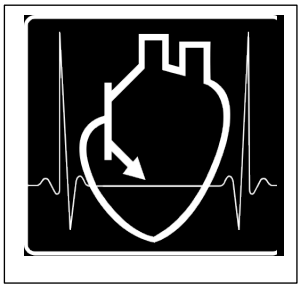
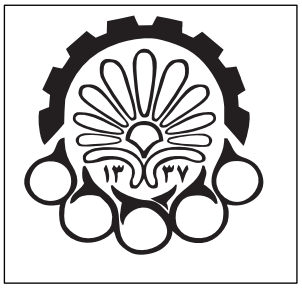




پروژه پایانی درس یادگیری ماشین

استاد درس: دکتر بغدادی

آریان افشار

۴۰۱۳۳۰۰۴

مقدمه

ابتدا تمام کتابخانه های لازم را در notebook آوردم و سپس دیتاست alzheimers diseases را از Kaggle دانلود و بارگزاری کردم.

در ابتدای کار، برای آشنایی دقیق با داده ها و ارزیابی وضعیت دیتاست آلزایمر، ابتدا ابعاد داده بررسی شد و مشخص گردید این دیتاست شامل ۲۱۴۹ نمونه و ۳۵ ویژگی مختلف است. این مقدار نسبتاً بزرگ بوده و امکان اعمال روش های یادگیری ماشین با دقت کافی را فراهم می کند. با مشاهده ساختار دیتاست به وسیله دستور info، به طور کامل جزییات هر ستون شامل نام و نوع داده (عدد صحیح، اعشاری، یا متنی)، و همچنین تعداد مقادیر غیرگمشده در هر ستون مشخص شد. مشاهده شد که در تمام ۳۵ ویژگی، هیچ مقدار گمشده (Null) وجود ندارد و تمام نمونه ها برای تمام ستون‌ها کامل هستند و این موضوع در بخش بررسی missing values نیز تایید شد زیرا خروجی این بخش یک سری خالی بود، به این معنی که هیچ ستونی با مقدار ناقص در داده‌ها وجود ندارد. این موضوع باعث می شود مرحله پاک سازی داده ساده تر شده و نیازی به اعمال روش های جایگذاری یا حذف داده‌های ناقص نباشد البته که من به منظور کامل پوشش دادن کلاس, روش های handling missing data را در ادامه پیاده کردم.

در میان ویژگی ها، انواع مختلفی از متغیرها مشاهده می شود: متغیرهای عدد صحیح مانند سن (Age)، جنسیت (Gender) ، سطح تحصیلات (EducationLevel)، یا وجود بیماری هایی نظیر دیابت و پرفشاری خون، متغیرهای اعشاری مثل BMI و سطوح مختلف کلسترول، و تنها یک متغیر متنی (DoctorInCharge) که به نظر می رسد نام پزشک یا مسئول پرونده است. همچنین برچسب تشخیص (Diagnosis) به‌عنوان متغیر هدف برای مدل های طبقه‌بندی حضور دارد.

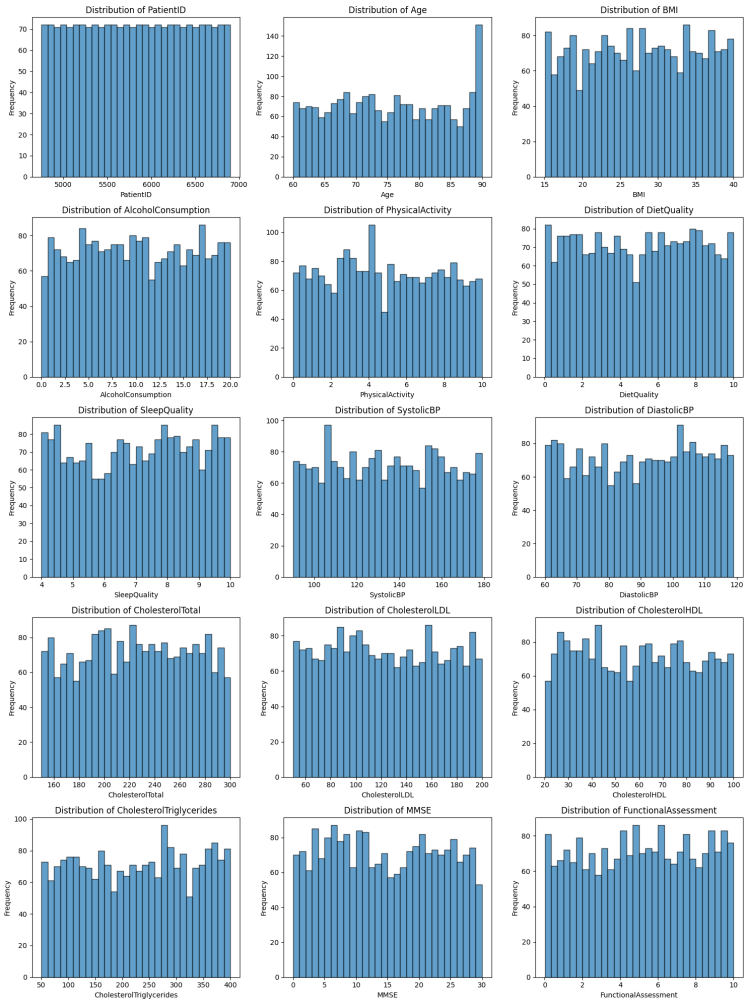
EDA(Exploratory data analysis)

در بخش بعدی با هدف آماده سازی داده ها برای تحلیل های آماری و مدل سازی، مرحله ای از تحلیل اکتشافی (EDA) با تمرکز بر شناسایی نوع ویژگی ها انجام شد. ابتدا برای تمایز بین ویژگی های عددی (numerical) و ویژگی های دسته ای (categorical) یک معیار ساده را به کار گرفتم که ویژگی هایی که تعداد مقادیر یکتای آن ها بیشتر از ۱۰ مقدار باشد، به عنوان ویژگی عددی در نظر گرفته میشوند. این کار معمولاً برای تفکیک متغیرهایی با مقادیر پیوسته از ویژگی‌های طبقه ای مناسب است. برای شناسایی دقیق تر ویژگی های دسته ای، ابتدا تمام ستون هایی که عددی محسوب نمی شوند فهرست شد و علاوه بر این ویژگی هدف (Diagnosis) نیز از این لیست حذف شد تا صرفاً ویژگی های ورودی مدل ها برای دسته بندی باقی بماند. در نهایت، با چاپ تعداد و نام هر گروه، توزیع کلی داده ها از این نظر مشخص شد: مثلا چند ویژگی عددی و چند ویژگی دسته ای داریم و نامشان چیست که به شرح زیر شد:

```
Numerical columns (16): ['PatientID', 'Age', 'BMI', 'AlcoholConsumption', 'PhysicalActivity',
'DietQuality', 'SleepQuality', 'SystolicBP', 'DiastolicBP', 'CholesterolTotal', 'CholesterolLDL',
'CholesterolHDL', 'CholesterolTriglycerides', 'MMSE', 'FunctionalAssessment', 'ADL']
```

```
Categorical columns (18): ['BehavioralProblems', 'CardiovascularDisease', 'Confusion',
'Depression', 'Diabetes', 'DifficultyCompletingTasks', 'Disorientation', 'DoctorInCharge',
'EducationLevel', 'Ethnicity', 'FamilyHistoryAlzheimers', 'Forgetfulness', 'Gender', 'HeadInjury',
'Hypertension', 'MemoryComplaints', 'PersonalityChanges', 'Smoking']
```

