به نام خدا

دوره جامع علم داده

دانشگاه تهران

پروژه پایانی

دوره مبانی علم داده

استاد: جناب آقای دکتر محتاط

دانش پذیران:

پروین شاهسوند

مجید ذوقی رودسری

پروژه پایانی:

علم داده[1] مجموعه‌ای از رویه‌ها، سیستم‌ها و فرآیندهاست که به منظور کشف دانش از داده‌های گردآوری شده که اصطلاحا به آن دیتاست می‌گویند، می‌باشد. ما در این پروژه از دیتاست دیابت[2] استفاده می‌کنیم که در پایگاه Kaggle[3] در دسترس می‌باشد. در واقع، این مجموعه داده توسط موسسه ملی دیابت و بیماری‌های گوارشی و کلیوی تهیه گردیده است و پس از بی نام سازی داده‌های مربوط به اشخاص مشارکت کننده در جمع‌آوری این دیتاست و با رعایت مقررات عمومی حفاظت از داده، به صورت همگانی به اشتراک گذاشته شده است. همچنین، محدودیت‌هایی در انتخاب نمونه‌ها اعمال گردیده و صرفا اطلاعات مراجعان زن سرخپوست[4] با حداقل ۲۱ سال سن در این دیتاست قرار داده شده است که در مجموع تعداد ۷۶۸ نمونه (۵۰۰ زن سالم و ۲۶۸ زن مبتلا به دیابت) در این دیتاست قرار داده شده است. هدف این پروژه پیش بینی احتمال ابتلا به بیماری دیابت در زنان بر اساس ویژگی‌ها و علائم اندازه گیری شده است. پروژه حاصل توسط ابزار IBM SPSS Modeler نسخه ۱۸.۰ پیاده سازی گردیده است که در ادامه مراحل انجام آن مطابق روش CRIP-DM شرح داده می‌شود.

هشت ویژگی که در جدول ذیل نشان داده شده است، بر اساس نتایج حاصله از آزمایش خون، اندازه گیری قد و وزن و پرسش از داوطلبان گردآوری گردیده است. احتمال ابتلا به دیابت با دو مقدار ۰ (نشان دهنده احتمال عدم ابتلا به دیابت) و ۱ (نشان دهنده احتمال ابتلا به دیابت) در ستون Outcome قرار گرفته است.

ID	Name	Description
1	Pregnancies	Number of times pregnant
2	Glucose	Plasma glucose concentration a 2-hours in an oral glucose tolerance test
3	BloodPressure	Diastolic blood pressure (mm Hg)
4	SkinThickness	Triceps skin fold thickness (mm)
5	Insulin	2-Hour serum insulin (mu U/ml)
6	BMI	Body mass index (weight in kg/(height in m)^2)
7	DiabetesPedigreeFunction	Diabetes pedigree function
8	Age	Age (years)

جدول شماره ۱: ریسک فاکتورهای تشخیصی بیماری دیابت

[1] Data Science
[2] Diabetes
[3] https://www.kaggle.com/datasets/mathchi/diabetes-data-set
[4] زنان سرخپوست از تبار پیما هستند که در آمریکای شمالی سکونت دارند.

درک داده[5]

بیماری دیابت زمانی به وجود می‌آید که بدن قادر به تولید انسولین نباشد یا هنگامی‌که بدن نمی‌تواند از انسولین تولید شده استفاده مؤثر داشته باشد. در یک تقسیم بندی، این بیماری به سه دسته تقسیم می‌شود. دیابت نوع یک یا دیابت وابسته به انسولین که بیشتر در کودکان دیده می‌شود. دیابت نوع دو یا دیابت غیر وابسته به انسولین که در ۹۰ تا ۹۵ درصد بیماران دیابتی مشاهده می‌شود و دیابت نوع سوم که بیشتر در زنان باردار مشاهده می‌شود که پس از بارداری به دیابت نوع دوم تبدیل می‌شود.

بیماری دیابت یکی از بیماری‌هایی است که انواع مختلفی دارد و براساس انواع آن علت‌های ایجاد آن نیز طبقه‌بندی می‌شوند. به عنوان مثال می‌توان گفت که دیابتی که در دوران بارداری ایجاد می‌شود با دیابتی که در دوران جوانی و کودکی بروز می‌کند تفاوت دارد و علت‌های ایجاد آنها نیز می‌توانند متنوع باشند. بنابراین در ادامه به تفکیک گروه‌هایی که علت ایجاد آنها می‌تواند مشابه باشد، دیابت را توضیح داده‌ایم.

دیابت نوع ۱: این نوع عنوان دیابت نوجوان نیز شناخته می‌شود، زمانی رخ می‌دهد که بدن نتواند انسولین تولید کند. در بیماری‌های خودایمنی مانند دیابت نوع ۱، سیستم ایمنی به اشتباه آنتی‌بادی‌ها و سلول‌های التهابی تولید می‌کند که سبب آسیب رساندن به بافت بدن انسان می‌شوند. در افراد مبتلا به نوع ۱ بیماری، سلول‌های بتای پانکراس، که مسئول تولید انسولین هستند، توسط سیستم ایمنی ناخواسته مورد حمله قرار می‌گیرند و تخریب می‌شوند. این امر باعث می‌شود که سطح انسولین خون شما کم شود یا اینکه انسولین تولید نشود. درابتلا به دیابت نوع۱، عواملی از جمله سابقه خانوادگی قند خون بالا بشدت موثر می‌باشند. همچنین دیده‌شده که اضافه وزن درابتلا به دیابت نوع ۱ یک تاثیری ندارد. دیابت نوع ۱ را دیابت وابسته به انسولین نیز می‌گویند به این معنی که فرد باید روزانه انسولین مصنوعی را دریافت کند تا زنده بماند.

