



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.

# 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	۱-۴-۱ انگیزه و اهداف انجام این پژوهش . . . . .
۱۳	۱-۴-۲ موارد انجام شده . . . . .
۱۴	۲ ادبیات پژوهش
۱۵	۲-۱ مقدمه . . . . .
۱۶	۲-۲ روش‌های داده‌کاوی . . . . .
۱۷	۲-۲-۱ طبقه‌بندی و خوشه‌بندی . . . . .
۱۸	۲-۲-۲ بیش برآش و کم برآش در فرایند‌ها . . . . .
۱۹	۲-۳ معرفی الگوریتم‌های طبقه‌بندی . . . . .
۲۰	۲-۳-۱ رگرسیون لجستیک . . . . .
۲۱	۲-۳-۲ درخت تصمیم . . . . .
۲۲	۲-۳-۳ جنگل درختان تصادفی . . . . .
۲۳	۲-۳-۴ AdaBoost . . . . .
۲۴	۲-۳-۵ Naive Bayes . . . . .
۲۵	۲-۴ دادگان، پیش‌پردازش و مصورسازی داده‌ها . . . . .
۲۶	۲-۴-۱ دادگان . . . . .
۲۷	۲-۴-۲ پیش‌پردازش داده‌ها . . . . .
۲۸	۲-۴-۳ مصورسازی . . . . .

۱۹	۴-۴-۲	انواع نمودارها و داده‌های اجمالی
۲۰	۵-۴-۲	اندازه گیری میزان خطای دقت
۲۲	۶-۴-۲	ROC و AUC
۲۳	۵-۲	نتیجه گیری
۲۴	۳	<b>کارهای پیشین</b>
۲۴	۱-۳	مقدمه
۲۴	۲-۳	مقاله ۱
۲۴	۳-۳	مقاله ۲
۲۵	۴-۳	نتیجه گیری
۲۶	۴	<b>روش‌ها و نتایج</b>
۲۶	۱-۴	مقدمه
۲۶	۲-۴	روش پیشنهادی
۲۶	۱-۲-۴	پیاده سازی
۴۰	۲-۲-۴	Arduino
۴۳	۳-۲-۴	ابزار Node Red
۴۴	۴-۲-۴	flask
۴۵	۳-۴	نتیجه گیری
۴۶	۵	<b>جمع‌بندی و کارهای آتی</b>
۴۶	۱-۵	جمع‌بندی
۴۶	۲-۵	کارهای آتی
۴۶	۱-۲-۵	راهاندازی سامانه ثبت گزارشات دیابت و پیش‌بینی
۴۶	۲-۲-۵	تولید کیت‌های ثبت نتایج دیابت جدید
۴۷	پیوست	
۴۷	۱-	کد استاندار دسازی متون فارسی آمیخته به عبارات انگلیسی
۴۸	واژه‌نامه	
۵۰	مراجع	

# فهرست تصاویر

۱۰	۱-۲ مقایسه خوشه بندی و طبقه بندی
۱۱	۲-۲ نمودار پراکنده‌گی
۱۲	۳-۲ نمودار رگرسیون خطی چندگانه در فضا
۱۳	۴-۲ مقایسه رگرسیون لجستیک و خطی
۱۴	۵-۲ نمودار تابع لجستیک
۱۴	۶-۲ مدل Logist Regression
۱۶	۷-۲ الگوریتم AdaBoost: در اینجا مراحل ذکر شده به صورت متناوب تکرار می‌شود.
۱۷	۸-۲ نمونه‌ای از داده‌های گم شده
۱۷	۹-۲ تعداد داده‌های گم شده در این پروژه به ازای هر ستون
۱۸	۰-۲ امتغیرهای دودویی
۱۹	۱-۲ نمونه‌ای از متغیرهای دودویی
۲۰	۲-۲ Pairplot
۲۲	۳-۲ اساختار ماتریس آشفتگی: پارامترهای مذکور در این بخش در این ماتریس قرار گرفته‌اند.
۲۲	۴-۲ اساختار ماتریس آشفتگی برای الگوریتم جنگل درختان تصادفی
۲۳	۵-۲: نمای کلی نمودار ROC
۲۷	۱-۴ جایگزینی مقادیر نامشخص با متوسط‌های آماری
۲۸	۲-۴ نمودار مقایسه ابتلا به دیابت در نژاد‌های مختلف
۲۹	۳-۴ ماتریس Scatter
۲۹	۴-۴ نمودار عوامل موثر در ابتلا دیابت (۱)
۳۰	۵-۴ نمودار عوامل موثر در ابتلا دیابت (۲)
۳۱	۶-۴ نقشه حرارتی
۳۵	۷-۴ نمودار ACC
۳۵	۸-۴ نمودار AUC
۳۶	۹-۴ نمودار ROC برای الگوریتم رگرسیون لجستیک
۳۶	۱۰-۴ نمودار ROC برای الگوریتم AdaBoost
۳۶	۱۱-۴ نمودار ROC برای الگوریتم جنگل درختان تصادفی
۳۷	۱۲-۴ نمودار ROC برای الگوریتم Naive Bayes
۳۷	۱۳-۴ نمودار ROC برای الگوریتم درخت تصمیم
۳۹	۴-۴ اساختار ماتریس آشفتگی برای الگوریتم رگرسیون لجستیک
۴۲	۵-۴ محیط توسعه Arduino در حال دریافت اعداد تولید شده توسط بورد

۴۳	۶-۴ اطراحی بخش به روزرسانی در NodeRed
۴۳	۷-۴ اطراحی بخش پیش بینی در NodeRed
۴۴	۸-۴ اصفحه داشبورد پیش بینی کاربر و به روزرسانی دادگان
۴۴	۹-۴ بخشی از کد flask

## 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 اندازه گیری 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...

...

---

<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>[۱۴]</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>[۱۴]</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. [۱۴] 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 ... [۳] [۱۴] [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

### ۲-۳-۱ 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.

### 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 \*\*\*فلان\*\*\* 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. [۱۳]

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 واضح‌تر 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

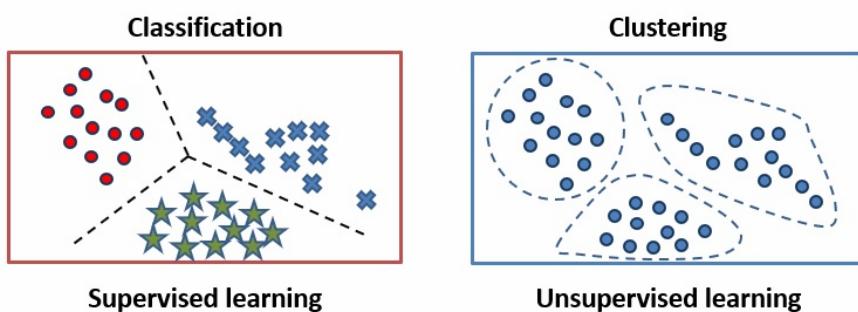
بازپرداخت کند، دیر بازپرداخت کند یا اعلام ورشکستگی کند. تراکنش کارت اعتباری می‌تواند عادی یا تقلیلی باشد. بسته‌ای از داده‌هایی که در یک شبکه حرکت می‌کنند می‌تواند عادی یا تهدیدکننده باشد. یک اتوبوس در سیستم حمل و نقل می‌تواند در دسترس باشد یا در دسترس نباشد. قربانی یک بیماری می‌تواند بهبود یابد، یا خیر. [۱۰]

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. [۱۰]

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. [۱۰]

- **خوشه‌بندی** تجزیه و تحلیل خوشه‌ای، برای تشکیل گروه‌ها یا خوشه‌هایی از رکوردهای مشابه بر اساس چندین اندازه‌گیری انجام شده بر روی این رکوردها استفاده 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 کرده‌اند. [۱۰]

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 می‌توانیم خوشه‌بندی را اعمال کنیم. (مثلاً یک فروشگاه اینترنتی که به خریداران مختلف، پیشنهاد‌های گوناگونی در ازای خرید‌های قبلی آن‌ها می‌دهد.)



شکل ۱-۲: مقایسه خوشه‌بندی و طبقه‌بندی

## ۲-۲-۲ بیش برآذش و کم برآذش در فرایندها

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 اضافه ای برای تشخیص لیوان بودن یا نبودن اشیاء منظور کرده است؛ و در این حالت می‌گوییم مدل ما دچار بیش برآذش شده است.

