به نام خدا

پروژه پایانی درس داده کاوی

نیم سال اول سال تحصیلی ۱۴۰۳-۱۴۰۴

دانشگاه تربیت دبیر شهید رجایی

فاطمه کوثر

۴۰۰۱۲۳۱۰۹۷

فهرست

مقدمه و هدف پروژه ۳

داده‌های ورودی و پیش پردازش داده‌ها..... ۳

آنالیز داده‌ها ۳

پیش پردازش داده‌ها ۷

مدیریت مقادیر گمشده ۷

مدیریت داده‌های پرت..... ۸

انکد کردن ۸

نرمال سازی داده‌ها ۸

الگوریتم‌های انتخابی و پیاده‌سازی..... ۹

طبقه بندی ۹

مدل درخت تصمیم..... ۹

مدل SVM..... ۱۰

مقایسه‌ی مدل‌ها ۱۱

خوشه بندی ۱۱

انتخاب ویژگی ۱۱

K-means..... ۱۲

خوشه بندی سلسله مراتبی..... ۱۳

مقایسه مدل‌ها..... ۱۶

نتیجه گیری..... ۱۶

مقدمه و هدف پروژه

در این پروژه قصد داریم یک دیتاست حاوی اطلاعات بیماران و نتیجه‌ی تست دیابت را ابتدا پیش پردازش کرده و تمیز کنیم سپس با استفاده از الگوریتم‌های انتخابی بررسی و طبقه بندی و دسته بندی کنیم. سپس نتیجه‌ی نهایی هر یک از این الگوریتم‌های را بررسی کرده و نحوه‌ی عملکرد آن‌ها را مقایسه و تحلیل می‌کنیم. در نهایت با بیان یک نتیجه‌ی کلی، الگوریتم مناسب را انتخاب می‌کنیم. انجام طبقه‌بندی و خوشه‌بندی، برای مقایسه و تشخیص داده‌هایی که در آینده به عنوان ورودی‌های جدید وارد می‌شوند اهمیت دارد. با استفاده از یک مدل مناسب می‌توان بیماری دیابت را در سطوح اولیه تشخیص داده و از پیشروی بیشتر آن جلوگیری کرد.

همچنین کد این پروژه در مخزن گیتهاب به آدرس https://github.com/FmKosar/DataMining قرار گرفت.

داده‌های ورودی و پیش پردازش داده‌ها

آنالیز داده‌ها

دیتاست ورودی با نام modified_diabetes_prediction_dataset.csv به عنوان ورودی در نظر گرفته شده است. این دیتاست شامل 100001 رکورد و ۹ ویژگی است. از ۹ ویژگی این دیتاست دو تا از جنس object، سه تا از جنس float 64 و چهار تا از جنس int 64 هستند (تصویر ۱).

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100001 entries, 0 to 100000
Data columns (total 9 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   gender               100001 non-null  object
 1   age                  99999 non-null   float64
 2   hypertension         100001 non-null  int64
 3   heart_disease        100001 non-null  int64
 4   smoking_history      100000 non-null  object
 5   bmi                  100001 non-null  float64
 6   HbA1c_level          100000 non-null  float64
 7   blood_glucose_level  100001 non-null  int64
 8   diabetes             100001 non-null  int64
dtypes: float64(3), int64(4), object(2)
memory usage: 6.9+ MB
```

تصویر ۱: اطلاعات یک دیتاست

همچنین این دیتاست شامل چند مقدار گمشده است که باید در مراحل پیش پردازش مدیریت شوند (تصویر ۲). در این دیتاست ستون age، دو مقدار گمشده، ستون smoking_history یک مقدار گمشده و ستون HbA1c_level یک مقدار گمشده دارند.

```
gender               0
age                  2
hypertension         0
heart_disease        0
smoking_history      1
bmi                  0
HbA1c_level          1
blood_glucose_level  0
diabetes             0
dtype: int64
```

تصویر ۲: داده‌های گمشده‌ی دیتاست

همچنین برای مشخص شدن ورودی‌هایی که از لحاظ منطق با ستون مورد نظر خوانایی ندارند، برای هر ستون مقادیر و تعداد تکرارشان را بررسی کردیم. در تصویر ۳ مشخص می‌شود که یک مقدار در جنسیت به عنوان unknown قرار دارد که داده‌ی گمشده محسوب می‌شود و باید مدیریت شود. همچنین طبق این خروجی، در ستون سن مقادیر منفی داریم که برای سن مقدار معتبری نیست.

```
Female     58552
Male       41430
Other         18
unknown        1
Name: gender, dtype: int64
 48.00    1591
 50.00    1586
 52.00    1568
 51.00    1560
 54.00    1543
          ...
 5.88        4
-0.60        4
-4.84        3
 0.40        2
-4.92        1
Name: age, Length: 222, dtype: int64
```

تصویر ۳: مقادیر ستون‌های جنسیت و سن

همچنین در تصویر ۴ مشخص است که در بین مقادیر معمول ستون سابقه‌ی سیگاری بودن، یک مقدار غیرمعمول (yes) وجود دارد. همچنین بیشترین مقدار این ستون متعلق به رکورد No Info است که اطلاعات مفیدی به ما نمی‌دهد و می‌توان گفت تاثیری در تحلیل ما ندارد و معادل مقدار پوچ است. به دلیل اینکه تعداد زیادی از این ستون به همین مقدار مربوط است، این ستون را از دیتاست حذف خواهیم کرد. همچنین با بررسی مقادیر bmi، می‌بینیم که یک مقدار ۱۰۱ داریم که بسیار پرت است. (البته مقادیر پرت با رسم نمودار جعبه‌ای بسیار بهتر مشخص خواهند شد.) این مقدارهای پرت که از لحاظ پزشکی هم بسیار نادر و پرت هستند را در قسمت پیش پردازش داده‌ها مدیریت خواهیم کرد.