برای دید بهتر از ویژگی ها کار بعدی که انجام دادم visualize کردن ویژگی ها و مقدار distribution آن ها بود که برای مقادیر و ویژگی های Numerical به شرح زیر شد:

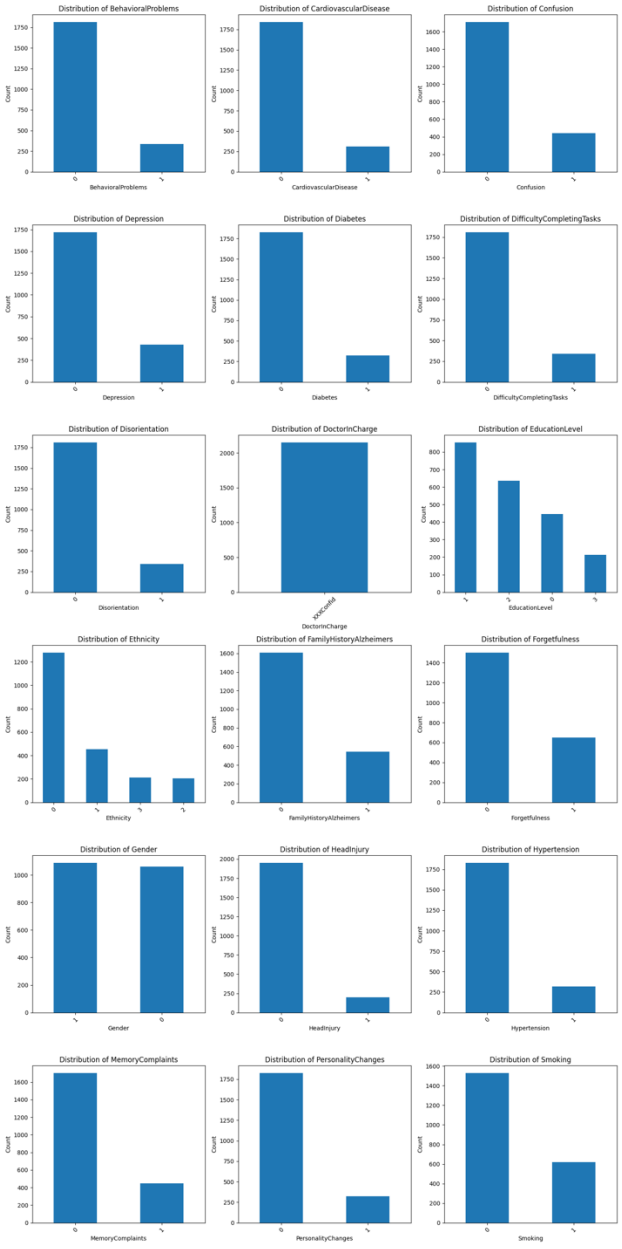


در هر نمودار، محور افقی بیانگر مقادیر هر ویژگی و محور عمودی بیانگر فراوانی تکرار نمونه‌هاست

برای مثال، شکل توزیع ویژگی «سن» (Age) نشان می‌دهد که تعداد زیادی از بیماران در محدوده سنی بالاتر (به ویژه نزدیک ۹۰ سال) تجمع دارند که به احتمال زیاد ناشی از ماهیت بیماری آلزایمر و شیوع بیشتر آن در سالمندان است. نقطه قابل توجه دیگر، توزیع متعادلی است که برای اغلب ویژگی های آزمایشگاهی و نمرات سنجش عملکردی مشاهده می شود، نظیر BMI، سطوح مختلف کلسترول ها، شیوه زندگی (مصرف الکل، فعالیت فیزیکی، کیفیت خواب) و غیره که رفتار یکنواخت تر و فاقد ناهنجاری خاص هستند.

در برخی ویژگی ها مانند ‘AlcoholConsumption’ یا ‘PhysicalActivity’ پراکندگی نسبتاً یکنواخت دیده می شود، اما در ویژگی هایی چون ‘SleepQuality’ یا ‘SystolicBP’ نقاط تجمع (Peak) و یا حتی وجود نمونه های پرت به چشم می خورد. با مشاهده هیستوگرام ‘MMSE’ توزیع نسبتاً یکنواخت با عدم وجود شیب تند قابل توجه دیده می شود که می تواند نشان دهنده پراکندگی نمرات بیماران باشد.

حال همین کار را برای مقادیر categorical و ویژگی های دسته بندی نیز انجام دادم که شکل پایین شد:



هر نمودار ستونی مربوط به یکی از ویژگی های دسته ای است و محور افقی نشان دهنده دسته ها (مثل بله/خیر، زن/مرد، یا درجات مختلف)، و محور عمودی بیانگر تعداد نمونه های هر دسته در دیتاست است.