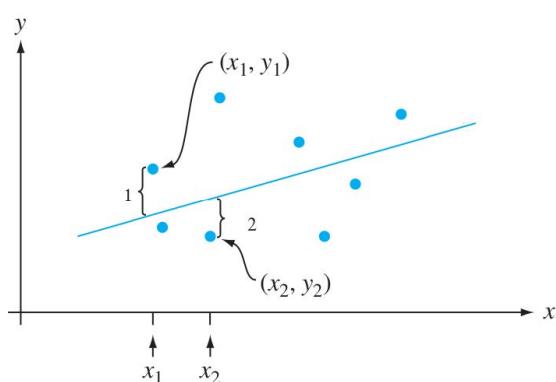
بر عکس در حالتی که فرض کنیم مدل ما اهمیتی برای داشتن ارتفاع نسبت به سطح را در نظر نگیرد، ممکن است یک بشقاب را به عنوان یک لیوان در نظر بگیرد و در این حالت می‌گوییم مدل ما دچار کم برآذش شده است.

## ۳-۲ معرفی الگوریتم‌های طبقه‌بندی

### ۱-۳-۲ رگرسیون لجستیک<sup>۱</sup>

خوب است در ابتدا مروری بر رگرسیون خطی<sup>۲</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 مقابل 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<sup>۲</sup> گوییم.



شکل ۲-۲: نمودار پراکندگی

حالا می‌توان خطی را در این صفحه مختصات درنظر گرفت که تا حد زیادی منطبق بر نقاط باشد که در واقع یک نمودار پیش‌بینی کننده Y بر مبنای X است که به آن معادله رگرسیون Y بر روی X

<sup>۱</sup>Logistic Regression

<sup>۲</sup>Linear Regression

گویند. [۲] 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. [۲] 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 خطا (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 خطا باشند برای ما کمک کننده است. [۲] بر اساس همین روش، با معادلات زیر به مقادیر بهینه  $\alpha$  و  $\beta$  دست می‌یابیم. [۲]

$$\alpha = \bar{y} - \beta \bar{x} \quad (۱-۲)$$

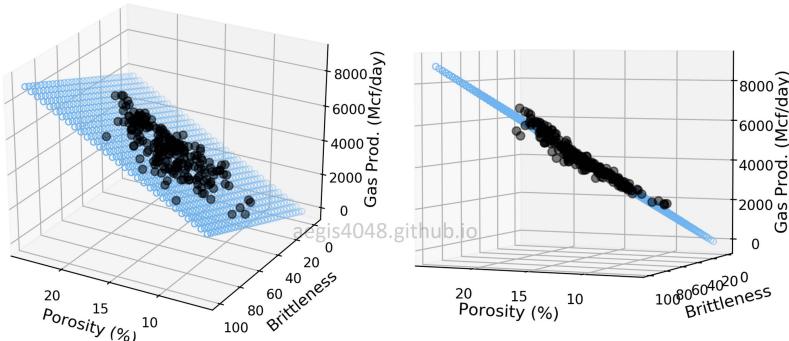
$$\beta = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2} \quad (۲-۲)$$

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 درنظر 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.

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} + \epsilon \quad i = 1, \dots, n \quad (۳-۲)$$

که در این رابطه،  $p$  ابعاد ما می‌باشد. [۲۰]

3D multiple linear regression model



شکل ۲-۳: 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.

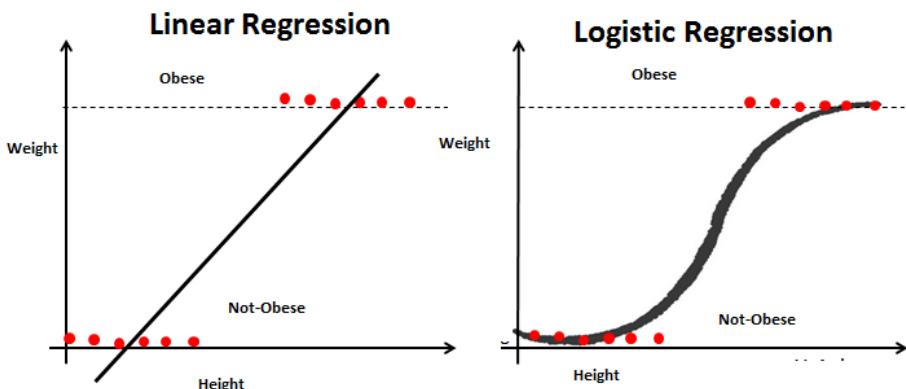
$$p(X, Y) = corr(X, Y) = \frac{Cov(x, y)}{(Var(x)Var(y))^{\frac{1}{2}}} \quad (۴-۲)$$

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.

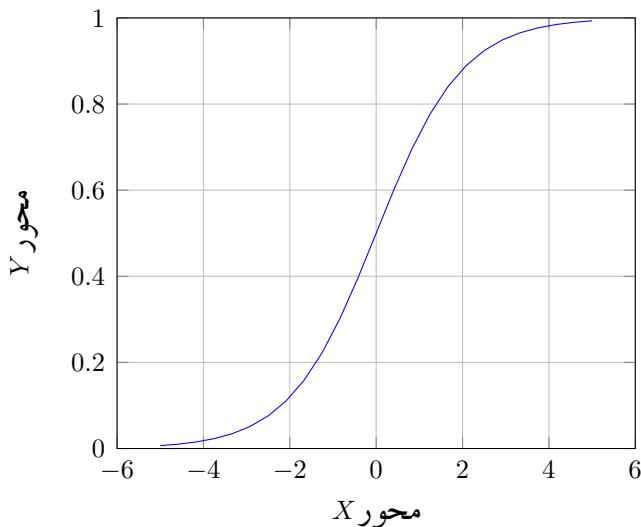
در این مدل به جای اینکه مقدار عددی برای متغیر وابسته تعریف شود، بر اساس احتمال متغیرهای دودویی را برای پیشگویی در نظر داریم [۱] که در ادامه این مسئله را بیشتر توضیح می‌دهیم. مثلاً کار ما، فاکتور BMI در افراد یک متغیر مستقل است و بر ابتلا به دیابت در افراد موثر بوده؛ همچنین در اینجا متغیر وابسته ما، همان 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 متغیرهای مانداریم. [۸][۹] ضمناً در حالت کلاس بندی، نمی‌توانیم حالت ۰ یا ۱ را به عنوان اعدادی در حالت رگرسیون خطی منظور کنیم؛ چرا که ۰ و ۱ را به عنوان مقادیر عدد در نظر گرفته می‌شود و امکان نمایش کلاس‌های گوناگون در در یک نمودار خطی مشخص وجود ندارد. [۹] اگر به شکل ۴-۲ دقت کنید، متوجه می‌شوید در حالت رگرسیون خطی برخی از نمونه‌ها در کلاس مربوطه قرار نگرفته‌اند.



شکل ۴-۲: 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 ۰ تا ۱ استفاده کنیم [۹]: رابطه ۵-۲ و شکل ۵-۲

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



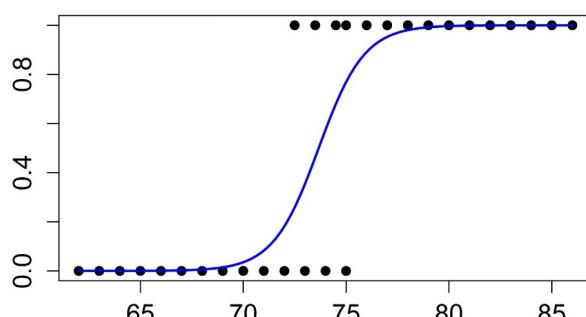
شکل ۲-۵: نمودار تابع لجستیک

سپس اگر سمت راست رابطه ۳-۲ را در جای  $X$  در معادله ۵-۲ (که آن را تابع سیگماوار<sup>۳</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]:

$$g(x) = \ln\left(\frac{p(x)}{1 - P(x)}\right) = \frac{\frac{e^{b_0 + b_1 x}}{1 + e^{b_0 + b_1 x}}}{1 - \frac{e^{b_0 + b_1 x}}{1 + e^{b_0 + b_1 x}}} = \ln(e^{b_0 + b_1 x}) = b_0 + b_1 x \quad (6-2)$$

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 قرار دارد.) [۸][۹] به بیان دیگر می‌توانیم بگوییم، پاسخ ۷، ترکیب خطی پیش‌بینی‌کننده‌ها (یا همان  $X$ ) است. [۶]



شکل ۶-۲: مدل Logist Regression

<sup>۳</sup>Sigmoid/ Logistic Function

### 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.

ادامه دارد ...