```
0    92516
1     7485
Name: hypertension, dtype: int64
0    96059
1     3942
Name: heart_disease, dtype: int64
No Info        35817
never          35094
former          9352
current         9285
not current     6447
ever            4004
yes                1
Name: smoking_history, dtype: int64
101.665015    1
26.628701     1
36.683750     1
20.367588     1
31.288412     1
             ..
26.142655     1
18.256319     1
31.582976     1
23.840415     1
25.698752     1
Name: bmi, Length: 100001, dtype: int64
```

تصویر ۴: مقادیر ستون‌های hypertension، سابقه‌ی بیماری قلبی، سابقه‌ی سیگار کشیدن و bmi

همچنین در تصویر ۵ مشخص است که مقدار ۹۹۹۹ یک داده‌ی پرت است که برای سطح قند خون در علم پزشکی یک مقدار بسیار بالاست و باید در بخش پیش پردازش داده‌ها مدیریت شود.

```
130     7794
159     7759
140     7732
160     7712
126     7702
145     7679
200     7600
155     7575
90      7112
80      7107
100     7025
158     7025
85      6901
280      729
300      674
240      636
260      635
220      603
9999       1
Name: blood_glucose_level, dtype: int64
```

تصویر ۵: مقادیر ستون سطح قند خون

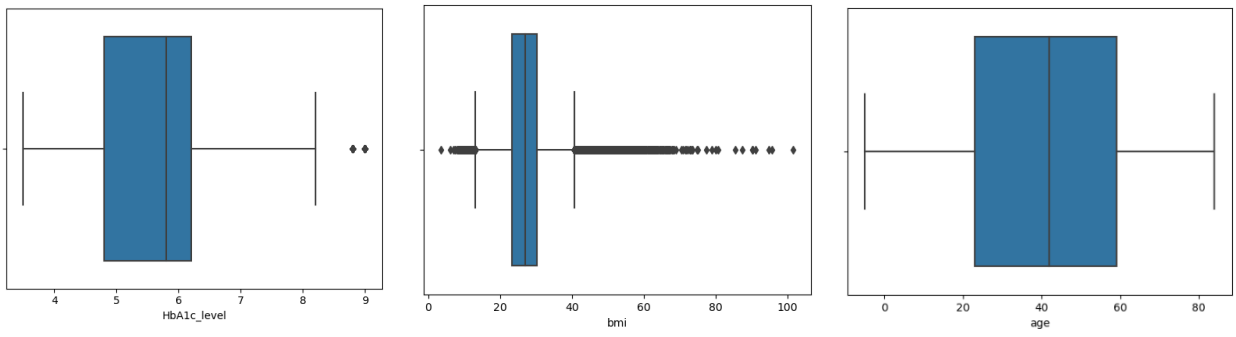
در قدم بعد داده‌ها را از نظر آماری بررسی می‌کنیم (تصویر ۶).

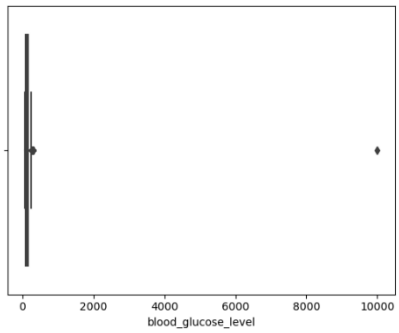
	gender	smoking_history
count	100001	100000
unique	4	7
top	Female	No Info
freq	58552	35817

	age	hypertension	heart_disease	bmi	HbA1c_level	blood_glucose_level	diabetes
count	60695.000000	60696.000000	60696.000000	60696.000000	60694.000000	60695.000000	60695.000000
mean	41.351717	0.074651	0.038932	27.325967	5.528705	138.162320	0.086811
std	22.660966	0.262829	0.193434	6.958973	1.071839	57.253239	0.281560
min	-4.920000	0.000000	0.000000	2.000000	3.500000	80.000000	0.000000
25%	23.000000	0.000000	0.000000	23.397091	4.800000	100.000000	0.000000
50%	42.000000	0.000000	0.000000	26.960196	5.800000	140.000000	0.000000
75%	59.000000	0.000000	0.000000	30.272423	6.200000	159.000000	0.000000
max	84.000000	1.000000	1.000000	101.665015	9.000000	9999.000000	1.000000

تصویر ۶: بررسی داده‌ها از نظر آماری

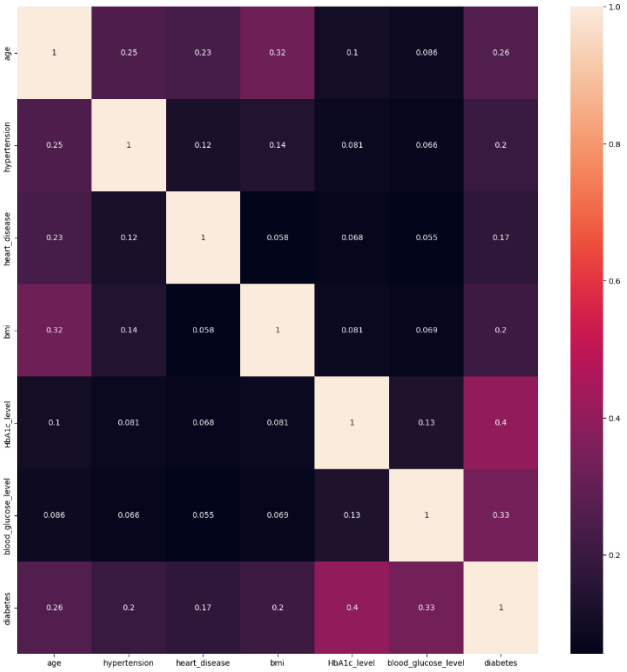
برای پیش پردازش داده‌ها نیاز داریم داده‌های پرت را مدیریت کنیم. برای تشخیص و تصمیم گیری درباره‌ی این داده‌های پرت نیاز داریم نمودار جعبه‌ای را برای ستون‌های عددی و غیربااینری رسم کنیم (تصویر ۷).





تصویر ۷: نمودارهای جعبه‌ای ستون‌های عددی

برای تصمیم‌گیری درباره‌ی داده‌های پرت لازم است ارتباط میان ویژگی‌ها را در نظر بگیریم. به همین دلیل ماتریس وابستگی را مطابق شکل زیر رسم کردیم (تصویر ۸). طبق این ماتریس بیشترین همبستگی با هدف، ۰.۴ است که متعلق به HbA1c_level است. دومین سطح متعلق به سطح قند خون است. پس وقتی داده‌های پرت این ستون‌ها را مدیریت می‌کنیم لازم است دقت بیشتری داشته باشیم.



تصویر ۸: ماتریس وابستگی

پیش پردازش داده‌ها

این مرحله شامل مراحل گوناگونی است که با توجه به نوع دیتاست و هدف قابل تغییر هستند. برای مثال طبق بررسی‌های ما این دیتاست هیچ داده‌ی تکراری‌ای ندارد پس ما نیازی به انجام مرحله‌ی حذف داده‌ی تکراری نداریم.

مدیریت مقادیر گمشده

در این مرحله ابتدا داده‌های گمشده را مدیریت کردیم. همان طور که توضیح دادیم ستون سابقه‌ی سیگار کشیدن به دلیل اینکه در تعداد زیادی از رکوردها با مقدار No Info مقداردهی شده است، اطلاعات مفیدی در اختیار ما نمی‌گذارد به همین دلیل این ستون را حذف می‌کنیم.

همان طور که بررسی کردیم از بین ستون‌ها، سن و HbA1c_level دارای مقادیر گمشده هستند. این مقادیر را با استفاده از KNN Imputer مقداردهی کردیم و همچنین مقدار unknown در ستون جنسیت که به نوعی گمشده محسوب می‌شود، مُد ستون را جایگزین می‌کنیم. در اینجا به دلیل اینکه تعداد مقادیر گمشده بسیار کم بود ترجیح دادیم با روش‌های موجود آن‌ها را جایگزین کنیم و حذف نکنیم.

در نهایت با بررسی دوباره می‌بینیم که مقادیر گمشده از بین رفته‌اند.

مدیریت داده‌های پرت

مدیریت داده‌های پرت در بحث پیش‌پردازش داده‌ها بسیار مهم است چرا که در برخی الگوریتم‌های طبقه بندی و دسته بندی این داده‌های پرت باعث خطای الگوریتم می‌شوند.