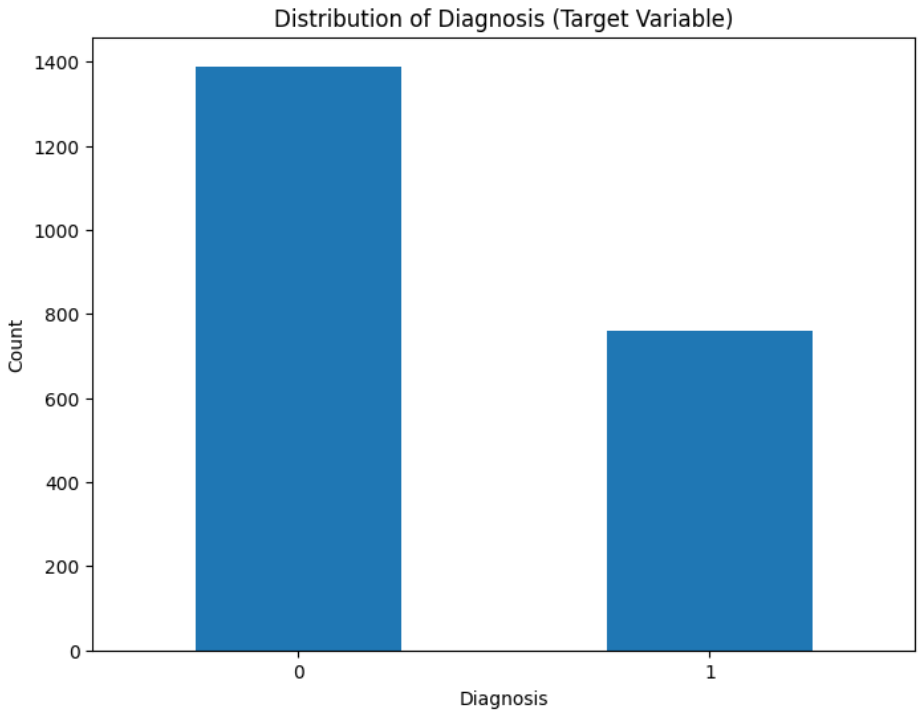
. اغلب ویژگی های دسته ای دارای دو مقدار (مثلاً صفر و یک) هستند که بیانگر وضعیت عدم وجود یا وجود یک ویژگی هستند. به عنوان نمونه، ویژگی هایی مانند ’BehavioralProblems‘، ’CardiovascularDisease‘، ’Confusion‘، ’Depression‘، ’Diabetes‘و ’Smoking‘ همگی نشان می دهند بیشترین تعداد نمونه ها مقدار صفر دارند (یعنی بیشتر بیماران فاقد این شرایط هستند)، در حالی که تعداد بیماران دارای این شرایط بسیار کمتر است و این مسئله باعث توزیع غیرمتعادل بین کلاس‌ها شده است.

برای برخی ویژگی ها مثل ‘Gender’، پراکندگی نمونه ها در دو دسته تقریباً برابر است و توازن جنسیتی مناسبی در داده‌ها مشاهده می شود. همچنین در ویژگی هایی نظیر ‘EducationLevel’ و ‘Ethnicity’، توزیع بین چند دسته با اختلافاتی همراه است، اما باز هم بیشتر نمونه‌ها متعلق به یک یا دو گروه اصلی هستند.

ویژگی هایی مانند ‘FamilyHistoryAlzheimers’ و ‘MemoryComplaints’ هم مشابه سایر ویژگی های دسته ای بیشتر نمونه ها دارای مقدار صفر هستند (یعنی سابقه خانوادگی یا شکایت حافظه در اکثر بیماران مشاهده نشده است)، و این نشان دهنده unbalanced بودن ویژگی‌ها در داده‌ها است.

حال که به دید کلی از ویژگی های دیتاست رسیدیم, میتوان ویژگی target یا همان Diagnosis را بررسی کرد و دید که آیا بالانس هستند دیتا ها یا خیر.

Target variable analysis




```
Class Distribution:
Diagnosis
0    1389
1     760
Name: count, dtype: int64

Class Balance Ratio: 1.83
```

براساس خروجی نمودار و مقادیر چاپ شده، می بینیم که تعداد افراد با مقدار ۰ (یعنی افراد بدون تشخیص آلزایمر) به صورت قابل توجهی بیشتر از افراد با مقدار ۱ (افراد مبتلا به آلزایمر) است. به طور دقیق، تعداد کلاس ۰ برابر ۱۳۹۰ نمونه و تعداد کلاس ۱ برابر ۷۵۹ نمونه است. این نسبت جزییات دقیق ترش با محاسبه نسبت تعداد کلاس غالب به کلاس کم تر به دست آمده و مشخص شد که نسبت تعادل کلاس ها حدود ۱.۸۳ به ۱ است. یعنی تقریباً به ازای هر ۱.۸۳ نفر سالم، یک نفر بیمار در داده‌ها ثبت شده است.

این عدم توازن کلاس ها (class imbalance) اهمیت زیادی در پروژه های یادگیری ماشین طبقه بندی به ویژه در حوزه های پزشکی پیدا می کند. چرا که مدل هایی مثل Decision Tree یا Logistic Regression در صورت نبود اقدامات جبرانی ممکن است به سمت پیش بینی کلاس اکثریت متمایل شوند و عملکرد ضعیفی روی کلاس اقلیت (بیماران) داشته باشند. به همین خاطر، در ادامه پروژه از تکنیک oversampling (SMOTE) استفاده کردم تا دیتا ها بالانس شوند که در ادامه به آن خواهم پرداخت.

حال نوبت به مشاهده missing values میرسد که درسته در این دیتاست ما مقادیر گمشده نداشتیم اما به منظور کاور دادن تمامی مباحث من از median imputation برای مقادیر عددی و mode imputation برای مقادیر دسته ای استفاده کردم و نتیجه چاپ شده آن به شرح زیر شد:

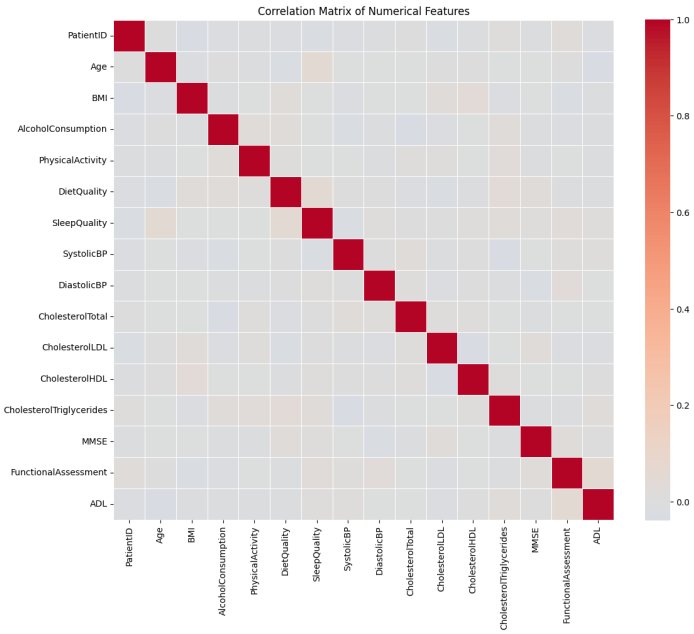
```
Missing values after imputation:
0
```

Correlation analysis

مرحله بعدی که انجام دادم تحلیل همبستگی (Correlation Analysis) بین ویژگی های عددی دیتاست بود که جهت این تحلیل، یک ماتریس همبستگی از تمامی ستون های عددی رسم گردید و با استفاده از نمودار heatmap (نقشه حرارتی)، میزان ارتباط بین جفت ویژگی ها ب ەصورت تصویری نمایش داده شد.