### AdaBoost ۴-۳-۲

این الگوریتم یک فرض ساده در نظر دارد و آن یادگیری گروهی است.<sup>[۱۲]</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>[۱۰]</sup>

این روند جداسازی داده‌های آموزشی و امتیاز دهی در تشخیص توسط طبقه بندی کننده‌های مختلف درون درختان تصمیم ضعیف، مکررا تکرار می‌شود تا درنهایت طبقه بندی‌های متعددی داشته باشیم که هر کدام بر اساس میزان سرآمدی در تست‌های مختلف، امتیازات مختلفی دریافت کنند.<sup>[۱۲]</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>[۱۰]</sup><sup>[۱۲]</sup>

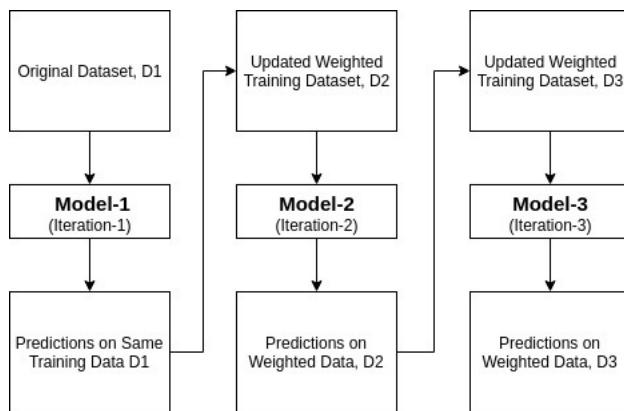
اگر از دید ریاضی الگوریتم را بررسی کنیم، برای محاسبه میزان خطای مدل،  $M_i$  وزن هر یک از تاپل‌های  $D_i$  را که  $M_i$  اشتباه طبقه بندی کرد، جمع می‌کنیم.

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

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>[۱۰]</sup> سپس یک مجموعه داده جدید به نام  $D_i$  تولید می‌کنیم و از آن یک

جدید استخراج می‌کنیم و دوباره این روند را ادامه می‌دهیم. هر چه میزان خطای طبقه‌بندی کننده کمتر باشد، دقیق‌تر است و بنابراین، وزن آن برای انتخاب شدن باید بیشتر باشد. از رابطه ۲-۸ برای محاسبه وزن هر طبقه‌بندی استفاده می‌شود. [۱۰]

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



شکل ۷-۲: الگوریتم AdaBoost: در اینجا مراحل ذکر شده به صورت متناوب تکرار می‌شود. [۱۷]

## Naive Bayes ۵-۳-۲

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

#### ۱-۴-۲ دادگان

دادگانی<sup>۴</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>۵</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 تغذیه برای رفع نیازهای نوظهور دارد. [۱۵]

صاحب‌جهه NHANES شامل سوالات جمعیت شناختی، اجتماعی-اقتصادی، رژیم غذایی و سلامتی است. جزء معاینه شامل اندازه گیری‌های پزشکی، دندانی و فیزیولوژیکی و همچنین تست‌های آزمایشگاهی است که توسط پرسنل پزشکی بسیار آموزش دیده انجام می‌شود. [۱۵]

#### ۲-۴-۲ پیش‌پردازش داده‌ها

##### پیش‌پردازش داده‌ها چیست؟

<sup>۴</sup>Dataset

<sup>۵</sup>US Centers for Disease Control and Prevention

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 گم شده چیست؟

وقتی دادگانمان را مورد بررسی قرار می‌دهیم، در برخی از سلول‌ها، جای برخی مقادیر خالی هستند (در پروسه جمع آوری داده‌ها این اطلاعات به هر دلیلی ثبت نشده‌اند) یا در اثر عملیات‌های مربوط به شناسایی داده‌های پرت، برخی از داده‌ها را حذف می‌کنیم و جای آن‌ها خالی می‌ماند. در این موقع باید با استفاده از تدابیری، داده‌های گم شده را با مقادیری جایگزین کنیم یا کلا آن رکوردها را حذف کنیم تا مدل‌های دقیق‌تری داشته باشیم. (شکل ۸-۲)

	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 ۸-۲: نمونه‌ای از داده‌های 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
<b>dtype:</b>	<b>int64</b>

شکل ۹-۲: تعداد داده‌های گم شده در این پروژه به ازای هر ستون

### راه حل‌های داده‌های گم شده

#### • حذف ردیف‌های حامل داده‌های گم شده

یکی از راهکارهایی که در هنگام کار با داده‌های گم شده انجام می‌شود، حذف کل رکوردهایی است که دارای این مقادیر هستند. این مسئله یک بدی دارد و بدی آن این است که داده‌های کم تری برای آموزش مدل‌مان در اختیار خواهیم داشت و مدل ما ضعیف‌تر خواهد بود. اما اگر به هر دلیلی نتوانیم این مقادیر را با مقادیری دارای تقریب خوب پر کنیم، چاره دیگری نداریم.

## • جایگزینی با متoste‌های آماری

یکی از راهکارهای دیگر برای مدیریت داده‌های گم شده، جایگزین کردن مقادیر گم شده با جایگزینی با متoste‌های آماری است. برخی از ستون‌ها را که زیاد اهمیت نداشته باشند می‌توان با مقادیر میانگین پر کنیم. مثلاً اگر یک دادگان داشته باشیم که حاوی اطلاعات مشتریان یک بانک باشد که بخواهیم از آن برای وام دادن به آن‌ها استفاده کنیم، میتوانیم در ستونی که مربوط به میزان حقوق ماهیانه هر فرد می‌باشد، درصورت مشاهده مقادیر NaN، آن‌ها را با میانگین حقوق مشتریان جایگزین کرد. این راهکار، صرفاً به ما امکان می‌دهد تا تحلیل را ادامه دهیم و اطلاعات موجود در این رکورد را برای متغیرهای دیگر از دست ندهیم. [۱۱]