برای مدیریت داده‌های پرت سه انتخاب داریم: رکوردهای دارای مقادیر پرت را حذف کنیم، مقادیر پرت را جایگزین کنیم (با استفاده از میانگین، مد یا رگرسیون) یا مقادیری که در نمودار جعبه‌ای به عنوان مقادیر پرت شناخته شده‌اند را به عنوان مقادیر معتبر بشناسیم. هر یک از این تصمیمات بستگی به دیتاست و مقادیر دارد.

درست است که در نمودار جعبه‌ای ستون سن هیچ مقدار پرتی نمی‌بینیم ولی می‌دانیم مقادیر منفی برای ویژگی سن بی‌معنا هستند به همین دلیل تمام رکوردهایی که مقدار منفی داشتند را حذف کردیم. حجم این دیتاست بسیار بزرگ است و این مقادیر کم هستند پس به جای آموزش یک مدل رگرسیون و جایگزینی مقادیر منفی سن، تصمیم گرفتیم مقادیر را حذف کنیم.

طبق علم پزشکی، مقادیر کوچک‌تر از ۱۰ یا بزرگ‌تر از ۸۰ برای bmi در انسان بسیار نادر هستند و احتمال دارد داده‌ی پرت باشد پس این گونه رکوردها را از دیتاست حذف می‌کنیم.

همچنین میزان قند خون بیش از ۳۰۰ غیرطبیعی است و احتمال بالایی دارد که یک داده‌ی پرت باشد. پس این مقدار را نیز حذف کردیم.

انکد کردن

ستون جنسیت از جنس object است، نه عدد. برای مدیریت این گونه داده‌ها باید آن‌ها را انکد کنیم. دو روش برای این کار داریم: one-hot encoding و label encoding.

در روش one-hot encoding، به ازای هر مقدار ممکن یک ستون اضافه می‌شود که مشخص می‌کند کدام موارد این ویژگی را دارند. از این روش وقتی استفاده می‌شود که ترتیب خاصی برای مقادیر موجود یک ستون قائل نباشیم. مانند جنسیت.

در روش label encoding، به هر مقدار ممکن یک عدد نسبت داده می‌شود و وقتی استفاده می‌شود که ترتیب منطقی‌ای در این مقادیر وجود داشته باشد. مانند سایز لباس (small، medium، large و...).

باید توجه داشت که از روش درستی در این قسمت استفاده کنیم زیرا اگر برای داده‌هایی که ترتیب خاصی ندارند از روش دوم استفاده کنیم در آموزش مدل‌ها طوری با این داده رفتار می‌شود که هر مقدار این ویژگی به مقدار دیگر برتری داشته باشد. پس در این قسمت که ترتیب مهم نیست و هیچ مقداری نسبت به مقدار دیگر برتری ندارد و به عبارتی جنسیت یک صفت اسمی است (نه ترتیبی) از روش اول برای انکد کردن استفاده کردیم.

نرمال سازی داده‌ها

ابتدا داده‌ها را به دو گروه X شامل ویژگی‌های مورد استفاده و y شامل نتیجه و هدف تقسیم کردیم.

نرمال سازی داده‌ها بسیار اهمیت دارد چرا که روی دقت الگوریتم‌های طبقه‌بندی (clustering) و سرعت همگرا شدن الگوریتم SVM بسیار تاثیر می‌گذارد. در این قسمت از standard scaler استفاده کردیم که به گونه‌ای داده‌ها را اسکیل می‌کند که میانگین آن‌ها ۱ و انحراف معیار صفر شود. همچنین دقت کنید که در این قسمت فقط داده‌های عددی را به عنوان داده‌هایی که باید اسکیل شوند استفاده می‌کنیم و ویژگی‌های باینری را دخیل نمی‌کنیم.

الگوریتم‌های انتخابی و پیاده‌سازی

طبقه بندی

در این بخش دو الگوریتم SVM و درخت تصمیم را پیاده سازی و ارزیابی کردیم. در هر دو الگوریتم داده‌ها به دو دسته‌ی آموزشی و تست تقسیم شدند. هر الگوریتم با داده‌های آموزشی، آموزش داده شد و با داده‌های تست ارزیابی شد. ارزیابی الگوریتم‌ها با محاسبه‌ی معیارهای دقت، دقت مثبت، حساسیت، امتیاز F1 و ماتریس گیج‌زنی انجام شد.

مدل درخت تصمیم

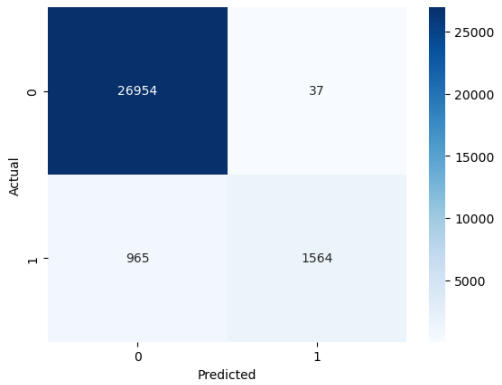
الگوریتم درخت تصمیم یک الگوریتم نظارت شده‌ی قابل تفسیر است که بر اساس ویژگی‌ها داده‌ها را به قسمت‌های مختلف تقسیم کرده و در نهایت در برگ‌ها برچسب را بررسی می‌کند. این مدل ممکن است دچار بیش برازش (overfit) شود. بیش برازش در این مدل به این صورت اتفاق می‌افتد که هر مسیر متعلق به تعداد خیلی کمی داده خواهد بود. برای جلوگیری از این اتفاق می‌توان از روش‌های از پس هرس کردن یا از پیش هرس کردن استفاده کرد. این روش‌ها در پیاده سازی با تنظیم پارامترهای مدل قابل انجام

است. در این پیاده سازی ما با تنظیم حداکثر عمق درخت و حداقل نمونه در هر برگ، از پیش هرس کردن را انجام دادیم. توجه کنید که این پارامترها به صورت تجربی و با امتحان کردن مقادیر متفاوت تنظیم شده‌اند.

نتایج ارزیابی این مدل به صورت زیر خواهد بود (تصویر ۹ و ۱۰). این ارزیابی‌ها نشان می‌دهد این مدل در تشخیص موارد منفی بهتر عمل کرده است. همچنین حساسیت کم نشان می‌دهد تعداد قابل توجهی از موارد مثبت، منفی تشخیص داده شده‌اند.

