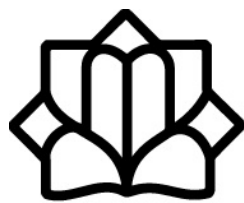


صلى الله عليه وسلم



دانشگاه کاشان

دانشکده مهندسی برق و کامپیوتر

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 کانایی می‌افشانند.

فهرست مطالب

آ	تقدیر و تشکر
چهار	فهرست تصاویر
۵	خلاصه
۶	۱ مقدمه
۶	۱-۱ تاریخچه
۷	۲-۱ علل ابتلا
۷	۱-۲-۱ فاکتورهای محیطی
۷	۲-۲-۱ عوامل ژنتیکی و غیر محیطی
۷	۳-۱ راهکارهای پیش‌گیری و استعدادسنجی
۷	۱-۳-۱ مبتلایان به پیش‌دیابت
۸	۲-۳-۱ سایر افراد جامعه
۸	۴-۱ انگیزه و اهداف انجام این پژوهش
۸	۵-۱ موارد انجام شده
۹	۲ ادبیات پژوهش
۹	۱-۲ مقدمه
۹	۲-۲ روش‌های داده‌کاوی
۹	۱-۲-۲ طبقه‌بندی و خوشه‌بندی
۱۰	۲-۲-۲ بیش‌برازش و کم‌برازش در فرایندها
۱۰	۳-۲ معرفی الگوریتم‌های طبقه‌بندی
۱۰	۱-۳-۲ رگرسیون لجستیک
۱۴	۲-۳-۲ درخت تصمیم
۱۴	۳-۳-۲ جنگل درختان تصادفی
۱۴	۴-۳-۲ AdaBoost
۱۵	۵-۳-۲ Naive Bayes
۱۵	۴-۲ دادگان، پیش‌پردازش و مصورسازی داده‌ها
۱۵	۱-۴-۲ دادگان
۱۵	۲-۴-۲ پیش‌پردازش داده‌ها
۱۹	۳-۴-۲ مصورسازی

۱۹	۴-۴-۲ انواع نمودارها و داده‌های اجمالی
۲۰	۵-۴-۲ اندازه‌گیری میزان خطای دقت
۲۲	۶-۴-۲ ROC و AUC
۲۳	۵-۲ نتیجه‌گیری
۲۴	۳ کارهای پیشین
۲۴	۱-۳ مقدمه
۲۴	۲-۳ مقاله ۱
۲۴	۳-۳ مقاله ۲
۲۵	۴-۳ نتیجه‌گیری
۲۶	۴ روش‌ها و نتایج
۲۶	۱-۴ مقدمه
۲۶	۲-۴ روش پیشنهادی
۲۶	۱-۲-۴ پیاده‌سازی
۳۹	۲-۲-۴ Arduino
۴۱	۳-۲-۴ ابزار Node Red
۴۱	۴-۲-۴ flask
۴۱	۳-۴ نتیجه‌گیری
۴۲	۵ جمع‌بندی و کارهای آتی
۴۲	۱-۵ جمع‌بندی
۴۲	۲-۵ کارهای آتی
۴۲	۱-۲-۵ راه‌اندازی سامانه ثبت گزارشات دیابت و پیش‌بینی
۴۲	۲-۲-۵ تولید کیت‌های ثبت نتایج دیابت جدید
۴۳	پیوست
۴۳	۱- کد استانداردسازی متون فارسی آمیخته به عبارات انگلیسی
۴۴	واژه‌نامه
۴۵	مراجع

فهرست تصاویر

۱-۲	مقایسه خوشه بندی و طبقه بندی	۹
۲-۲	نمودار پراکندگی	۱۰
۳-۲	نمودار رگرسیون خطی چندگانه در فضا	۱۱
۴-۲	مقایسه رگرسیون لجستیک و خطی	۱۲
۵-۲	نمودار تابع لجستیک	۱۳
۶-۲	مدل Logist Regression	۱۳
۷-۲	الگوریتم AdaBoost: در اینجا مراحل ذکر شده به صورت متناوب تکرار می شود. [۱۶]	۱۵
۸-۲	نمونه ای از داده های گم شده	۱۶
۹-۲	تعداد داده های گم شده در این پروژه به ازای هر ستون	۱۶
۱۰-۲	حذف مقادیر نامشخص در ستون متغیر وابسته	۱۷
۱۱-۲	افراز داده ها	۱۷
۱۲-۲	متغیرهای دودویی	۱۸
۱۳-۲	نمونه ای از متغیرهای دودویی	۱۸
۱۴-۲	آکد مربوط به مقیاس بندی داده ها	۱۸
۱۵-۲	Pairplot	۲۰
۱۶-۲	اساختار ماتریس آشفتگی: پارامترهای مذکور در این بخش در این ماتریس قرار گرفته اند.	۲۲
۱۷-۲	اساختار ماتریس آشفتگی برای الگوریتم جنگل درختان تصادفی	۲۲
۱۸-۲	نمای کلی نمودار ROC	۲۳
۱-۴	جایگزینی مقادیر نامشخص با متوسط های آماری	۲۷
۲-۴	نمودار مقایسه ابتلا به دیابت در نژاد های مختلف	۲۸
۳-۴	ماتریس Scatter	۲۹
۴-۴	نمودار عوامل موثر در ابتلا دیابت (۱)	۲۹
۵-۴	نمودار عوامل موثر در ابتلا دیابت (۲)	۳۰
۶-۴	نمودار ACC	۳۴
۷-۴	نمودار AUC	۳۵
۸-۴	نمودار ROC برای الگوریتم رگرسیون لجستیک	۳۵
۹-۴	نمودار ROC برای الگوریتم AdaBoost	۳۵
۱۰-۴	نمودار ROC برای الگوریتم جنگل درختان تصادفی	۳۶
۱۱-۴	نمودار ROC برای الگوریتم Naive Bayes	۳۶
۱۲-۴	نمودار ROC برای الگوریتم درخت تصمیم	۳۶

۳-۴	ساختار ماتریس آشفتگی برای الگوریتم رگرسیون لجستیک	۳۹
-----	---	----

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 جمع‌آوری شده‌اند می‌تواند 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 کرده‌ایم، از مجموعه داده‌گان بیماران آمریکایی^۱ است که در سال ۲۰۰۹ تا ۲۰۱۲ جمع‌آوری 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...

...

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. [۵] 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. [۱۳] آمار ابتلا و مرگ و میر بسیار بالایی از این بیماری در جهان وجود دارد و متأسفانه 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 عدد زیادی از افراد هم به پیش‌دیابت مبتلا هستند. [۶] 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. [۵] اگر پیش‌دیابت در افراد وجود داشته باشد، یک برنامه برای تغییر سبک زندگی، می‌تواند به افراد در جلوگیری از این بیماری کمک کند. [۱۳]

این بیماری سه نوع دارد: [۵]

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 بایستی 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 رایج تر 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.