### افراز داده‌ها

بخش کردن داده‌ها یا افراز<sup>۶</sup>

### 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 ۲-۱۰) و مقادیر ستون‌های جدید را با بله/خیر (۰ و ۱) پر می‌کنیم. [۱۱] به عنوان مثال در ستون مربوط به وضعیت تا هل فرد، وضعیت های گوناگون را به ستون‌های معجزا تقسیم کردیم و برای هر کدام مقادیری در نظر گرفته شد. (شکل ۲-۱۱)

```

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 ۲-۱۰: متغیرهای دودویی

<sup>۶</sup>Data Splitting

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 ۱۱-۲: نمونه‌ای از متغیرهای دودویی

### مقیاس‌بندی

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.<sup>[۱۱]</sup>

متغیرهایی که در مقیاس‌های مختلف اندازه گیری می‌شوند به طور یکسان در برازش مدل و تابع آموخته شده مدل نقش ندارند و ممکن است منجر به ایجاد یک سوگیری شوند.

<https://towardsdatascience.com/how-and-why-to-standardize-your-data-996926c2c832>

**Cross validation** <https://www.geeksforgeeks.org/normalization-vs-standardization/>

### ۳-۴-۲ مصورسازی

**تعریف:** به طور ساده می‌توانیم بگوییم زمانی که داده‌هایمان را به صورت انواع نمودارها، نقشه‌ها و شکل‌های مختلف بصری دریاباوریم تا نتیجه گیری و تحلیل آن‌ها توسط مغز جهت شناسایی الگوها و نقاط پرت در داده آسان‌تر شود، این کار انجام می‌گیرد.<sup>[۱۶]</sup>

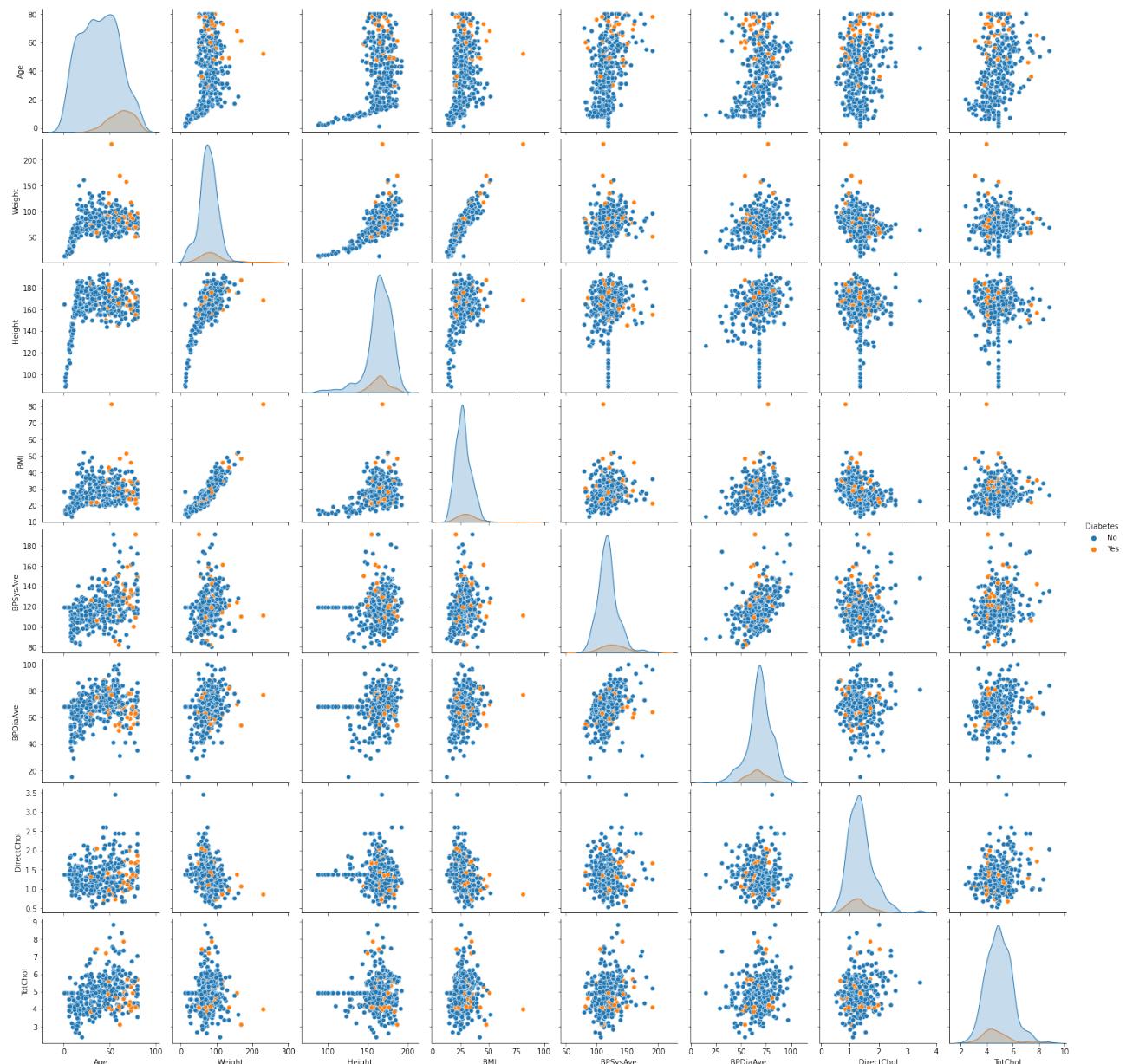
**اهمیت:** یک تصویر هزاران برابر بیشتر از کلمات ارزش دارد.

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.<sup>[۱۶]</sup>

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

#### Pairplot



شکل ۲-۱۲

در شکل ۱۲-۲ نمودار Pairplot را مشاهده می‌کنیم.

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.

ادامه دارد....

## ۵-۴-۲ اندازه گیری میزان خطای دقت

به دلیل پدید آمدن مشکلاتی از قبیل بیش برآریش و کم برآریش داده‌ها که در بخش‌های قبلی به آن اشاره شد مدل‌های ما دچار اشتباهاتی در پیش‌بینی‌ها می‌شود. در این بخش به بیان انواع خطاهای و روابطی

که برای سنجش آنها به کار گرفته ایم می‌پردازیم. [۱۰]  
ابتدا به معروفی <sup>۴</sup> تاپل مختلف می‌پردازیم که بعدا از آنها در محاسبه میزان خطا استفاده خواهیم کرد : [۱۰]

- **TP<sup>۷</sup>**: این تاپل موارد مثبت واقعی را علامت گذاری می‌کند. مثلا در این پروژه مواردی که احتمال ابتلا به دیابت در آنها مثبت پیش‌بینی شده و در دادگان هم مثبت بوده در این دسته قرار می‌گیرد.
- **TN<sup>۸</sup>**: این تاپل موارد منفی واقعی را علامت گذاری می‌کند. مثلا 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>۹</sup>**: این تاپل موارد مثبت کاذب را علامت گذاری می‌کند. مثلا 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>۱۰</sup>**: این تاپل موارد منفی کاذب را علامت گذاری می‌کند. مثلا 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.

• **FPR** یا (1 - Specificity) رابطه ۹-۲ نشان دهنده نسبت افراد سالم شناسایی شده واقعی به کل افراد سالم (افراد اشتباهها بیمار تشخیص داده شده به علاوه افراد سالم شناسایی شده واقعی) است.

$$1 - Specificity = FPR = \frac{FP}{FP + TN} \quad (9-2)$$

• **Sensitivity** یا TPR :

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

$$Sensitivity(Recall) = TPR = \frac{TP}{TP + FN} \quad (10-2)$$

تعدادی از این مقادیر را در ماتریس آشفتگی <sup>۱۱</sup> نشان می‌دهند [۱۹] و ساختار آن در شکل ۱۳-۲ است. همچنین این ماتریس را برای مدل جنگل درختان تصادفی رسم کردیم (شکل ۱۴-۲) :

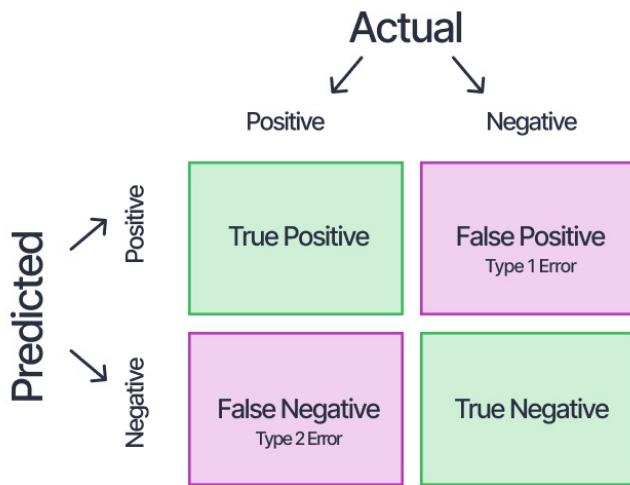
<sup>۷</sup>True positives

<sup>۸</sup>True negatives

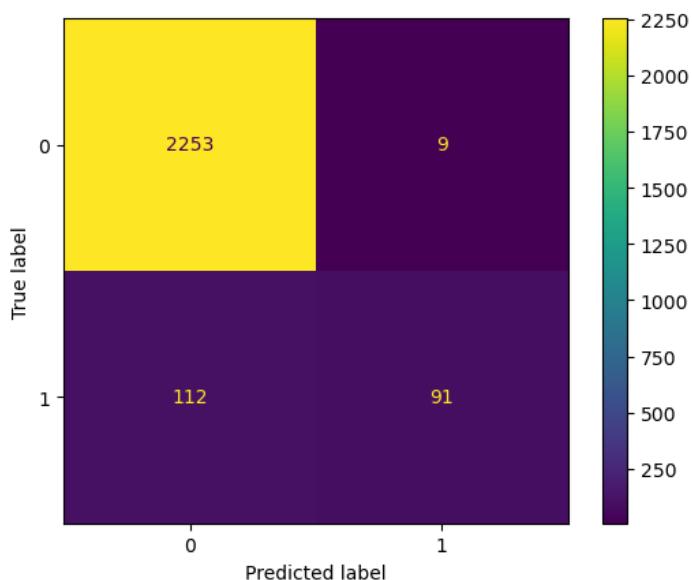
<sup>۹</sup>False positives

<sup>۱۰</sup>False negatives

<sup>۱۱</sup>Confusion



شکل ۲-۱۳: ساختار ماتریس آشفتگی: پارامترهای مذکور در این بخش در این ماتریس قرار گرفته اند.



شکل ۲-۱۴: ساختار ماتریس آشفتگی برای الگوریتم جنگل درختان تصادفی

به طور کلی این ماتریس به ما خلاصه‌ای از درستی نتایج پیش‌بینی هایمان را نشان می‌دهد. [۱۹] حال میزان میزان دقت را با رابطه ۲-۱۱ محاسبه می‌کنیم.

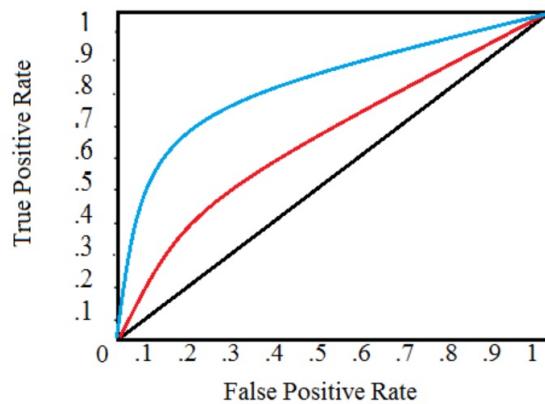
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (11-2)$$

## 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. [۱۰] ما در اینجا مقادیر مختلف

برای هر مرز (اشاره بهتابع Sigmid که در بخش‌های قبلی مطرح شد) پیش‌بینی را از ۰ تا ۱ اندازه‌گیری و در نمودار علامت می‌زنیم.<sup>۱۰</sup> 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.<sup>۱۰</sup> مثلاً در یک مرز در نظر می‌گیریم اگر مقدارتابع Sigmid از ۷٪ بیشتر تر بود، آن مورد را یک مورد مثبت شناسایی کن. در نمودار ROC<sup>۱۲</sup> مولفه‌ی طول‌ها برابر با مقدار FPR است و مولفه‌ی عرض‌ها برابر TPR در آستانه‌های گوناگون است.<sup>۱۰</sup>

سطح زیر این نمودار را با AUC<sup>۱۳</sup> نمایش می‌دهیم که بالا بودن این مساحت بیانگر این است که پیش‌بینی‌های بیشتری در این مدل درست بوده است.<sup>۱۰</sup>



شکل ۲-۱۵: نمای کلی نمودار ROC

## ۵-۲ نتیجه‌گیری

<sup>۱۰</sup>Receiving Operating Characteristic

<sup>۱۳</sup>Area Under the Curve

## فصل ۳

### کارهای پیشین

#### ۱-۳ مقدمه

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 رهنمودهای مفید برای کارهای آتی طلقی شوند.

#### 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. [۱۸]

#### 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 سنجدی 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 گرفتند که نتایج آن‌ها داری دقت مناسبی نبوده است.

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

<sup>۱</sup>PIDD

### ۴-۳ نتیجه‌گیری

در کل با مطالعه مقالات اساسی که در گذشته تالیف شده بودند، سعی شد روش‌های مناسب به کار گرفته شوند و در روند توسعه استفاده 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. [۱۰]

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.

#### ۲-۴ روش پیشنهادی

##### ۱-۲-۴ پیاده سازی

پس از Import 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 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=[

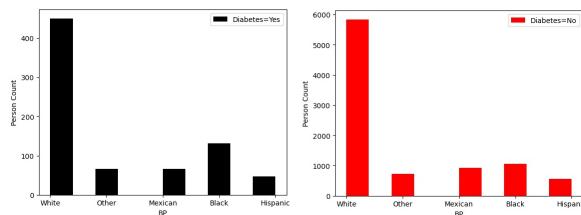
'Red'], label=['Diabetes>No'])

plt.xlabel('BP')

