"پروژه تحلیل و پیش‌بینی بیماری قلبی با استفاده از یادگیری ماشین"

نام دانشگاه: شهید بهشتی

نام دانشکده: علوم ریاضی

نام دانشجو: مبینا امینی پارسا

نام استاد راهنما: دکتر سکینه دهقان

تاریخ: پاییز ۱۴۰۴

فهرست مطالب

۱. مقدمه

بیماری‌های قلبی یکی از شایع‌ترین دلایل مرگ و میر در سطح جهان هستند و تشخیص به موقع و دقیق آن‌ها اهمیت بسیار زیادی دارد. هدف این پروژه، ساخت یک مدل طبقه‌بندی است که بتواند احتمال ابتلای یک فرد به بیماری قلبی را بر اساس ویژگی‌های بالینی و جمعیتی او پیش‌بینی کند.

با استفاده از روش‌های یادگیری ماشین، داده‌های بیماران مورد بررسی قرار گرفته و مدل‌هایی آموزش داده شده‌اند تا بتوانند دقت، حساسیت و قابلیت اعتماد مناسبی در پیش‌بینی داشته باشند .این کار می‌تواند به تصمیم‌گیری‌های پزشکی کمک کرده و ریسک تشخیص نادرست را کاهش دهد.

۲. اهداف تحقیق

- استفاده از روش‌های مختلف طبقه‌بندی که عبارت اند از :

Logistic Regression ,LDA ,QDA ,Navie Bayes , Decision Tree ،Random Forest ، KNN, SVM

- ارزیابی و مقایسه مدل‌ها با معیارهای:

Accuracy ،Precision ،Recall ،F1 و ROC/AUC

- انتخاب بهترین مدل و بهینه‌سازی آن با استفاده از:

Cross-Validation , Hyperparameter Tuning

- تحلیل اکتشافی داده‌ها و شناسایی ویژگی‌های کلیدی

EDA (Exploratory Data Analysis)

- پیش‌پردازش داده‌ها شامل مقیاس‌بندی ویژگی‌های عددی و تبدیل ویژگی‌های دسته‌ای

۳. معرفی داده ها

برای این پروژه، از دیتاست Heart Disease استفاده شده است که نسخه‌ای از UCI Heart Disease Datasetبوده و در سایت Kaggleدر دسترس است. این دیتاست شامل اطلاعات کلینیکی و جمعیتی بیماران است که هدف آن، پیش‌بینی ابتلای افراد به بیماری قلبی می‌باشد.

ویژگی‌های دیتاست به دو دسته اصلی تقسیم می‌شوند که عبارتند از عددهای پیوسته و ویژگی‌های دسته‌ای . ویژگی‌های عددی شامل سن بیماران(age) ، فشار خون استراحتی(trestbps) ، سطح کلسترول خون(chol) ، حداکثر ضربان قلب ثبت شده در تست ورزش(thalach) ، میزان افت ST در ECG (oldpeak) و تعداد عروق اصلی که با ماده حاجب مشخص شده‌اند (ca) هستند.

ویژگی‌های دسته‌ای شامل جنسیت(sex) ، نوع درد قفسه سینه(cp) ، قند خون ناشتا(fbs) ، وضعیت ECG در استراحت(restecg) ، شیب ST در حین ورزش(slope) ، نوع تالاسمی (thal) و وجود آنژین ناشی از ورزش (exang)هستند.

ستون هدف (target) نشان‌دهنده وضعیت بیماری قلبی هر بیمار است، که عدد ۱ نشان‌دهنده بیمار قلبی و عدد ۰ نشان‌دهنده فرد سالم است. این دیتاست دارای حدود ۳۰۲ نمونه می‌باشد و ترکیبی از ویژگی‌های عددی و دسته‌ای است که نیازمند پیش‌پردازش مناسب مانند مقیاس‌بندی و تبدیل ویژگی‌های دسته‌ای به فرمت قابل استفاده برای مدل‌های یادگیری ماشین است.

داده‌ها کمی نامتوازن هستند؛ برای مثال برخی ویژگی‌های دسته‌ای مانند جنسیت یا قند خون دارای تعداد نمونه‌های مختلف در هر دسته هستند، که در ارزیابی مدل و تحلیل عملکرد آن باید مورد توجه قرار گیرد. استفاده از این دیتاست به ما امکان می‌دهد با روش‌های یادگیری ماشین، مدل‌های طبقه‌بندی برای پیش‌بینی بیماری قلبی بسازیم و عملکرد آن‌ها را با معیارهای مختلف سنجیده و بهینه کنیم.

٤. روش شناسی

- ✓ تحلیل اکتشافی داده ها

در این بخش در ابتدا نگاهی کلی به داده ها انداختیم به این صورت که ابتدا پنج سطر از دیتاست را همراه تمام ۱۴ ستون که ۱۳ ستون ویژگی های دیتاست ما است و ستون اخر متغیر پاسخ یا هدف ما است.

سپس با استفاده از دستور shape مشاهده کردیم که فایل مجموعه داده ۱۰۲۵ مشاهده و ۱۴ ستون دارد، با دستور info متوجه میشویم که در هر ستون چه تعداد مشاهده غیرخالی داریم و تایپ هر متغیر چیست که در دیتاست ما در هر ستون ۱۰۲۵ مشاهده بود که نشان دهنده این است که خانه خالی وجود ندارد و نوع همه متغیر ها عدد صحیح هست بجز ستون oldpeak که از نوع اعداد اعشاری است. و اینجا متوجه میشویم همه متغیر های دسته ای از قبل کدگزاری شده اند.

سپس با دستور describe آماره های توصیفی هر ستون یا متغیر را میتوان مشاهده کرد که برای مثال میتوان با نگاه کردن به مقدار میانگین و میانه متوجه شد که اگر این دو مقدار مساوی باشند یا بهم نزدیک باشند متغیر دارای توزیع نرمال است اگر مقدارهایشان خیلی تفاوت داشته باشد توزیعشان نرمال نیست و همچنین میتوان چارک ها را مشاهده کرد و مقدار حداقل و حداکثر هر متغیر را.

در مرحله بعد کدی را اجرا می کنیم که بررسی کنیم آیا ردیف تکراری در داده ها وجود دارد یا خیر که در دیتاست ما ۷۲۳ ردیف تکراری وجود داشت و آنها را حذف کردیم و در نهایت ۳۰۲ مشاهده با ۱۴ ویژگی بدست آمد.

در مرحله بعد ستون متغیر های دسته ای را با این شرط که هر متغیری که مقدارهای کمتر از ۱۰ دارد را به عنوان متغیر دسته ای تعیین کن مشخص کردیم و با دستور value_counts تعداد مشاهده ها در هر دسته را مشاهده کردیم تا توازن و ناتوازنی دسته ها برای هر ویژگی را بررسی کنیم . به ناتوازنی ویژگی ها نمیتوانیم دست بزنیم چون مدل باید از روی همین ویژگی ها آموزش ببیند تا بتواند در واقعیت عملکرد خوبی داشته باشد اما در متغیر هدف توازن بررسی میشود که اگر ناتوازن بود با روش های موجود ناتوازنی را رفع کنیم که در این دیتاست مشاهده کردیم متغیر هدف ما توازن در دسته هایش برقرار است.

در مرحله بعد تحلیل تک متغیره را انجام می دهیم :

در تحلیل تک متغیره برای تک تک متغیر های عددی نمودار هیستوگرام کشیده میشود زیرا در مرحله بعد یعنی پیش پردازش داده ها بسیار ضروری است به این علت که توزیع متغیر ها و چولگی آنها را بررسی کنیم

که اگر نرمال نباشند و چولگی داشته باشند توزیعشان را تقریبا به نرمال تبدیل کنیم زیرا برخی از الگوریتم های یادگیری ماشین مبتنی بر فرض نرمال بودن هستند مانند LDA,QDA و همچنین در این مرحله باکس پلات های متغیر های عددی هم رسم میشود تا نقاط پرت ر متغیر مشخص شود و برای متغیر های دسته ای نمودار میله ای آنها رسم شد.

همچنین نمودار حرارتی دیتاست هم رسم کردیم تا میزان همبستگی میان متغیر های عددی را به صورت نمودار حرارتی مشاهده کنیم.

- ✓ پیش پردازش داده ها

در این بخش ابتدا باید بررسی شود که داده های گمشده در دیتاست ما وجود دارد یا خیر که سپس با استفاده از روش های جایگزینی داده های گمشده آنها را پر کنیم.

در دیتاست ما داده گمشده ای وجود نداشت بنابراین سراغ مرحله بعد میرویم که تقسیم داده ها به دو مجموعه آموزش و آزمون است این مرحله را در پیش پردازش داده ها قبل از بقیه مراحل برای مثال مقیاس بندی داده ها انجام میدهیم چون مجموعه داده های آزمون ما باید دست نخورده باقی بمانند و تا انتهای مدلسازی نباید آنها را لمس کنیم چون مدل ما فقط باید روی داده های مجموعه آموزش داده ها را یاد بگیرد و اموزش ببیند و برازش دهیم و هیچ اطلاعاتی در مجموعه ازمون ما که در اینجا از ۲۰ درصد داده ها تشکیل شده است نباید لو برود بنابراین ۳۰۲ مشاهده ما ۸۰ درصد یعنی ۲۴۱ مشاهده در مجموعه آموزش و ۲۰ درصد یعنی ۶۱ مشاهده در مجموعه آزمون قرار گرفت و اینکار به صورت تصادفی انجام می شود.