دیابت نوع ۲: در مرحله پیش‌دیابتی که می‌تواند به دیابت نوع ۲ منجر شود، سلول‌ها به اثر انسولین مقاوم می‌شوند در واقع بر خلاف نوع ۱ در حالی‌که غده پانکراس هنوز انسولین را تولید می‌کند، سلول‌های بدن به طور موثری به آن پاسخ نمی‌دهند و به جای انتقال گلوکز به سلول‌هایی که به انرژی نیاز دارند، گلوکز در جریان خون باقی می‌ماند. عوامل زیادی در ابتلا به این نوع دیابت نقش ایفا می‌کنند ازجمله فشار خون بالا، بافت چربی اضافه در بدن که باعث می‌شود سلولها به انسولین مقاوم شوند. همچنین سابقه خانوادگی در ابتلا به این نوع از دیابت بشدت تاثیرگذار می‌باشد. یکی دیگر از عوامل مهم در ابتلا به این نوع دیابت افزایش سن می‌باشد که دلیل آن می‌تواند کاهش میزان فعالیت بدن و همچنین افزایش شاخص توده بدن در سنین بالا باشد. سابقه دیابت بارداری هم در

[5] Understand data

ابتلا به دیابت نوع ۲، موثر است به این دلیل که احتمال ابتلا به دیابت نوع ۲ بعد از بارداری برای زنانی که در دوران بارداری دچار دیابت بوده‌اند چهار برابر بیشتر از زنانی است که در دوران بارداری به دیابت مبتلا نبوده‌اند.

دیابت حاملگی: این نوع از مشکل قند خون در دوران بارداری ایجاد می‌شود. در واقع در طول بارداری، جفت، هورمون‌هایی را تولید می‌کند تا بارداری را حفظ کند. این هورمون‌ها سلول‌ها را به انسولین مقاوم‌تر می‌کنند. به طور معمول، پانکراس با تولید انسولین بیشتر به اندازه کافی برای غلبه بر این مقاومت پاسخ می‌دهد. اما گاهی اوقات پانکراس نمی‌تواند این کار را به درستی انجام دهد. در این حالت گلوکز بسیار کمی به سلول‌ها منتقل می‌شود و سطح آن در خون زیاد می‌شود و منجر به دیابت حاملگی می‌شود. این نوع بیماری تنها در برخی از زنان باردار دیده می‌شود و معمولا پس از زایمان برطرف می‌شود. سن بالای ۲۵ سال، قند خون بالا، سابقه خانوادگی، وزن بالا و نژاد از عوامل بسیار موثر در ابتلا به این نوع دیابت می‌باشد.

این مجموع داده شامل بانوانی می‌باشد که تعدادی از آنها دیابت نوع ۱، نوع ۲ و یا دیابت بارداری مبتلا هستند. ویژگی‌هایی که در اینجا مورد بررسی می‌شود شامل موارد ذیل می‌گردد:

۱- تعداد دفعات وضع حمل

۲- غلظت گلوکز پالسمای خون

۳- فشار خون دیاستولیک

۴- ضخامت پوست ماهیچه سه سر بازویی

۵- انسولین سرم دوساعته

۶- شاخص توده بدنی

۷- داشتن سابقه دیابت

۸- سن

آماده سازی دادگان[6]

مولفه Read File: برای آماده سازی داده‌ها در ابتدا باید مجموعه داده‌های موضوعی را در ابزار IBM SPSS Modeler وارد کرد. از آنجا که این فایل از نوع CSV[7] است باید با استفاده از Var. File که از قسمت Source قابل دسترس است نسبت به وارد کرد فایل[8] اقدام نمود. پس از آنکه آدرس فایل را در قسمت File و از طریق دکمه[9] Browse for file تعیین کردیم، باید از قسمت Field Delimiters نسبت به تعیین جدا کننده‌ها اقدام نماییم. از آنجا که فایل ما از نوع CSV بوده است و به این معنی است که نمونه‌ها از طریق خط جدید از هم جدا شده‌اند و متغیرها[10] با علامت , جدا شده‌اند، پس در این قسمت Comma و Newline را تیک می‌زنیم. بهتر است از نوع کد گذاری UTF-8 در قسمت Encoding استفاده نماییم تا در صورتی وجود کاراکترهای غیر لاتین با مشکل مواجه نگردد و همچنین پیشنهاد می‌شود Double quotes را در حالت Pair and discard قرار دهیم تا در صورت وجود علامت نقل قول آن را نادیده گرفته و کاراکترها را بصورت یکپارچه برگرداند. تعیین نوع داده‌ها در شناسایی داده پرت و مفقوده و همچنین مدلسازی بسیار با اهیمیت می‌باشد، می‌توانیم در این مرحله از برگ نشان[11] Types نسبت به تعیین نوع هر یک از متغیرها اقدام نماییم. برای این کار گزینه Read Values را کلیک می‌کنیم و نوع داده‌ها بصورت خودکار شناسایی می‌گردند و در قسمت Values بازه مقادیر آن نشان داده می‌شوند. بهتر است قبل از شناسایی داده پرت، تمامی متغیرهای وابسته را از نوع Continuous در نظر بگیریم تا قادر به شناسایی باشد (همچنین پیشنهاد می‌گردد متغیرهای گسسته را نیز در مدل‌های مبتنی بر فاصله از نوع پیوسته در نظر بگیریم). با این وجود تنها متغیر Outcome را با Measurement از نوع Flag (دو مقداره) تعیین می‌کنیم و Role آن را به Target تغییر می‌دهیم.

مولفه For test: برای مشاهده داده‌ها می‌توانستیم در قسمت قبلی از دکمه Preview استفاده کنیم. در این حالت صرفا ۱۰ نمونه از کل داده‌ها نشان داده می‌شود. برای مشاهده کل داده‌ها می‌توانیم از Table که از قسمت Output قابل دسترسی است استفاده نماییم. تنها کافی است آن را به Read File متصل کرده و Run را بزنیم. جدولی از کلیه داده‌ها نشان داده می‌شود.

---

[6] Data Preparation
[7] Comma Separated Values
[8] Import
[9] Button
[10] Field
[11] Tab

نکته ۱: تمامی مولفه‌های شرح داده شده در ذیل در صورتی فعال می‌شوند که به مرحله قبل خود متصل[۱۲] باشند که به اختصار در توضیحات لحاظ نشده است.