```

```
plt.ylabel('Person Count')
plt.legend()
plt.show()
```

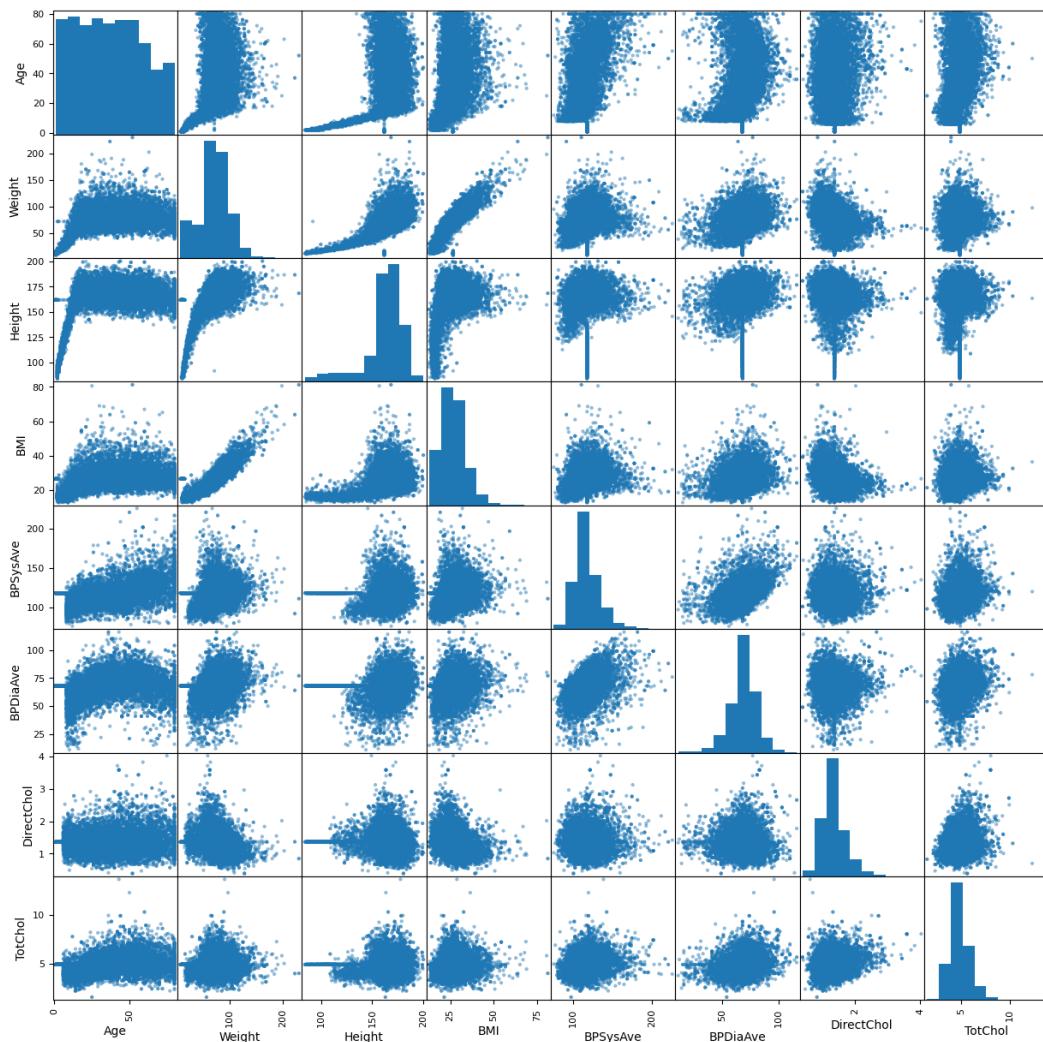
نتیجهٔ عملیات فوق، نمودار‌های زیر می‌باشد (در این نمودار، تاثیر نژاد در ابتلا به دیابت مشخص است):



شکل ۴-۲: نمودار مقایسهٔ ابتلا به دیابت در نژاد‌های مختلف

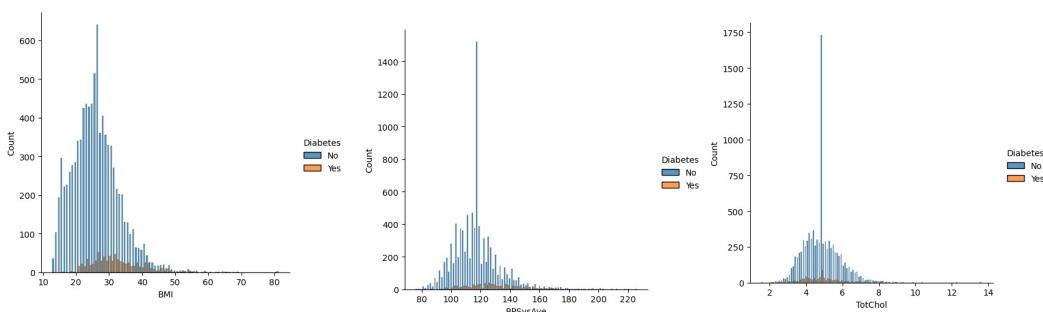
سپس با کد زیر، ماتریس scatter برای نمایش همبستگی میان داده‌ها رسم می‌کنیم:

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

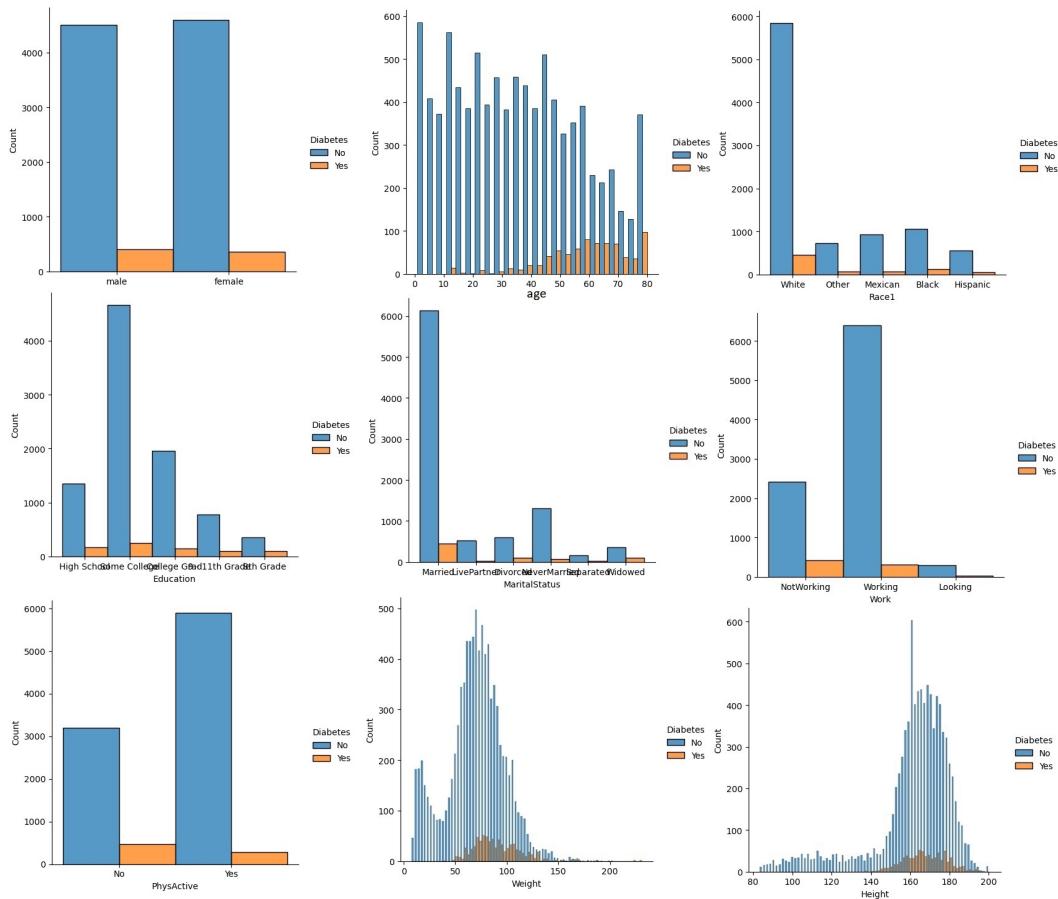


شکل ۳-۴: ماتریس Scatter

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 ۴-۶: نقشه حرارتی

حال پس از بخش مصور سازی به سراغ آماده سازی داده ها برای ایجاد مدل ها پرداختیم. پس متغیر های 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 اعتبارسنجی 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 [۱۱] 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  $20$  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 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 Fold‌های ۵، ۲ و ۱۰ می‌کنیم:

```

```

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

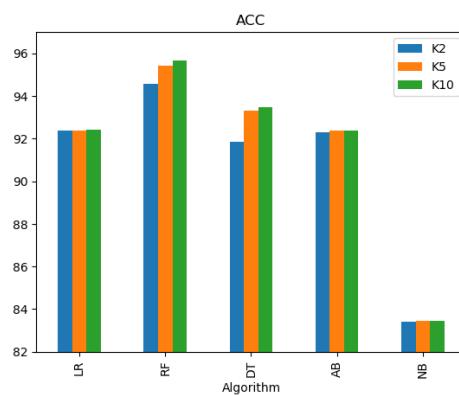
```

```

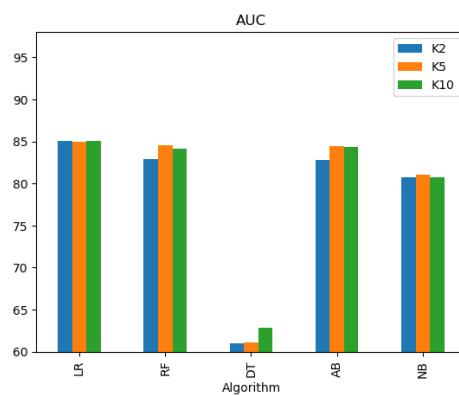
barauc['K5'] = barauc['K5'].astype('float64')
barauc['K10'] = barauc['K10'].astype('float64')
# plot grouped bar chart
barauc.plot(x='Algorithm',
            kind='bar',
            stacked=False,
            ylim=[60, 98],
            title='AUC')

```

نمودار ها:

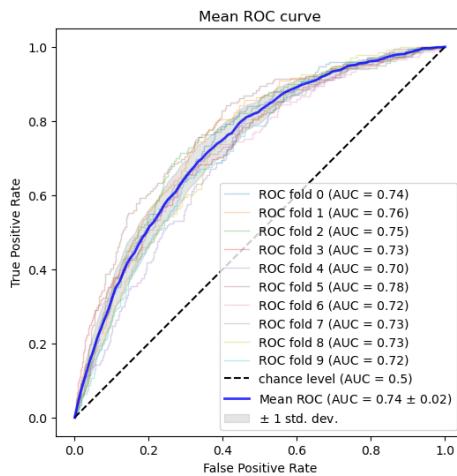


شکل ۷-۴: نمودار ACC

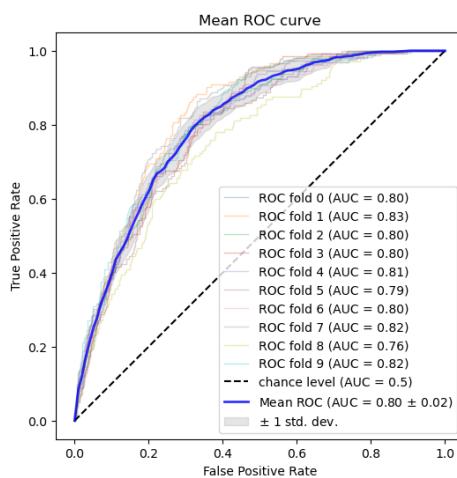


شکل ۸-۴: نمودار AUC

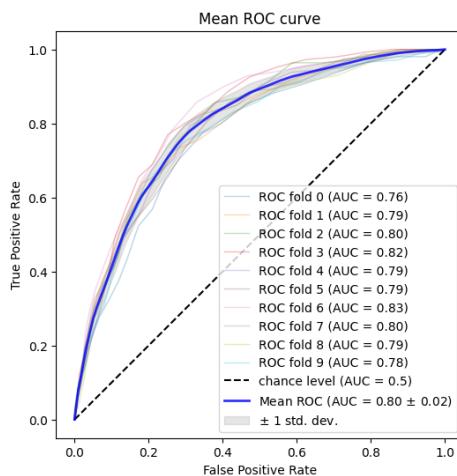
حال برای الگوریتم های مختلف، نمودار ROC را در Fold های مختلف رسم می کنیم و میانگین آن نیز مشاهده می شود:



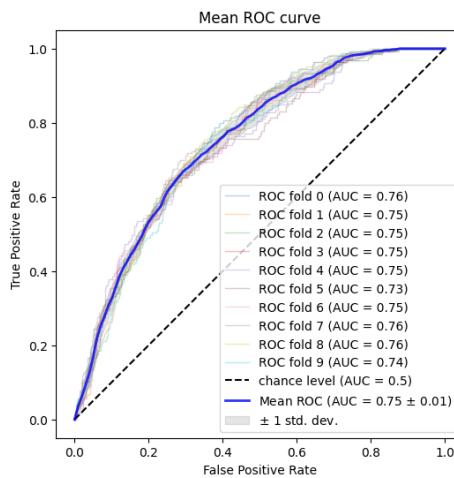
شکل ۹-۴: نمودار ROC برای الگوریتم رگرسیون لجستیک



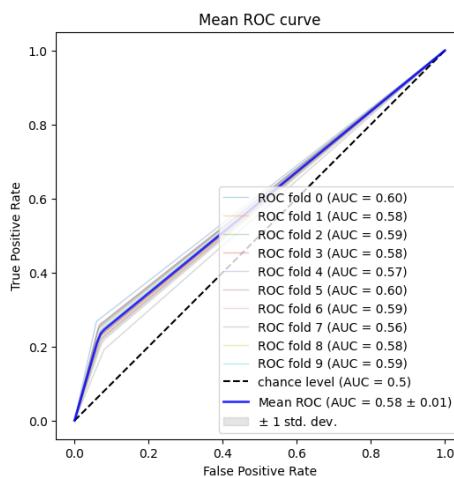
شکل ۱۰-۴: نمودار ROC برای الگوریتم AdaBoost



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 نمودار های 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)

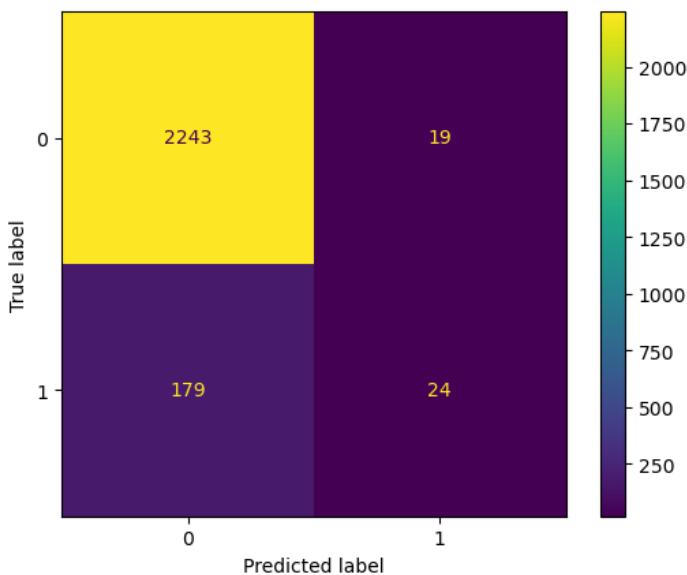
```