۲-۱ 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. [۱۳] 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]

۲-۲-۱ عوامل ژنتیکی و غیر محیطی

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 می‌تواند در اثر بیماری‌های خانوادگی و اثری دیگری در افراد وجود داشته باشد. مثل برخی ویروس‌ها، وجود کلسترول، چربی و فشار خون و ... [۱۳] [۳] [۴]

۳-۱ 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 پیش‌گیری 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 می شوند. [۳] در این جا با تحلیل برخی فاکتورهای سلامتی می توان پیش بینی 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 بررسی های مختلفی انجام شده تا بتوانیم با اندازه گیری برخی فاکتورهای کمی و کیفی در افراد، مسئله استعداد در ابتلا به این بیماری را در آن ها بررسی کنیم. چالشی که در این زمینه وجود دارد این است که بسیاری از داده های غیرخطی و غیراستاندارد پزشکی با ارتباطات و ساختارهای پیچیده وجود دارند که این بررسی ها را دشوار می سازد. [۶]

۱-۴ انگیزه و اهداف انجام این پژوهش

در راستای همه موارد مطرح شده در بخش های 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 انسان های بیشتر با در نظر گیری مناسب ترین الگوریتم ها باشیم.

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

۱-۵ موارد انجام شده

در ابتدا مقالات مختلفی را مطالعه کردم و درمورد الگوریتم هایی 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 سال های ۲۰۰۹ تا ۲۰۱۲ بوده است، با به کار گیری کتابخانه های مختلف پایتون از جمله scikit learn، Pandas، seaborn، numpy و ... عملیات های گوناگونی بر روی داده ها 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 الگوریتم ها در این پیش بینی، شناخته شد.

فصل ۲

ادبیات پژوهش

۱-۲ مقدمه

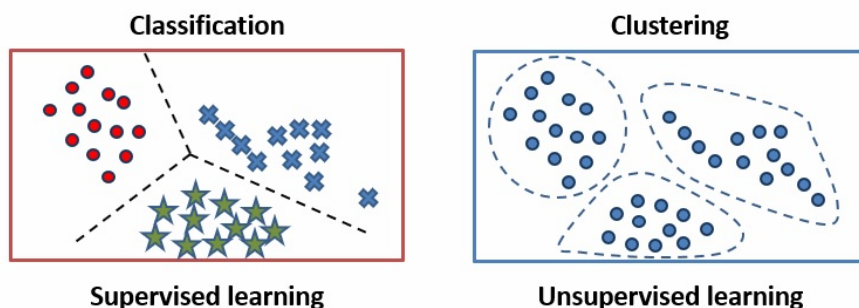
روش‌های نظارت شده‌ای مانند طبقه بندی و تخمین تلاش می‌کنند تا رابطه‌ای میان صفات خاصه ورودی (که گاه متغیرهای مستقل نامیده می‌شوند) را با یک یا چند صفت خاصه هدف (که گاه متغیر وابسته نامیده می‌شود) کشف کنند. در نهایت این رابطه با یک ساختار به عنوان مدل نمایش داده می‌شود. [۱]

... ..

۲-۲ روش‌های داده کاوی

۱-۲-۲ طبقه بندی و خوشه بندی

توضیح درمورد یادگیری ماشین و خوشه بندی و طبقه بندی و ...
از دید دیگر می‌توانیم بگوییم هرگاه حسایت ما برای تشخیص نمونه های گوناگون زیاد باشد



شکل ۲-۱: مقایسه خوشه بندی و طبقه بندی

۲-۲-۲ بیش برازش و کم برازش در فرایندها

یک مثال خوب که می توان 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 می رسد شرط اضافه ای برای تشخیص لیوان بودن یا نبودن اشیاء منظور کرده است؛ و در این حالت می گوییم مدل ما دچار بیش برازش شده است.

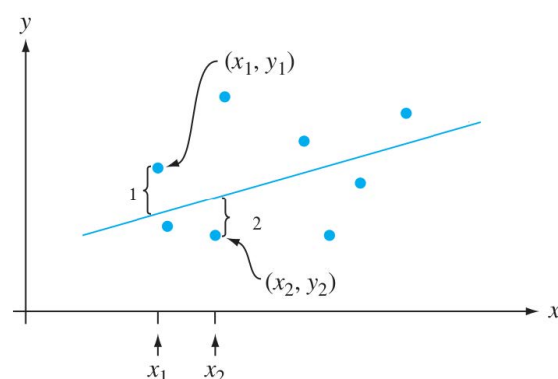
برعکس در حالتی که فرض کنیم مدل ما اهمیتی برای داشتن ارتفاع نسبت به سطح را در نظر نگیرد، ممکن است یک بشقاب را به عنوان یک لیوان در نظر بگیرد و در این حالت می گوییم مدل ما دچار کم برازش شده است.

۳-۲ معرفی الگوریتم های طبقه بندی

۱-۳-۲ رگرسیون لجستیک^۱

خوب است در ابتدا مروری بر رگرسیون خطی^۲ داشته باشیم تا در ادامه بتوانیم رگرسیون لجستیک را بهتر درک کنیم.
رگرسیون خطی:

در بسیاری از بررسی های آماری، لازم است یک متغیر وابسته را از روی یک یا چند متغیر مستقل پیش بینی کنیم که اصطلاحاً به آن رگرسیون یا برگشت می گوییم. [۲] 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 نمونه ای از جمعیت را در نظر گرفته و در آن مقادیرهای X_1 تا X_n در متغیر مستقل خود مقابل مقادیر نظیر در متغیر وابسته از Y_1 تا Y_n قرار می دهیم. [۲] سپس آن ها را مثل یک نمودار در صفحه مختصات به یکدیگر متصل کرده که به آن نمودار پراکندگی می گوییم. [۲]



شکل ۲-۲: نمودار پراکندگی

حالا می توان خطی را در این صفحه مختصات در نظر گرفت که تا حد زیادی منطبق بر نقاط باشد که در واقع یک نمودار پیش بینی کننده Y بر مبنای X است که به آن معادله رگرسیون Y بر روی X

^۱ Logistic Regression

^۲ Linear Regression

گویند. [۲] حالا رابطه این نقاط و منحنی را با $\mu_{Y|x} = E(Y|x) = \alpha + \beta x$ مشخص می‌کنیم و α و β 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 α و β که دارای بیشترین انطباق و کمترین خطا باشند برای ما کمک کننده است. [۲] بر اساس همین روش، با معادلات زیر به مقادیر بهینه α و β دست می‌یابیم. [۲]

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

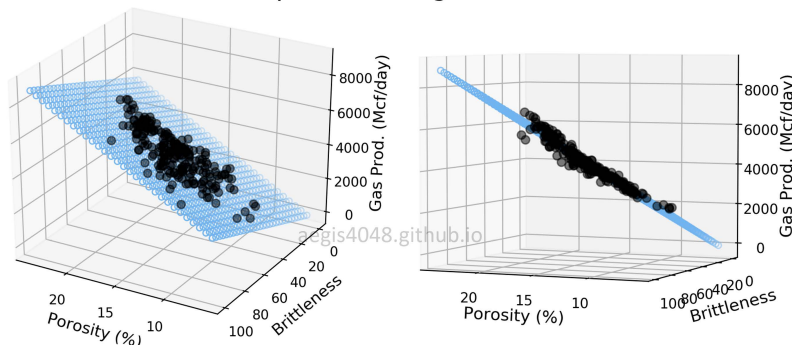
$$\beta = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2} \quad (۲-۲)$$

که در روابط فوق \bar{x} و \bar{y} میانگین x و y 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.

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

که در این رابطه، p ابعاد ما می‌باشد. [۱۹]

3D multiple linear regression model



شکل ۳-۲: 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.