بعد از آنها از روش های مقیاس بندی متغیر ها استفاده می کنیم زیرا برخی متغیر ها دارای بعضی مقادیر خیلی بزرگ یا خیلی کوچک هستند نسبت به سایر مقدار هایی که در آن متغیر است و برخی از الگوریتم های یادگیری ماشین مانند رگرسیون خطی ،رگرسیون لوژستیک ،شبکه های عصبی و تحلیل مولفه های اصلی به مقیاس بندی حساس هستند برای مثال رگرسیون خطی و رگرسیون لوژستیک ، این الگوریتم‌ها بر اساس بهینه‌سازی یک تابع هزینه عمل می‌کنند که معمولاً مجموع مربعات خطا یا Log Lossاست. اگر ویژگی‌ها در مقیاس‌های بسیار متفاوت باشند:

- ویژگی‌هایی با دامنه بزرگ‌تر وزن بزرگ‌تری می‌گیرند، حتی اگر اهمیت واقعی آن‌ها کم باشد.
- این موضوع باعث می‌شود الگوریتم به اشتباه اهمیت برخی ویژگی‌ها را بیش از حد در نظر بگیرد و همگرایی بهینه‌سازی کندتر شود یا حتی به سمت یک مینیمم نامناسب برود.

- استانداردسازی ویژگی‌ها باعث می‌شود همه متغیرها در یک مقیاس مشابه باشند و فرآیند آموزش عادلانه و سریع‌تر انجام شود.

تمام این الگوریتم‌ها به مقیاس داده‌ها حساس هستند چون عملیات ریاضی آن‌ها (گرادیان، کوواریانس، محاسبه فاصله و ...) مستقیماً تحت تأثیر دامنه اعداد قرار می‌گیرد. مقیاس‌بندی باعث می‌شود همه ویژگی‌ها به طور متعادل و با سرعت و دقت بهتر در مدل استفاده شوند.

در نهایت در این پروژه، برای آماده‌سازی دیتاست بیماری قلبی جهت مدل‌سازی، یک خط لوله پیش‌پردازش داده‌ها (Preprocessing Pipeline) طراحی شد که شامل مراحل زیر است:

۱ .شناسایی نوع متغیرها و توزیع آنها

ویژگی‌های عددی و دسته‌ای از هم جدا شدند. برای ویژگی‌های عددی، ابتدا توزیع هر متغیر بررسی شد:

- متغیرهایی که تقریبا نرمال بودند(age, thalach)
- متغیرهایی که چوله بودند و توزیع آنها نامتقارن بود(chol, oldpeak)
- متغیرهایی که دارای داده‌های پرت بودند(trestbps)

۲ .ساخت خط لوله‌های پیش‌پردازش برای هر نوع متغیر

- ویژگی‌های عددی نرمال: با استفاده از StandardScalerاستانداردسازی شدند تا میانگین صفر و انحراف معیار یک داشته باشند. این کار باعث می‌شود الگوریتم‌های حساس به مقیاس مانند رگرسیون خطی و شبکه‌های عصبی به درستی یادگیری کنند.
- ویژگی‌های چوله: ابتدا با PowerTransformer(method='yeo-johnson')به توزیع نزدیک به نرمال تبدیل شدند و سپس با StandardScalerاستانداردسازی شدند. این روش باعث کاهش اثر چولگی و بهبود عملکرد الگوریتم‌هایی که فرض نرمال بودن داده‌ها دارند می‌شود.
- ویژگی‌های دارای داده پرت: از RobustScalerاستفاده شد که نسبت به داده‌های پرت مقاوم است و مقیاس‌بندی را بدون تاثیر شدید از مقادیر دورافتاده انجام می‌دهد.
- ویژگی‌های دسته‌ای: با OneHotEncoderبه فرم عددی تبدیل شدند تا برای مدل‌های یادگیری ماشین قابل استفاده باشند و همچنین گزینه handle_unknown='ignore'تضمین می‌کند که اگر در مجموعه آزمون دسته‌ای جدید دیده شود، باعث خطا نشود.

۳ .ترکیب خط لوله‌ها

تمام این خط لوله‌ها با استفاده از ColumnTransformerترکیب شدند تا یک پیش‌پردازنده یکپارچه ایجاد شود که بتواند به صورت همزمان همه ویژگی‌ها را بر اساس نوع و خصوصیاتشان پردازش کند. این کار باعث می‌شود پیش‌پردازش داده‌ها سیستماتیک و قابل بازتولید باشد.

۴ .آموزش و تبدیل داده‌ها

- پیش‌پردازنده ابتدا روی مجموعه آموزش آموزش داده شد (fit) تا پارامترهای مقیاس‌بندی و ترنسفورم‌ها بر اساس داده‌های آموزش محاسبه شوند.
- سپس همان پیش‌پردازنده برای تبدیل مجموعه آموزش و مجموعه آزمون استفاده شد (transform) تا تضمین شود که مجموعه آزمون دست‌نخورده باقی بماند و اطلاعات آن در آموزش مدل لو نرود.

۵ .خروجی پیش‌پردازش

- پس از پیش‌پردازش، داده‌ها به فرم عددی و استاندارد درآمدند و تمام ویژگی‌ها در مقیاس مشابه قرار گرفتند.
- ویژگی‌های دسته‌ای نیز به فرم عددی باینری در آمدند و تعداد کل ویژگی‌ها پس از پیش‌پردازش افزایش یافت که آماده استفاده در مدل‌های یادگیری ماشین شد.

چرا این روش‌ها انتخاب شدند:

- رعایت توزیع داده‌ها و مدیریت چولگی باعث افزایش دقت مدل و بهبود همگرایی الگوریتم‌ها می‌شود.
- استانداردسازی ویژگی‌ها، عملکرد مدل‌های حساس به مقیاس مانند رگرسیون و شبکه‌های عصبی را بهبود می‌بخشد.
- استفاده از RobustScalerاز تاثیر داده‌های پرت جلوگیری می‌کند.
- تبدیل ویژگی‌های دسته‌ای با OneHotEncoderامکان استفاده از آن‌ها در الگوریتم‌های ریاضی-عددی را فراهم می‌کند.
- ترکیب همه مراحل در یک خط لوله قابل بازتولید بودن و امنیت داده‌ها را تضمین می‌کند.

- ✓ مدل سازی

در این بخش در این پروژه برای پیش‌بینی بیماری قلبی، مجموعه‌ای از الگوریتم‌های یادگیری ماشین قابل استناد و رایج مورد استفاده قرار گرفت. برای هر الگوریتم، یک خط لوله (Pipeline) ساخته شد که شامل پیش‌پردازش داده‌ها و مدل یادگیری ماشین بود. استفاده از Pipeline باعث شد که پیش‌پردازش و آموزش مدل به‌صورت یکپارچه و قابل بازتولید انجام شود و هیچ اطلاعاتی از مجموعه آزمون در حین آموزش لو نرود. الگوریتم‌های مورد استفاده شامل موارد زیر هستند:

Logistic Regression: یک مدل خطی برای طبقه‌بندی که احتمال وقوع یک کلاس را پیش‌بینی می‌کند.

LDA , QDA : مدل‌های آماری کلاسیک که برای تفکیک کلاس‌ها و مدل‌سازی توزیع شرطی ویژگی‌ها استفاده می‌شوند.

Naive Bayes: یک مدل مبتنی بر احتمال که فرض می‌کند ویژگی‌ها شرطی بر کلاس مستقل هستند.

KNN : یک الگوریتم مبتنی بر فاصله که کلاس هر نمونه جدید را بر اساس نزدیک‌ترین نمونه‌ها تعیین می‌کند.

Decision Tree , Random Forest : مدل‌های درختی که روابط غیرخطی و تعامل بین ویژگی‌ها را می‌توانند به خوبی یاد بگیرند.

SVM : یک الگوریتم قدرتمند برای جداسازی کلاس‌ها با پیدا کردن ابرصفحه‌ای که بیشترین فاصله را بین کلاس‌ها داشته باشد.