```
Accuracy: 0.9660569105691057
Precision: 0.976889444097439
Recall: 0.6184262554369316
F1 Score: 0.7573849878934624
```

تصویر ۹: نتیجه ارزیابی مدل درخت تصمیم‌گیری



تصویر ۱۰: ماتریس گیج‌زنی مدل درخت تصمیم‌گیری

مدل SVM

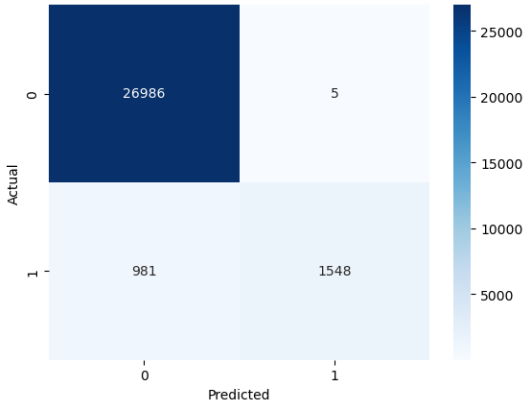
مدل SVM مدلی است که در آن یک مرز تصمیم‌گیری برای جداسازی کلاس‌ها رسم می‌شود و داده‌های جدید بر اساس موقعیت نسبت به این مرزها کلاس‌بندی می‌شوند. رسم این خط نیازمند بهینه کردن پارامترهای مشخصی است. همچنین پاک سازی صحیح داده بسیار در عملکرد این مدل تاثیر گذار است و نرمال سازی صحیح در افزایش سرعت همگرایی این مدل بسیار اهمیت دارد.

در این مدل پارامترهای متفاوتی را می‌توان تنظیم کرد که پارامترهای انتخاب شده به صورت تجربی و با انتخاب چند ترکیب پارامتر مختلف انتخاب شده‌اند. یکی از پارامترهایی که تنظیم می‌شوند، کرنل است. در این دیتاست طبق مشاهدات کرنل خطی عملکرد بسیار ضعیفی دارد که نشان می‌دهد داده‌ها به صورت خطی تفکیک‌پذیر نیستند. در نهایت این مدل با کرنل چند جمله‌ای و درجه‌ی ۵ نتیجه‌ی نسبتاً خوبی به ما ارائه داد.

نتایج ارزیابی این مدل به صورت زیر خواهد بود (تصاویر ۱۱ و ۱۲). طبق ماتریس گیج‌زنی، ۹۸۱ داده که در اصل مثبت بوده‌اند منفی تشخیص داده شده‌اند که نشان می‌دهد عملکرد این مدل چندان خوب نبوده است. این مدل توانسته موارد منفی را خیلی بهتر تشخیص بدهد و فقط ۵ مورد که منفی بوده‌اند به اشتباه مثبت تشخیص داده شده‌اند. در SVM یک معیار ارزیابی دیگر به نام میانگین مربع خطاها داریم که در اینجا تقریبا ۰.۰۳ است که نشان می‌دهد مدل دقت نسبتاً خوبی دارد.