مولفه Zero Removal: همانطور که در مولفه Read File و در هنگام تعیین نوع داده‌ها محدوده داده‌ها و همچنین در مولفه For test مشاهده کردیم، کمترین مقدار متغیرهای Pregnancies، Glucose، BloodPressure، SkinThickness، Insulin، BMI و DiabetesPedigreeFunction مقدار صفر بوده است. در مورد تعداد وضع حمل مقدار صفر منطقی است اما در بقیه موارد این مقدار نشان دهنده وجود نویز در داده‌هاست و می‌توان حدس زد که مقادیر Null به اشتباه با صفر تکمیل گردیده است. لذا در مرحله بعد آماده سازی باید به سراغ آن رفت و مقادیر صفر را با Null جایگزین کرد تا در مرحله داده‌های مفقوده آن را مدیریت نمود. برای این کار از Filler که از قسمت Field Ops قابل دسترسی است می‌توان استفاده نمود. ابتدا فیلدهای دارای صفر غیر مجاز را در قسمت Fill in fields انتخاب می‌کنیم و با انتخاب حالت Always در قسمت Replace دستور زیر را در قسمت Replace with می‌نویسیم.

```
if @FIELD = 0 then undef else @FIELD endif
```

شایان ذکر است @FIELD نشان دهنده کلیه متغیرهای انتخاب شده در قسمت قبل است و undef نشان دهنده Null است.

مولفه Normal Distribution Test: در این مرحله باید داده‌های پرت[۱۳] را مدیریت کنیم. برای شناسایی داده‌های پرت اغلب از دو روش Z-Score و IQR استفاده می‌شود. زمانی روش اول را انتخاب می‌کنیم که متغیر دارای توزیع نرمال یا گاما باشد و در غیر اینصورت معمولا از روش دوم بهره‌گیری می‌شود. برای درک بهتر نوع توزیع هر یک از متغیرها از Sim Fit که از قسمت Output قابل دستیابی است، استفاده می‌کنیم. در اینجا گزینه‌ها را بصورت پیش فرض بدون تغییر قرار داده و با زدن Run گزارش مربوطه را ایجاد می‌کنیم. در گزارش ایجاد شده در قسمت Simulated Fields تمامی متغیرها و نوع توزیع هریک نشان داده شده است. متغیر BloodPressure از نوع Normal و متغیر BMI از نوع Gamma تشخیص داده شده است و مابقی از توزیع‌های دیگری تبعیت می‌کنند. پس برای شناسایی داده‌های پرت برای متغیرهای اشاری از روش Z-Score استفاده می‌کنیم و برای مابقی از روش IQR بهره گیری می‌شود.

[12] Connect
[13] Outliers

مولفه Read Values: پس از استفاده از Filler و حذف مقادیر صفر نامعتبر، بهتر است مجددا نوع داده‌ها خوانده و تعیین گردد، لذا از Type (از قسمت Field Ops) برای این کار استفاده می‌گردد که در اینجا به عنوان مولفه Read Values نام گذاری شده است.

مولفه‌های Generating Z-Score و Generating IQR: جهت مدیریت داده‌های پرت باید از Data Audit که از قسمت Output قابل دسترسی است، استفاده کرد. از آنجا که دو روش مختلف برای مدیریت داده پرت مد نظر است باید از دو مولفه جداگانه بهره برد و دو سوپرنود[۱۴] جداگانه با نام‌های Z-Score Supernode و IQR Supernode ایجاد کرد.

نکته۲: شایان ذکر است نیازی به نگهداری Data Audit پس از ایجاد سوپرنود مربوطه نیست و می‌توان آن را حذف نمود اما در اینجا به دلیل مشخص نمودن تمامی مراحل آن را حفظ کردیم.

نکته۳: IBM SPSS Modeler داده‌های پرت را در دو دسته جداگانه داده پرت Outlier و داده خیلی پرت Extreme در نظر گرفته و می‌توان رفتار جداگانه‌ای با آن داشت.

نکته۴: پس از ایجاد این سوپرنودها برای هریک الزامی است مجددا نوع داده‌ها خوانده و تعیین گردد، لذا از Type (از قسمت Field Ops) برای این کار استفاده می‌گردد که در اینجا به عنوان مولفه Read Values نام گذاری شده است.

به منظور شناسایی داده‌های پرت در برگ نشان Quality از پنجره Data Audit و در قسمت Outliers & Extreme Values روش شناسایی داده پرت را مشخص می‌کنیم. جهت روش Z-Score گزینه Standard deviation from mean را انتخاب می‌کنیم و مقادیر Outliers و Extremes را به ترتیب برابر ۳ و ۵ (که در واقع به معنای 3δ و 5δ می‌باشد) قرار می‌دهیم. جهت روش IQR گزینه Interquartile ranges from upper/lower quartiles را اینخاب می‌کنیم و مقادیر Outliers و Extremes را به ترتیب برابر ۱/۵ و ۳ (که در واقع به معنای 1.5 IQR و 3 IQR می‌باشد) قرار می‌دهیم. در هر دو روش پس از زدن Run پنجره ای گشوده می‌شود که در برگ نشان Quality می‌توان تعداد داده‌های پرت و داده‌های خیلی پرت را در مقابل هر یک از متغیرها مشاهده نمود. جهت مدیریت داده‌های پرت به راحتی می‌توان از قسمت Action روش مورد نظر خود را به تفکیک متغیرها معین نمود.

---

[۱۴] SuperNode : مولفه‌ای است که خود دارای چند مولفه دیگر در خود می‌باشد. برای مشاهدهدرون این مولفه می‌توان پس از گشودن آن از Zoom In استفاده کرده تا درون این نود و رفتار آن را مشاهده کنیم.

همانطور که در قبل گفته شد، برای دو متغیر BloodPressure و BMI از روش Z-Score برای شناسایی داده پرت استفاده می‌کنیم. از آنجا که BloodPressure دارای مقدار Extreme نیست از متد Coerce استفاده می‌کنیم. این روش در واقع داده‌های پرت را با حد بالا یا حد پایین جایگزین می‌نماید. در متغیر BMI به دلیل دارا بودن یک مقدار Extreme از متد Coerce outlier / nullify extreme برای آن استفاده می‌کنیم. این روش داده‌های پرت را با حد بالا یا حد پایین جایگزین کرده و داده‌های خیلی پرت را با Null جایگزین می‌کند تا در مرحله مدیریت داده‌های مفقوده مورد بررسی واقع گردد. سایر متغیرهایی که داده‌های پرت آن‌ها با روش IQR شناسایی شده‌اند و متد بکارگرفته برای هریک در ذیل شرح داده شده است.