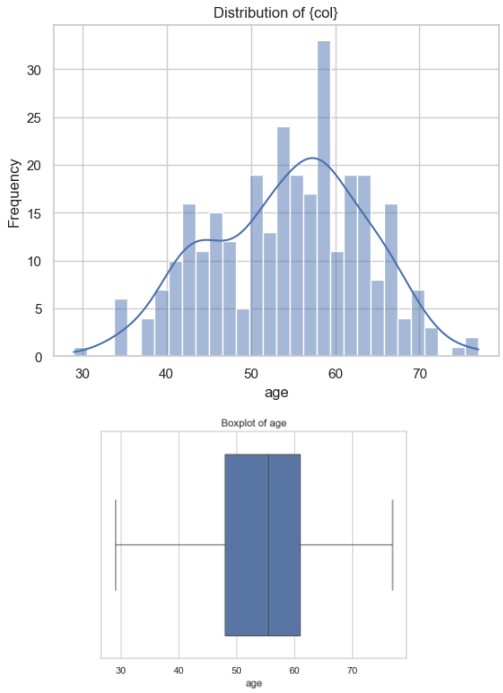
پس از تعریف خط لوله‌ها، تمام مدل‌ها روی مجموعه آموزش برازش داده شدند و پس از آموزش، برای هر مدل پیش‌بینی‌های کلاس‌ها و احتمال وقوع کلاس مثبت روی مجموعه آزمون انجام شد. این روش امکان مقایسه عملکرد الگوریتم‌ها و تحلیل نتایج را فراهم می‌کند و تضمین می‌کند که فرآیند مدل‌سازی سیستماتیک، قابل بازتولید و بدون نشت اطلاعات از مجموعه آزمون باشد.

سپس با استفاده از روش ارزیابی عملکرد مدل به نام cross-validation عملکرد مدل ها را ارزیابی کردیم یا به عبارتی برآورد خطای تست را برای هر مدل بدست اوردیم که در ادامه نتایج آمده است و همچنین در این روش از K-Fold Cross-Validation با K=10 استفاده کردیم که یکی از بهترین روش های ارزیابی عملکرد مدل است و به این سوال پاسخ می دهد که آیا عملکرد مدل وابسته به یک Train/Test خاص است یا واقعا پایدار است؟

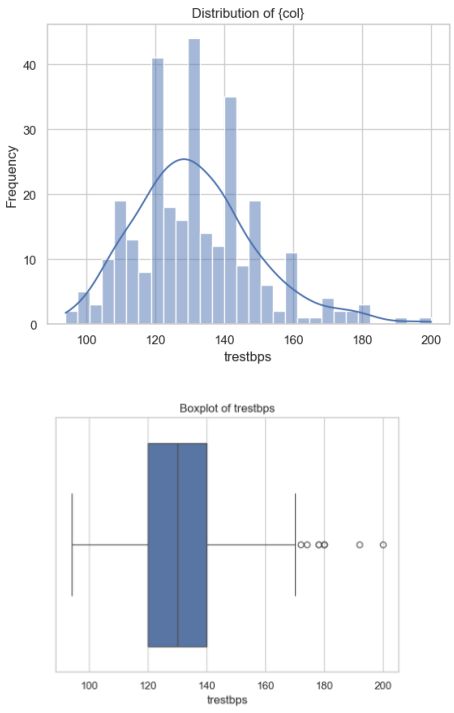
در نهایت تنظیم هایپرپارامترها انجام شد. به منظور بهبود عملکرد مدل‌ها و جلوگیری از بیش‌برازش، فرآیند تنظیم هایپرپارامترها با استفاده از روش Grid Searchهمراه با اعتبارسنجی متقاطع انجام شد. در این راستا، از الگوریتم GridSearchCVاستفاده گردید که با بررسی سیستماتیک تمام ترکیب‌های ممکن از هایپرپارامترهای از پیش تعیین‌شده، بهترین ترکیب را بر اساس معیار ارزیابی انتخاب می‌کند. تنظیم هایپرپارامترها بر روی مدل‌های منتخب شامل رگرسیون لجستیک، جنگل تصادفی و K نزدیک‌ترین همسایه انجام شد و در نهایت، مدلی با بالاترین میانگین دقت در فرآیند اعتبارسنجی به‌عنوان مدل بهینه انتخاب گردید.

استفاده از GridSearchCV به جای RandomizedSearchCV به دلیل محدود بودن فضای هایپرپارامترها و قابل تفسیر بودن فرآیند جستجو انتخاب شد، به‌گونه‌ای که امکان بررسی دقیق اثر هر پارامتر بر عملکرد مدل فراهم گردد.

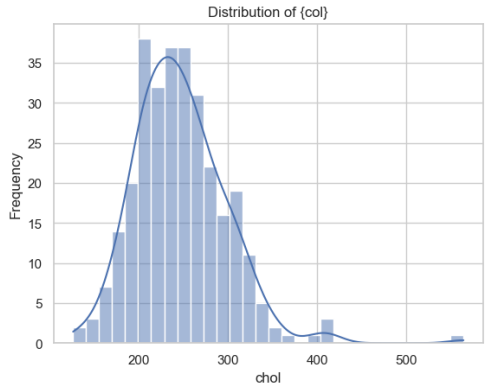
۵. نتایج و آزمایشات تجربی

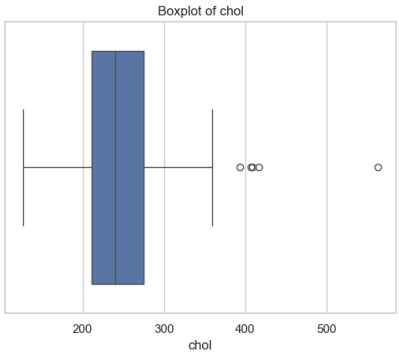


در نمودار هیستوگرام توزیع متغیر سن ،توزیع سن تقریبا تک قله ای است. قله اصلی توزیع حدود ۵۵ تا ۶۰ سال قرار دارد تفسیر علمی آن بیان میکند که تمرکز اصلی نمونه ها در بازه میانسالی تا سالمندی اولیه است ،که با ماهیت دیتاست بیماری قلبی سازگار است. توزیع کاملا نرمال نیست اما به نرمال نزدیک است. کمی چولگی راست دیده می شود زیرا دم توزیع در سمت سنین بالاتر (۷۰+) کشیده تر است. تفسیر علمی آن بیان میکند که چولگی خفیف یعنی تعداد افراد با سن بالا کمتر اما پراکندگی آن ها بیشتر است. حداقل سن ۳۰ و حداکثر سن ۷۵ سال و بیشترین تراکم داده ها بین سن ۴۵ تا ۶۰ سال است که تفسیر علمی آن بیان میکند که داده ها دامنه سنی وسیعی دارند، اما تمرکز اصلی روی بازه ای خاص است که برای تحلیل بیماری قلبی منطقی است. در هیستوگرام توزیع پیوسته است و شکاف غیرطبیعی ندارد و نبود پرت های شدید نشان می دهد که متغیر سن نیازی به روش های مقاوم به پرت مانند RobustScaler ندارد. با توجه به این ویژگی‌ها، متغیر سن برای ورود مستقیم به مدل‌های یادگیری ماشین مناسب بوده و استفاده از روش استانداردسازی (StandardScaler) برای آن کفایت می‌کند.

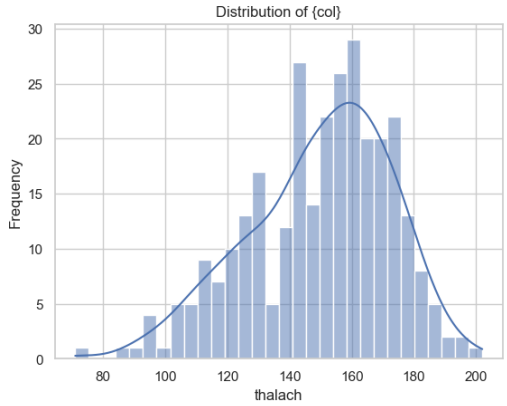


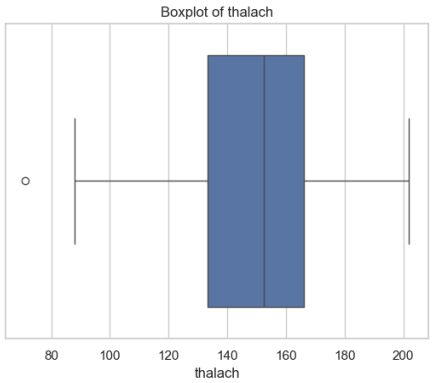
در نمودار هیستوگرام توزیع متغیر فشار خون در حالت استراحت ،توزیع این متغیر تک قله ای است. قله توزیع حدود فشار خون ۱۲۰ تا ۱۴۰ قرار دارد یعنی تمرکز اصلی فشار خون نمونه ها در این بازه است. توزیع کاملا نرمال نیست و دارای چولگی راست است زیرا دم توزیع از فشارخون ۱۷۰ به بالا کشیده شده است یعنی تعداد کمی از افراد فشار خون بسیار بالا دارند، که ممکن است داده های پرت یا گروه خاصی باشند حداقل فشارخون ۹۰ و حداکثر ۲۰۰ و بیشترین تراکم بین ۱۲۰ تا ۱۵۰ است. دامنه وسیع است ولی تمرکز اصلی روی فشار های نسبتا بالا قرار دارد که برای تحلیل بیماری قلبی منطقی است. به دلیل چولگی و احتمال وجود داده های پرت RobustScaler نسبت به StandardScaler انتخاب بهتری است.