و در نهایت ماتریس آشتفتگی را با استفاده از کد زیر برای الگوریتم های 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 ۴-۱۴: ساختار ماتریس آشتفتگی برای الگوریتم رگرسیون لجستیک

**Arduino ۲-۲-۴**

```
espmqttclient
http://arduino.esp8266.com/stable/package_esp8266com_index.json
file>performances> additional board manager url
npm install node-red-dashboard
mosquitto -c
/etc/mosquitto
c:\mosquitto\
sc stop mosquitto
sc start mosquitto
allow_anonymous true
listener 1883 0.0.0.0
```

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);
}

```

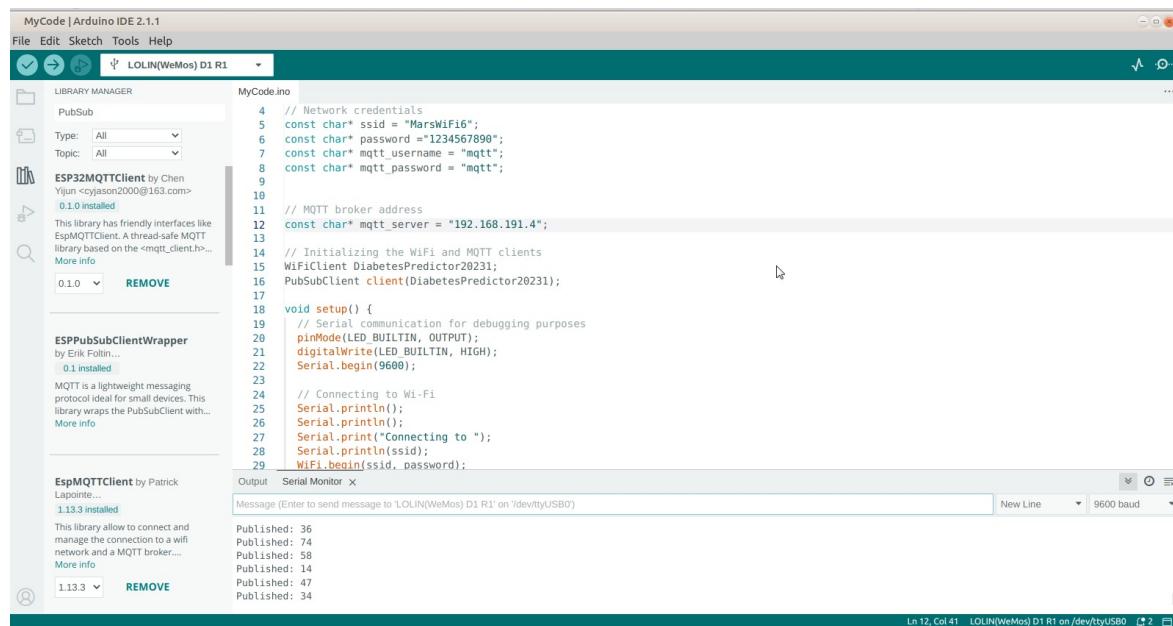
```

delay(400);

digitalWrite(LED_BUILTIN, HIGH);

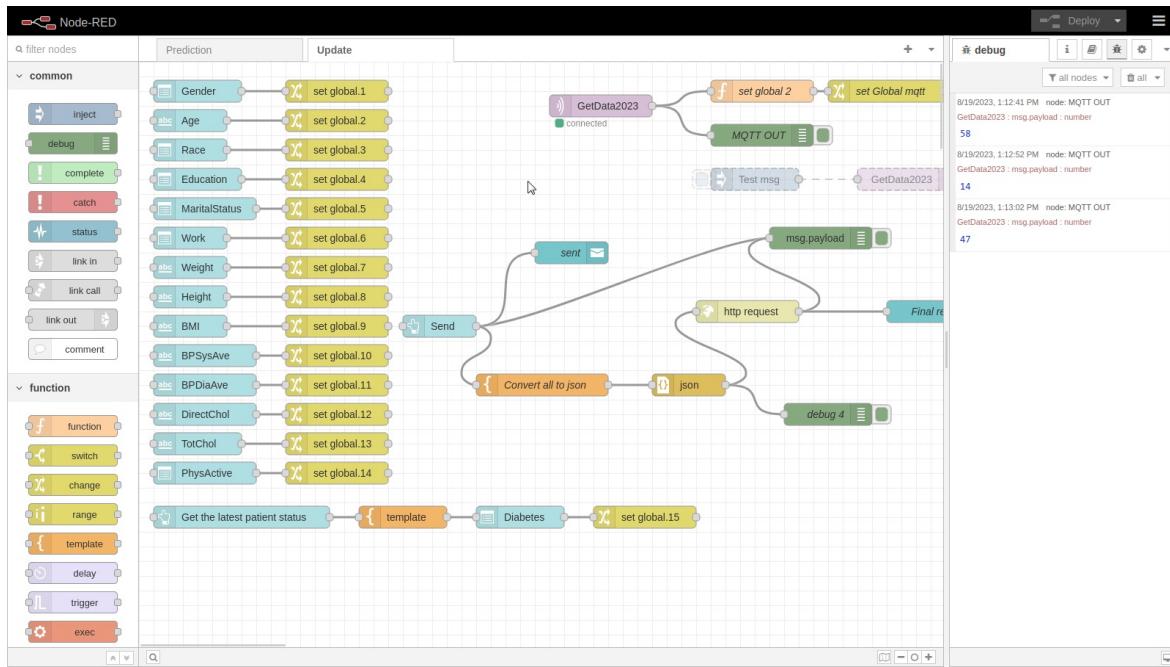
// Printing the payload and wait for 10 seconds before publishing again
delay(10000);}

```

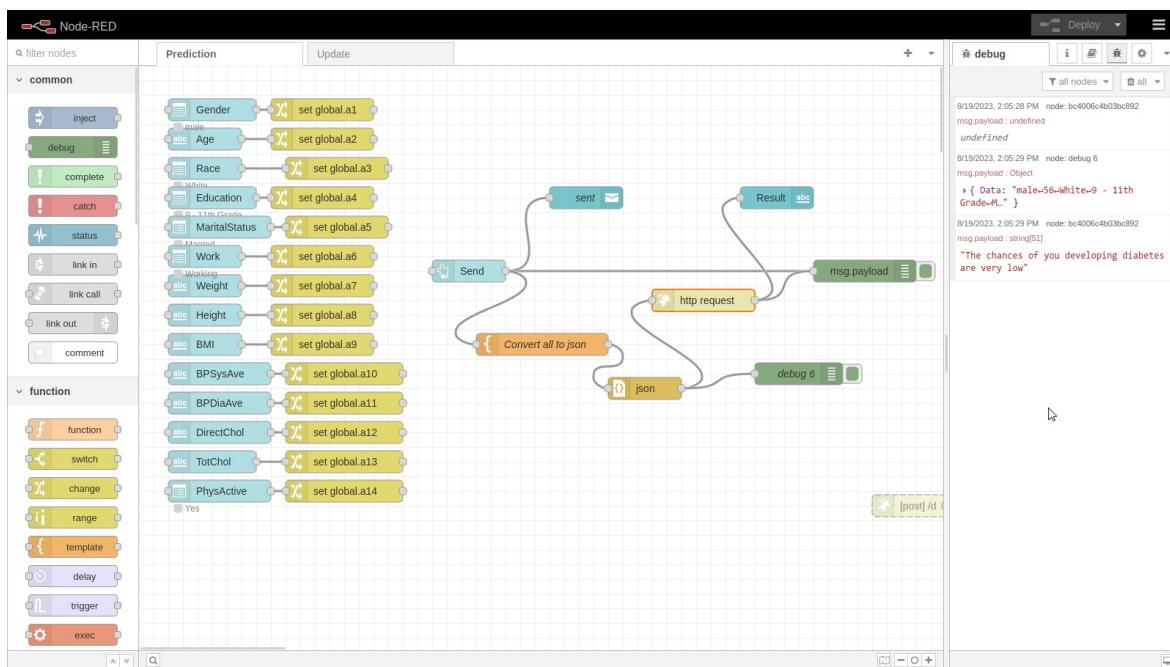


شکل ۱۵-۴: محیط توسعه Arduino در حال دریافت اعداد تولید شده توسط بورد

## Node Red ۳-۲-۴ ابزار



شکل ۱۶-۴: طراحی بخش به روزرسانی در NodeRed



شکل ۱۷-۴: طراحی بخش پیش‌بینی در NodeRed

Prediction of diabetes

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

شکل ۱۸-۴: صفحه داشبورد پیش‌بینی کاربر و به روزرسانی دادگان

flask ۴-۲-۴

MainApp.py - Flask - Visual Studio Code

