گرم مشخص اند.

	Age	Weight	Height	BMI	BPSysAve	BPDiaAve	DirectChol	TotChol
Age	1.000000	0.485982	0.448773	0.396693	0.436330	0.192314	0.087136	0.279955
Weight	0.485982	1.000000	0.722345	0.870981	0.211238	0.239157	-0.256111	0.109253
Height	0.448773	0.722345	1.000000	0.434615	0.099839	0.156478	-0.091657	0.056475
BMI	0.396693	0.870981	0.434615	1.000000	0.231158	0.213611	-0.268621	0.130599
BPSysAve	0.436330	0.211238	0.099839	0.231158	1.000000	0.426362	0.004474	0.202014
BPDiaAve	0.192314	0.239157	0.156478	0.213611	0.426362	1.000000	-0.019679	0.250050
DirectChol	0.087136	-0.256111	-0.091657	-0.268621	0.004474	-0.019679	1.000000	0.221467
TotChol	0.279955	0.109253	0.056475	0.130599	0.202014	0.250050	0.221467	1.000000

شکل ۶-۴: نقشه حرارتی

حال پس از بخش مصور سازی به سراغ آماده سازی داده ها برای ایجاد مدل ها پرداختیم. پس متغیرهای دودویی را با کد زیر ایجاد کردیم:

```
df3 = dataset_new.copy()
df3 = pd.get_dummies(df3,columns = ['Gender', 'Race1','Education',
'MaritalStatus','Work','PhysActive','Diabetes'], drop_first = True)
print(df3.columns)
```

سپس ستون Y و X هایمان مشخص کردیم تا داده هایمان را به بخش های آموزشی و سنجش تقسیم کنیم. چون در ساخت مدل ها، باید داده ها را به چهار دسته های X آموزشی، های X سنجش، Y های آموزشی و Y های سنجش، تقسیم کنیم. [۱۲] مقیاسی که برای داده های سنجش در نظر گرفته شده ۲۵٪ است.

```
X = dataset_new.iloc[:, :-1].values
Y = dataset_new.iloc[:, -1].values
```

```
XTrain, XTest, YTrain, YTest = train_test_split(X,Y, test_size = 0.25,
random_state = 0)
```

در مرحله بعد، داده ها را مقیاس بندی می کنیم تا در مدل های موردنظرمان مورد استفاده قرار گیرند:

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
XTrain = sc.fit_transform(XTrain)
XTest = sc.transform(XTest)
```

با توجه به اینکه برای هر مدل باید الگوریتم K-Fold تکرار شونده برای  $K=2,5,10$  به تعداد ۲۰ مرتبه انجام شود و مقادیر AUC و ACC به دست آیند، دوتابع ایجاد می کنیم:  
ساخت لیست مربوط به مقادیر ACC :

```
#Importing required libraries
from sklearn.model_selection import RepeatedKFold
from sklearn.metrics import accuracy_score
```

```

from numpy import mean
from numpy import std
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score

def Kfold_modulation(input_model):
    #Implementing cross validation
    k_list = [2,5,10]
    acc_list=[]
    for k in k_list:
        kf = KFold(n_splits=k,shuffle=False, random_state=None)
        model = input_model
        acc_score = []
        scores = cross_val_score(model, X, Y, scoring='accuracy', cv=kf)
        avg_acc_score = mean(scores)
        # print('Avg acc : avg_acc_score')
        acc_list.append(avg_acc_score)
    return acc_list

```

ساخت لیست مربوط به مقادیر : AUC

```

from sklearn.model_selection import cross_val_score
def Kfold_modulation2(input_model):
    #Implementing cross validation
    k_list = [2,5,10]
    auc_list=[]
    for k in k_list:
        mean_score = cross_val_score
        (input_model, X, Y, scoring="roc_auc", cv = k).mean()
        auc_list.append(mean_score)
    return auc_list

```

سپس هر مدل را ساخته و میزان AUC و ACC را برای هر کدام در یک لیست می‌ریزیم:

```

from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
logreg_acc=Kfold_modulation(logreg)
logreg_auc=Kfold_modulation2(logreg)
logreg.fit(XTrain, YTrain)
logreg_pred=logreg.predict(XTest)

```

```

print(logreg_acc)

from sklearn.ensemble import RandomForestClassifier
ranfor = RandomForestClassifier()
ranfor_acc=Kfold_modulation(ranfor)
ranfor_auc=Kfold_modulation2(ranfor)
ranfor.fit(XTrain, YTrain)
ranfor_pred=ranfor.predict(XTest)

from sklearn.tree import DecisionTreeClassifier
DecTree = DecisionTreeClassifier()
DecTree_acc=Kfold_modulation(DecTree)
DecTree_auc=Kfold_modulation2(DecTree)
DecTree.fit(XTrain, YTrain)

from sklearn.ensemble import AdaBoostClassifier
AdaBoost = AdaBoostClassifier()
AdaBoost_acc=Kfold_modulation(AdaBoost)
AdaBoost_auc=Kfold_modulation2(AdaBoost)
AdaBoost.fit(XTrain, YTrain)
AdaBoost_pred=AdaBoost.predict(XTest)

from sklearn.naive_bayes import GaussianNB
nb=GaussianNB()
nb_acc=Kfold_modulation(nb)
nb_auc=Kfold_modulation2(nb)
nb.fit(XTrain, YTrain)
nb_pred=nb.predict(XTest)

```

سپس مقادیر AUC و ACC را که برای هر الگوریتم را که در لیست های جداگانه قرار دارند، را باهم ترکیب می کنیم و یک دیتابست حاوی تمامی مقادیر AUC و ACC ایجاد می کنیم تا با استفاده از آن، نمودار های ACC، AUC و ROC را رسم می کنیم.