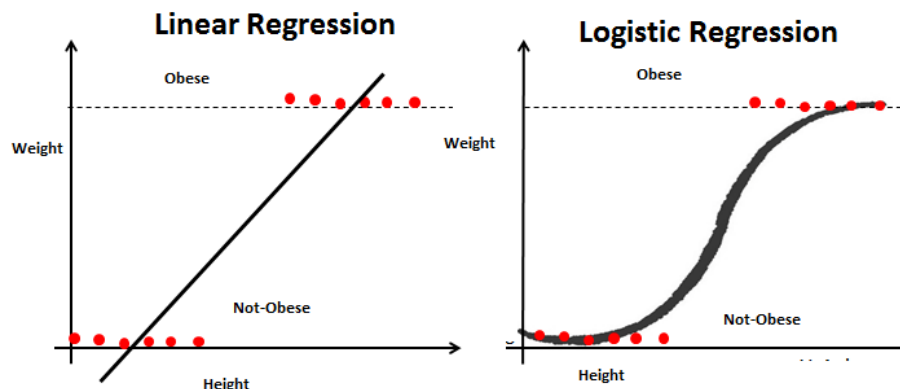
$$p(X, Y) = \text{corr}(X, Y) = \frac{\text{Cov}(x, y)}{(\text{Var}(x)\text{Var}(y))^{\frac{1}{2}}} \quad (۴-۲)$$

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.

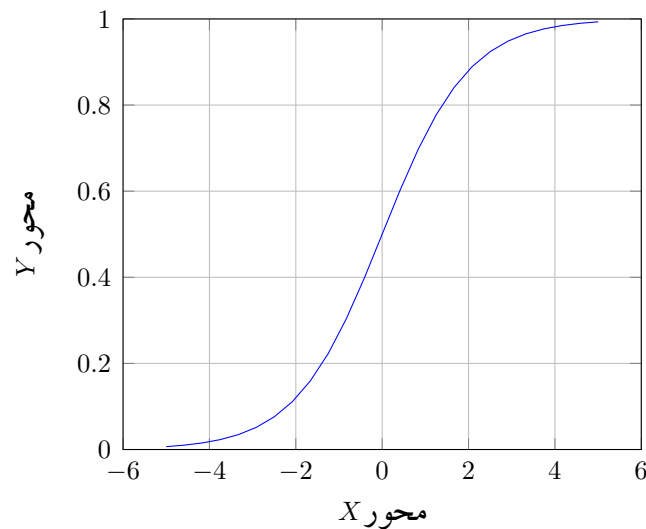
در این مدل به جای اینکه مقدار عددی برای متغیر وابسته تعریف شود، بر اساس احتمال متغیرهای دودویی را برای پیشگویی در نظر داریم [۱] که در ادامه این مسئله را بیشتر توضیح می‌دهیم. مثلاً کار ما، فاکتور BMI در افراد یک متغیر مستقل است و بر ابتلا به دیابت در افراد موثر بوده؛ همچنین در اینجا متغیر وابسته ما، همان 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 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 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 ۲-۴ دقت کنید، متوجه می‌شوید در حالت رگرسیون خطی برخی از نمونه‌ها در کلاس مربوطه قرار نگرفته‌اند.



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

$$\text{logistic}(x) = \frac{1}{1 + \exp^{-x}} \quad (۲-۵)$$



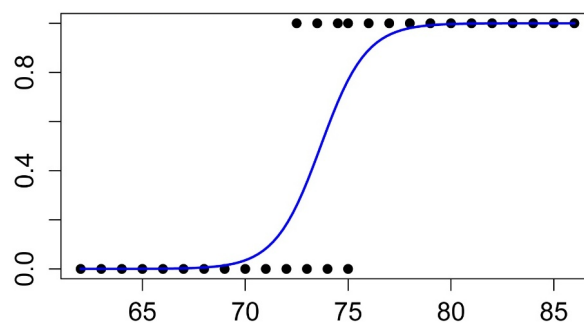
شکل ۲-۵: نمودار تابع لجستیک

سپس اگر سمت راست رابطه ۲-۳ را در جای X در معادله ۲-۵ (که آن را تابع سیگماوار^۳ 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 ویژگی‌هایمان 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]:

$$g(x) = \ln\left(\frac{p(x)}{1 - P(x)}\right) = \frac{\frac{e^{b_0 + b_1 x}}{1 + e^{b_0 + b_1 x}}}{1 - \frac{e^{b_0 + b_1 x}}{1 + e^{b_0 + b_1 x}}} = \ln(e^{b_0 + b_1 x}) = b_0 + b_1 x \quad (۲-۶)$$

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 ۲-۶: مدل Logist Regression

^۳Sigmoid/ 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 تصمیم‌گیری را انجام می‌دهد. ممکن است پرسید 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 رای‌گیری 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...

۴-۳-۲ AdaBoost

این الگوریتم یک فرض ساده در نظر دارد و آن یادگیری گروهی است. [۱۲] 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 هر پزشکی معیارهای مختلفی را برای تشخیص بیماری مراجعه‌کننده 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. [۱۰]

این روند جداسازی داده‌های آموزشی و امتیاز دهی در تشخیص توسط طبقه بندی‌کننده‌های مختلف درون درختان تصمیم ضعیف، مکرراً تکرار می‌شود تا در نهایت طبقه بندی‌های متعددی داشته باشیم که هر کدام بر اساس میزان سرآمدی در تست‌های مختلف، امتیازات مختلفی دریافت کنند. [۱۲] بعد از ساخته شدن مدل، حالا می‌توانیم به نسبت امتیاز هر طبقه بندی‌کننده داده‌های دریافتی را به صورت شانس 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 تقویت‌کننده تطبیقی گویند. [۱۰] [۱۲]

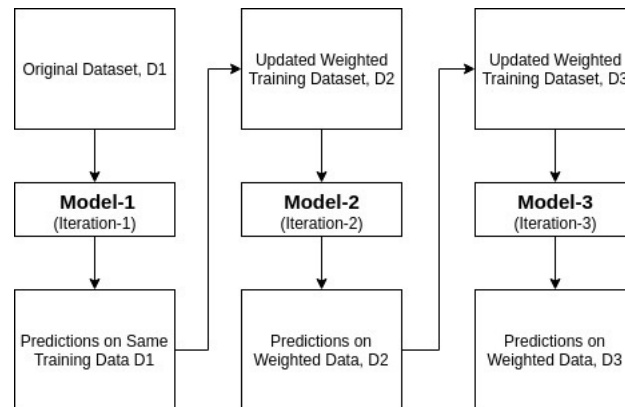
اگر از دید ریاضی الگوریتم را بررسی کنیم، برای محاسبه میزان خطای مدل، M_i وزن هر یک از تاپل‌های D_i را که M_i اشتباه طبقه بندی کرد، جمع می‌کنیم.

$$error(M_i) = \sum_{j=1}^d w_i \times err(X_j) \quad (۷-۲)$$

حالا اگر طبقه بندی‌کننده X_j اشتباه کند، میزان ارور برای آن ۱ اندازه‌گیری می‌شود و در غیر این صورت ۰ است. اگر یک طبقه بندی‌کننده آن قدر ضعیف باشد که خطایش از ۰.۵ بیشتر شود آن را دیگر در نظر نمی‌گیریم. [۱۰] 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، دقیق تر است و بنابراین، وزن آن برای انتخاب شدن باید بیشتر باشد. از رابطه ۲-۸ برای محاسبه وزن هر طبقه بندی استفاده می‌شود. [۱۰]

$$\log\left(\frac{1 - \text{error}(M_i)}{\text{error}(M_i)}\right) \quad (۲-۸)$$



شکل ۲-۷: الگوریتم AdaBoost: در اینجا مراحل ذکر شده به صورت متناوب تکرار می‌شود. [۱۶]