- رنگ های نزدیک به قرمز پررنگ نشان دهنده همبستگی مثبت قوی (مقادیر نزدیک به ۱)
- رنگ های نزدیک به آبی/خاکستری نشان دهنده همبستگی ضعیف یا منفی (مقادیر نزدیک به صفر یا منفی)

- قطر اصلی ماتریس، که متعلق به همبستگی هر ستون با خودش است، همیشه مقدار ۱ دارد و به صورت قرمز پررنگ دیده می شود.
- تقریباً اکثر خانه های خارج از قطر رنگی نزدیک به خاکستری یا قرمز خیلی کم رنگ دارند که نشان دهنده همبستگی نسبتاً ضعیف بین بیشتر ویژگی های عددی دیتاست است.
- به عبارت دیگر، اکثر متغیرهای عددی مستقل از هم هستند و ارتباط خطی قوی بین آن ها وجود ندارد. این مسئله می تواند برای مدل های مبتنی بر یادگیری ماشین مفید باشد چراکه فرض Multicollinearity شدید در داده وجود ندارد.



در ادامه ، میزان همبستگی هر یک از ویژگی های عددی با متغیر هدف با استفاده از ضریب همبستگی پیرسون محاسبه و مرتب شد.

```
Features most correlated with target:
Diagnosis                 1.000000
CholesterolHDL            0.042584
PatientID                 0.041019
BMI                       0.026343
CholesterolTriglycerides  0.022672
DietQuality               0.008506
CholesterolTotal          0.006394
PhysicalActivity          0.005945
DiastolicBP               0.005293
Age                      -0.005488
Name: Diagnosis, dtype: float64
```

- **هیچ یک از ویژگی های عددی، ضریب همبستگی چشمگیر با Diagnosis ندارد.** بزرگ ترین مقدار غیر از سطر اول، فقط حدود ۰.۰۴ است، که عملاً تقریباً هیچ وابستگی خطی وجود ندارد. این موضوع مهم است، چون نشان می دهد ساختار داده های این پروژه از نظر همبستگی خطی، ساده نیست و شاید اطلاعات پیش بینی Diagnosis بیشتر در ویژگی‌های غیرخطی، ترکیبی یا ستون های غیراعدادی پنهان باشد.
- همبستگی کم به معنای بی ارزشی این ویژگی‌ها نیست. ممکن است مدل های غیرخطی (مثل Random Forest, و ...) همچنان بتوانند رابطه های پنهان را کشف کنند.
- **PatientID** همیشه باید حذف شود، چون شناسه صرفاً ابزاری برای ایندکس و فاقد نقش معنایی در پیش‌بینی است که البته این اتفاق در feature selection به صورت خودکار خواهد افتاد.

پیش پردازش دیتا

حال میرسیم به مرحله PreProcessing

کد گذاری متغیر های دسته ای (Categorical Variables Encoding)

- تمامی ستون های دسته ای با استفاده از **LabelEncoder** به مقدار های عددی تبدیل شدند. این کار باعث شد که ستون های داده های متنی به فرمت قابل استفاده برای الگوریتم های یادگیری ماشین تبدیل شوند.
- ستون‌های جدیدی با پسوند _encoded ایجاد شدند تا نسخه کد گذاری‌شده هر ستون موجود باشد و مخدوش شدن داده های اصلی رخ ندهد.

حذف داده های پرت (Outlier Removal)

- **روش IQR** (Interquartile Range) برای شناسایی داده های پرت استفاده شد. این روش به صورت زیر عمل کرد:
  - **چارک اول (Q1)** و **چارک سوم (Q3)** برای تمامی ویژگی های عددی محاسبه شد.
  - محلی که داده ها مورد تایید هستند با استفاده از **رابطه IQR** تعریف شد.
  - سطرهایی که خارج از این بازه قرار داشتند به عنوان داده پرت شناسایی شده و حذف شدند.
  - در مجموع، **۱۸۹۵ داده پرت** حذف گردید که باعث بهبود کیفیت داده های باقی مانده شد.

متعادل سازی کلاس ها با SMOTE

- برای مدیریت **عدم توازن کلاس ها** در متغیر هدف (Diagnosis)، از روش SMOTE استفاده شد.
  - SMOTE یک تکنیک oversampling است که نمونه های مصنوعی کلاس اقلیت (Class 1 - مبتلا به آلزایمر) را بر اساس داده های موجود ایجاد می کند.
  - این فرآیند باعث شد هر دو کلاس (۰ و ۱) در تعداد نمونه ها برابر باشند، که می تواند دقت و کارایی مدل سازی را افزایش دهد.
- نمودار ها نشان می دهند که پس از اعمال SMOTE، کلاس‌های ۰ و ۱ به طور کامل متوازن شده اند.