```

acc_list0=[logreg_acc,ranfor_acc,DecTree_acc,AdaBoost_acc,nb_acc]
acc_list=[]
for i in acc_list0 :
    my_formatted_list = [ '%.4f' % elem for elem in i ]
    list=[]
    acc_list.append(my_formatted_list)
auc_list0=[logreg_auc,ranfor_auc,DecTree_auc,AdaBoost_auc,nb_auc]

```

```

auc_list=[]
for i in auc_list0 :
    my_formatted_list = [ '%.4f' % elem for elem in i ]
    list=[]
    auc_list.append(my_formatted_list)
print(auc_list)

bar=pd.DataFrame([acc_list[0],acc_list[1],acc_list[2],acc_list[3],acc_list[4]])
bar['algo'] = ['LR','RF','DT','AB','NB']
bar.columns=['K2' , 'K5' , 'K10','Algorithm']

bar['K2']=bar['K2'].astype('float64')*100
bar['K5']=bar['K5'].astype('float64')*100
bar['K10']=bar['K10'].astype('float64')*100

print('ACC','\n',bar,'\n')
barauc=pd.DataFrame([auc_list[0],auc_list[1],auc_list[2],auc_list[3],auc_list[4]])
barauc['algo'] = ['LR','RF','DT','AB','NB']
#barauc.insert()=['logreg_auc','ranfor_auc','DecTree_auc','AdaBoost_auc','nb_auc']
barauc.columns=['K2' , 'K5' , 'K10','Algorithm']
barauc['K2']=barauc['K2'].astype('float64')*100
barauc['K5']=barauc['K5'].astype('float64')*100
barauc['K10']=barauc['K10'].astype('float64')*100
print('\n','AUC','\n',barauc)

```

سپس با کد زیر، نمودار مقایسه ACC و AUC را برای الگوریتم‌های مختلفمان در Fold های ۲، ۵ و ۱۰ رسم می‌کنیم:

```

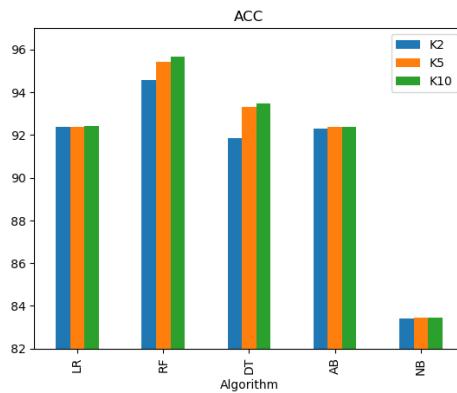
bar['K2'] = bar['K2'].astype('float64')
bar['K5'] = bar['K5'].astype('float64')
bar['K10'] = bar['K10'].astype('float64')
# plot grouped bar chart
bar.plot(x='Algorithm',
          kind='bar',
          stacked=False,
          ylim=[82, 97],
          title='ACC')

barauc['K2'] = barauc['K2'].astype('float64')
barauc['K5'] = barauc['K5'].astype('float64')
barauc['K10'] = barauc['K10'].astype('float64')

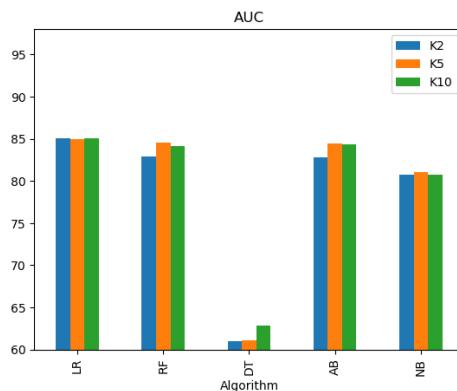
```

```
# plot grouped bar chart
barauc.plot(x='Algorithm',
             kind='bar',
             stacked=False,
             ylim=[60, 98],
             title='AUC')
```

نمودار ها:

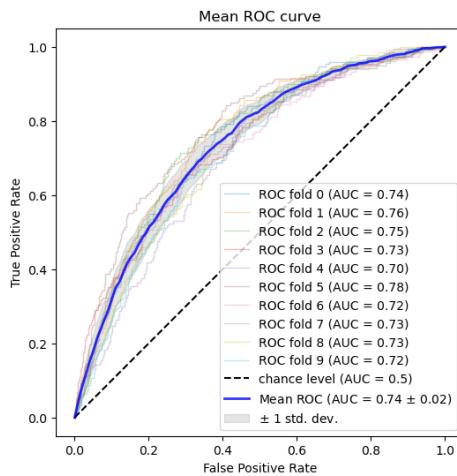


شکل ۷-۴: نمودار ACC

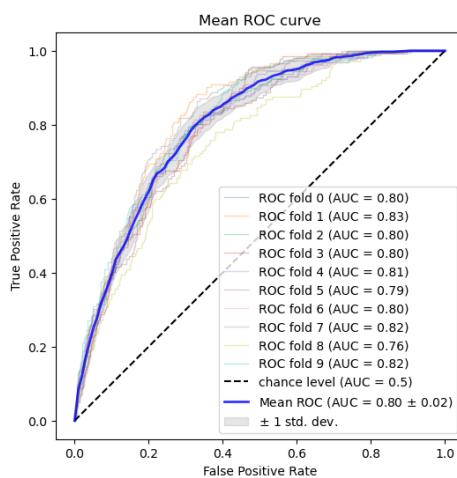


شکل ۸-۴: نمودار AUC

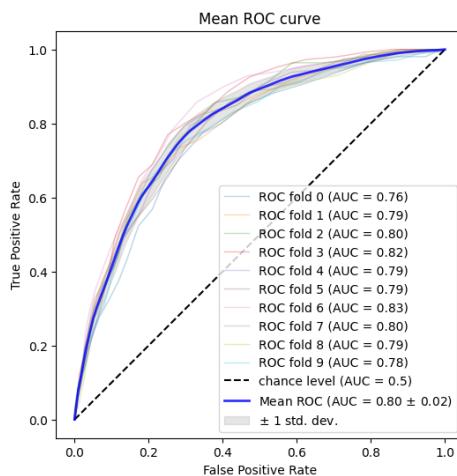
حال برای الگوریتم های مختلف، نمودار ROC را در Fold های مختلف رسم می کنیم و میانگین آن نیز مشاهده می شود:



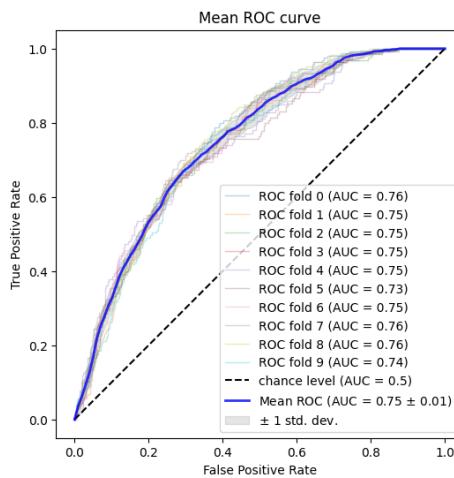
شکل ۹-۴: نمودار ROC برای الگوریتم رگرسیون لجستیک



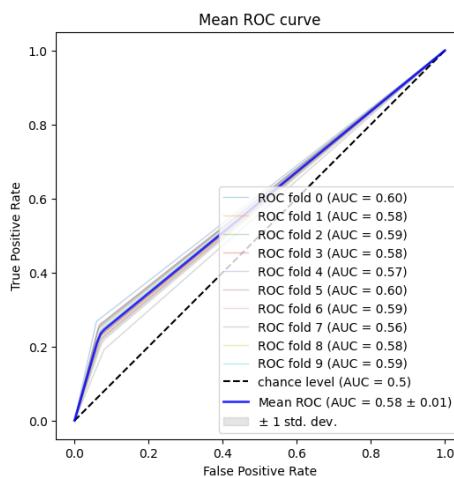
شکل ۱۰-۴: نمودار ROC برای الگوریتم AdaBoost



شکل ۱۱-۴: نمودار ROC برای الگوریتم جنگل درختان تصادفی



شکل ۱۲-۴ : نمودار ROC برای الگوریتم Naive Bayes



شکل ۱۳-۴ : نمودار ROC برای الگوریتم درخت تصمیم

از کد زیر برای رسم نمودارهای فوق استفاده کردیم:

```

XR, YR = X, Y
XR, YR = XR[YR != 2], YR[YR != 2]
n_samples, n_features = XR.shape
random_state = np.random.RandomState(0)
XR = np.concatenate([XR, random_state.randn(n_samples, 200 * n_features)], axis=1)

import matplotlib.pyplot as plt
from sklearn import svm
from sklearn.metrics import auc
from sklearn.metrics import RocCurveDisplay
from sklearn.model_selection import StratifiedKFold

#cv = StratifiedKFold(n_splits=5)

```

```

cv=RepeatedKFold(n_splits=5, n_repeats=2, random_state=None)
classifier = logreg

tprs = []
aucs = []
mean_fpr = np.linspace(0, 1, 100)

fig, ax = plt.subplots(figsize=(6, 6))
for fold, (train, test) in enumerate(cv.split(XR, YR)):
    classifier.fit(XR[train], YR[train])
    viz = RocCurveDisplay.from_estimator(
        classifier,XR[test],YR[test],name=f"ROC fold {fold}",alpha=0.3,lw=1,ax=ax,)

    interp_tpr = np.interp(mean_fpr, viz.fpr, viz.tpr)
    interp_tpr[0] = 0.0
    tprs.append(interp_tpr)
    aucs.append(viz.roc_auc)

ax.plot([0, 1], [0, 1], "k--", label="chance level (AUC = 0.5)")
print(len(aucs))
#print(tprs)
print(aucs[5])
mean_tpr = np.mean(tprs, axis=0)
mean_tpr[-1] = 1.0
mean_auc = auc(mean_fpr, mean_tpr)

std_auc = np.std(aucs)

ax.plot(mean_fpr,mean_tpr,color="b",
label=r"Mean ROC (AUC = %0.2f $\pm$ %0.2f)" % (mean_auc, std_auc),lw=2,alpha=0.8,)

std_tpr = np.std(tprs, axis=0)
tprs_upper = np.minimum(mean_tpr + std_tpr, 1)
tprs_lower = np.maximum(mean_tpr - std_tpr, 0)
ax.fill_between(mean_fpr,tprs_lower,tprs_upper,color="grey",alpha=0.2,
label=r"$\pm$ 1 std. dev.),ax.set(xlim=[-0.05, 1.05],ylim=[-0.05, 1.05],
xlabel="False Positive Rate",ylabel="True Positive Rate",title=f"Mean ROC curve",)

ax.axis("square")
ax.legend(loc="lower right")
plt.show()
print(type(aucs[0]))
aucs2=[]

```

```

for i in aucs:
    for u in aucs:
        aucs2.append(u.item())

def Average(lst):
    return sum(lst) / len(lst)
average1 = Average(aucs2)
print(average1)

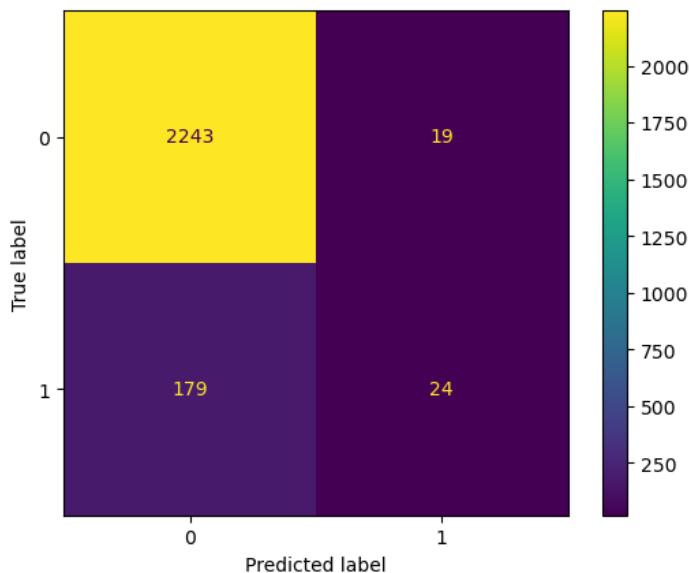
```

و در نهایت ماتریس آشتفتگی را با استفاده از کد زیر برای الگوریتم‌های مختلف رسم کردیم:

```

from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
#In the models, we put test data to make predictions for us.
YP_ranfor = ranfor.predict(XTest)
cm = confusion_matrix(YTest, YP_ranfor, labels=None)
print(cm)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot()

```



شکل ۱۴-۴: ساختار ماتریس آشتفتگی برای الگوریتم رگرسیون لجستیک

## Arduino ۲-۲-۴

آردوینو یک بستر منبع باز است که برای ساخت پروژه‌های الکترونیکی استفاده می‌شود و شامل یک برد مدار فیزیکی قابل برنامه‌ریزی و یک نرم افزار به نام محیط توسعه یکپارچه (IDE) است که روی کامپیوتر اجرا می‌شود و برای نوشتن و آپلود کدهای کامپیوتربی روی برد استفاده می‌شود. پلتفرم آردوینو به این دلیل محبوب است که برای بارگذاری کد جدید روی برد نیازی به برنامه نویس جداگانه

ندارد و از نسخه ساده شده C++ برای برنامه نویسی استفاده می‌کند. این پلتفرم یک فرم فاکتور استاندارد را فراهم می‌کند که استفاده از عملکردهای میکروکنترلر را آسان تر می‌کند. [۳۰]

بوردهای آردوینو در اتصالات برق، پین‌ها برای اتصال سیم‌ها، دکمه تنظیم مجدد، انواع نشانگرهای LED برای ارتباطات گوناگون از جمله سریال، I<sub>C</sub> اصلی و تنظیم‌کننده ولتاژ در مدل‌های مختلف تفاوت دارند. [۳۰] آردوینو می‌تواند با قطعات و سنسورهای مختلف تعامل داشته باشد و از طریق پروتکل MQTT می‌تواند با دستگاه‌های مختلف مبتنی بر اینترنت اشیاء تعامل داشته باشد. [۳۱]

درمورد راه اندازی Arduino IDE دو نکته ضروری است:

۱. حتماً باید پکیج مربوط به بورد مد نظر را بر روی آن نصب کنید. (برای مدلی که استفاده می‌کردم (WeMos D1 R1) آن را بالینک<sup>۱</sup> در file> performances> additional board manager url وارد کردم.)

۲. برای استفاده از MQTT باید بسته espmqttclient را از طریق Arduino IDE نصب کرد.

#### MQTT

این پروتکل، یک پروتکل پیام‌رسانی است که برای ارتباط دستگاه‌ها در اینترنت اشیا (IoT) استفاده می‌شود. سیک، کارآمد و مقیاس‌پذیر است و برای انتقال داده‌ها از طریق شبکه‌هایی با محدودیت منابع، ایده‌آل است. MQTT از پیام‌رسانی بین دستگاه‌ها و ابر پشتیبانی می‌کند و قابلیت اطمینان و ویژگی‌های امنیتی را ارائه می‌دهد. [۲۸]

به جای ارتباط مستقیم بین مشتریان و سرورها، MQTT از یک واسطه پیام برای مدیریت ارتباط بین فرستنده‌گان و دریافت کننده‌گان استفاده می‌کند. کارگزار پیام‌های فرستنده‌گان را فیلتر کرده و بین گیرنده‌گان توزیع می‌کند. این کار فرستنده و گیرنده پیام‌ها را از نظر مکان و زمان جدا می‌کند. [۲۸]

اجزای MQTT شامل کلاینت‌های MQTT است که دستگاه‌هایی هستند که با استفاده از MQTT ارتباط برقرار می‌کنند و یک کارگزار MQTT که پیام‌ها را بین مشتریان هماهنگ می‌کند. کاربران می‌توانند پیام‌هایی را با موضوعات و داده‌ها منتشر کنند و همچنین می‌توانند برای دریافت پیام‌ها مشترک موضوعات خاصی شوند. [۲۸]

همچنین (MQTT over WebSockets (WSS)) یک پیاده‌سازی از MQTT است که اجازه می‌دهد داده‌ها مستقیماً در یک مرورگر وب دریافت شوند. ارتباط MQTT ایمن است و می‌توان با استفاده از پروتکل SSL، گواهی‌ها و رمزهای عبور محافظت کرد. [۲۸]

برای راه اندازی یک سرور MQTT به صورت محلی، می‌توان از ابزاری به نام mosquitto<sup>۲</sup> که در سیستم عامل‌های مختلف در دسترس است، استفاده کرد.

درمورد راه اندازی mosquitto نکته‌ای که قابل توجه است این است که بایستی پورت 1883 در تنظیمات این برنامه باز باشد و امکان اتصال به صورت ناشناس نیز ممکن باشد تا بتوان به راحتی و بدون دردسر به آن متصل شد برای این منظور مراحل زیر بایستی طی شوند.

۱. ابتدا آن را متوقف می‌کنیم:

<sup>۱</sup>[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

<sup>۲</sup>[mosquitto.org](http://mosquitto.org)

در لینوکس : sudo systemctl stop mosquitto

در ویندوز : sc stop mosquitto

۲. سپس به فایل mosquitto.conf دسترسی پیدا می‌کنیم و آن را با یک ویرایشگر باز می‌کنیم. این فایل در سیستم عامل‌های مختلف مسیر متفاوت دارد:

در لینوکس : /etc/mosquitto

در ویندوز : C:\Program Files\mosquitto\

۳. پس از بازگشایی این فایل، دو خط زیر را به آن می‌افزاییم:

```
allow_anonymous true
listener 1883
```

۴. دستور c mosquitto را در خط فرمان وارد می‌کنیم.

۵. سپس آن را مجدداً راه اندازی می‌کنیم:

در لینوکس : sudo systemctl start mosquitto

در ویندوز : sc start mosquitto

در کدی که برای بورد آردوینو نوشتم، ابتدا تنظیمات اتصال به WiFi را انجام و تنظیمات مربوط به اتصال MQTT را با وارد کردن IP سرور و تعیین نام و گذر واژه برای آن انجام داده و پس از مشخص کردن پین‌های مربوط به LED، تعیین می‌کنیم در هر ۱۰ ثانیه یک عدد دو رقمی تولید و از طریق پروتکل مورد بحث به سرور ارسال شود که توسط NodeRed دریافت شده و زوج و فرد بودن آن به اصطلاح مشخص می‌کند که مثلاً فرد مورد نظر ما دیابت دارد یا ندارد. این یک مثال است که مدل ساده‌ای را ساخته‌ایم که با فرض اتصال ابزار‌های سنجش گرفتار خون به صورت آنلاین که مبتنی بر اینترنت اشیاء باشند، می‌توان آخرین اطلاعات فرد بیمار را دریافت کرد.

کد بورد آردوینو:

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

// Network credentials
const char* ssid = "WiFi6";
const char* password = "*****";
const char* mqtt_username = "mqtt";
const char* mqtt_password = "mqtt";

// MQTT broker address
const char* mqtt_server = "192.168.191.3";

// Initializing the WiFi and MQTT clients
WiFiClient DiabetesPredictor20231;
PubSubClient client(DiabetesPredictor20231);
```