۵-۳-۲ Naive Bayes

۴-۲ دادگان، پیش‌پردازش و مصورسازی داده‌ها

۱-۴-۲ دادگان

دادگانی^۴ که از آن استفاده کردیم، برگرفته از یک برنامه مطالعاتی به نام NHANES بوده که جهت بررسی سلامتی کودکان و بزرگسالان از سوی CDC^۵ 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 اندازه‌گیری‌های سلامت و تغذیه برای رفع نیازهای نوظهور دارد. [۱۴]

مصاحبه NHANES شامل سوالات جمعیت‌شناختی، اجتماعی-اقتصادی، رژیم غذایی و سلامتی است. جزء معاینه شامل اندازه‌گیری‌های پزشکی، دندان‌پزشکی و فیزیولوژیکی و همچنین تست‌های آزمایشگاهی است که توسط پرسنل پزشکی بسیار آموزش دیده انجام می‌شود. [۱۴]

۲-۴-۲ پیش‌پردازش داده‌ها

پیش‌پردازش داده‌ها چیست؟

^۴Dataset

^۵US 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 گم‌شده 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 گم‌شده را با مقادیری جایگزین کنیم یا کلاً آن رکوردها را حذف کنیم تا مدل‌های دقیق‌تری داشته باشیم. (شکل ۲-۸)

	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 ۲-۸: نمونه‌ای از داده‌های گم‌شده

```
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 ۲-۹: تعداد داده‌های گم‌شده در این پروژه به ازای هر ستون

راه حل‌های داده‌های گم‌شده

• حذف ردیف‌های حامل داده‌های گم‌شده

یکی از راهکارهایی که در هنگام کار با داده‌های گم‌شده انجام می‌شود، حذف کل رکورد‌هایی است که دارای این مقادیر هستند. این مسئله یک بدی دارد و بدی آن این است که داده‌های کم‌تری 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.

در این پروژه، ردیف هایی که ستون آخرشان (یعنی Diabetes) دارای مقادیر نامشخص بودند را حذف کردیم. زیرا تمام مدل سازی های ما وابسته به آخرین ستون است و اگر این ستون مقادیر نادرستی داشته باشد، مدل هایی که می سازیم دقت پایینی خواهند داشت.

```
Replace undefined values

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

Python
```

شکل ۲-۱۰: حذف مقادیر نامشخص در ستون متغیر وابسته

• جایگزینی با متوسط های آماری

یکی از راهکارهای دیگر برای مدیریت داده های گم شده، جایگزین کردن مقادیر گم شده با جایگزینی با متوسط های آماری است. برخی از ستون ها را که زیاد اهمیت نداشته باشند می توان 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 آن ها استفاده کنیم، میتوانیم در ستونی که مربوط به میزان حقوق ماهیانه هر فرد می باشد، در صورت مشاهده مقادیر NaN، آن ها را با میانگین حقوق مشتریان جایگزین کرد. این راهکار، صرفاً به ما امکان می دهد تا تحلیل را ادامه دهیم و اطلاعات موجود در این رکورد را برای متغیرهای دیگر از دست ندهیم. [۱۱]

افراز داده ها

بخش کردن داده ها یا افراز^۶

در ساخت مدل مربوط به رگرسیون لجستیک که از K-Fold استفاده نکردیم، باید داده ها را به چهار دسته^۷ های X آموزشی، Y های سنجش، Y های آموزشی و Y های سنجش، تقسیم کنیم. [۱۱] مقیاسی که برای داده های سنجش در نظر گرفته شده ۲۵٪ است.

```
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
XTrain, XTest, YTrain, YTest = train_test_split(X,Y, test_size = 0.25, random_state = 0)
```

شکل ۲-۱۱: افراز داده ها

استاندارد سازی داده ها

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

متغیرهای ساختگی (دودویی)

در مدل های 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 نمی تواند متغیرهای

^۶Data Splitting

رشته ای را تشخیص دهد. پس از یک راهکار استفاده می کنیم. ستون هایی که حاوی مقادیر گسسته هستند را به چندین ستون دیگری تقسیم می کنیم (شکل ۲-۱۲) و مقادیر ستون های جدید را با بله/خیر (۱/۰) پر می کنیم. [۱۱] به عنوان مثال در ستون مربوط به وضعیت تاهل فرد، وضعیت های گوناگون را به ستون های مجزا تقسیم کردیم و برای هر کدام مقادیری در نظر گرفته شد. (شکل ۲-۱۳)

```
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 ۲-۱۲: متغیر های 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 ۲-۱۳: نمونه ای از متغیر های دودویی

مقیاس بندی

برخی از الگوریتم ها نیاز دارند که داده ها قبل از پیاده سازی مؤثر الگوریتم نرمال سازی شوند. برای نرمال سازی یک متغیر، میانگین را از هر مقدار کم می کنیم و سپس بر انحراف استاندارد تقسیم می کنیم. این عملیات گاهی اوقات استاندارد سازی نیز نامیده می شود. [۱۱]

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

```
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()

XTrain = sc.fit_transform(XTrain)
XTest = sc.transform(XTest)
```

شکل ۲-۱۴: کد مربوط به مقیاس بندی داده ها

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

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

۲-۴-۳ مصورسازی

تعریف: به طور ساده می‌توانیم بگوییم زمانی که داده‌هایمان را به صورت انواع نمودارها، نقشه‌ها و شکل‌های مختلف بصری دریاوریم تا نتیجه‌گیری و تحلیل آن‌ها توسط مغز جهت شناسایی الگوها و نقاط پرت در داده آسان تر شود، این کار انجام می‌گیرد. [۱۵]