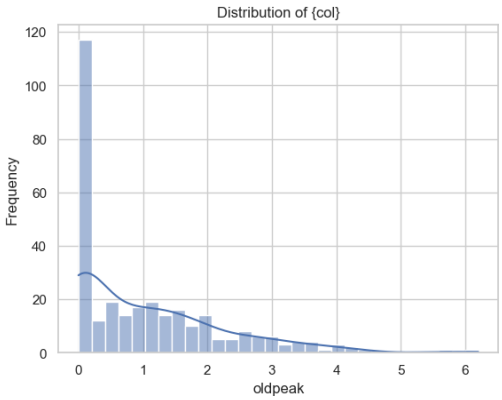


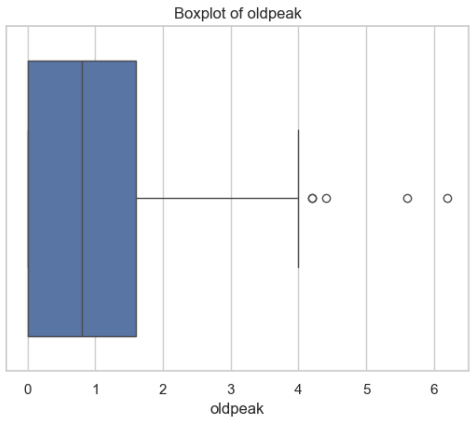
در نمودار هیستوگرام توزیع متغیر سطح کلسترول ،توزیع متغیر تک قله ای با کشیدگی زیاد است. قله اصلی در بازه ۲۰۰ تا ۲۵۰ است. دم توزیع به سمت راست کشیده شده است یعنی سطوح کلسترول بالا که تفسیر آن عبارت است از اکثر افراد سطح کلسترول متوسط دارند ،اما تعداد کمی با کلسترول بسیار بالا (بالای ۴۰۰) وجود دارند که باعث کشیدگی توزیع شده اند. توزیع نرمال نیست و چولگی مثبت شدید دارد این چولگی می تواند روی مدل های آماری تاثیر منفی بگذارد. حداقل کلسترول ۱۰۰ و حداکثر بالای ۵۰۰ و بیشترین تراکم بین ۲۰۰ تا ۳۰۰ است. دامنه وسیع است و پراکندگی زیاد نشان دهنده تنوع بالا در سطح کلسترول افراد است. وجود داده های پرت شدید در کلسترول بالا می تواند باعث نرمال نبودن توزیع و کاهش دقت مدل شود . برای مقیاس بندی این متغیر از روش PowerTransformer + StandardScaler استفاده شود بهتر است.





در نمودار هیستوگرام توزیع متغیر حداکثر ضربان قلب ثبت شده در تست ورزش ،توزیع این متغیر تک قله ای و نسبتا متقارن است. قله اصلی در بازه ۱۴۰ تا ۱۷۰ قرار دارد. منحنی KDE شکل زنگوله ای دارد و به نرمال نزدیک است. تفسیر علمی آن بیان می کند اکثر افراد در تست ورزش ضربان قلبی بین ۱۴۰ تا ۱۷۰ داشته اند که نشان دهنده عملکرد قلبی نسبتا نرمال در جمعیت مورد بررسی است. توزیع نزدیک به نرمال است و چولگی بسیار کم و تقریبا صفر است یعنی این ویژگی برای مدل سازی بسیار مناسب است چون توزیع نرمال باعث می شود مدل های آماری بهتر عمل کنند. مقدار حداقل برای این متغیر ۷۰ حداکثر ۲۰۰ و بیشترین تراکم بین ۱۴۰ تا ۱۷۰ است یعنی دامنه نسبتا وسیع است ولی تراکم روی محدوده ای است که با عملکرد قلبی سالم در تست ورزش همخوانی دارد . نبود داده های پرت شدید نشان می دهد که این ویژگی نیازی به روش هالی مقاوم به پرت ندارد. با توجه به این ویژگی ها، مقیاس بندی مناسب آن StandardScaler است.