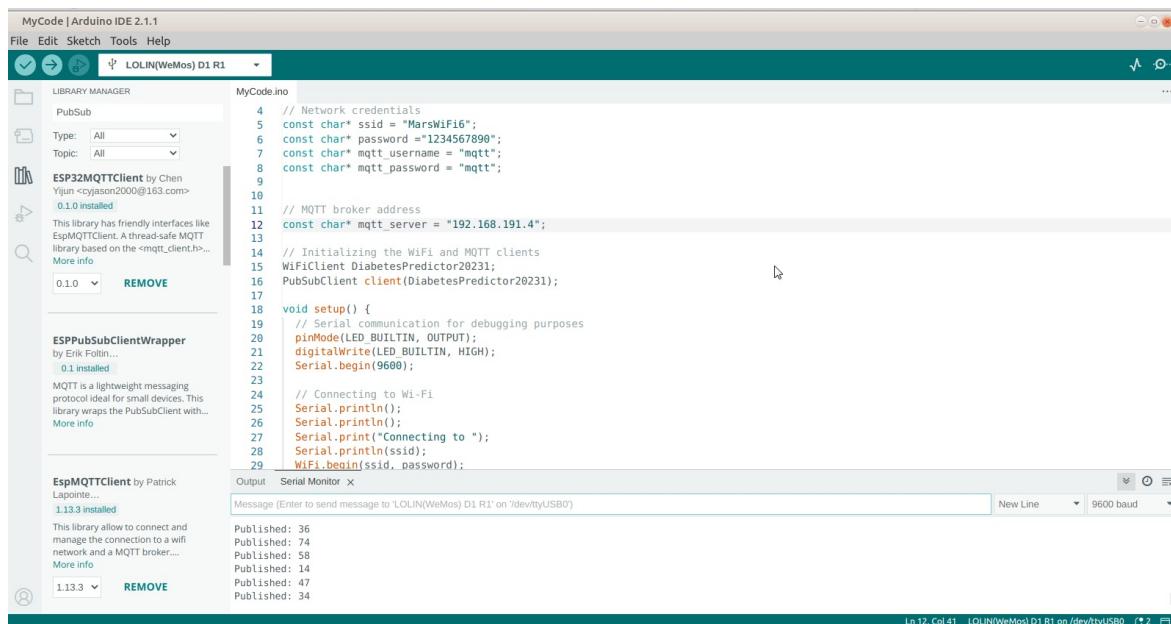
```
void setup() {  
    // Serial communication for debugging purposes  
    pinMode(LED_BUILTIN, OUTPUT);  
    digitalWrite(LED_BUILTIN, HIGH);  
    Serial.begin(9600);  
  
    // Connecting to Wi-Fi  
    Serial.println();  
    Serial.println();  
    Serial.print("Connecting to ");  
    Serial.println(ssid);  
    WiFi.begin(ssid, password);  
    while (WiFi.status() != WL_CONNECTED) {  
        delay(500);  
        Serial.print(".");  
    }  
    Serial.println("");  
    Serial.println("WiFi connected");  
    Serial.println(WiFi.localIP());  
    Serial.println('\n');  
  
    // Connecting to MQTT broker  
    client.setServer(mqtt_server, 1883);  
    while (!client.connected()) {  
        Serial.print("Connecting to MQTT broker...");  
        if (client.connect("DiabetesPredictor2023", mqtt_username, mqtt_password)) {  
            Serial.println("connected");  
            //client.subscribe("GetData2023", 2);  
        } else {  
            Serial.print("failed with state ");  
            Serial.print(client.state());  
            Serial.print("\n");  
            delay(2000);  
        }  
    }  
}  
void loop() {  
    // Generating a two-digit random number and convert it to a string  
    int random_num = random(10, 100);  
    String payload = String(random_num);
```

```

// Publishing the payload to the MQTT topic
client.publish("GetData2023", payload.c_str());
digitalWrite(LED_BUILTIN, LOW);
Serial.print("Published: ");
Serial.println(payload);
delay(400);
digitalWrite(LED_BUILTIN, HIGH);

// Printing the payload and wait for 10 seconds before publishing again
delay(10000);}
```

ادامه توسعه این پروژه در بخش پایانی مطرح می‌شود.



شکل ۱۵-۴: محیط توسعه Arduino در حال دریافت اعداد تولید شده توسط بورد

### ابزار ۳-۲-۴ Node Red

ابزار Node-RED یک چارچوب نرم افزاری مبتنی بر جریان<sup>۳</sup> همه کاره و کاربرپسند<sup>۴</sup> است که توسط IBM توسعه یافته است که توسعه اینترنت اشیا را ساده می‌کند. این چارچوب مبتنی بر Node.js است و امکان ادغام یکپارچه با دستگاه‌های مختلف، API‌ها و انواع خدمات آنلاین را فراهم می‌کند. با رابط بصری جذاب آن، کاربران می‌توانند به سادگی با کشیدن و رها کردن، گره<sup>۵</sup>‌هایی را روی یک صفحه ایجاد و به هم متصل کنند تا رفتار مورد نظر برنامه خود را تعریف کنند. [۲۹] این ابزار با ادغام یکپارچه با پلتفرم‌هایی مانند Raspberry Pi، AWS، Arduino، Node.js و Flow-Based<sup>۶</sup>، این ابزار با ادغام یکپارچه با پلتفرم‌هایی مانند Raspberry Pi، AWS، Arduino، Node.js و Flow-Based<sup>۶</sup>، این ابزار با ادغام گرینه‌ای محبوب برای اینترنت اشیا و برنامه‌های اتوماسیون خانگی تبدیل کرده است. [۲۹]

<sup>3</sup>Flow-Based

<sup>4</sup>User friendly

<sup>5</sup>Node

همچنین نصب داشبورد در این بستر، با وارد کردن دستور `npm install node-red-dashboard` در خط فرمان انجام می‌شود.

در این ابزار، یک داشبورد (شکل ۱۶-۴) طراحی کردیم که دو بخش دارد.

The screenshot shows a Node-RED dashboard titled "Prediction of diabetes". It features two parallel forms, one for "Prediction" and one for "Update". Both forms contain fields for gender, age, race, education, marital status, work status, weight, height, BMI, blood pressure (BPSysAve, BPDiaAve), cholesterol levels (DirectChol, TotChol), physical activity (PhysActive), and a result section. The "Update" form has different values filled in compared to the "Prediction" form. At the bottom of each form are "SEND" and "GET THE LATEST PATIENT STATUS" buttons.

Field	Prediction (Value)	Update (Value)
Gender	male	male
Age	120	32
Race	Mexican	Other
Education	9 - 11th Grade	9 - 11th Grade
MaritalStatus	LivePartner	LivePartner
Work	NotWorking	NotWorking
Weight	2	563
Height	54	45
BMI	9	
BPSysAve	9	
BPDiaAve	9	
DirectChol	9	
TotChol	9	
PhysActive	Yes	Select option
Result	You may develop diabetes!	Diabetes
<b>SEND</b>		<b>GET THE LATEST PATIENT STATUS</b>
		<b>SEND</b>