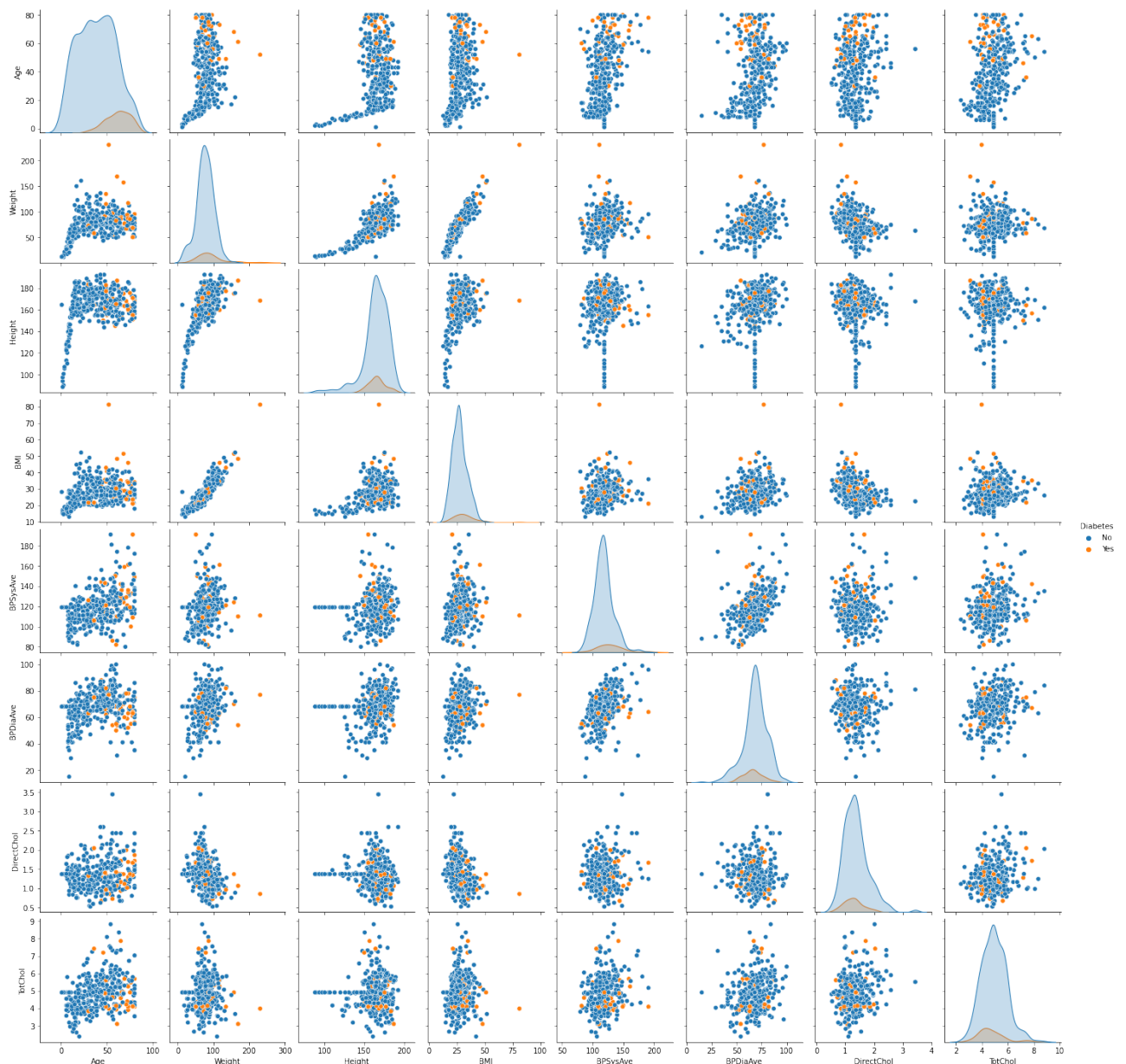
اهمیت: یک تصویر هزاران برابر بیشتر از کلمات ارزش دارد.

جمله فوق، به نوعی اهمیت مصورسازی را برای ما نمایان می‌کند. درواقع ما با انجام این کار، درک 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.

ضمناً در کنار این موارد، با مصور سازی داده‌ها دیگر نیازی به توضیحات اضافه نخواهیم داشت و بسیاری از افراد عادی قادر به درک موضوع مطرح شده خواهد بود. [۱۵]

۲-۴-۴ انواع نمودارها و داده‌های اجمالی

Pairplot



شکل ۲-۱۵: Pairplot

در شکل ۲-۱۵ 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....

۲-۴-۵ اندازه گیری میزان خطای دقت

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

که برای سنجش آن‌ها به کار گرفته ایم می‌پردازیم. [۱۰]

ابتدا به معرفی ۴ تاپل مختلف می‌پردازیم که بعداً از آن‌ها در محاسبه میزان خطا استفاده خواهیم کرد [۱۰]:

- **TP**^۷: این تاپل موارد مثبت واقعی را علامت گذاری می‌کند. مثلاً 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**^۸: 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**^۹: 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**^{۱۰}: 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.

• **FPR** یا $(1 - \text{Specificity})$:

رابطه ۲-۹ نشان دهنده نسبت افراد سالم شناسایی شده واقعی به کل افراد سالم (افراد اشتباه بیمار تشخیص داده شده به علاوه افراد سالم شناسایی شده واقعی) است.

$$1 - \text{Specificity} = \text{FPR} = \frac{FP}{FP + TN} \quad (۲-۹)$$

• **TPR** یا **Sensitivity**:

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

$$\text{Sensitivity}(\text{Recall}) = \text{TPR} = \frac{TP}{TP + FN} \quad (۲-۱۰)$$

تعدادی از این مقادیر را در ماتریسی به نام ماتریس آشفتگی^{۱۱} نشان می‌دهند [۱۸] و ساختار آن در شکل ۲-۱۶ است. همچنین این ماتریس را برای مدل جنگل درختان تصادفی رسم کردیم (شکل ۲-۱۷):

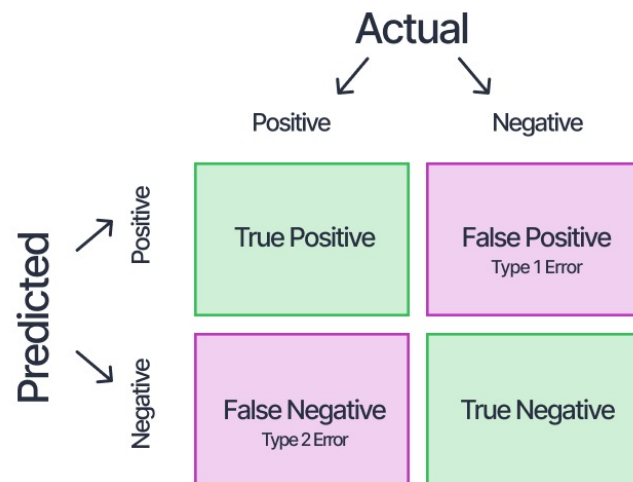
^۷True positives

^۸True negatives

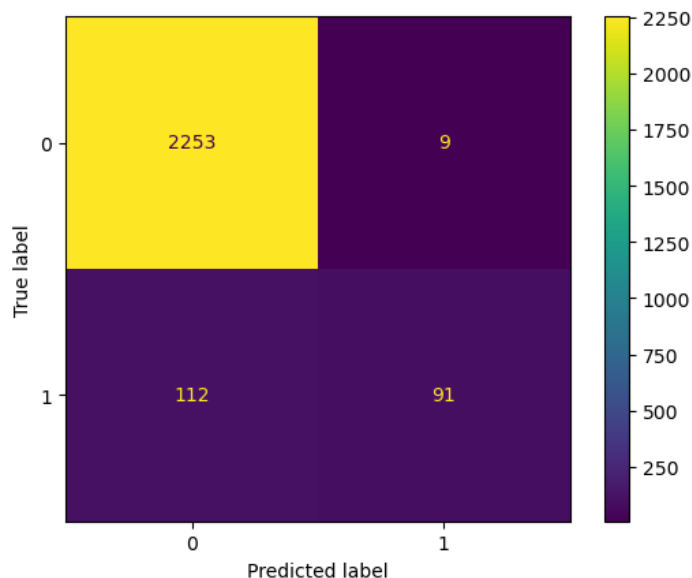
^۹False positives

^{۱۰}False negatives

^{۱۱}Confusion



شکل ۲-۱۶: ساختار ماتریس آشفته‌گی: پارامترهای مذکور در این بخش در این ماتریس قرار گرفته اند.



شکل ۲-۱۷: ساختار ماتریس آشفته‌گی برای الگوریتم جنگل درختان تصادفی

به طور کلی این ماتریس به ما خلاصه‌ای از درستی نتایج پیش‌بینی هایمان را نشان می‌دهد. [۱۸] حال میزان میزان دقت را با رابطه ۲-۱۱ محاسبه می‌کنیم.