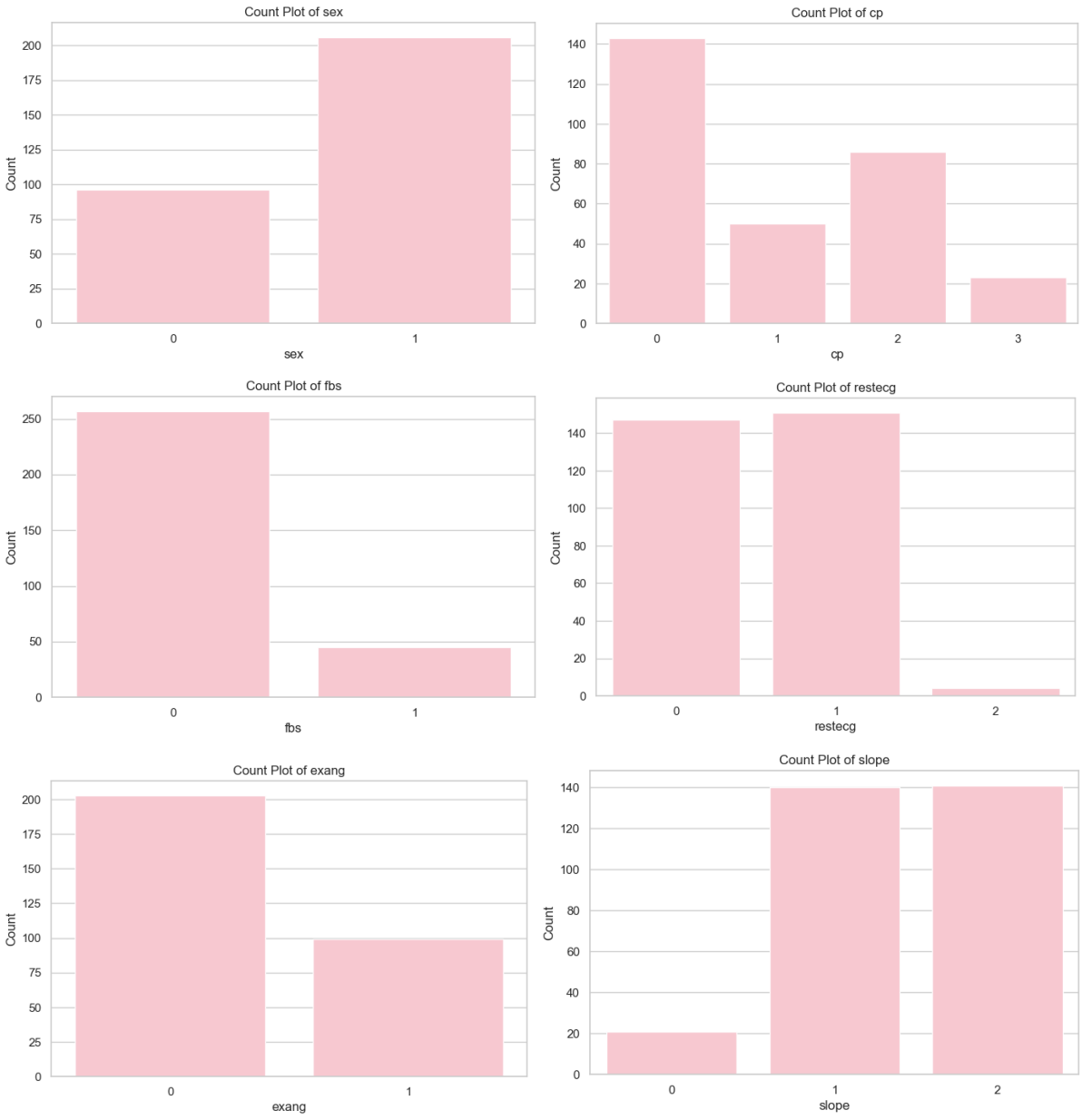


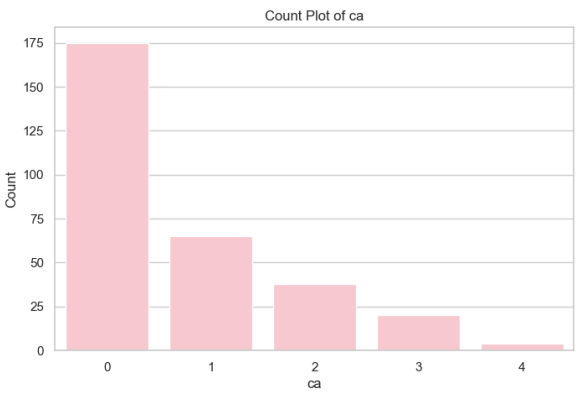


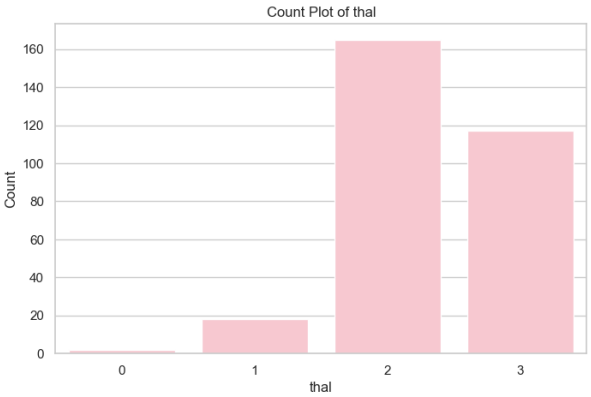
نمودار هیستوگرام توزیع متغیر افت در تست ورزش ،توزیع این متغیر توزیع تک قله ای با چولگی شدید به راست است . قله اصلی در بازه ۰ تا ۱ قراردارد. منحنی KDE به وضوح به سمت راست کشیده شده است که تفسیر علمی آن بیان می کند ST اکگثر افراد افت کمی دارند اما تعداد کمی با افت شدید وجود دارند که باعث کشیدگی توزیع شده اند. توزیع نرمال نیست و چولگی شدید مثبت دارد که یعنی ST بیشتر افراد افت کمی دارند ولی تعداد کمی با افت بالا (مثلا ST بالای ۴) وجود دارند که ممکن است داده های پرت یا بیماران خاص باشند. حداقل این ویژگی ۰ و حداکثر حدود ۶.۵ و بیشترین تراکم بین ۰ تا ۱.۵ است که یعنی دامنه وسیع است ولی تمرکز اصلی روی افت های کم قرار دارد، که در جمعیت عمومی قابل انتظار است. در سمت راست نمودار ۴ نمونه پرت وجود دارد که می تواند باعث کاهش دقت مدل شود بنابراین برای مقیاس بندی مناسب انجام شود که PowerTransformer + StandardScaler

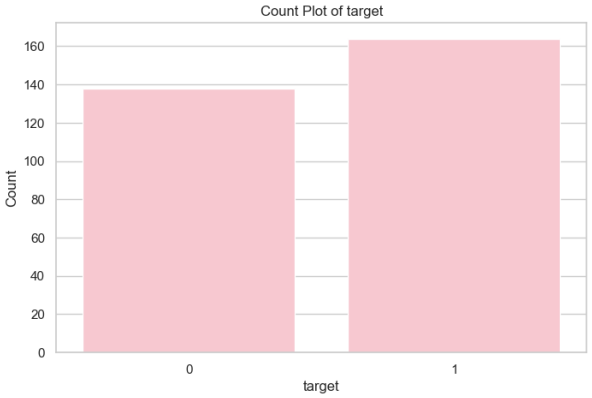
بهترین انتخاب است.

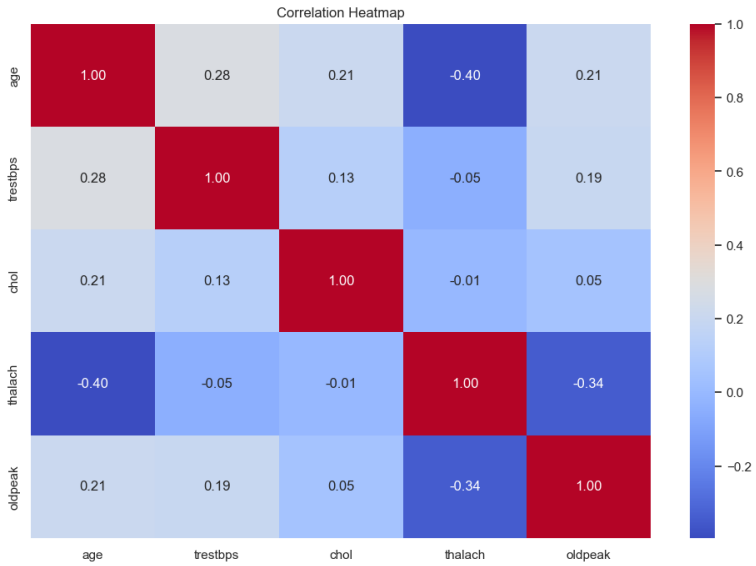
نمودارهای متغیرهای کیفی :



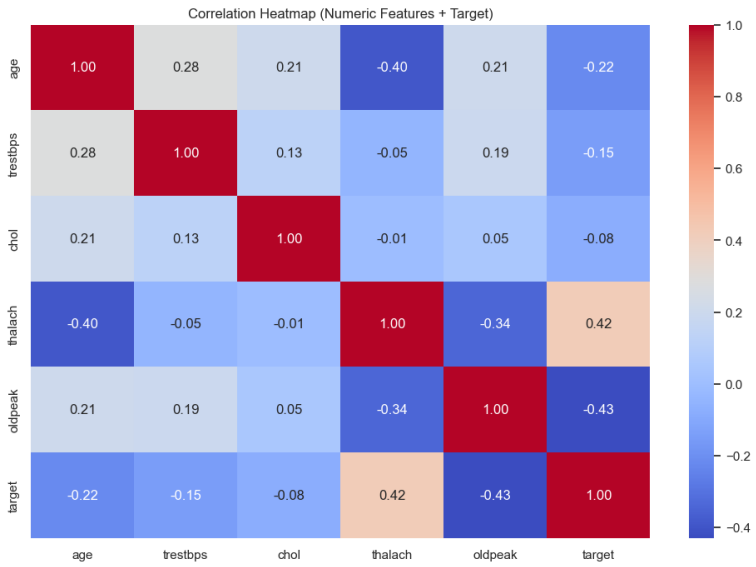

Count Plot of ca
Count
ca
0
1
2
3
4
0
25
50
75
100
125
150
175


Count Plot of thal
Count
thal
0
1
2
3
0
20
40
60
80
100
120
140
160


Count Plot of target
Count
target
0
1
0
20
40
60
80
100
120
140
160



نمودار حرارتی یا نقشه حرارتی همبستگی ، این نمودار به ما نشان می دهد که که کدام ویژگی ها با هم رابطه دارند و چقدر این رابطه قوی یا ضعیف است. در این نمودار نمایش ضریب همبستگی بین ویژگی های عددی و رابطه خطی مثبت یا منفی مشاهده می شود. نقشه حرارتی همبستگی بین ویژگی‌های عددی نشان می‌دهد که متغیر age با thalach همبستگی منفی قابل توجهی دارد (-۰.۴۰)، که نشان‌دهنده کاهش ضربان قلب با افزایش سن است. همچنین، thalach با oldpeak نیز همبستگی منفی دارد (-۰.۳۴)، که می‌تواند نشان‌دهنده ارتباط فیزیولوژیکی بین عملکرد قلب و افت ST باشد. سایر روابط مانند age با trestbps (۰.۲۸) نیز قابل توجه هستند. این تحلیل به انتخاب ویژگی‌های مؤثر در مدل‌سازی کمک کرده و از هم‌خطی بین ویژگی‌ها جلوگیری می‌کند.



نمودار حرارتی مقابل همبستگی بین ویژگی های عددی و متغیر هدف را نشان میدهد که برای EDA این مرحله یکی از مهم ترین بخش های انتخاب ویژگی های موثر در مدل سازی است. ویژگی هایی با همبستگی بالا با متغیر هدف عبارت اند از ضربان قلب که با سلامت قلب مرتبط است و افت ST بیشتر معمولا نشانه بیماری قلبی است و افراد مسن تر بیشتر در معرض بیماری هستند و فشار خون بالا ممکن است با بیماری مرتبط باشد و همچنین متوجه میشویم کلسترول ارتباط ضعیفی با بیماری قلبی دارد.