شکل ۱۶-۴: صفحه داشبورد پیش‌بینی کاربر و به روزرسانی دادگان

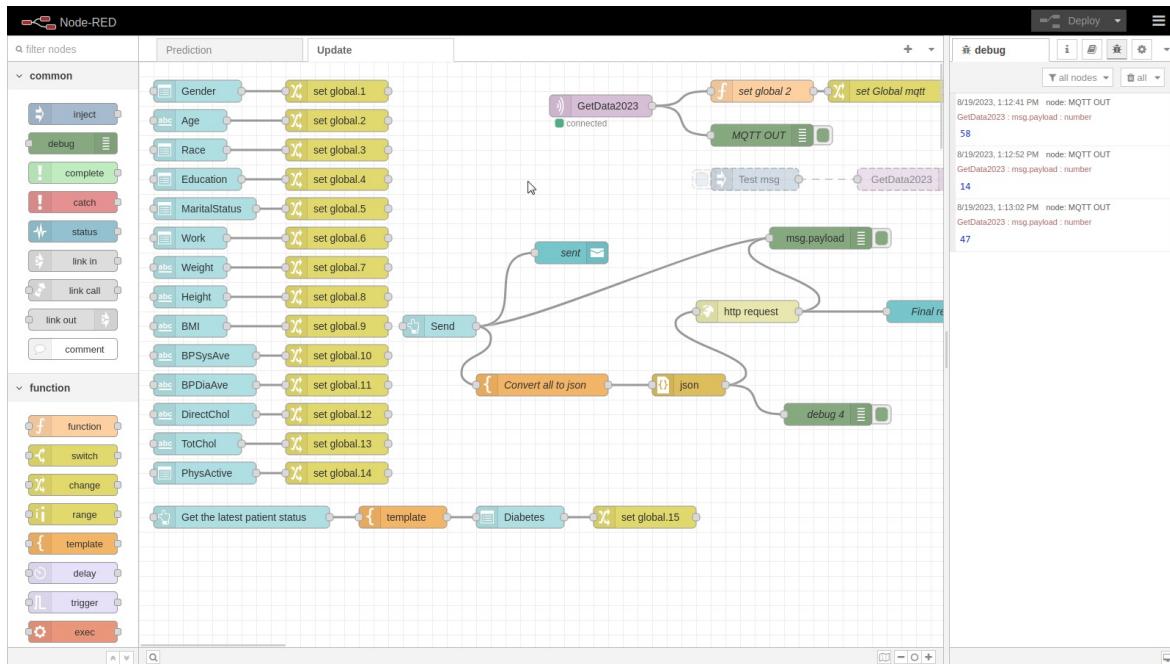
#### • به روزرسانی دادگان:

در این بخش داده‌های جدید از یک سامانه دیگر دریافت شده و از طریق چارچوب Flask در فایل CSV دادگان اضافه می‌شوند. بدیهیست که گزینه GET THE LAST PATIENT STATUS آخرين وضعیت بیمار را درمورد داشتن یا نداشتن دیابت را از طریق اینترنت اشیاء دریافت می‌کند. بدیهیست داشتن دادگان به روز، به صحت مدل و افزایش دقت آن، کمک شایانی می‌کند.

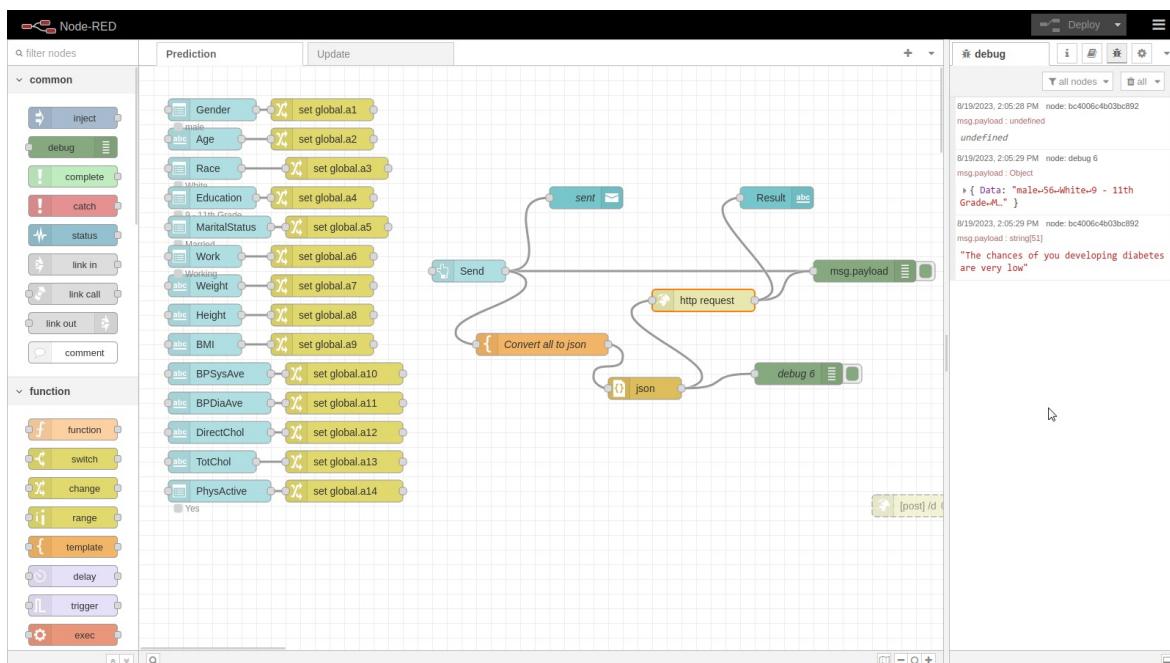
• پیش‌بینی: در این بخش، افراد با وارد کردن ویژگی‌های سلامتی خود، می‌توانند بینند که آیا در آینده ممکن است به دیابت مبتلا شوند یا خیر.

در هر کدام از بخش‌های مطرح شده، گره‌های مربوط به دریافت مشخصات را ایجاد و مقادیر از طریق مرورگر دریافت و در آن ثبت می‌شود. سپس در ساختار داده json ثبت شده و در نهایت از طریق یک http request برای Flask ارسال می‌شود.

بعد از دریافت صحیح اطلاعات، یک اعلان درمورد دریافت صحیح داده‌ها نیز دریافت می‌شود.