- متغیر Pregnancies: دارای ۴ داده پرت است و از متد Coerce استفاده می‌کنیم.
- متغیر SkinThickness: دارای ۲ داده پرت و ۱ داده خیلی پرت است و از متد Coerce outlier / nullify extreme استفاده می‌کنیم.
- متغیر Insulin: دارای ۱۶ داده پرت و ۸ داده خیلی پرت است و از متد Coerce outlier / nullify extreme استفاده می‌کنیم.
- متغیر DiabetesPedigreeFunction: دارای ۲۳ داده پرت و ۶ داده خیلی پرت است و از متد Coerce outlier / nullify extreme استفاده می‌کنیم.
- متغیرAge: دارای ۹ داده پرت است و از متد Coerce استفاده می‌کنیم.

سایر متغیرهایی که به آن‌ها اشاره نشد، دارای داده پرت نیستند. پس از تشخیص داده‌های پرت و تعیین روش مدیریت آن، جهت ایجاد سوپرنود مربوطه باید از قسمت Generate گزینه Outlier & Extreme SuperNode را کلیک کنیم.

مولفه Miss-values MNG: پس از مدیریت داده‌های پرت، در این مرحله داده‌های مفقوده[15] را شناسایی و مدیریت می‌نماییم. برای اینکار مجددا از Data Audit استفاده می‌کنیم و آن را با گزینه‌های پیش فرض و بدون تغییر آن Run می‌کنیم. پیش از بررسی داده‌های مفقوده در پنجره گشوده شده، از آنجا که در نظر داریم با توجه به نوع توزیع متغیرها (که به دلیل مدیریت داده‌های پرت در مرحله قبل، احتمال تغییر در آن وجود دارد) روش‌های متفاوتی را در مدیریت داده‌های بکار گیریم، از Sim Fit استفاده می‌کنیم. با انجام این کار در می‌یابیم دو متغیر BloodPressure و BMI دارای توزیع نرمال هستند. حال مجدد به گزارش ارائه شده توسط Data Adit مراجعه می‌کنیم. در قسمت Complete % می‌توانیم درصد تکمیل بودن هر متغیر را مشاهده کنیم. برای متغیرهایی که دارای داده مفقوده هستند (درصد تکمیل کمتر از ۱۰۰ دارند) در قسمت Impute Missing گزینه Black &

[15] Missing Values

Null Value را انتخاب می‌کنیم. در واقع با این کار می‌خواهیم برای سیستم تعیین کنیم چه نوع داده مفقوده‌ای را می‌خواهیم مدیریت کنیم. سپس در قسمت Method تعیین می‌کنیم از چه روشی برای پر کردن داده‌های مفقوده استفاده نماید. برای دو متغیر BloodPressure و BMI از روش Random و از نوع Normal استفاده می‌کنیم. این روش با در نظر گرفتن میانگین و انحراف معیاری که از داده‌های آن متغیر محاسبه می‌کند، مقادیر تصادفی تولید کرده و با مقدار خالی جایگزین می‌نماید. برای متغیرهای Glucose، Skin و Insulin از روش Random و از نوع Uniform استفاده می‌شود. این روش با در نظر گرفتن کمترین و بیشترین مقدار که از داده‌های آن متغیر محاسبه می‌کند، مقادیر تصادفی تولید کرده و با مقدار خالی جایگزین می‌نماید. پس از این‌کار باید از قسمت Generate گزینه Missing Values SuperNode را انتخاب کنیم تا سوپر نود جدیدی ایجاد نماید.

نکته۵: همانند مرحله مدیریت داده‌های پرت پس از ایجاد این سوپرنودها مجددا نوع داده‌ها خوانده و تعیین می‌گردد، لذا از Type (از قسمت Field Ops) برای این کار استفاده می‌گردد که در اینجا به عنوان مولفه Read Values نام گذاری شده است.

مولفه Data Scaling: در الگوریتم‌های طبقه بندی مبتنی بر فاصله الزامیست داده‌ها هم مقیاس گردند و در سایر الگوریتم‌ها اگرچه الزامی نیست لیکن بهتر است هم مقیاس سازی انجام پذیرد. برای این منظور از Auto Data Prep که در قسمت Field Ops قابل دسترس است، بهره گیری می‌نماییم. اگرچه Auto Data Prep صرفا به هم مقیاس نمودن داده‌ها محدود نمی‌گردد و کاربردهای بیشتری دارد، اما به دلیل اینکه هر نوع آماده سازی داده‌ها را در مراحل قبل انجام داده‌ایم، در این مرحله تنها به هم مقیاس سازی داده‌ها می‌پردازیم. بنابراین سایر قسمت‌ها را غیرفعال نموده و تنها به سراغ قسمت Transform Continuous Field از بخش Prepare Inputs & Target از برگ نشان Settings می‌رویم. در اینجا دو روش هم مقیاس سازی در نظر گرفته شده است، Min/Max transformation که نیاز به مقادیر Minimum و Maximum دارد و دیگری z-score transformation است که نیاز به وارد کردن دو مقدار Final mean و Final Standard deviation دارد. ما از روش Min/Max استفاده می‌کنیم و بازه آن را بین ۰ تا ۱۰۰ در نظر می‌گیریم. سپس گزینه Analyze Data را می‌زنیم تا داده‌ها را آماده سازی کند، با این کار آیکن آن با تیک آبی نشان داده می‌شود.

مولفه Balance Checking: تا اینجا کار آماده سازی داده‌ها به اتمام رسیده است و باید به سراغ مدلسازی رفت. اما قبل از انجام مدلسازی متوازن بودن داده‌ها[16] را بررسی می‌نماییم و تا در صورت نیاز دیتاست را متوازن نماییم. برای این کار از Distribution که از قسمت Graphs قابل دسترس است، استفاده می‌نماییم. با انتخاب متغیر

[16] Data Balancing

Outcome در قسمت Field به عنوان متغیر هدف و کلیک بر روی Run، تعداد و درصد هر یک از مقادیر هدف را نشان داده می‌شود که به ترتیب برای مقادیر ۰ و ۱، تعداد ۵۰۰ و ۲۶۸ و با درصد ۶۵/۱ و ۳۴/۹ می‌باشد. این اعداد نشان دهنده این هستند که تعداد نمونه‌های دارای دیابت کمتر از نمونه‌های دیگر است. جهت متوازن سازی آن با دو روش رایج Over Sampling (افزودن داده‌های دارای متغیر هدف کمتر) و Under Sampling (حذف داده‌های دارای متغیر هدف بیشتر) می‌توانیم به ترتیب از گزینه‌های Balace Node (boost) و Balance Node (reduce) در قسمت Generate استفاده کرد که به ترتیب نودهای Over Sampling و Under Sampling را استفاده می‌نماییم.