پس از برازش مدل ها روی مجموعه آموزش برای یادگیری مدل، پیشبینی را با استفاده از مدل های برازش داده شده، حال روی مجموعه تست برازش داده می شوند بدست می آوریم و نتایج به دست امده عبارت است از ستون اول شمارش ،ستون دوم مقدار واقعی ،ستون سوم مقدار پیشبینی هر مدل برای متغیر هدف و ستون چهارم احتمال تعلق هر مشاهده به کلاس مثبت :

```
=== LogisticRegression – Example Predictions (first 20 rows) ===
    Actual  Predicted  Predicted_Probability
0        0          0               0.192612
1        0          1               0.520765
2        0          0               0.004224
3        0          1               0.618032
4        0          1               0.950852
5        0          0               0.010422
6        1          1               0.847458
7        0          0               0.092787
8        1          1               0.983297
9        0          0               0.273636
10       1          1               0.940230
11       1          1               0.638497
12       0          1               0.620576
13       1          1               0.952971
14       1          0               0.092183
15       1          1               0.947591
16       1          1               0.909764
17       1          0               0.058218
18       1          1               0.994779
19       1          1               0.709575
```

```
=== LDA – Example Predictions (first 20 rows) ===
    Actual  Predicted  Predicted_Probability
0        0          0               0.158923
1        0          1               0.721797
2        0          0               0.001382
3        0          1               0.723068
4        0          1               0.974540
5        0          0               0.006847
6        1          1               0.935042
7        0          0               0.065127
8        1          1               0.994925
9        0          0               0.488628
10       1          1               0.973819
11       1          1               0.863587
12       0          1               0.776209
13       1          1               0.986404
14       1          0               0.082204
15       1          1               0.973940
16       1          1               0.955850
```

```
17       1          0               0.049246
18       1          1               0.998551
19       1          1               0.802934
```

```
=== QDA – Example Predictions (first 20 rows) ===
    Actual  Predicted  Predicted_Probability
0        0          0           1.970247e-02
1        0          1           9.998854e-01
2        0          0           5.006765e-08
3        0          1           9.789784e-01
4        0          1           9.981795e-01
5        0          0           8.726803e-05
6        1          1           9.985659e-01
7        0          0           3.932774e-03
8        1          1           9.999557e-01
9        0          1           6.642431e-01
10       1          1           9.999952e-01
11       1          0           4.514766e-02
12       0          1           9.999568e-01
13       1          1           9.999947e-01
14       1          0           1.229960e-03
15       1          1           9.999849e-01
16       1          1           9.943022e-01
17       1          0           1.967104e-04
18       1          1           9.999988e-01
19       1          1           9.899097e-01
```

```
=== NaiveBayes – Example Predictions (first 20 rows) ===
    Actual  Predicted  Predicted_Probability
0        0          0           8.170122e-05
1        0          1           9.997260e-01
2        0          0           4.360781e-12
3        0          1           9.162103e-01
4        0          1           9.999688e-01
5        0          0           2.292256e-08
6        1          1           9.999207e-01
7        0          0           3.685003e-07
8        1          1           9.999641e-01
9        0          1           9.344149e-01
10       1          1           9.999036e-01
11       1          0           1.445683e-01
12       0          1           8.799397e-01
13       1          1           1.000000e+00
14       1          0           9.219633e-04
15       1          1           1.000000e+00
16       1          1           9.971304e-01
17       1          0           1.728254e-05
18       1          1           1.000000e+00
19       1          1           9.803359e-01
```

```
=== KNN – Example Predictions (first 20 rows) ===
    Actual  Predicted  Predicted_Probability
0        0          0                    0.0
1        0          0                    0.4
2        0          0                    0.0
3        0          0                    0.4
4        0          1                    1.0
5        0          0                    0.0
6        1          1                    0.8
7        0          0                    0.2
8        1          1                    1.0
9        0          0                    0.4
10       1          1                    0.8
11       1          0                    0.0
12       0          1                    0.6
13       1          1                    1.0
14       1          0                    0.2
15       1          1                    1.0
16       1          1                    0.8
17       1          0                    0.0
18       1          1                    1.0
19       1          1                    1.0
```

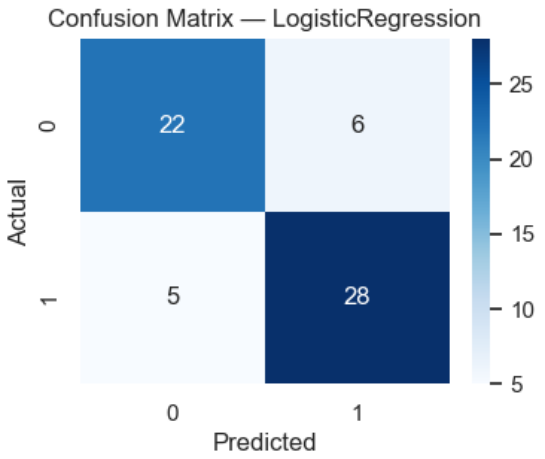
```
=== DecisionTree – Example Predictions (first 20 rows) ===
    Actual  Predicted  Predicted_Probability
0        0          0                    0.0
1        0          0                    0.0
2        0          0                    0.0
3        0          1                    1.0
4        0          1                    1.0
5        0          0                    0.0
6        1          1                    1.0
7        0          0                    0.0
8        1          1                    1.0
9        0          0                    0.0
10       1          1                    1.0
11       1          1                    1.0
12       0          1                    1.0
13       1          1                    1.0
14       1          1                    1.0
15       1          1                    1.0
16       1          1                    1.0
17       1          0                    0.0
18       1          1                    1.0
19       1          1                    1.0
```

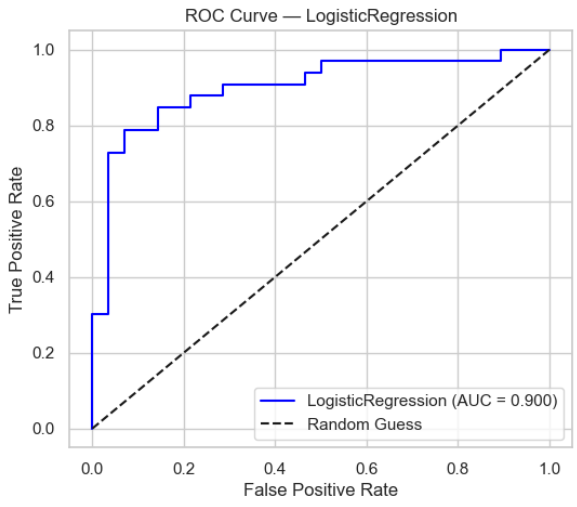
```
=== RandomForest – Example Predictions (first 20 rows) ===
    Actual  Predicted  Predicted_Probability
0        0          0                   0.06
1        0          1                   0.77
2        0          0                   0.03
3        0          1                   0.67
4        0          1                   0.93
5        0          0                   0.07
6        1          1                   0.89
7        0          0                   0.10
8        1          1                   0.90
9        0          0                   0.44
10       1          1                   0.91
11       1          0                   0.39
12       0          1                   0.55
13       1          1                   1.00
14       1          0                   0.32
15       1          1                   0.89
16       1          1                   0.87
17       1          0                   0.21
18       1          1                   1.00
19       1          1                   0.72
```

```
=== SVM – Example Predictions (first 20 rows) ===
    Actual  Predicted  Predicted_Probability
0        0          0               0.107286
1        0          1               0.788331
2        0          0               0.018806
3        0          1               0.790526
4        0          1               0.951620
5        0          0               0.099951
6        1          1               0.910778
7        0          0               0.249364
8        1          1               0.968129
9        0          0               0.395840
10       1          1               0.891733
11       1          0               0.331592
12       0          1               0.627850
13       1          1               0.958620
14       1          0               0.197365
15       1          1               0.899576
16       1          1               0.899101
17       1          0               0.117316
18       1          1               0.978749
19       1          1               0.757279
```

در این بخش، عملکرد مدل‌های مختلف یادگیری ماشین روی مجموعه آزمون ارزیابی می شود. برای این منظور از معیارهای دقت، Precision، Recall، F1-score، ماتریس سردرگمی و منحنی ROC به همراه معیار AUC استفاده می شود.

```
--- LogisticRegression ---

Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.79      0.80        28
           1       0.82      0.85      0.84        33

    accuracy                           0.82        61
   macro avg       0.82      0.82      0.82        61
weighted avg       0.82      0.82      0.82
```





بر اساس گزارش طبقه بندی، مدل رگرسیون لوژستیک دقت کلی بالایی (۸۲٪) نشان می دهد، اما بررسی معیار recall که معیار پوشش است و تمرکز اصلی آن روی داده هایی است که واقعا مثبت بوده اند برای کلاس مثبت نشان می دهد که توانایی مدل در شناسایی بیماران واقعی مناسب تر از معیار precision که معیار صحت است و تمرکز اصلی آن بر روی درستی تشخیص های مثبت توسط الگوریتم است. این موضوع نشان می دهد که مدل تمایل بیشتری به تولید مثبت کاذب دارد تا منفی کاذب. معیار f1-score تابعی از recall و precision است که میزان تعادل در مدل را نشان می دهد.

در پزشکی recall بالا مهم است زیرا به معنای از دست ندادن بیمار است. Precision بالا وقتی هزینه مثبت کاذب بالاست مهم است.

ماتریس سردرگمی نشان می دهد:

True Positive (TP) = 28

True Negative (TN) = 22

False Positive (FP) = 6

False Negative (FN) = 5

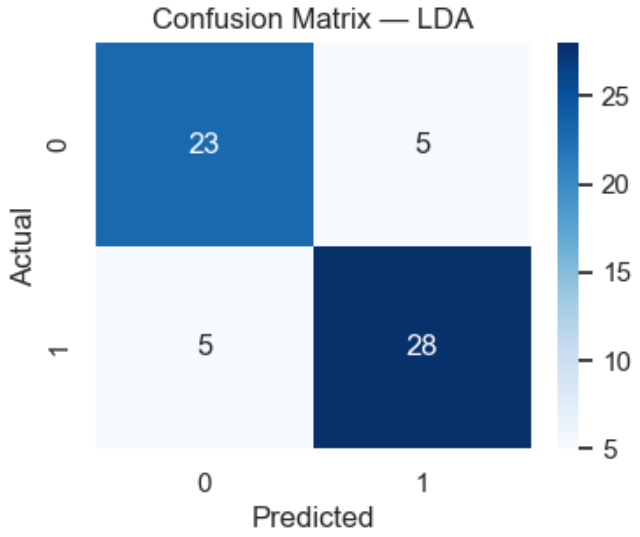
ماتریس سردرگمی نشان می دهد که تعداد خطاهای منفی کاذب کمتر از مثبت های کاذب است. این ویژگی در مسئله تشخیص بیماری قلبی اهمیت بالایی دارد، زیرا شناسایی نکردن بیمار واقعی می تواند پیامد های جدی تری نسبت به تشخیص اشتباه بیماری داشته باشد.