شکل ۱۷-۴: طراحی بخش به روزرسانی در NodeRed



شکل ۱۸-۴: طراحی بخش پیش‌بینی در NodeRed

## Flask ۴-۲-۴

می‌توان گفت Flask یک چارچوب توسعه وب سبک و کاربرپسند است که به زبان پایتون نوشته شده است. این سادگی، انعطاف پذیری و ساختار مدولار را برای ساخت برنامه‌های کاربردی مقیاس پذیر ارائه می‌دهد. مستندات گسترده و جامعه پر جنب و جوش این چارچوب، یادگیری و استفاده از

آن را آسان می‌کند. علاوه بر این، Flask به خوبی با کتابخانه‌ها و ابزارهای دیگر ادغام می‌شود و از پایگاه داده‌های مختلف و فناوری‌های وب پشتیبانی و پردازش انواع درخواست‌ها را آسان می‌کند. این یک انتخاب ارجح برای توسعه دهنده‌گان با تمام سطوح تجربه است و به آنها اجازه می‌دهد تا به طور کارآمد برنامه‌های کاربردی وب قوی بسازند. [۳۲]

لذا در این پروژه از این چارچوب برای استفاده از توابع و کتابخانه‌های پایتون جهت پردازش درخواست‌های کاربران استفاده کردیم. وقتی از طریق داشبورد در NodeRed یک درخواست مبتنی بر به روز رسانی داده‌های دادگان یا درخواست از طرف یک فرد درمورد پیش‌بینی احتمال ابتلای آن فرد با مشخصاتش در چارچوب Flask دریافت شود، در تابع مربوطه این درخواست مورد پردازش قرار می‌گیرد و نتیجه به سادگی برای کاربر ارسال می‌شود.

```

MainApp.py - Flask - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER FLASK MainApp.py M x
MainApp.py > Prediction
dataset_new=dataset
df3 = dataset_new.copy()
# These columns must be converted
df3=df3.dropna()

a=inputData

#print(len(df3.index))
df3 = df3.append(pd.DataFrame([a], columns=df3.columns), ignore_index=True)
#print(df3.iloc[-1,:])

df3 = pd.get_dummies(df3,columns = ['Gender','Race1','Education','MaritalStatus','Work','P'])
#print(df3.columns)
x=df3.iloc[-1,:].tolist()
x.pop()
#print(len(df3.index))
dataset_new=df3.copy()
df3=df3.drop(len(df3)-1)
#print(len(df3.index))
#print(x)

# Selecting X & Y
X = dataset_new.iloc[:, :-1] values
Y = dataset_new.iloc[:, -1]

127.0.0.1 - - [19/Aug/2023 13:44:01] "POST /b HTTP/1.1" 500 -
You may develop diabetes!
127.0.0.1 - - [19/Aug/2023 14:04:31] "POST /b HTTP/1.1" 200 -
The chances of you developing diabetes are very low
127.0.0.1 - - [19/Aug/2023 14:05:28] "POST /b HTTP/1.1" 200 -

```

شکل ۴-۱۹: بخشی از کد Flask

### ۳-۴ نتیجه‌گیری

در این بخش، کوشیدیم تا ضمن بررسی روش انتخاب بهترین الگوریتم پیش‌بینی دیابت با پایتون و سنجش میزان صحت و دقت روش‌ها، یک روش پیاده سازی مختصر نیز برای استفاده همگان از این سیستم معرفی شود.

این سنجش دقت از طریق مقایسه ROC، AUC و نمودارهای آن‌ها انجام شد که در فصل قبلی نیز مقالاتی که ارائه شده بودند نیز بر این اساس مقایسه شدند که نتیجه کلی آن روش جنگل درختان تصادفی بود که به عنوان مدل برتر برگزیده شد.

بدترین عملکرد مربوط به الگوریتم‌های بیز ساده و درخت تصمیم بود که مطابق نمودارهای مشاهده شده عملکرد پایین تری نسبت به سایرین داشتند.

و در نهایت به بررسی روش طراحی یک مدل ساده برای سامانه آنلاین پیش‌بینی دیابت با ابزارهای

اینترنت اشیاء و دو چارچوب NodeRed و Flask پرداختیم.

## فصل ۵

### جمع‌بندی و کارهای آتی

#### ۱-۵ جمع‌بندی

بر هیچ کس پوشیده نیست که پیشگیری بهتر از درمان است. لذا با استفاده از راهکارهای علمی، توانستیم مدلی را طراحی کنیم تا با دقت خوبی برای افراد جامعه، بتواند پیش‌بینی کند که آیا در آینده به دیابت مبتلا می‌شوند یا خیر تا در صورت لزوم، اقدامات پیش‌گیرانه را سریع‌تر آغاز کنند.

در نتیجه انجام فرایند‌های بسیار، مطابق نمودارهای ایجاد شده در فصل قبلی، الگوریتم جنگل درختان تصادفی در اعتبارسنجی متقابل با  $K=10$  Fold،  $K=10$  برای ما بهترین نتیجه را با میزان حدودی  $ACC=96\%$  در این پیش‌بینی به ارمغان آورد که نشان می‌دهد دفعات بیشتری که الگوریتم‌های اعتبارسنجی متقابل مدل ما را مورد آزمایش قرار دهند، دقت ارزیابی بالاتری برایمان به دنبال خواهد داشت.

لازم به ذکر است از طریق بررسی عملکرد با شاخصه AUC نمودارهای رگرسیون لجستیک و جنگل درختان تصادفی تقریباً در یک عرض قرار گرفتند.

البته به خوبی مشخص است که هرچه دفعات انجام الگوریتم‌های اعتبارسنجی متقابل بیشتر باشد، زمان و هزینه کار نیز بیشتر خواهد بود.

#### ۲-۵ کارهای آتی

در مورد روش‌های توسعه این سیستم می‌توانیم به مواردی چون سیستم‌های یادگیری ماشین آنلاین اشاره کنیم که دادگان همواره با اطلاعات جدید به روزرسانی می‌شوند و در بازه‌های مختلف توسط ناظر سیستم، بهترین الگوریتم‌ها بر رویشان اعمال می‌گردد تا بهترین نتایج ارائه شوند. ضمن اینکه می‌توانیم از اینترنت اشیا و امکاناتی از این قبیل استفاده کنیم. به تازگی گجت‌ها و کیت‌های مخصوصی برای گوشی‌های هوشمند طراحی شده اند که برای سنجش پارامترهای گوناگون سلامتی مورد استفاده قرار می‌گیرند و با استفاده از گسترش شبکه‌های پرسرعت اینترنت نظری 5G به سرعت می‌توانیم حجم عظیمی از داده‌های جدید را دریافت کنیم.

به عنوان یک نمونه ساده و سمبولیک می‌توان از برد الکترونیکی آردوبینو که قابلیت نصب گجت‌های

مختلف و سنسور های گوناگون را فراهم می‌آورد، استفاده کرد و یک سیستم یادگیری ماشین آنلاین را طراحی کرد که با سرور ما (مثلا برای آردوینو NodeRed می‌باشد) تبادل دارد.