نکته ۶: با آزمودن مدل‌ها در هر سه روش داده‌های اصلی، داده‌های Over Sample و داده‌های Under Sample، به مدل‌های بهتری در روش دوم رسیدیم و همچنین به دلیل اندک بودن داده‌های آموزشی و حذفی بخشی از داده‌ها در روش Under Sampling، در اینجا از داده‌های بدست آمده از روش Over Sampling بهره‌گیری نمودیم.

مدلسازی[17]:

مولفه Partitioning 80/20: پیش از آنکه سراغ مدلسازی و بهره گیری از الگوریتم‌های یادگیری ماشین برویم، نیاز است تا داده‌ها را به دو دسته آموزشی[18] و آزمایشی[19] تقسیم کنیم. این کار به ما در ارزیابی مدل کمک می‌کند و می‌توانیم با تشکیل ماتریس اغتشاشات[20] به محاسبه شاخص‌های ارزیابی مانند صحت[21] بپردازیم. جداسازی داده‌ها اغلب به روش Holdout و یا Cross-Validation انجام می‌پذیرد. در روش اول داده‌ها را به دو دسته آموزشی و آزمایشی تقسیم شده و این تقسیم بندی اغلب به این صورت است که مدل از روی ۸۰ درصد داده‌ها آموزش دیده و با ۲۰ درصد داده‌های باقی‌مانده ارزیابی می‌گردد. این عمل اینگونه انجام می‌پذیرد که ۲۰ درصد داده‌ها بدون برچسب شده و به مدل طراحی شده سپرده می‌شود تا تشخیص دهد به آن نمونه چه برچسبی اعطا نماید و در نهایت آن را با متغیر هدف آن نمونه چک می‌کنند تا مشاهده نمایند مدل به درستی پیش بینی کرده است یا خیر. چالش این روش این است که طراحی مدل تنها بر اساس ۸۰ درصد داده‌هاست و تمامی داده‌ها در آموزش شرکت نمی‌کنند و این کار ریسک طراحی بهترین مدل را از بین می‌برد. بنا بر این اغلب پس از ارزیابی مدل و کسب نتیجه مطلوب، سعی می‌شود مدل مجددا با کل داده‌ها آموزش داده شود. این عمل در مدل Cross-Validation در چند مرحله (k) به تعداد k دیتاست ایجاد کرده به طوری که در هر یک قانون ۸۰-۲۰ رعایت می‌شود و هیچ بخشی از داده‌ها دوبار به عنوان داده تست انتخاب نمی‌شوند. از آنجا که روش دوم بسیار زمانبر است، در اینجا از روش Holdout استفاده شده است. برای این کار از Partition که از قسمت Field Ops قابل دسترسی است، بهره‌گیری شده است. تنها کافی است در قسمت Partitions سایز دو بخش Training partition size و Testing partition size را مشخص کرده که به تزتیب ۸۰ و ۲۰ می‌گذاریم و OK را کلیک می‌کنیم. شایان ذکر است در قسمت Seed عددی نوشته شده که نشانگر نحوه انتخاب نمونه‌هاست و با کلیک بر بروی Generate می‌توان آن را تغییر داد و دسته‌های جداگانه‌ای از داده بدست آورد. این کار را اغلب زمانی انجام می‌دهیم که مدل ما مطلوب به نظر می‌رسد و می‌خواهیم اطمینان حاصل کنیم که در تمامی سناریوها به طور مناسبی پیش بینی می‌کند و جهت اطمینان از مدل با تغییر Seed آن را می‌آزماییم.

مولفه Type for other approaches: برای الگوریتم‌هایی که مبتنی بر فاصله نیستند نوع داده‌ها را مجددا بررسی کرده و دو متغیر Pregnancies و Age را از نوع Nominal دسته بندی کردیم. این کار برای الگوریتم‌های مبتنی بر فاصله انجام نشد و نوع این دو متغیر Continuous باقی ماند. الگورتیم‌های مبتنی بر فاصله بررسی شده در

[17] Modeling
[18] Training Set
[19] Testing Set
[20] Confusion Matrix
[21] Accuracy

اینجا عبارتند از KNN، Neural Network و SVM و همچنین سایر الگوریتم‌هایی که در اینجا مورد بررسی قرار می‌گیرند شامل CART، QUEST، CHAID، C5 و Random Forest می‌باشند که در ادامه به تفسیر آن می‌پردازیم.

نکته ۷: در تمام مدل‌هایی که در ذیل به شرح جزئیات آن می‌پردازیم، از برگ نشان Annotations به تغییر نام آن مبادرت می‌نماییم تا اسم آن مشخص گردد.

نکته ۸: در بیان شرح انجام کار صرفا مقادیر یافته از حالت پیش فرض مدل در توضیحات آمده است و شرحی از مقادیر پیش فرض در ذیل نیامده است.

مولفه KNN:

الگوریتم k نزدیک ترین همسایه[۲۲] یکی از الگوریتم‌های ساده یادگیری ماشین[۲۳] با ناظر[۲۴] می‌باشد که در مسائل طبقه بندی[۲۵] مورد بهره‌برداری قرار می‌گیرد. این روش که از مدل‌های مبتنی بر حافظه[۲۶] می‌باشد، نزدیک ترین همسایه را پیدا و با اکثریت آرا نزدیک‌ترین همسایگان کلاس را پیش‌بینی می‌کند. برای یافتن k نزدیک‌ترین همسایه از تکنیک‌های مختلفی نظیر محاسبه فاصله‌ی اقلیدسی[۲۷]، فاصله‌ی منهتن[۲۸] و یا فاصله‌ی مینکوفسکی[۲۹] استفاده می‌شود.