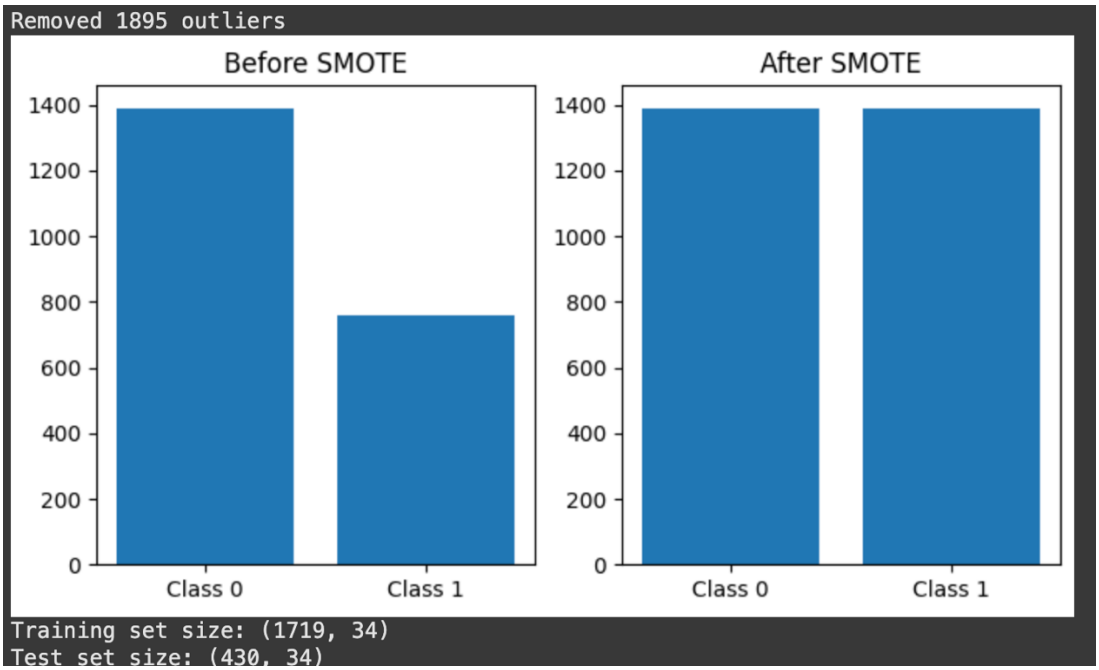
تقسیم داده به مجموعه های آموزشی و تستی

- پس از حذف داده های پرت و اعمال SMOTE، مجموعه داده ها به ترتیب **80% برای آموزش** و **20% برای تست** تقسیم شدند:
  - اندازه مجموعه آموزشی: 1719.34
  - اندازه مجموعه تستی: 430.34
- این تقسیم بندی به صورت Stratified انجام شد تا توزیع کلاس ها در هر مجموعه (آموزشی و تستی) حفظ شود.

رسم نمودار کلاس ها قبل و بعد از SMOTE

- نمودار ها تصویری از وضعیت کلاس ها را پیش و پس از اعمال SMOTE نشان می دهند:
  - **قبل از SMOTE:** کلاس ۰ تعداد بیشتری دارد.

- **بعد از SMOTE:** هر دو کلاس تعداد برابر دارند (~۱۴۰۰ نمونه در هر دسته).



Normalization and Standardization

در مرحله بعدی، داده های عددی انتخاب شده با دو رویکرد مجزا تحت مقیاس دهی **نرمال سازی** (Normalization) و **استاندارد سازی** (Standardization) قرار گرفتند. ابتدا لیستی از ستون هایی که نیازمند مقیاس دهی هستند (مانند سن، BMI، مصرف الکل، سطح فعالیت بدنی، فشار خون و شاخص‌های بیوشیمیایی مانند کلسترول و نمرات شناختی) تعیین شد(بر اساس خود دیتاست و توضیحاتش در Kaggle). سپس بررسی شد که کدام یک از این ستون ها واقعاً در داده های آموزشی وجود دارند و فقط ستون‌های موجود انتخاب شدند تا پردازش بدون خطا انجام شود.

برای هر دو مجموعه داده آموزش و تست، دو نسخه جدید ایجاد شد:

یکی برای نرمال سازی و دیگری برای استانداردسازی. در مرحله **نرمال سازی**، از روش **Min-Max Scaler** استفاده شد تا مقدار هر ویژگی در بازه بین صفر تا یک قرار گیرد. این روش مقادیر هر ستون را به نسبت فاصله از کمترین تا بیشترین مقدار آن ستون تبدیل می‌کند. در مقابل، در مرحله **استانداردسازی** از روش **Z-score** استفاده شد که باعث می شود میانگین مقادیر هر ویژگی برابر با صفر و انحراف معیار آن برابر با یک قرار بگیرد. این کار به مدل های یادگیری ماشین کمک می کند تا از اثرگذاری مقیاس متفاوت ویژگی ها جلوگیری شود و مدل آموزش پایدارتر و دقیق تری داشته باشد.

در پایان، تعداد ستون هایی که تحت مقیاس دهی قرار گرفتند و نام آن‌ها نمایش داده شد تا شفافیت پردازش و صحت اجرای مراحل برای ادامه کار حفظ شود.

```
Scaled 15 columns: ['Age', 'BMI', 'AlcoholConsumption', 'PhysicalActivity', 'DietQuality',
'SleepQuality', 'SystolicBP', 'DiastolicBP', 'CholesterolTotal', 'CholesterolLDL',
'CholesterolHDL', 'CholesterolTriglycerides', 'MMSE', 'FunctionalAssessment', 'ADL']
```

انتخاب ویژگی

در قسمت بعدی به بخش مهم feature selection میپردازم.

برای این قسمت من از ۵ روشی که در کلاس گفته شد استفاده کردم.

روش های مورد استفاده برای انتخاب ویژگی:

اطلاعات متقابل (Mutual Information):

- میزان وابستگی و اطلاعات مشترک بین هر ویژگی و متغیر هدف را اندازه گیری می‌کند.
- بر اساس این روش، ویژگی های FunctionalAssessment, ADL, MMSE, و MemoryComplaints_encoded بالاترین امتیاز را کسب کردند.

آزمون کای-دو (Chi-Square Test):

- برای داده های نرمالایز شده و غیرمنفی اجرا شد تا قدرت تفکیک هر ویژگی نسبت به متغیر هدف را بسنجد.
- مهم‌ترین ویژگی ها شامل PatientID, MemoryComplaints_encoded, BehavioralProblems_encoded, Fun ctionalAssessment, ADL و MMSE بودند.

آزمون ANOVA (F-value):

- برای سنجش تفاوت میانگین هر ویژگی عددی بین گروه‌های هدف.
- ستون‌های FunctionalAssessment, ADL, MemoryComplaints_encoded, MMSE و Be havioralProblems_encoded بالاترین F-Score را کسب کردند.

اهمیت ویژگی با مدل جنگل تصادفی (Random Forest Feature Importance):

- اهمیت هر ویژگی را بر اساس نقش آن در مدل Random Forest اندازه گیری می‌کند.
- مهم‌ترین ویژگی‌ها: FunctionalAssessment, ADL, MMSE, MemoryComplaints_encoded، و BehavioralProblems_encoded.

حذف بازگشتی ویژگی (RFE):

- با استفاده از یک مدل Random Forest، به صورت بازگشتی ضعیف ترین ویژگی را حذف کرده و ۱۵ ویژگی برتر را نگه داشت.
- خروجی RFE مشابه نتایج دیگر بود و ویژگی های کلیدی را تایید کرد.