```
Accuracy: 0.9665989159891599
Precision: 0.9967804249839022
Recall: 0.6120996441281139
F1 Score: 0.7584517393434591
MSSE: 0.03340108401084011
```

تصویر ۱۱: نتایج ارزیابی مدل SVM



تصویر ۱۲: ماتریس گیج‌زنی مدل SVM

مقایسه‌ی مدل‌ها

قبل از هر گونه تحلیل بهتر است اول معیار خود را برای مقایسه‌ی مدل‌ها مشخص کنیم. در این مسئله که یک کاربرد پزشکی است، حساسیت معیار بهتری است چرا که برای ما مهم است تعداد بیشتری موارد مثبت را مثبت تشخیص دهیم تا بتوانیم در مراحل اولیه‌ی بیماری با تشخیص صحیح، از پیشرفت بیماری جلوگیری کنیم.

بین دو مدل طبقه بندی که بررسی شدند، درخت تصمیم با حساسیت تقریبا ۶۱.۸ بهتر از SVM با حساسیت ۶۱.۲ عمل کرده است و طبق ماتریس گیج‌زنی تعداد کمتری از موارد مثبت را به اشتباه، منفی تشخیص داده است. پس مدل درخت تصمیم بهتر از مدل SVM عمل کرده است. عملکرد ضعیف‌تر SVM می‌تواند به دلیل وابستگی این مدل به پاک‌سازی صحیح داده‌ها باشد در صورتی که مدل درخت تصمیم به انجام درست این مرحله حساسیت کمتری دارد و چندان تحت تاثیر آن قرار نمی‌گیرد.

خوشه بندی

انتخاب ویژگی

در این قسمت باید ویژگی‌هایی که در خوشه بندی‌ها استفاده می‌کنیم، انتخاب شوند. یکی از مواردی که در این قسمت بررسی می‌کنیم، ماتریس همبستگی است. ویژگی‌هایی که همبستگی زیادی دارند (بیش از ۰.۹) را از ویژگی‌ها حذف می‌کنیم. در این دیتاست چنین همبستگی‌ای بین ویژگی‌ها مشاهده نمی‌شود.

همچنین در این قسمت می‌توان بر اساس اهمیت ویژگی‌ها که در مدل درخت تصمیم محاسبه شده است، ویژگی‌ها را انتخاب کنیم (تصویر ۱۳).

	Feature	Importance
5	blood_glucose_level	0.573168
4	HbA1c_level	0.369547
0	age	0.021369
3	bmi	0.020171
1	hypertension	0.008102
2	heart_disease	0.006120
7	gender_Male	0.001194
6	gender_Female	0.000330
8	gender_Other	0.000000

تصویر ۱۳: اهمیت ویژگی‌ها، استخراج شده از درخت تصمیم

در این مرحله ویژگی‌هایی را انتخاب کردیم که اهمیت آن‌ها بیش از ۰.۰۱ بود (تصویر ۱۴).

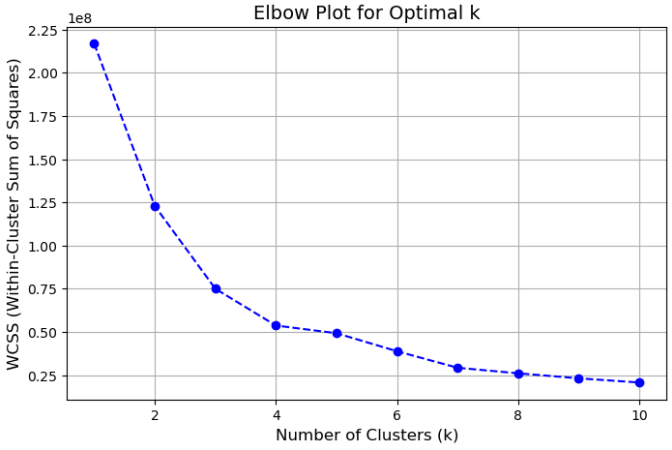
```
Selected Features for Clustering: ['blood_glucose_level', 'HbA1c_level', 'age', 'bmi']
```

تصویر ۱۴: ویژگی‌های انتخابی برای خوشه‌بندی

K-means

در این الگوریتم، داده‌ها با توجه به شباهتشان در k خوشه قرار می‌گیرند. این الگوریتم یک الگوریتم بدون نظارت است به این معنا که دسته‌ها از پیش مشخص نشده‌اند و مبنای تصمیم‌گیری فاصله‌ی داده‌هاست.

یکی از مراحل مهم، انتخاب k یا تعداد دسته‌هاست. در این پیاده سازی از نمودار elbow استفاده کردیم (تصویر ۱۵). طبق این نمودار عددی برای تعداد دسته مناسب است که در محل شکستگی باشد یعنی با اعداد قبل خود اختلاف چشمگیر و با اعداد بعدی اختلاف کمتری داشته باشد که در اینجا k را ۳ انتخاب کردیم.



تصویر ۱۵: نمودار elbow

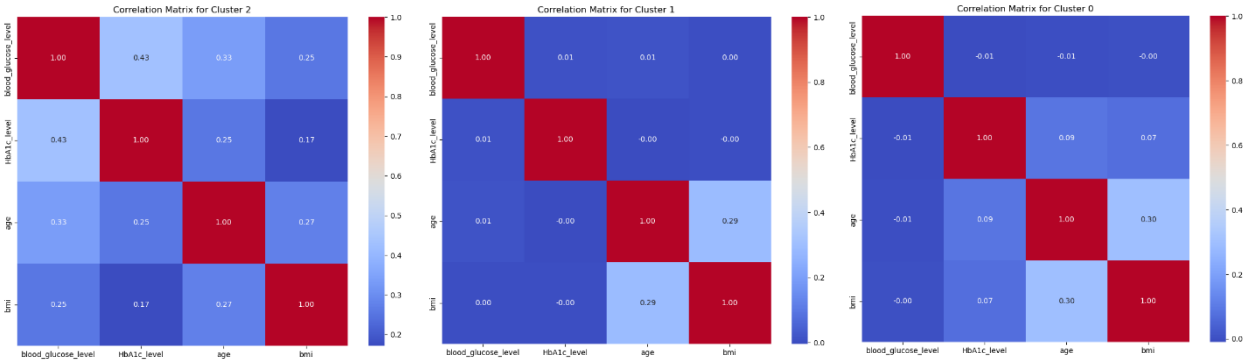
حالا می‌توانیم با تفسیر میزان اهمیت در هر یک از این خوشه‌ها ببینیم هر کدام چه ویژگی‌های برجسته‌ای دارند (تصویر ۱۶). این اطلاعات نشان می‌دهد بیشترین اختلاف بین هر دسته مربوط به میزان قند خون بیماران است. گروه آخر شامل بیمارانی است که قند خون بسیار بالایی دارند و احتمال ابتلای آن‌ها به دیابت بسیار بالاست، گروه اول شامل بیمارانی است که قند خون سطح متوسط دارند و احتمال ابتلای آن‌ها به دیابت کمتر از گروه اول است ولی همچنان نیاز به مراقبت دارند و گروه دوم افرادی با سطح قند خون نرمال هستند که احتمال ابتلای آن‌ها به دیابت بسیار پایین است.