برای ایجاد این مدل از KNN که در قسمت Modeling و برگ نشان Classification در دسترس است، استفاده کرده و تغییرات ذیل را برای ساختن مدل خود اعمال می‌نماییم:

- برگ نشان Objectives: در این قسمت هدف را بر روی Custom analysis قرار می‌دهیم.

- برگ نشان Settings: در قسمت Neighbors گزینه Automatically select k را در بازه ۲ و ۳۰ قرار می‌دهیم تا برای K همسایه تعیین شده بررسی گردد و بهترین آن در نتیجه نشان داده شود. سپس گزینه

[22] K-Nearest Neighbors
[23] Machine Learning
[24] Supervised
[25] Classification
[26] Instance-based learning
[27] Euclidean distance
[28] Manhattan distance
[29] Minkowski distance

Weight features by importance when computing distances را تیک می‌زنیم تا به هر یک از متغیرها بر اساس درجه اهمیت آن وزنی دهد.

نکته ۹: این الگوریتم از قابلیت Cross-Validation پشتیبانی می‌کند و تعداد fold ها ۱۰ در نظر گرفته شده است. این کار سبب ایجاد ده دسته آموزشی گردد تا با حداکثر بهره‌گیری از داده‌ها، عمل آموزش و آزمایش به منظور کشف مدل بهتر انجام پذیرد.

مولفه Neural Network:

شبکه عصبی مصنوعی[۳۰] که به اختصار به آن شبکه عصبی گوین، بهره‌گیری از شیوه کارکرد سیستم عصبی زیستی برای پردازش داده‌ها به منظور یادگیری است. شبکه عصبی شامل لایه‌ها، وزن‌ها و نورون‌ها می‌باشد. هر شبکه عصبی شامل یک لایه ورودی با نورون‌هایی که نشان دهنده متغیرهای ورودی، یک یا چند لایه پنهان با نورون‌هایی که وظیفه استخراج ویژگی‌های مناسب را به عهده دارند و یک لایه خروجی با نورون‌هایی که نشان دهنده متغیر(های) هدف می‌باشد.

برای ایجاد این مدل از Neural Net که در قسمت Modeling و برگ نشان Classification در دسترس است، استفاده کرده و تغییرات ذیل را برای ساختن مدل خود اعمال می‌نماییم:

- برگ نشان Build Options:

- در قسمت Objectives پس از تست حالت‌های Standard model، Boosting و Bagging، گزینه Enhance model stability را انتخاب می‌کنیم تا مدل از طریق روش bagging، خود را بهبود دهد. تکنیک bagging که با نام bootstrap aggregating هم شناخته می‌شود یکی از روش‌های یادگیری تجمیعی[۳۱] است، که برای حداقل کردن واریانس مدل استفاده می شود. در تکنیک bagging برای آموزش هر مدل، یک بخشی از داده به صورت تصادفی انتخاب می شود و در پروسه تصمیم گیری، نظر مدل‌ها باهم ترکیب می‌شود.
- در قسمت Basics مدل شبکه عصبی را Multilayer Perceptron (MLP) در نظر گرفته و تنها به یک لایه پنهان با تعداد ۵ نورون اکتفا نمودیم.
- در قسمت Stopping Rules هیچ گونه محدودیتی در ساخت مدل اعمال نکردیم.

[30] Artificial Neural Network (ANN)
[31] Ensemble Learning

نکته ۱۰: از آنجا که روش Bagging را انتخاب کردیم در قسمت Ensembles باید تعدادBag ها و روش انتخاب نتیجه را معین نمود که آن را به حالت پیش فرض رای گیری از ۱۰ Bag قرار دادیم.

مولفه SVM[32]:

ماشین بردار پشتیبان یکی از الگوریتم‌های نظارت‌شده یادگیری ماشین است که برای طبقه بندی استفاده می‌شود. این الگوریتم نمونه‌ی داده‌هایی را به‌صورتی نقاطی در فضا نشان داده شده است، با استفاده از یک خط یا هایپرپلین[۳۳]، از هم جدا می‌کند. این جداسازی به‌گونه‌ای است که نقاط داده‌ای که در یک طرف خط هستند مشابه به هم و در یک گروه قرار می‌گیرند.

برای ایجاد این مدل از SVM که در قسمت Modeling و برگ نشان Classification در دسترس است، استفاده کرده و تغییرات ذیل را برای ساختن مدل خود اعمال می‌نماییم:

- برگ نشان Expert: در این قسمت مدل را از نوع Expert انتخاب کرده و مقدار C را ۱۰ و گاما را ۱ تعیین نمودیم. لازم به ذکر است که نوع kernel را RBF انتخاب کردیم.

مولفهCART، QUEST و CHAID:

هر چهار مدل CART، Quest، Chaid و C5 از نوع درخت تصمیم[۳۴] هستند. این الگوریتم از پرکاربردترین روش‌های داده کاوری است و در طبقه بندی استفاده می‌گردد. در ساختار درخت تصمیم، نتایج حاصله از درخت در قالب یک سری قواعد توضیح داده می‌شود. هر مسیر از ریشه تا یک برگ درخت تصمیم، یک قانون را بیان می‌کند و در نهایت برگ با کلاسی که بیشترین مقدار رکورد در آن تعلق گرفته برچسب می‌خورد. هنگامی که یک درخت تصمیم ساخته می‌شود، تعدادی از شاخه‌ها ناهنجاری‌هایی در داده‌های آموزش منعکس می‌کنند که ناشی از داده‌های پرت و یا نویز است که اغلب برای رفع مشکل از هرس کردن استفاده می‌شود.