یافتن ویژگی های مشترک:

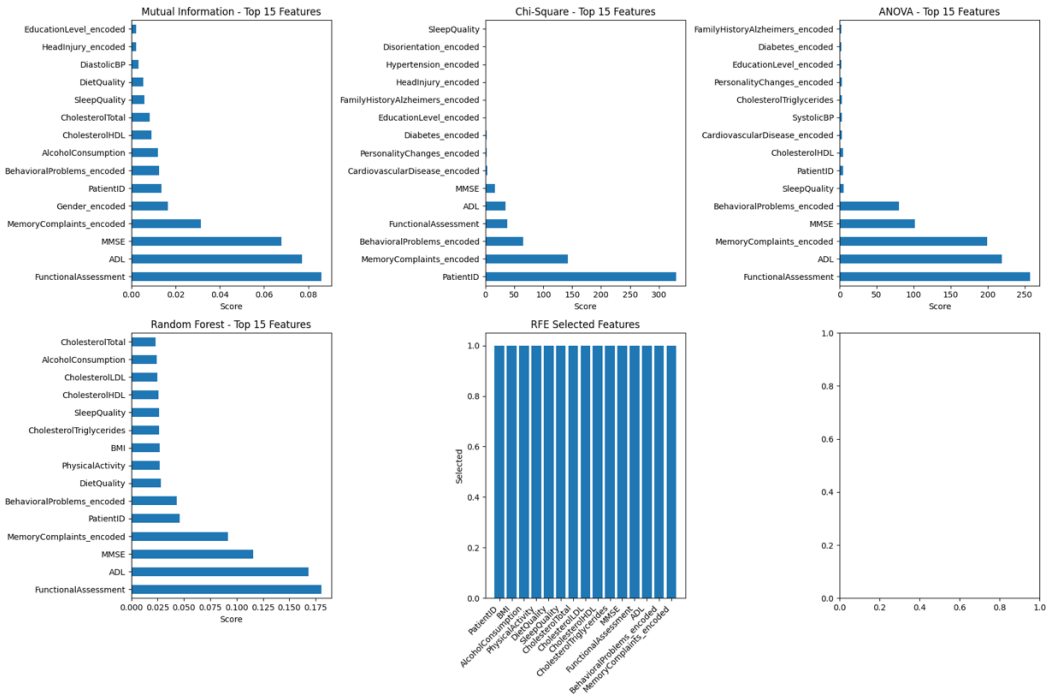
پس از اجرای همه روش های بالا، برای هر روش ۱۵ ویژگی برتر انتخاب شد و سپس ویژگی هایی که حداقل در سه روش مختلف بین ۱۵ ویژگی برتر قرار داشتند شناسایی شدند. جمعاً ۱۳ ویژگی مشترک به دست آمد که شامل موارد زیر است:

['FunctionalAssessment', 'ADL', 'MMSE', 'MemoryComplaints_encoded', 'PatientID', 'BehavioralProblems_encoded', 'AlcoholConsumption', 'CholesterolHDL', 'CholesterolTotal', 'SleepQuality', 'DietQuality', 'EducationLevel_encoded', 'CholesterolTriglycerides']

در نهایت، بر اساس **اهمیت ویژگی در مدل Random Forest** که معمولاً روشی قابل اعتماد محسوب می شود، ۱۵ ویژگی نهایی برای ادامه فرآیند مدل سازی انتخاب شد:

['FunctionalAssessment', 'ADL', 'MMSE', 'MemoryComplaints_encoded', 'PatientID', 'BehavioralProblems_encoded', 'DietQuality', 'PhysicalActivity', 'BMI', 'CholesterolTriglycerides', 'SleepQuality', 'CholesterolHDL', 'CholesterolLDL', 'AlcoholConsumption', 'CholesterolTotal']

این ویژگی ها، ورودی نهایی مدل های طبقه بندی خواهند بود.



1. Mutual Information Scores (Top 10):
FunctionalAssessment 0.086064
ADL 0.077301
MMSE 0.067915
MemoryComplaints_encoded 0.031435
Gender_encoded 0.016643
PatientID 0.013719
BehavioralProblems_encoded 0.012573
AlcoholConsumption 0.011914
CholesterolHDL 0.009035
CholesterolTotal 0.008191
dtype: float64
2. Chi-Square Scores (Top 10):
PatientID 329.330417
MemoryComplaints_encoded 142.269386
BehavioralProblems_encoded 65.326114
FunctionalAssessment 37.385703
ADL 34.277787
MMSE 16.045227
CardiovascularDisease_encoded 2.844915
PersonalityChanges_encoded 2.275388
Diabetes_encoded 1.670498
EducationLevel_encoded 1.553112
dtype: float64
3. ANOVA F-Scores (Top 10):
FunctionalAssessment 257.318763
ADL 219.327424
MemoryComplaints_encoded 199.616266
MMSE 101.512032
BehavioralProblems_encoded 79.965086
SleepQuality 5.240384
PatientID 4.978942
CholesterolHDL 4.667156
CardiovascularDisease_encoded 3.306843
SystolicBP 3.168299
dtype: float64
4. RFE Selected Features (15):
['PatientID', 'BMI', 'AlcoholConsumption', 'PhysicalActivity', 'DietQuality', 'SleepQuality', 'CholesterolTotal', 'CholesterolLDL', 'CholesterolHDL', 'CholesterolTriglycerides', 'MMSE', 'FunctionalAssessment', 'ADL', 'BehavioralProblems_encoded', 'MemoryComplaints_encoded']

```
5. Random Forest Feature Importance (Top 10):
FunctionalAssessment          0.180777
ADL                           0.168314
MMSE                          0.115669
MemoryComplaints_encoded      0.091840
PatientID                     0.045995
BehavioralProblems_encoded    0.043322
DietQuality                   0.028330
PhysicalActivity              0.027073
BMI                           0.026989
CholesterolTriglycerides      0.026312

Features selected by at least 3 methods (13):
['FunctionalAssessment', 'ADL', 'MMSE', 'MemoryComplaints_encoded', 'PatientID',
'BehavioralProblems_encoded', 'AlcoholConsumption', 'CholesterolHDL', 'CholesterolTotal',
'SleepQuality', 'DietQuality', 'EducationLevel_encoded', 'CholesterolTriglycerides']
Final selected features using Random Forest importance:
['FunctionalAssessment', 'ADL', 'MMSE', 'MemoryComplaints_encoded', 'PatientID',
'BehavioralProblems_encoded', 'DietQuality', 'PhysicalActivity', 'BMI',
'CholesterolTriglycerides', 'SleepQuality', 'CholesterolHDL', 'CholesterolLDL',
'AlcoholConsumption', 'CholesterolTotal']
```

Modeling and Grid Search(3-folds)

در اینجا من از مدل های طبقه بندی گفته شده در کلاس استفاده کردم و با انجام grid search روی آن ها بهترین هایپرپارامتر ها را پیدا کردم. با توجه به محدودیت پردازش google collab من مجبور به انتخاب مهمترین هایپرپارامتر ها برای grid search شدم اما با این حال به دقت خوبی رسیدم.