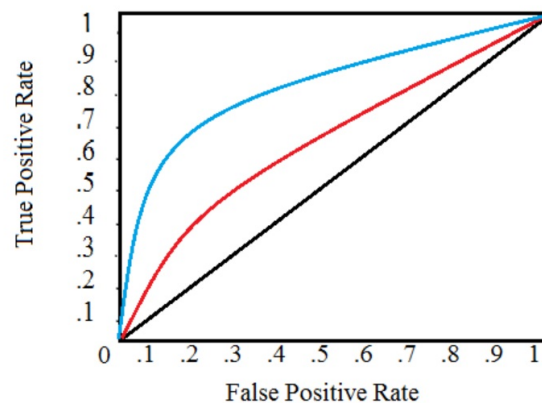
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (۲-۱۱)$$

۲-۴-۶ ROC و AUC

برای اینکه مقادیر پیش‌بینی‌های صحیح و غلط را به صورت یک نمودار نمایش دهیم، از دو شاخصه FPR و TPR که در بخش قبلی آن‌ها را معرفی کردیم استفاده می‌کنیم. [۱۰] ما در اینجا مقادیر مختلف

برای هر مرز (اشاره به تابع Sigmoid که در بخش‌های قبلی مطرح شد) پیش‌بینی را از ۰ تا ۱ اندازه‌گیری و در نمودار علامت می‌زنیم. [۱۰] اگر مقدار TPR برای یک مرز برابر با ۱ باشد یعنی در آن مرز مدل خیلی خوب عمل کرده و اگر برابر ۰/۵ باشد یعنی تصادفی عمل شده است. [۱۰] مثلاً در یک مرز در نظر می‌گیریم اگر مقدار تابع Sigmoid از ۰/۷ بیش تر بود، آن مورد را یک مورد مثبت شناسایی کن. در نمودار ROC^{۱۲} مولفه‌ی طول‌ها برابر با مقدار FPR است و مولفه‌ی عرض‌ها برابر TPR در آستانه‌های گوناگون است. [۱۰]

سطح زیر این نمودار را با AUC^{۱۳} نمایش می‌دهیم که بالا بودن این مساحت بیانگر این است که پیش‌بینی‌های بیشتری در این مدل درست بوده است. [۱۰]



شکل ۲-۱۸: نمای کلی نمودار ROC

۵-۲ نتیجه‌گیری

^{۱۲}Receiving Operating Characteristic

^{۱۳}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 تلقی شوند.

۲-۳ مقاله ۱

مقاله [۷] در سال ۲۰۱۸ با استفاده از داده‌های بیماران هندی^۱ 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 موثرترین الگوریتم در این بررسی در نظر 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. [۱۷]

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 گرفتند که نتایج آن‌ها داری دقت مناسبی نبوده است.

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

^۱PIDD

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 ییز ساده، جنگل درختان تصادفی و الگوریتم درخت تصمیم و رگرسیون لجستیک اشاره کرد.

فصل ۴

روش‌ها و نتایج

۴-۱ 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 پروژه، 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 خطاها 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.

۴-۲ روش پیشنهادی

۴-۲-۱ پیاده‌سازی

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.

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 ۰ با NaN در ستون‌هایی با مقادیر پیوسته بود مثل سن. زیرا به خوبی مشخص است این موارد از جمله داده‌های پرت محسوب می‌شوند که دقت مدل‌های ما را 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]
```

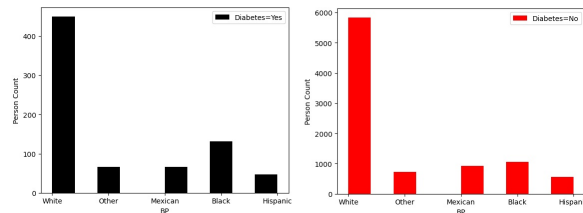
```
Race1 = YesDia['Race1'].tolist()
Race0 = NoDia['Race1'].tolist()
```