از آنجا که سه مدل CART، Quest و Chaid دارای تنظیمات مشابه هستند، توضیحات آن در این قسمت بصورت مشترک آمده است و مدل C5 بصورت جداگانه توضیح داده شده است. برای ایجاد مدل‌ها به ترتیب از C&R Tree، Quest و CHAID که در قسمت Modeling و برگ نشان Classification در دسترس است، استفاده کرده و تغییرات ذیل را برای ساختن آن‌ها اعمال می‌نماییم:

[32] Support Vector Machines
[33] Hyperplane
[34] Decision Tree

- برگ نشان Build Options:

- در قسمت Objectives پس از تست حالت‌های Standard model، Boosting و Bagging، گزینه Enhance model accuracy را انتخاب می‌کنیم تا مدل از طریق روش boosting، خود را بهبود دهد. تکنیک boosting یکی دیگر از روش‌های یادگیری تجمیعی است و برای بهبود دقت یادگیری توسط یک پروسه تکرارشونده بکار برده می‌شود. این الگوریتم از کل مجموعه داده به منظور آموزش هر دسته‌کننده استفاده می‌کند، اما بعد از هر بار آموزش، بیشتر بر روی داده‌های سخت تمرکز می‌کند تا به درستی کلاسه بندی شوند. این روش تکراری تغییر انطباقی به توزیع داده ها آموزش با تمرکز بیشتر بر روی نمونه‌هایی است که قبلا بطور صحیح دسته بندی نشده‌اند.
- در قسمت Basics مقدار بیشترین عمق درخت[۳۵] را برای CART مقدار ۵ و برای دو روش QUEST و CHAID مقدار ۴ تعیین می‌نماییم.
- در قسمت Stopping Rules تعیین می‌کنیم تعداد نمونه‌ها در گره والد[۳۶] حداقل ۴ درصد کل نمونه‌ها و در گره فرزند[۳۷] حداقل ۲ درصد کل نمونه‌ها باشد. پر واضح است که مدل با گره‌های کمتر از این مقدار ایجاد نمی‌کند.

نکته ۱۰: از آنجا که روش Boosting را انتخاب کردیم در قسمت Ensembles باید تعداد آزمون و روش انتخاب نتیجه را معین نمود که آن را به حالت پیش فرض رای گیری از ۱۰ آزمون قرار دادیم.

مولفه C5:

همانطور که توضیح داده شد، C5 یکی از انواع درخت تصمیم می‌باشد. برای ایجاد این مدل از C5 که در قسمت Modeling و برگ نشان Classification در دسترس است، استفاده کرده و تغییرات ذیل را برای ساختن مدل خود اعمال می‌نماییم:

- برگ نشان Model: نوع خروجی را بر روی Decision tree گذاشته و Mode را به Expert تغییر می‌دهیم و سپس مقدار ۹۵ برای هرس[۳۸] در نظر می‌گیریم. با تیک زدن گزینه Use boosting تعیین می‌نماییم که از این روش برای بهبود مدل بهره‌گیری شود و تعداد آزمون‌ها را برای ۱۰ در نظر می‌گیریم.

[35] Maximum Tree Depth
[36] Parent branch
[37] Child branch
[38] Pruning severity

مولفه Random Forest:

الگوریتم جنگل تصادفی یکی از الگوریتم‌های تجمیعی استکه در آن مدل نهایی از تجمیع تعداد زیادی درخت تصمیم ساخته می‌گردد. به بیان ساده، جنگل تصادفی چندین درخت تصمیم ساخته و آن‌ها را با یکدیگر ادغام می‌کند تا پیش‌بینی‌های صحیح‌تر و پایدارتری حاصل شوند.

برای ایجاد مدل از Random Trees که در قسمت Modeling و برگ نشان Classification در دسترس است، استفاده می‌کنیم. در اینجا هیچ تغییری انجام نداده و بر اساس موارد پیش فرض، تعداد ۱۰۰ مدل با حداکثر عمق ۱۰ ایجاد کرده و از بین آن بهترین مدل را به خروجی ارسال می‌کند.

مولفه Find Better Classifier:

جهت بررسی و مقایسه چند مدل و یافتن بهترین مدل، می‌توان از Auto Classifier که از قسمت Analytic Server قابل دسترسی است، استفاده نمود. در این ماژول امکان انتخاب روش‌های طبقه بندی مختلف به همراه انتخاب پارامترهای مختلف وجود دارد. در اینجا تمامی مدل‌های فوق الذکر را انتخاب نموده و پارامترهای گوناگون را تعیین نمودیم و معین گردیده ۶ روش با میزان صحت بالاتر در داده آزمایش را به عنوان نتیجه نشان دهد. مدل C5 به عنوان بهترین روش انتخاب شده است که صحت ۸۴ درصد برای داده‌های آزمایشی را به ثبت رسانده است. از آنجا که این مدل به نظر بیش برازش[۳۹] می‌باشد، به این اکتفا نکرده و مدل‌های ساخته شده توسط خودمان را در مرحله ارزیابی مورد بازبینی قرار داده‌ایم.

[39] Over Fitting

ارزیابی[40]:

مولفه Analysis Test: برای ارزیابی مدل‌های طبقه بندی که در مراحل قبل ایجاد شد، از معیار Accuracy استفاده می‌کنیم. این معیار در واقع درصد میزان تشخیص صحیح مدل را اعلام می‌نماید. با استفاده از Analysis که از قسمت Output قابل دسترسی است، می‌توانیم با کلیک بر روی Run نتایج را مشاهده نماییم. در پنجره ظاهر شده دو درصد در قسمت Correct به تفکیک Training و Testing لحاظ گردیده است که با مشاهده مقادیر آن و بررسی اختلاف آن‌ها بصورت نسبی مطلوب بودن یا نبودن مدل را ارزیابی می‌نماییم که در ذیل بر اساس هر کدام از مدل‌های ساخته شده در مراحل قبل، ارزیابی حاصله شرح داده شده است.

ارزیابی مدل‌ها:

جهت ارزیابی نهایی، مدل‌ها را با ۸ دسته[41] مختلف با داده‌های جداگانه آموزش و تست بررسی نمودیم که نتایج آن به تفکیک در ذیل آمده است.