تنظیم و انتخاب بهترین پارامترهای مدل ها با GridSearchCV

در این مرحله، برای افزایش دقت طبقه بندی و بهینه سازی عملکرد مدل‌های مختلف یادگیری ماشین، با استفاده از **GridSearchCV** مجموعه‌ای از پارامترهای مهم هر مدل روی داده های آموزش و با اعتبارسنجی متقاطع (۳-فولدی) بررسی و انتخاب شدند. هدف، شناسایی بهترین تنظیمات (Hyperparameters) برای هر یک از الگوریتم‌ها بود.

پارامترهای تنظیم شده برای هر مدل:

- **درخت تصمیم (Decision Tree):** پارامترها: حداکثر عمق درخت (max_depth)، حداقل نمونه برای تقسیم یا برگ (min_samples_split ,min_samples_leaf)، معیار انشعاب (criterion)
- **جنگل تصادفی (Random Forest):** پارامترها: تعداد درخت ها (n_estimators)، حداکثر عمق، حداقل نمونه برگ

- **نایو بیز (Gaussian Naive Bayes):** پارامتر: مقدار صاف سازی واریانس (var_smoothing)
- **ماشین بردار پشتیبان (SVM):** پارامترها: میزان جریمه (C)، نوع کرنل، پارامتر گاما (gamma)
- **رگرسیون لجستیک:** پارامترها: جریمه (C)، نوع جریمه (penalty)، حل کننده
- **K نزدیک‌ترین همسایه (KNN):** پارامترها: تعداد همسایه‌ها، وزن‌دهی، نوع معیار فاصله

```
Training Decision Tree...
Fitting 3 folds for each of 90 candidates, totalling 270 fits
Best params for Decision Tree: {'criterion': 'entropy', 'max_depth': 5, 'min_samples_leaf': 2, 'min_samples_split': 10}
Best CV score: 0.9471

Training Random Forest...
Fitting 3 folds for each of 8 candidates, totalling 24 fits
Best params for Random Forest: {'max_depth': 10, 'min_samples_leaf': 1, 'n_estimators': 100}
Best CV score: 0.9244

Training Naive Bayes...
Fitting 3 folds for each of 3 candidates, totalling 9 fits
Best params for Naive Bayes: {'var_smoothing': 1e-07}
Best CV score: 0.8237

Training SVM...
Fitting 3 folds for each of 2 candidates, totalling 6 fits
Best params for SVM: {'C': 10, 'gamma': 'scale', 'kernel': 'linear'}
Best CV score: 0.8278

Training Logistic Regression...
Fitting 3 folds for each of 3 candidates, totalling 9 fits
Best params for Logistic Regression: {'C': 10, 'penalty': 'l2', 'solver': 'lbfgs'}
Best CV score: 0.8208

Training K-Nearest Neighbors...
Fitting 3 folds for each of 12 candidates, totalling 36 fits
Best params for K-Nearest Neighbors: {'metric': 'manhattan', 'n_neighbors': 7, 'weights': 'uniform'}
Best CV score: 0.6102
```

تحلیل عملکرد:

- **درخت تصمیم (Decision Tree)** با معیار تقسیم entropy ، عمق ۵ و تنظیمات ارسال شده، بالاترین دقت (۰.۹۴) را نسبت به سایر الگوریتم ها در داده آموزش به دست آورد.
- **جنگل تصادفی (Random Forest)** نیز با عمق ۱۰ و ۱۰۰ درخت، دقت بسیار خوبی (۰.۹۲) داشت و اختلاف کمی با درخت تصمیم داشت؛ وجود چند مدل تصادفی باعث پایداری و کاهش واریانس شد.
- مدل‌های مبتنی بر احتمال **Naive Bayes** و **SVM** هر دو در حوالی دقت حدود ۰.۸۲–۰.۸۳ عملکرد داشتند.
- **رگرسیون لجستیک** با جریمه بیشتر (C=10) کمی پایین تر از SVM نتیجه داد.
- **KNN** با دقت ۰.۶۱ نشان داد که ساختار داده برای این الگوریتم چندان مناسب نیست و احتمالاً داده ها پراکندگی یا نویز دارند.

Model Evaluation

و میرسیم به بخش آخر که همان model evaluation است که با آن میتوان به بهترین مدل دست یافت.

در این بخش، پس از آموزش مدل ها با بهترین پارامترهای به دست آمده از مرحله تنظیم پارامتر ها (Hyperparameter Tuning)، عملکرد نهایی هر مدل روی داده تست به صورت، مورد ارزیابی قرار گرفت. اهداف این ارزیابی عبارت بودند از:

۱) مقایسه دقیق مدل ها از نظر شاخص های مختلف، ۲) انتخاب مدل نهایی برای پیاده سازی و استفاده واقعی.

۱. محاسبه شاخص های ارزیابی مدل ها

برای هر یک از شش مدل، شاخص های اصلی زیر محاسبه و ثبت شد:

- **Accuracy (دقت کلی)**
- **Precision (دقت مثبت)**
- **Recall (حساسیت)**
- **F1-Score (میانگین متوازن دقت و حساسیت)**

نتایج به صورت زیر بود:

Model Evaluation:

Decision Tree:
Accuracy: 0.956
Precision: 0.956
Recall: 0.956
F1-Score: 0.956

Random Forest:
Accuracy: 0.930

Precision: 0.931
Recall: 0.930
F1-Score: 0.929

Naive Bayes:
Accuracy: 0.812
Precision: 0.809
Recall: 0.812
F1-Score: 0.809

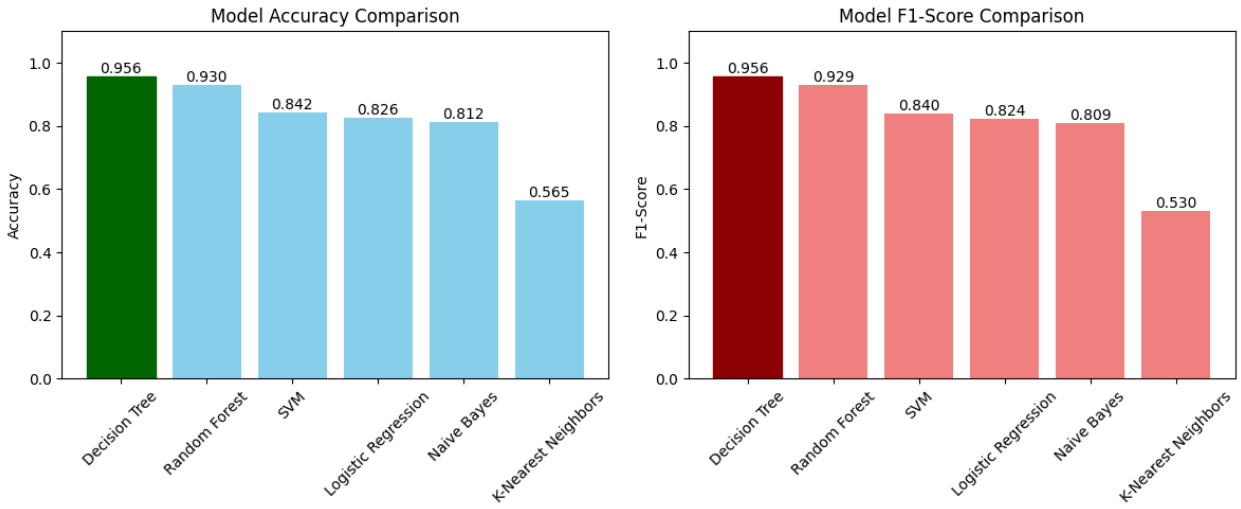
SVM:
Accuracy: 0.842
Precision: 0.840
Recall: 0.842
F1-Score: 0.840

Logistic Regression:
Accuracy: 0.826
Precision: 0.823
Recall: 0.826
F1-Score: 0.824

K-Nearest Neighbors:
Accuracy: 0.565
Precision: 0.517
Recall: 0.565
F1-Score: 0.530

Comparison Table:

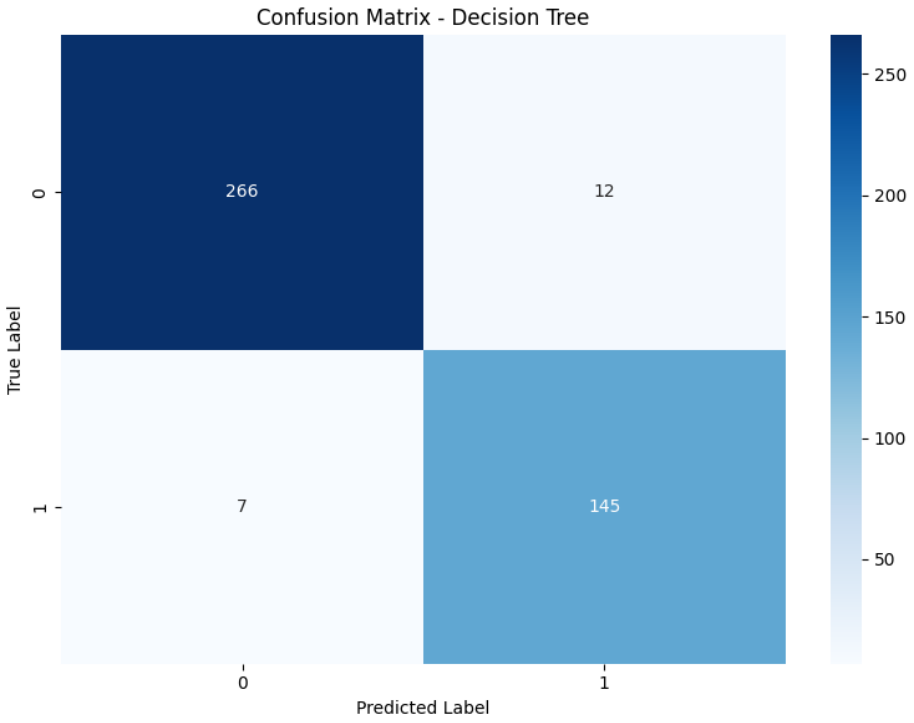
Model	Accuracy	Precision	Recall	F1-Score
Decision Tree	0.956	0.956	0.956	0.956
Random Forest	0.930	0.931	0.930	0.929
SVM	0.842	0.840	0.842	0.840
Logistic Regression	0.826	0.823	0.826	0.824
Naive Bayes	0.812	0.809	0.812	0.809
K-Nearest Neighbors	0.565	0.517	0.565	0.530



مطابق نمودارها، مقایسه عملکرد مدل ها بر اساس دقت و F1-Score به خوبی مشخص است:

- **درخت تصمیم (Decision Tree)** با فاصله قابل توجهی، هم در Accuracy و هم F1-Score، بالاتر از سایر مدل ها قرار دارد.
- مدل **جنگل تصادفی** نیز بسیار نزدیک به مدل برتر قرار دارد.
- سایر مدل ها مانند SVM و رگرسیون لجستیک و Naive Bayes، عملکرد متوسطی داشته اند.
- مدل **KNN** با دقت و F1 پایین، مناسب استفاده عملی نیست.

برای مدل Decision Tree، ماتریس درهم ریختگی به صورت زیر به دست آمد:



- بخش عمده ای از نمونه های هر دو کلاس به درستی پیش‌بینی شده اند.
- تعداد موارد **False Positive** کلاس سالم به اشتباه بیمار پیش بینی شده: ۱۲ مورد
- تعداد موارد **False Negative** کلاس بیمار به اشتباه سالم پیش بینی شده: ۷ مورد
- این اعداد نشان دهنده **ریز خطای بسیار پایین** در هر دو کلاس است.

نتیجه گیری

نتایج تحلیل‌ها نشان داد که مدل **درخت تصمیم (Decision Tree)** با پیاده سازی و تنظیمات مناسب، **توانست بالاترین دقت و کمترین نرخ خطا را در پیش‌بینی افراد مبتلا به آلزایمر در داده‌های تست مستقل ارائه دهد** .مدل جنگل تصادفی (Random Forest)نیز به‌عنوان یک گزینه جایگزین قوی عمل کرد و پایین ترین دقت برای KNN بود که احتمالا ناشی از وجود نویز زیاد است. همچنین در بخش آخر مقدار نمونه هایی misclassified هم آورده ام که ۱۹ عدد بود با rate 4.42% که عدد بسیار خوبی برای این پروژه بود.

با تشکر