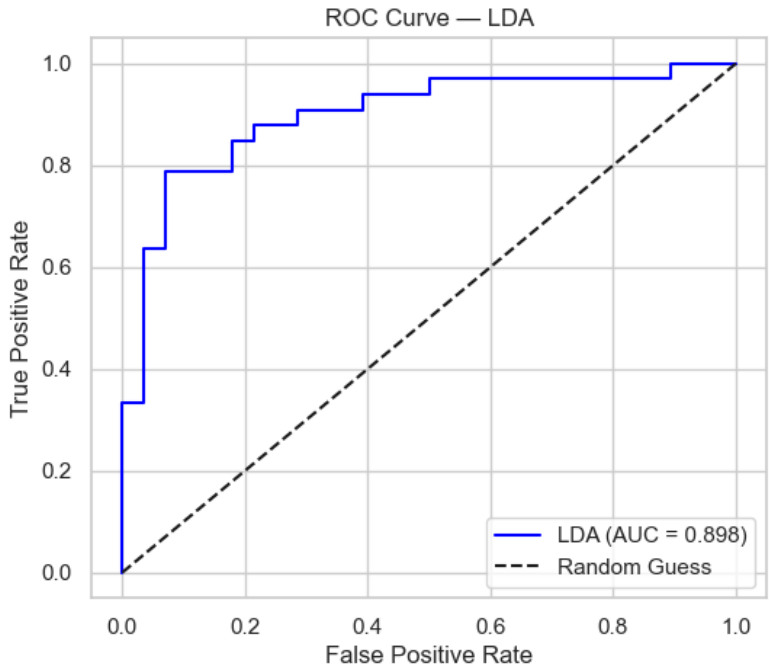
منحنی ROC به این صورت است که هرچه منحنی به گوشه بالا-چپ نزدیک تر باشد مدل بهتر است و در مقایسه بصری مدل ها بسیار مهم است. در مدل رگرسیون لوژستیک منحنی بالا به میزان خوبی به گوشه بالا-چپ نزدیک است.

منحنی ROC مدل رگرسیون لوژستیک نشان می دهد که مدل رگرسیون لوژستیک در اکثر آستانه ها نرخ تشخیص صحیح بالایی دارد که در ادامه میتوانیم این نرخ تشخیص صحیح را نسبت به سایر مدل ها مقایسه کنیم .

مقدار AUC که مساحت زیر منحنی ROC است برای این مدل برابر با ۰.۹۰ بسیار عالی است و بیانگر قدرت تفکیک بالای این مدل بین بیماران و افراد سالم است.

```
--- LDA ---

Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.82      0.82        28
           1       0.85      0.85      0.85        33

    accuracy                           0.84        61
   macro avg       0.83      0.83      0.83        61
weighted avg       0.84      0.84      0.84        61
```





بر اساس گزارش طبقه بندی، مدل LDA یا تحلیل ممیزی خطی دقت کلی بالا، ۸۴٪ را نشان می دهد، اما بررسی معیار recall برای کلاس مثبت با معیار precision برای کلاس مثبت باهم برابر است. وقتی این دو برابر باشند یعنی مدل نه تنها بیماران واقعی را خوب شناسایی میکند بلکه اشتباهاتش در پیش بینی مثبت هم کم هستند. این تعادل نشان می دهد که مدل در تشخیص کلاس مثبت که بیماران هستند هم حساس و هم دقیق است.

Recall بالا یعنی مدل اکثر بیماران واقعی را پیدا کرده است.

Precision بالا یعنی اکثر مواردی که مدل گفته "بیمار هست" واقعا بیمار بوده اند.

تفسیر دقیق تر عملکرد این مدل می گوید مدل نه بیش از حد محافظه کار است که فقط موارد مطمئن را بیمار اعلام کند و خیلی ها را از دست بدهد و نه بیش از حد جسور است که موارد زیادی را بیمار اعلام کند ولی اشتباه زیاد داشته باشد.

ماتریس سردرگمی نشان می دهد:

True Positive (TP) = 28

True Negative (TN) = 23

False Positive (FP) = 5

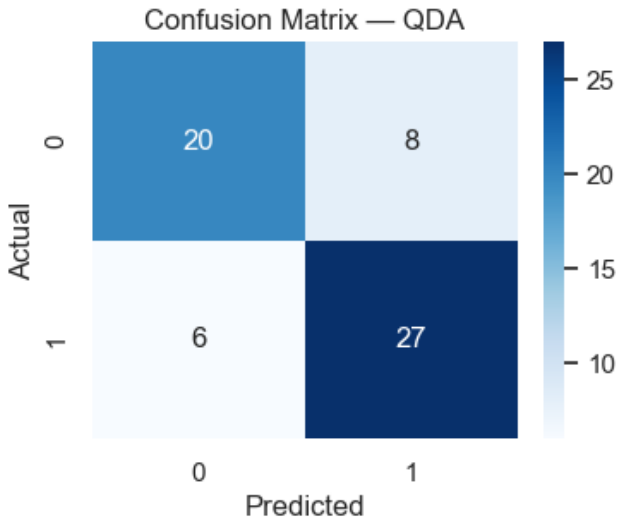
False Negative (FN) = 5

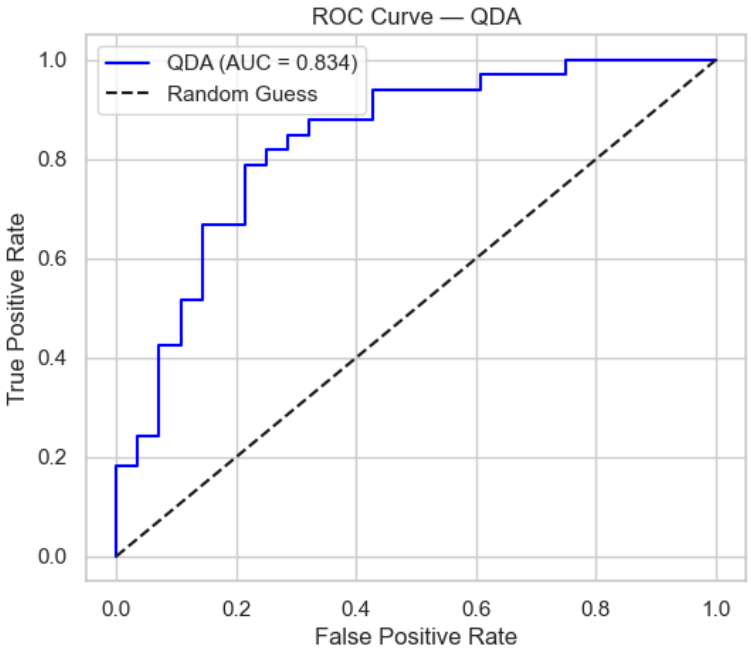
ماتریس سردرگمی نشان می دهد تعداد خطای منفی کاذب با تعداد خطای مثبت کاذب باهم برابر است و مدل در دو جنبه ی اصلی تشخیص کلاس مثبت یعنی بیماران نوعی تعادل برقرار کرده است یعنی مواردی که بیمار واقعی بوده ولی مدل نتونسته تشخیص دهد با مواردی که فرد سالم بوده است ولی مدل اشتباها بیمار پیش بینی کرده باهم برابر است وقتی این دو برابر باشند یعنی مدل به همان اندازه ای که بیماران واقعی را از دست می دهد که حساسیت پایین تر است ، به همان اندازه هم افراد سالم را اشتباه بیمار اعلام می کند که اختصاصیت پایین تر است. این تعادل می تواند به عنوان نشانه ای از بی طرفی مدل در خطاها ذکر شود.

منحنی ROC نشان می دهد که مدل LDA در اکثر آستانه ها نرخ تشخیص صحیح بالایی دارد.

مقدار AUC که مساحت زیر نمودار ROC است برابر با ۰.۸۹۸ خوب است و بیانگر قدرت تفکیک بالای این مدل بین بیماران و افراد سالم است.

```
--- QDA ---

Classification Report:
              precision    recall  f1-score   support

           0       0.77      0.71      0.74        28
           1       0.77      0.82      0.79        33

    accuracy                           0.77        61
   macro avg       0.77      0.77      0.77        61
weighted avg       0.77      0.77      0.77        61
```





بر اساس گزارش طبقه بندی، مدل QDA یا تحلیل ممیزی درجه دوم به دقت کلی ۷۷٪ روی مجموعه آزمون دست یافته است.

Recall نسبتا بالا برای کلاس مثبت نشان می دهد که مدل QDA توانایی مناسبی در شناسایی بیماران واقعی دارد و درصد نسبتا کمی از بیماران واقعی را به عنوان سالم طبقه بندی می کند. این ویژگی در مسائل پزشکی اهمیت بالایی دارد، زیرا خطای منفی کاذب می تواند پیامد های جدی تری نسبت به خطای مثبت کاذب داشته باشد. در مقابل، Precision متوسط برای کلاس مثبت نشان می دهد که بخشی از پیش بینی های مثبت مدل ممکن است نادرست باشند. پس نتایج نشان می دهد که مدل تمایل بیشتری به تولید مثبت کاذب تا منفی کاذب دارد.