مدل KNN:

KNN	Training	Test	Diff
Seed 1	76.91	81.25	4.34
Seed 2	80.05	80	0.05
Seed 3	80.22	84.27	4.05
Seed 4	79.44	79.82	0.38
Seed 5	80.59	80.56	0.03
Seed 6	79.23	81.02	1.79
Seed 7	79.46	75.13	4.33
Seed 8	79.93	81.77	1.84
AVG	80.4775		

جدول شماره ۲: ارزیابی مدل KNN

[40] Evaluation
[41] Seed

مدل Neural Network:

NN	Training	Test	Diff
Seed 1	78.92	84.09	5.17
Seed 2	80.37	81.14	0.77
Seed 3	80.1	78.09	2.01
Seed 4	80.69	80.09	0.6
Seed 5	83.33	81.48	1.85
Seed 6	81.05	81.02	0.03
Seed 7	81.51	79.37	2.14
Seed 8	80.62	84.9	4.28
AVG	81.2725		

جدول شماره ۳: ارزیابی مدل NN

مدل SVM:

SVM	Training	Test	Diff
Seed 1	79.61	80.34	0.73
Seed 2	79.95	82.95	3
Seed 3	81.29	77.53	3.76
Seed 4	80.76	80.37	0.39
Seed 5	82.46	78.34	4.12
Seed 6	81.51	79.63	1.88
Seed 7	79.93	82.98	3.05
Seed 8	80.62	81.25	0.63
AVG	80.42375		

جدول شماره ۴: ارزیابی مدل SVM

مدل CART:

CART	Training	Test	Diff
Seed 1	79.39	78.74	0.65
Seed 2	79.98	80.9	0.92
Seed 3	80.49	77.27	3.22
Seed 4	78.89	79.53	0.64
Seed 5	79.39	85.25	5.86
Seed 6	77.69	80.09	2.4
Seed 7	78.24	85.86	7.62
Seed 8	79.75	80.95	1.2
AVG	81.07375		

جدول شماره ۵: ارزیابی مدل CART

مدل QUEST:

QUEST	Training	Test	Diff
Seed 1	77.7	80.34	2.64
Seed 2	77.49	80.11	2.62
Seed 3	79.42	76.67	2.75
Seed 4	76.88	81.19	4.31
Seed 5	78.51	77.31	1.2
Seed 6	77.84	73.02	4.82
Seed 7	78.15	76.04	2.11
Seed 8	77.45	79.89	2.44
AVG	78.07125		

جدول شماره ۶: ارزیابی مدل QUEST

مدل CHAID:

CHAID	Training	Test	Diff
Seed 1	81.97	86.93	4.96
Seed 2	81.17	86.36	5.19
Seed 3	83.93	82.86	1.07
Seed 4	83.44	84.09	0.65
Seed 5	84.02	83.41	0.61
Seed 6	82.91	84.02	1.11
Seed 7	82.24	86.98	4.74
Seed 8	81.83	82.98	1.15
AVG	84.70375		

جدول شماره ۷: ارزیابی مدل CHAID

مدل C5:

C5	Training	Test	Diff
Seed 1	82.56	86.93	4.37
Seed 2	81.96	81.03	0.93
Seed 3	82.1	88.64	6.54
Seed 4	82.86	84.26	1.4
Seed 5	84.12	81.02	3.1
Seed 6	82.71	86.57	3.86
Seed 7	83.42	87.77	4.35
Seed 8	82.55	88.89	6.34
AVG	85.63875		

جدول شماره ۸: ارزیابی مدل C5

مدل Random Forest:

Random Forest	Training	Test	Diff
Seed 1	88.58	85.8	2.78
Seed 2	86.13	86.93	0.8
Seed 3	89.54	85.8	3.74
Seed 4	87.21	87.56	0.35
Seed 5	88.25	89.91	1.66
Seed 6	88.39	82.87	5.52
Seed 7	88.31	89.95	1.64
Seed 8	87.42	87.5	0.08
AVG	87.04		

جدول شماره ۹: ارزیابی مدل Random Forest

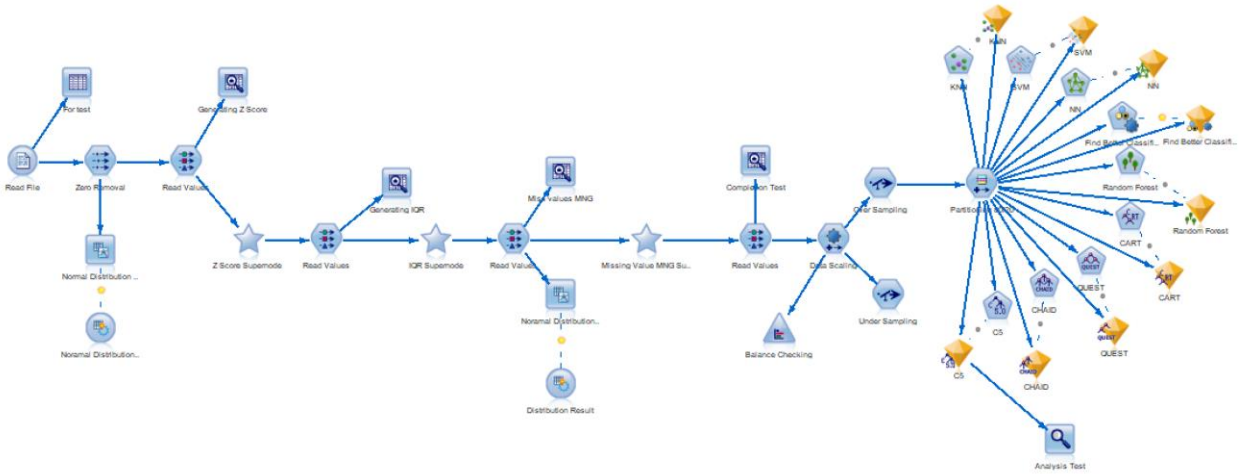
ارزیابی‌ها نشان می‌دهد مدل‌ها به ترتیب ذیل از بالا به پایین دارای بهترین نتایج بوده‌اند.

Random Forest	87.04
C5	85.63875
CHAID	84.70375
NN	81.2725
CART	81.07375
KNN	80.4775
SVM	80.42375

جدول شماره ۱۰: ارزیابی تجمیعی کلیه مدل‌ها بر اساس مقدار میانگین آن‌ها

بیشترین صحت کسب شده متعلق به الگوریتم Random Forest با مقدار ۸۹.۹۵ درصد و کمترین صحت کسب شده متعلق به الگوریتم QUEST با مقدار ۷۳.۰۲ درصد می‌باشد.

در ذیل نمایی کلی از پیاده سازی مدل‌های فوق در ابزار IBM SPSS Modeler نشان داده شده است.



شکل شماره ۱: نمایی از پیاده سازی پروژه در ابزار IBM SPSS Modeler