```
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()
```

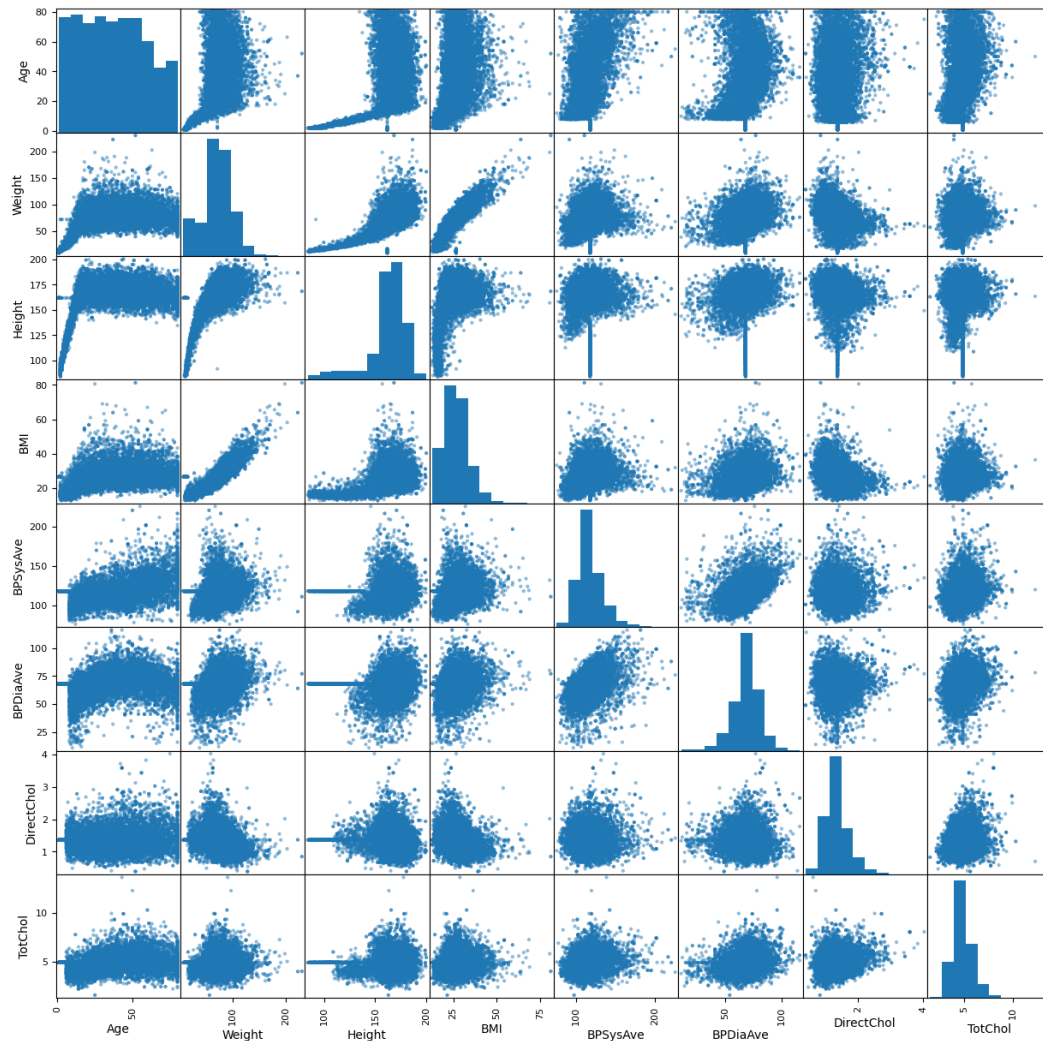
نتیجه عملیات فوق، نمودارهای زیر می باشد (در این نمودار، تاثیر نژاد در ابتلا به دیابت مشخص است):



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

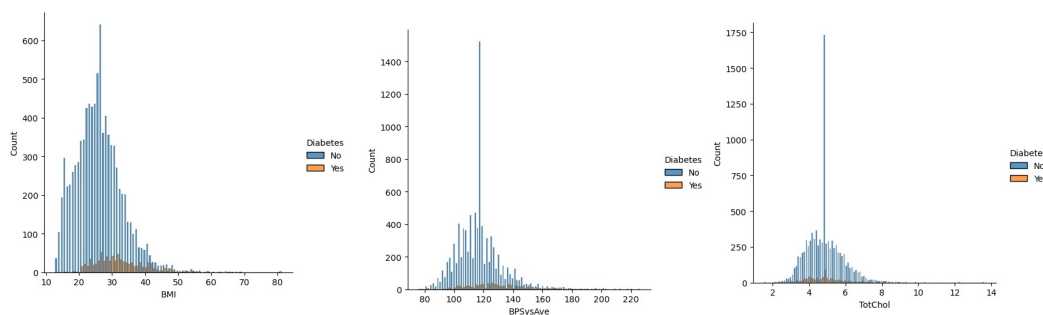
سپس با کد زیر، ماتریس scatter برای نمایش همبستگی میان داده‌ها رسم می‌کنیم:

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

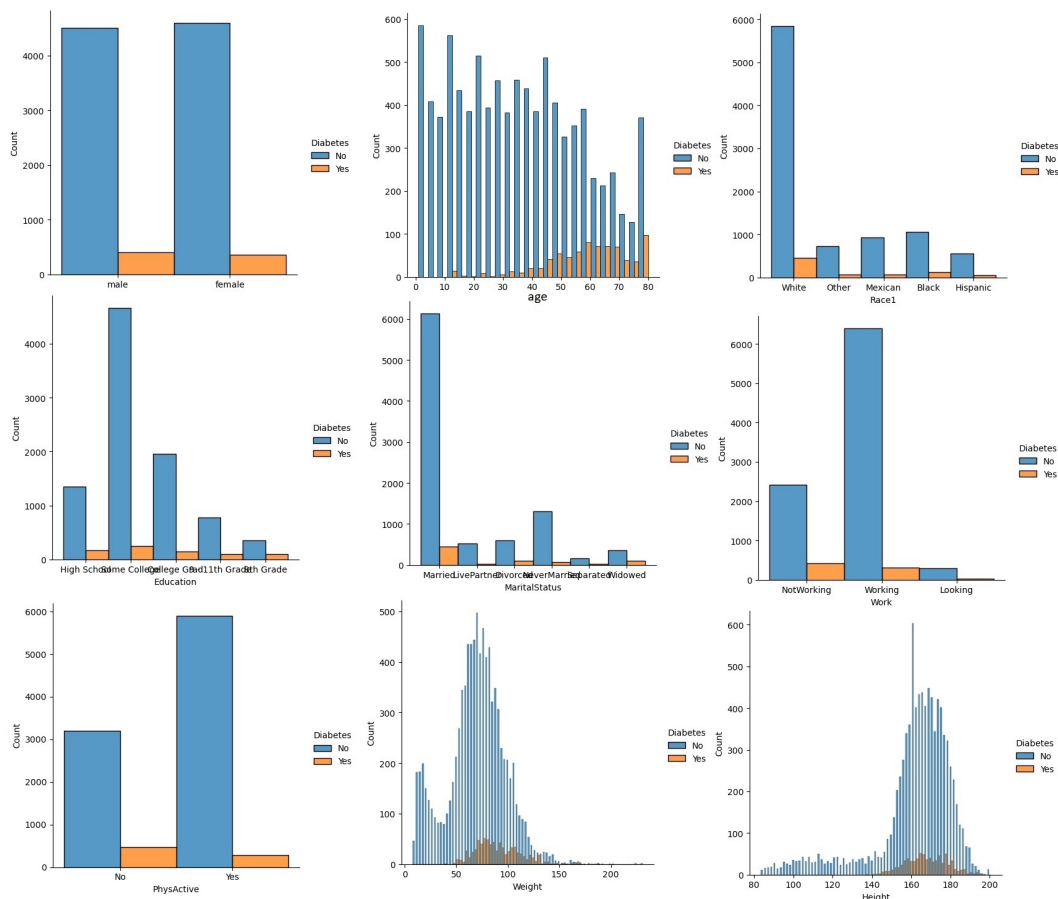



شکل ۴-۳: ماتریس Scatter

در شکل ۴-۳، برخی عوامل مرتبط با هم مشخص اند. (مثل رابطه کلسترول و وزن) در نمودارهایی که در ادامه خواهید دید، تاثیر پارامترهای گوناگون در ابتلای افراد به دیابت مورد بررسی است (شاخص)، BMI فشارخون، کلسترول خون، جنسیت، سن، نژاد، وضعیت تاهل، 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 نمودار های 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")
```

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)
```

و بعد از آن ستون Y و X هایمان مشخص کردیم تا داده هایمان را به بخش های آموزشی و اعتبارسنجی تقسیم کنیم.

```
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، داده‌ها را مقیاس‌بندی می‌کنیم تا در مدل‌های 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، دو تابع ایجاد می‌کنیم:

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 دیتاست حاوی تمامی مقادیر AUC و ACC ایجاد می کنیم تا با استفاده از آن، نمودار های AUC ، ACC 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 ۵، ۱۰ و ۲۰ 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')