```
Cluster 0:
blood_glucose_level    146.474133
age                     41.837717
bmi                     26.912617
HbA1c_level              5.116283
Name: 0, dtype: float64

Cluster 1:
blood_glucose_level    88.749277
age                    40.355505
bmi                    26.532733
HbA1c_level             5.008970
Name: 1, dtype: float64

Cluster 2:
blood_glucose_level    218.718020
age                     47.882594
bmi                     28.329240
HbA1c_level              5.566564
Name: 2, dtype: float64
```

تصویر ۱۶: میانگین ویژگی‌ها در هر دسته

همان طور که قابل تشخیص است به دلیل اهمیت بالاتر ویژگی قند خون و همبستگی نسبتاً بالا با ویژگی دیابت، این گروه بندی به طور کلی با توجه به سطح قند خون انجام شده است. این روش خوشه‌بندی توانسته با رابطه‌ای منطقی این داده‌ها را خوشه‌بندی کند.

همچنین در هر خوشه قصد داریم وابستگی بین ویژگی‌ها را بررسی کنیم. این کار با استفاده از ماتریس همبستگی قابل انجام است.



تصویر ۱۷: ماتریس همبستگی هر خوشه

پیش از این گفتیم خوشه‌ی ۰ متعلق به افرادی با سطح قند خون متوسط است. این ماتریس همبستگی نشان می‌دهد در این خوشه همبستگی زیادی بین سن و bmi وجود دارد و همچنین همبستگی بسیار ضعیفی بین سن و HbA1c_level و bmi و HbA1c_level وجود دارد. همچنین گفتیم خوشه‌ی ۱ متعلق به افرادی با سطح قند خون نرمال است که فقط بین bmi و سن این همبستگی وجود دارد و بقیه ویژگی‌ها وابستگی چشمگیری به هم ندارند. همچنین خوشه‌ی ۲ متعلق به افرادی با قند خون بالا است که ماتریس همبستگی نشان می‌دهد وابستگی زیادی میان ویژگی‌های HbA1c_level و سطح قند خون وجود دارد پس افزایش هر یک باعث افزایش دیگری می‌شود. همچنین به طور کلی وابستگی بین بقیه ویژگی‌ها و ویژگی قند خون نسبت به گروه‌های دیگر بسیار بیشتر است به طوری که نشان می‌دهد با افزایش هر یک از ویژگی‌های سن و bmi، قند خون هم افزایش می‌یابد.

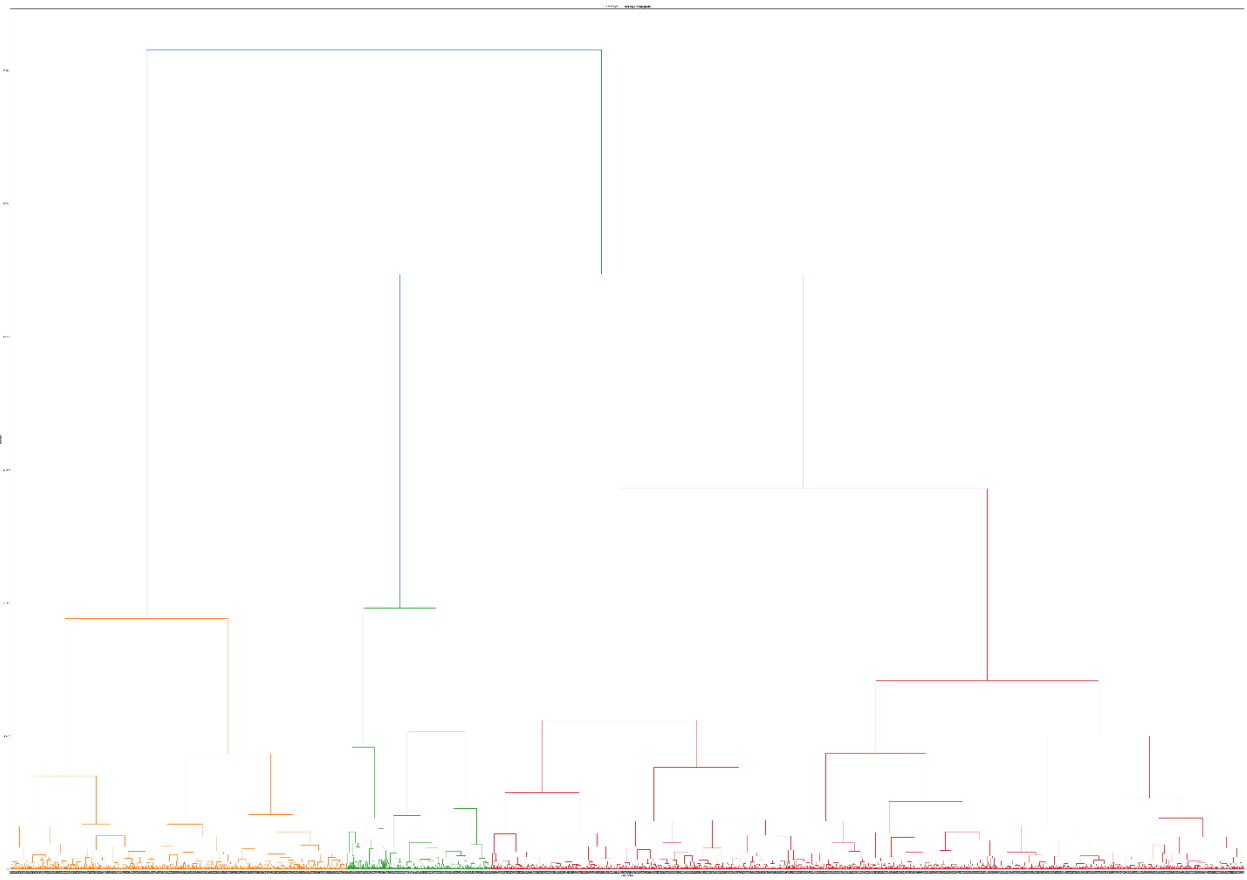
خوشه بندی سلسله مراتبی

در این خوشه‌بندی داده‌ها بر حسب شباهت در کنار هم قرار می‌گیرند و خوشه‌ها را می‌سازند. چون تعداد داده‌ها بسیار زیاد هستند ابتدا ۵۰۰۰ داده را نمونه برداری کردیم تا این خوشه بندی را روی آن‌ها انجام دهیم.