```

MainApp.py - Flask - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER FLASK
MainApp.py M ...
MainApp.py > Prediction
111 dataset_new=dataset
112 df3 = dataset_new.copy()
113 # These columns must be converted
114 df3=df3.dropna()
115
116 a=inputData
117
118 #print(len(df3.index))
119 df3 = df3.append(pd.DataFrame([a], columns=df3.columns), ignore_index=True)
120 #print(df3.iloc[-1,:])
121
122 df3 = pd.get_dummies(df3,columns = ['Gender','Race1','Education','MaritalStatus','Work','P'
123 #print(df3.columns)
124 x=df3.iloc[-1,:].tolist()
125 x.pop()
126 #print(len(df3.index))
127 dataset_new=df3.copy()
128 df3=df3.drop(len(df3)-1)
129 #print(len(df3.index))
130 #print(x)
131
132 # Selecting X & Y
133 Y = dataset_new.iloc[-1,:-11].values
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
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 -
Ln 120, Col 31 Spaces:4 UTF-8 CRLF Python 3.8.10 64-bit Go Live Prettier

```

شکل ۱۹-۴: بخشی از کد flask

٣-٤ نتیجه‌گیری

## فصل ۵

### جمع‌بندی و کارهای آتی

#### ۱-۵ جمع‌بندی

#### 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 نظیر 5G به سرعت می‌توانیم حجم عظیمی از داده‌های جدید را دریافت کنیم.

به عنوان یک نمونه ساده و سمبولیک می‌توان از برد الکترونیکی آردوینو که قابلیت نصب گجت‌های مختلف و سنسور‌های گوناگون را فراهم می‌آورد، استفاده کرد و یک سیستم یادگیری ماشین آنلاین را طراحی کرد که با سرور ما (مثلاً برای آردوینو NodeRed می‌باشد) تبادل دارد.

همچنین می‌توانیم سامانه‌های 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 پارامترهای خودش می‌تواند نسبت به وضعیت سلامتی خود در آینده برآورده تقریبی داشته باشد.

#### ۱-۲-۵ راه‌اندازی سامانه ثبت گزارشات دیابت و پیش‌بینی

#### ۲-۲-۵ تولید کیت‌های ثبت نتایج دیابت جدید

## پیوست

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

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.

[.]+?\S

```
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	اختصاصی
Sigmoid	سیگماوار	اعتبارسنجی
BMI	شاخص توده بدنی	افزار
Classification	طبقه بندی	آشتگی
PhysActive	فعالیت فیزیکی	ب
TotChol	کلسترول کل	بیز ساده
DirectChol	کلسترول مستقیم	بیش برآذش
Underfitting	کم برآذش	بیماری های زمینه ای PIDD
Scatter matrix	ماتریس پراکندگی	پ
FP	مبثت کاذب	پیش پردازش
TP	مبثت واقعی	Preprocessing
IDE	محیط توسعه	تقویت کننده انطباقی AdaBoost
Mode	مد (آمار)	ج
Visualization	تصویرسازی	جنگل تصادفی
Scaling	مقیاس بندی	ت
	منحنی مشخصه	
	م	
	Sensitivity	حساست
	Dataset	دادگان
	NaN	داده نامشخص
	Training data	داده های آموزشی
	Testing data	داده های سنجش
	Missing data	داده های گم شده

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	BPDiaAve	میانگین فشار خون
	و	BPSysAve	دیاستولیک
MaritalStatus	وضعیت تاہل		میانگین فشار خون
Feature	ویژگی		سیستولیک
			ن
		AUC	ناحیه زیر منحنی
		FPR	نرخ مثبت کاذب

## مراجع

- [۱] اسماعیلی.مهدی، مفاهیم و تکنیک‌های داده‌کاوی
- [۲] نعمت‌الهی.نادر، آمار و احتمالات مهندسی

- [۳] What is diabetes?, Aoife M Egan, Sean F Dinneen
- [۴] Epidemiology of diabetes,Nita Gandhi Forouh,Nicholas J Wareha
- [۵] Diabetes Cookbook FOR DUMMIES 3RD EDITION, by Alan L. Rubin, MD with Cait James, MS
- [۶] Classification and prediction of diabetes disease using machine learning paradigm,Md.Maniruzzaman, Md. Jahanur Rahman, Benojir Ahammed and Md. Menhazul Abedin
- [۷] Prediction of Diabetes using Classification Algorithms, Deepti Sisodia ,Dilip Singh Sisodia
- [۸] Logistic Regression,Lynne Connelly
- [۹] Interpretable Machine Learning,Christoph Molnar
- [۱۰] Data Mining Concepts and Techniques ,Jiawei Han,Micheline Kamber,Jian Pei
- [۱۱] DATA MINING FOR BUSINESS ANALYTICS , GALIT SHMUELI, PETER C. BRUCE,PETER GEDECK,NITIN R. PATEL
- [۱۲] Top 10 algorithms in data mining,Xindong Wu,Jiannong Cao
- [۱۳] Machine Learning for Predictive Analysis, Amit Joshi, Mahdi Khosravy, Neeraj Gupta
- [۱۴] <https://www.cdc.gov/diabetes/basics/diabetes.html>
- [۱۵] <https://www.kaggle.com/datasets/cdc/national-health-and-nutrition-examination-survey>
- [۱۶] <https://www.techtarget.com/searchbusinessanalytics/definition/data-visualizatio>
- [۱۷] <https://www.datacamp.com/tutorial/adaboost-classifier-python>
- [۱۸] <https://scikit-learn.org/stable/modules/svm.html>
- [۱۹] <https://machinelearningmastery.com/confusion-matrix-machine-learning/>
- [۲۰] <https://www.investopedia.com/terms/m/mlr.asp>

[21] <https://nodered.org/docs/>