همچنین می‌توانیم سامانه های موازی با این سیستم را نیز راه اندازی کرد مثل یک سایت پیش‌بینی کننده احتمال ابتلا به بیماری دیابت در افراد که هر شخصی با وارد کردن پارامتر های خودش می‌تواند نسبت به وضعیت سلامتی خود در آینده برآورده تقریبی داشته باشد.

## پیوست

### ۱- کد استانداردسازی متن فارسی آمیخته به عبارات انگلیسی

با توجه به اینکه این پایان نامه در LATEX نوشته شده شد، یکی از معضلات مربوط به تایپ فارسی و انگلیسی در این سیستم، این است که حتماً عبارات انگلیسی بایستی درون تگ lr نوشته شوند تا فونت تعیین شده بر روی آن اعمال شود. پس یک برنامه ساده به زبان پایتون نوشتیم تا عبارات انگلیسی درون متن های فارسی که را برایمان در این تگ قرار دهد.

در این کد پایتونی از مبحث RegEx که در درس کامپایلر با آن آشنا شدیم، استفاده کردم. فایل ورودی متن عادی است که پس از انجام عملیات، در فایل خروجی شاهد قرار گرفتن عبارات انگلیسی درون تگ lr خواهیم بود.

```
import re
filename='import.txt'
lines=[]
with open(filename) as file:
    lines = [str(line.rstrip()) for line in file]

def myfun(a):
    e=a.group(0)
    e=' \lr{'+e+'}'
    return e

w=[]
for i in lines:
    x =re.findall(r"[a-zA-Z0-9\s]+\b(?=\\s[^a-zA-Z0-9]*)", i)
    tempi=re.sub(r"[a-zA-Z0-9\s]+\b(?=\\s[^a-zA-Z0-9]*)", myfun, i)
    w.append(tempi)
    print(tempi)

with open(r'export_text.txt', 'w+') as fp:
    for item in w:
        fp.write("%s\n" % item)
print('Done')
```

## واژه‌نامه

ACC(Accuracy)	دقت		الف
Dummy	ساختگی	Specificity	اختصاصی
Sigmoid	سیگماوار	Validation	اعتبارسنجی
		Split	افراز
		Confusion	آشتفتگی
BMI	شاخص توده بدنی		ب
Classification	طبقه بندی	Naive Bayes	بیز ساده
		Overfitting	بیش برآذش
		PIDD	بیماری های زمینه ای
FrameWork	چارچوب	Preprocessing	پیش پردازش
PhysActive	فعالیت فیزیکی		ت
		AdaBoost	تقویت کننده انطباقی
TotChol	کلسترول کل		ج
DirectChol	کلسترول مستقیم	Random forest	جنگل تصادفی
Underfitting	کم برآذش		
Scatter matrix	ماتریس پراکندگی	Sensitivity	حساسیت
FP	مثبت کاذب		د
TP	مثبت واقعی	Dataset	دادگان
IDE	محیط توسعه	NaN	داده نامشخص
Mode	مد (آمار)	Training data	داده های آموزشی
Visualization	تصویرسازی	Testing data	داده های سنجش
Scaling	مقیاس بندی	Missing data	داده های گم شده

FPR	نرخ مثبت کاذب		منحنی مشخصه
TPR	نرخ مثبت واقعی	ROC	عملکرد سیستم
Normalization	نرمال سازی	FN	منفی کاذب
Race	نژاد	TN	منفی واقعی
Heatmap	نقشه حرارتی	Mean	میانگین
Lineplot	نمودار میله ای		میانگین فشار خون
	و	BPDiaAve	دیاستولیک
MaritalStatus	وضعیت تاہل	BPSysAve	میانگین فشار خون
Feature	ویژگی		سیستولیک
			ن
		AUC	ناحیه زیر منحنی

## مراجع

- [۱] اسماعیلی . مهدی، مفاهیم و تکنیک‌های داده کاوی
- [۲] نعمت الهی . نادر، آمار و احتمالات مهندسی

- [3] What is diabetes?, Aoife M Egan, Sean F Dinneen
- [4] Epidemiology of diabetes,Nita Gandhi Forouh,Nicholas J Wareha
- [5] Diabetes Cookbook FOR DUMMIES 3RD EDITION, by Alan L. Rubin, MD with Cait James, MS
- [6] Classification and prediction of diabetes disease using machine learning paradigm,Md.Maniruzzaman, Md. Jahanur Rahman, Benojir Ahammed and Md. Menhazul Abedin
- [7] Prediction of Diabetes using Classification Algorithms, Deepti Sisodia ,Dilip Singh
- [8] Logistic Regression,Lynne Connelly
- [9] Interpretable Machine Learning,Christoph Molnar
- [10] An Overview of Big Data Visualization Techniques in Data Mining ,Samuel Soma Ajibade, Anthonia Adediran
- [11] Data Mining Concepts and Techniques ,Jiawei Han,Micheline Kamber,Jian Pei
- [12] DATA MINING FOR BUSINESS ANALYTICS , GALIT SHMUELI, PETER C. BRUCE,PETER GEDECK,NITIN R. PATEL
- [13] Top 10 algorithms in data mining,Xindong Wu,Jiannong Cao
- [14] Machine Learning for Predictive Analysis, Amit Joshi, Mahdi Khosravy, Neeraj Gupta
- [15] Cross-validation,Daniel Berrar
- [16] The Working Principle Of An Arduino, Yusuf Abdullahi Badamasi
- [17] <https://www.cdc.gov/diabetes/basics/diabetes.html>
- [18] <https://www.kaggle.com/datasets/cdc/national-health-and-nutrition-examination-survey>
- [19] <https://towardsdatascience.com/overfitting-vs-underfitting-a-complete-example-d05dd7e19765>
- [20] <https://towardsdatascience.com/how-to-split-data-into-three-sets-train-validation-and-test-and-why-e50d22d3e54c>

- 
- [21] <https://www.techtarget.com/searchbusinessanalytics/definition/data-visualizatio>
  - [22] [https://scikit-learn.org/stable/modules/cross\\_validation.html#cross-validation](https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation)
  - [23] <https://www.blog.trainindata.com/feature-scaling-in-machine-learning/>
  - [24] <https://www.datacamp.com/tutorial/adaboost-classifier-python>
  - [25] <https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/>
  - [26] <https://machinelearningmastery.com/confusion-matrix-machine-learning/>
  - [27] <https://www.investopedia.com/terms/m/mlr.asp>
  - [28] <https://aws.amazon.com/what-is/mqtt/>
  - [29] <https://nodered.org/docs/>
  - [30] <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all>
  - [31] <https://docs.arduino.cc/>
  - [32] <https://pythonbasics.org/what-is-Flask-python/>