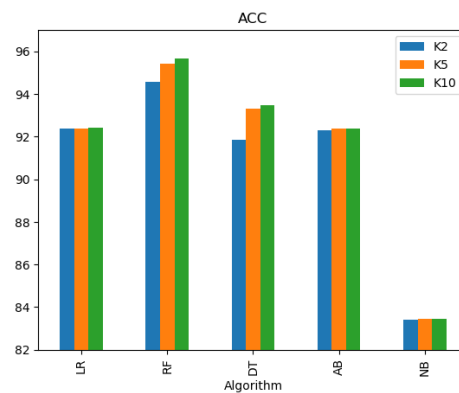
```

```

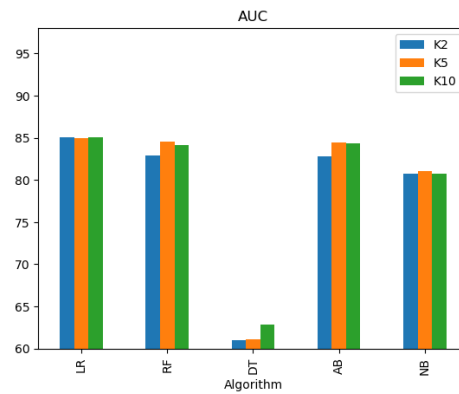
barauc['K2'] = barauc['K2'].astype('float64')
barauc['K5'] = barauc['K5'].astype('float64')
barauc['K10'] = barauc['K10'].astype('float64')
# plot grouped bar chart
barauc.plot(x='Algorithm',
            kind='bar',
            stacked=False,
            ylim=[60, 98],
            title='AUC')

```

نمودارها:

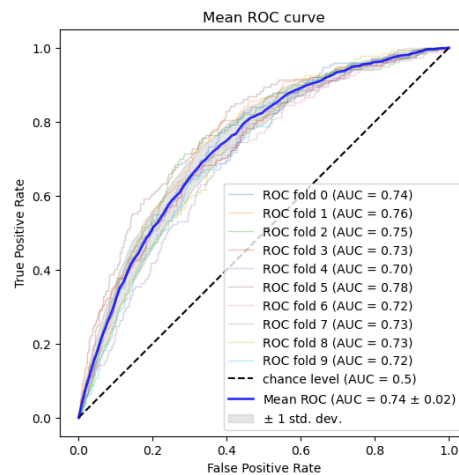


شکل ۴-۶: نمودار ACC

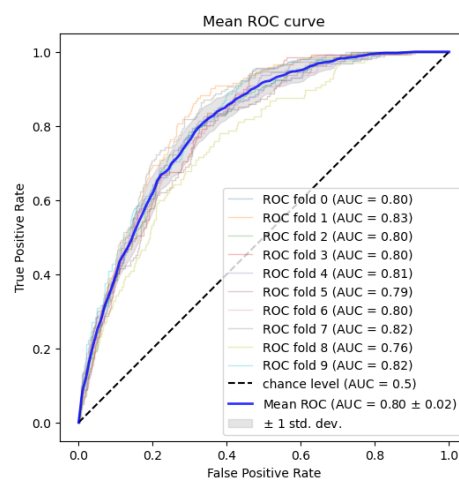


شکل ۴-۷: نمودار AUC

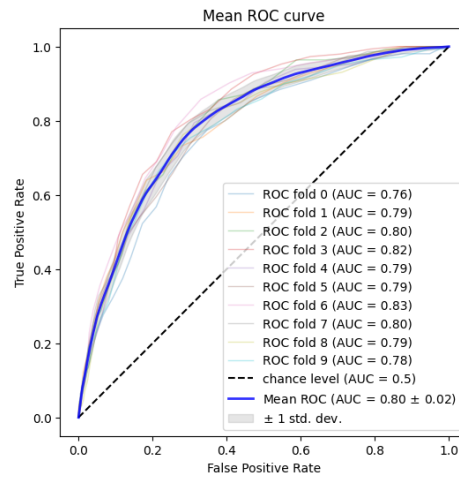
حال برای الگوریتم‌های مختلف، نمودار ROC را در Fold های مختلف رسم می‌کنیم و میانگین آن نیز مشاهده می‌شود:



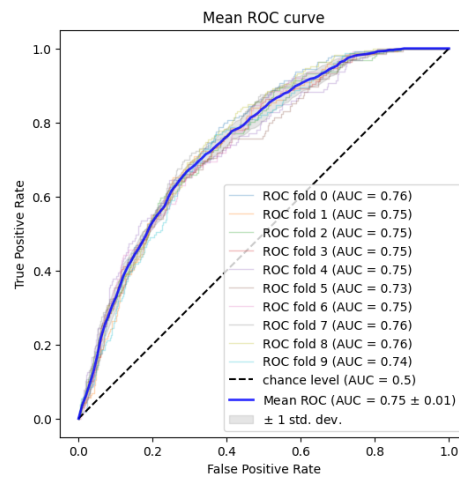
شکل ۴-۸: نمودار ROC برای الگوریتم رگرسیون لجستیک



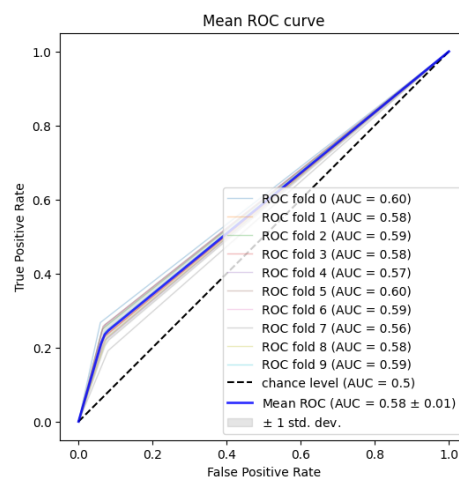
شکل ۴-۹: نمودار ROC برای الگوریتم AdaBoost



شکل ۴-۱۰: نمودار ROC برای الگوریتم جنگل درختان تصادفی



شکل ۴-۱۱: نمودار ROC برای الگوریتم Naive Bayes



شکل ۴-۱۲: 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)

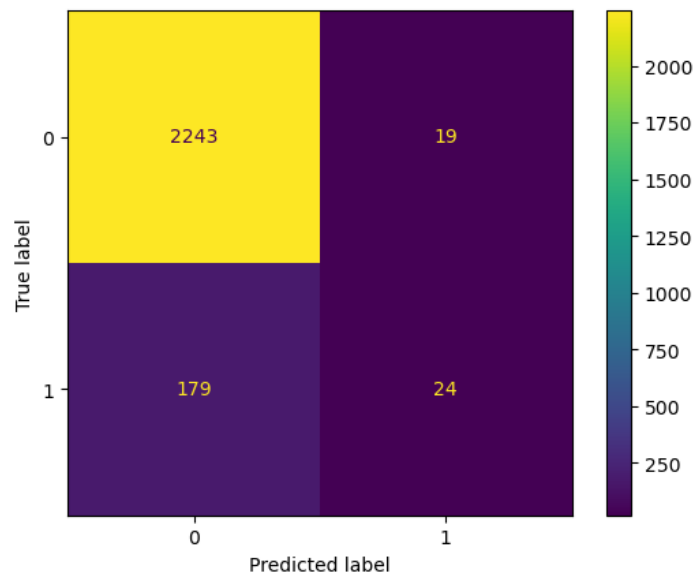
```

و در نهایت ماتریس آشفتگی 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 ۴-۱۳: ساختار ماتریس آشفته‌گی برای الگوریتم رگرسیون لجستیک

۴-۲-۲ Arduino

کد برد 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);}
```

۳-۲-۴ ابزار Node Red

۴-۲-۴ flask

۳-۴ 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 شده‌اند 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 سمبلیک 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 راه‌اندازی 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 برآوردی 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 متون 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

جاده‌ها	Data	ب	
جاده‌ها	Data	داده‌ها	Data
جاده‌ها	Data	داده‌ها	Data
جاده‌ها	Data	داده‌ها	Data
		داده‌ها	Data
		د	

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 Sisodia
- [8] Logistic Regression, Lynne Connelly
- [9] Interpretable Machine Learning, Christoph Molnar
- [10] Data Mining Concepts and Techniques, Jiawei Han, Micheline Kamber, Jian Pei
- [11] DATA MINING FOR BUSINESS ANALYTICS, GALIT SHMUELI, PETER C. BRUCE, PETER GEDECK, NITIN R. PATEL
- [12] Top 10 algorithms in data mining, Xindong Wu, Jiannong Cao
- [13] <https://www.cdc.gov/diabetes/basics/diabetes.html>
- [14] <https://www.kaggle.com/datasets/cdc/national-health-and-nutrition-examination-survey>
- [15] <https://www.techtarget.com/searchbusinessanalytics/definition/data-visualization>
- [16] <https://www.datacamp.com/tutorial/adaboost-classifier-python>
- [17] <https://scikit-learn.org/stable/modules/svm.html>
- [18] <https://machinelearningmastery.com/confusion-matrix-machine-learning/>
- [19] <https://www.investopedia.com/terms/m/mlr.asp>
- [20] <https://nodered.org/docs/>