ابتدا ماتریس لینکج را تشکیل دادیم که یک ماترین (n-1)*4 است که n تعداد داده‌هاست (در اینجا ۵۰۰۰) و هر سطر ماتریس نشان‌دهنده‌ی یک مرحله ادغام خوشه‌ها است و شامل ۴ مقدار زیر می‌شود:

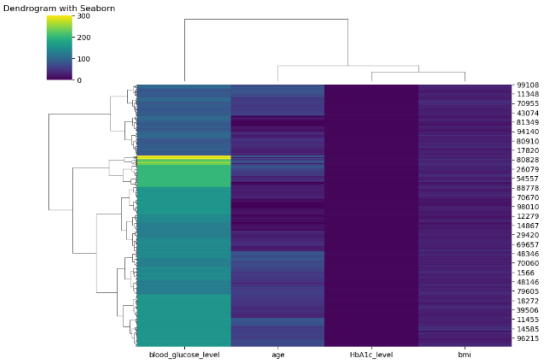
۱. اندیس اولین خوشه‌ای که با هم ترکیب شده‌اند.
۲. اندیس دومین خوشه‌ای که با هم ترکیب شده‌اند.
۳. فاصله یا معیار لینکج بین دو خوشه‌ی ترکیب شده.
۴. تعداد نقاط موجود در خوشه‌ی جدید.

با توجه به همین ماتریس می‌توان نمودار دندوگرام را برای این خوشه‌بندی رسم کرد (تصویر ۱۸). دندوگرام یک نمودار درختی است که سلسله مراتب خوشه‌ها را در خوشه‌بندی سلسله مراتبی نمایش می‌دهد. این نمودار نشان می‌دهد که چگونه داده‌ها مرحله به مرحله با یکدیگر ترکیب شده‌اند. در دندوگرام هر گره نشان‌دهنده‌ی یک ادغام بین دو خوشه است. ارتفاع هر ادغام نشان‌دهنده‌ی فاصله بین دو خوشه است.



تصویر ۱۸: نمودار دندوگرام

تصویر زیر clustermap را به همراه دندوگرام نشان می‌دهد (تصویر ۱۹). ماتریس رنگی نشان‌دهنده‌ی مقادیر متفاوت ویژگی‌ها برای هر داده است. این رنگ‌ها نشان‌دهنده‌ی پایین و بالا بودن این مقادیر هستند که مقادیر بالاتر با زرد و مقادیر پایین‌تر با بنفش مشخص شده‌اند. همچنین در بالا و چپ ماتریس دندوگرام رسم شده است. دندوگرام سمت چپ نشان‌دهنده‌ی سلسله مراتب هر داده است و دندوگرام پایین نشان‌دهنده‌ی خوشه‌بندی ویژگی‌هاست. بلندی شاخه‌ها همان طور که گفته شد، نشان‌دهنده‌ی فاصله‌ی هر خوشه است.



تصویر ۱۹: clustermap

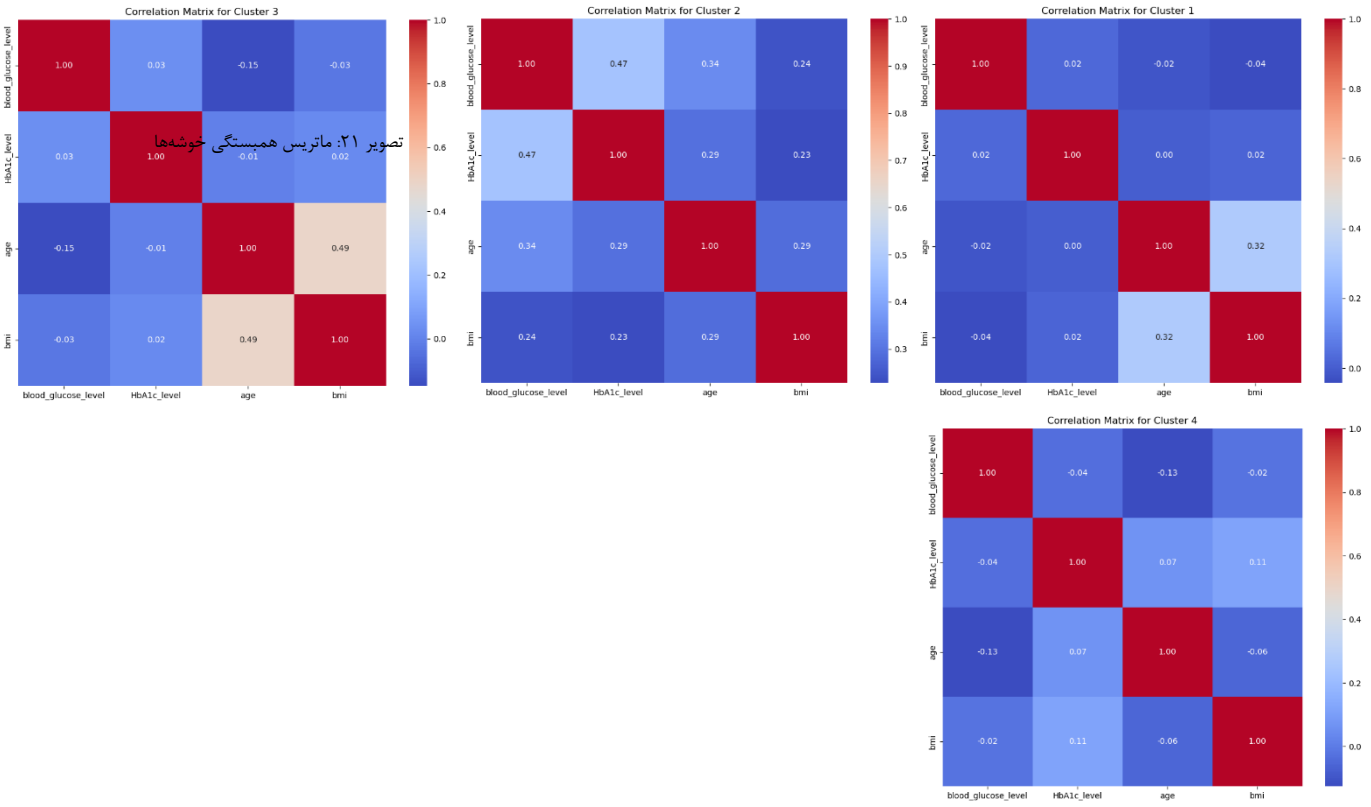
در این ماتریس می‌بینیم که ویژگی‌های مشابه در کنار هم قرار گرفته‌اند و همچنین داده‌های نزدیک به هم در کنار هم قرار گرفته‌اند. می‌توان با محدود کردن بلندی شاخه تعداد خوشه‌ها را محدود کرد یعنی داده‌هایی که فاصله‌ی آن‌ها از مقدار مشخصی کم‌تر است را در یک خوشه قرار می‌دهیم.

در این پیاده‌سازی ما فاصله را به ۱۰۰۰ محدود کردیم و ۴ خوشه به دست آمد. در نهایت میانگین ویژگی در هر خوشه را بررسی کردیم که به صورت زیر است (تصویر ۲۰). این میانگین‌ها مشخص می‌کند که گروه اول مربوط به افرادی با سطح گلوکز پایین است، گروه دوم مربوط به افرادی با سطح گلوکز بسیار بالا است، گروه سوم مربوط به افراد با سن پایین و سطح قند خون متوسط است و در نهایت گروه چهارم متعلق به افرادی با سطح قند خون متوسط و میانگین سن ۵۶ سال است.

Cluster	blood_glucose_level	HbA1c_level	age	bmi
1	89.145268	4.981658	40.155539	26.368305
2	217.649063	5.620102	48.408859	28.051107
3	145.495942	5.030844	19.296266	24.318182
4	147.684268	5.150165	56.284928	28.668867

تصویر ۲۰: میانگین ویژگی‌ها در هر خوشه

در نهایت با بررسی ماتریس همبستگی هر خوشه، وابستگی ویژگی‌ها را بررسی می‌کنیم (تصویر ۲۱). این ماتریس‌ها نشان می‌دهند که در خوشه‌ی اول، ارتباط چشمگیری بین سن و bmi وجود دارد، در گروه دوم ارتباط چشمگیری بین سطح قند خون و HbA1c_level وجود دارد و همچنین بین سطح قند خون و سن و bmi ارتباط قوی‌ای وجود دارد. یعنی با افزایش این موارد احتمال اینکه یک فرد در گروه دوم که شامل افراد با قند خون خیلی بالا است قرار بگیرد، افزایش می‌یابد. در گروه سوم نیز ارتباط زیادی بین سن و bmi مشاهده می‌شود. در گروه چهارم بیشترین همبستگی بین bmi و HbA1c_level وجود دارد.



تصویر ۲۱: ماتریس همبستگی خوشه‌ها

تصویر ۲۱: ماتریس همبستگی هر خوشه

این خوشه بندی توانسته داده‌ها را به ۴ قسمت تقسیم کند که ممکن است با افزایش محدوده، تعداد این خوشه‌ها کمتر شود و خوشه‌ی سوم و چهارم با هم ادغام شوند. این تقسیم بندی توانسته داده‌ها را به ۴ دسته با ویژگی‌های معقول تقسیم کند که البته این تعداد خوشه می‌توانست کمتر باشد.

مقایسه مدل‌ها

مدل K-means توانسته داده‌ها را به سه دسته با میانگین ویژگی‌های معقول تقسیم کند و همچنین مدل خوشه‌بندی سلسله مراتبی به حافظه‌ی زیادی نیاز دارد تا حدی که ناچار بودیم از داده‌ها نمونه برداری کنیم. پس مدل K-means مدل بهتری از بین دو مدل خوشه‌بندی است.

نتیجه گیری

در این پروژه سعی کردیم با پیاده سازی ۴ الگوریتم متفاوت و اندازه‌گیری دقت‌ها و تحلیل نتایج بررسی کنیم بهترین مدل تصمیم‌گیری برای دیتاست ارائه شده کدام است؟ طبق نتایج و اینکه این دیتاست شامل داده‌های برچسب‌دار است، به نظر می‌آید انتخاب یک الگوریتم طبقه بندی که بانظارت است، انتخاب بهتری باشد. از بین دو الگوریتم پیاده شده در قسمت قبل دیدیم که

درخت تصمیم به دلیل حساسیت بیشتر، بهتر عمل کرده است. پس از بین ۴ الگوریتم پیاده سازی شده، درخت تصمیم را انتخاب می‌کنیم.