ماتریس سردرگمی نشان می دهد:

True Positive (TP) = 27

True Negative (TN) = 20

False Positive (FP) = 8

False Negative (FN) = 6

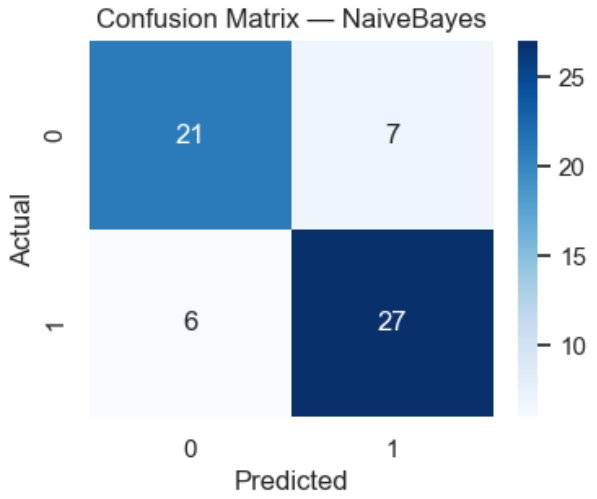
ماتریس سردرگمی نشان می دهد تعداد خطای منفی کاذب کمتر از خطای مثبت کاذب است که نشان می دهد این مدل تمایل دارد بیماران واقعی را بهتر شناسایی کند تا افراد سالم را به اشتباه بیمار تشخیص دهد. این رفتار برای کاربرد های تشخیص پزشکی قابل قبول تر است، هرچند افزایش خطای مثبت کاذب می تواند منجر به آزمایش های غیرضروری شود.

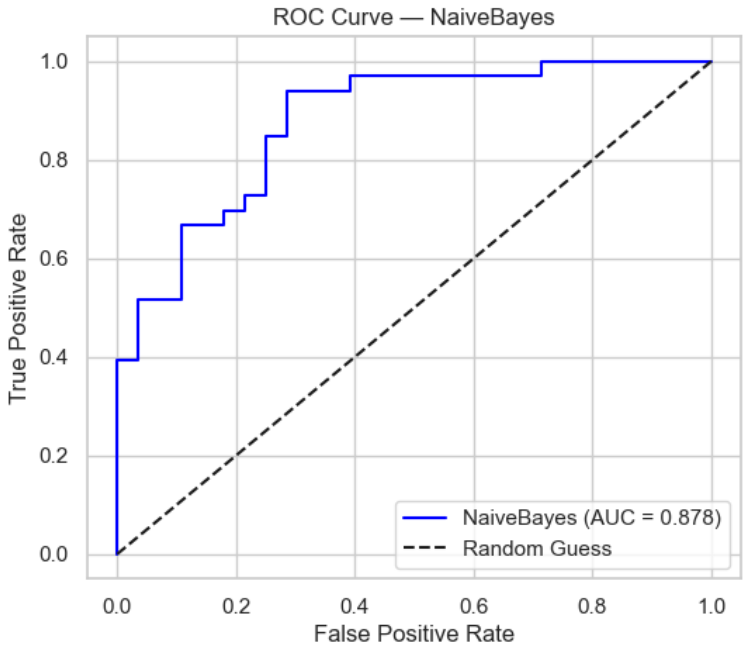
منحنی ROC مدل QDA فاصله مناسبی از خط حدس تصادفی دارد و به گوشه بالا-چپ نزدیک تر است بنابراین در اکثر آستانه ها نرخ تشخیص صحیح مناسبی دارد اما از دو مدل قبلی کمتر است.

مقدار AUC برابر با ۰.۸۳۴ بالاتر از ۰.۸ نشان دهنده قدرت تفکیک خوب مدل در تمایز بین بیماران و افراد سالم است که در ادامه با مدل های غیرخطی پیچیده تر مقایسه می شود.

مدل QDA با وجود سادگی نسبی، عملکرد قابل قبولی در تشخیص بیماری قلبی ارائه داده است. با این حال، حساسیت این مدل به مفروضات آماری مثل نرمال بودن و کوواریانس متفاوت کلاس ها می تواند دلیل عملکرد پایین تر آن نسبت به برخی مدل ها باشد.

```
--- NaiveBayes ---

Classification Report:
              precision    recall  f1-score   support

           0       0.78      0.75      0.76        28
           1       0.79      0.82      0.81        33

    accuracy                           0.79        61
   macro avg       0.79      0.78      0.78        61
weighted avg       0.79      0.79      0.79        61
```





بر اساس گزارش طبقه بندی، مدل Naive Bayes یا بیز ساده به دقت کلی ۷۹٪ روی مجموعه آزمون دست یافته است.

معیار Recall برای کلاس مثبت نسبتا بالا است و مدل توانایی مناسبی در شناسایی بیماران واقعی دارد و درصد کمی از بیماران واقعی را به عنوان افراد سالم طبقه بندی می کند. این ویژگی در مسائل پزشکی اهمیت بالایی دارد زیرا خطای منفی کاذب می تواند پیامد های جدی تری از خطای مثبت کاذب داشته باشد. در مقابل معیار precision متوسط برای کلاس مثبت نشان می دهد که بخشی از پیش بینی های مثبت مدل یعنی مواردی که مدل آن ها را به عنوان بیمار تشخیص داده است ممکن است اشتباه باشند. پس مدل تمایل بیشتری به تولید مثبت کاذب دارد.

ماتریس سردرگمی نشان می دهد:

True Positive (TP) = 27

True Negative (TN) = 21

False Positive (FP) = 7

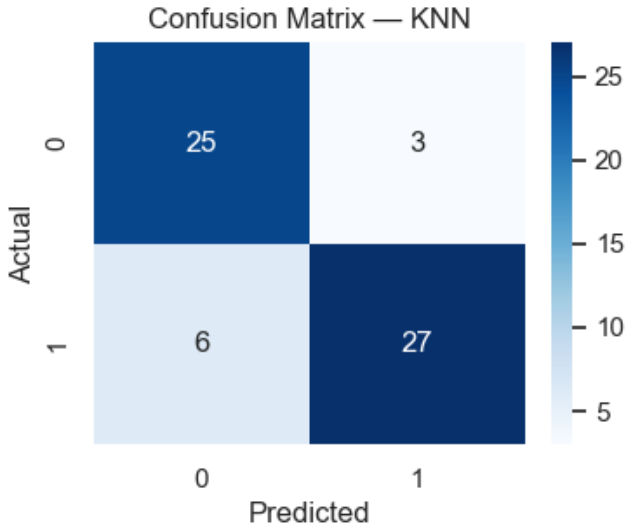
False Negative (FN) = 6

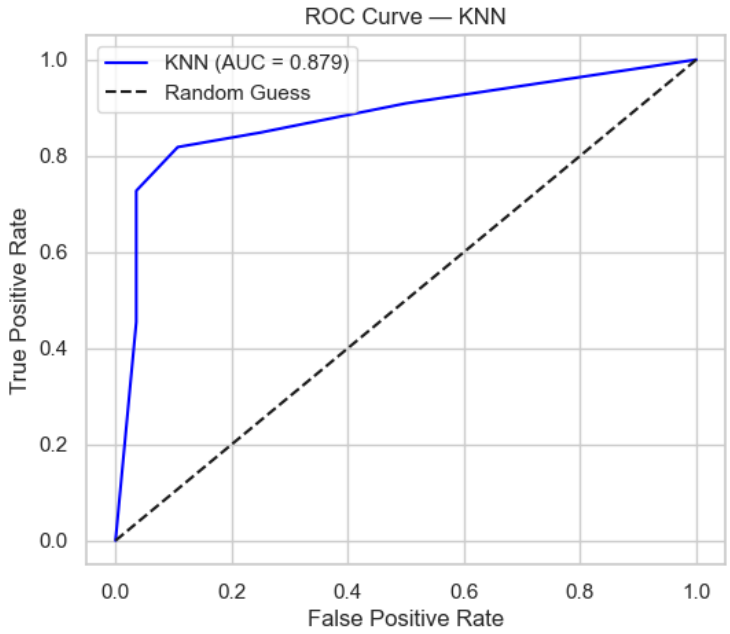
ماتریس سردرگمی نشان می دهد تعداد خطای منفی کاذب کمتر از خطای مثبت کاذب است که نشان می دهد این مدل تمایل دارد بیماران واقعی را بهتر شناسایی کند و تمایل دارد بیماران را از دست ندهد حتی اگر در برخی موارد افراد سالم را به اشتباه بیمار تشخیص دهد.

منحنی ROC در این مدل فاصله مناسبی از خط حدس تصادفی دارد و نرخ تشخیص صحیح مناسبی در طیف وسیعی از آستانه ها دارد.

مقدار AUC برابر با ۰.۸۷۸ بیانگر قدرت تفکیک خوب مدل در تمایز بین بیماران و افراد سالم (کلاس مثبت و منفی) است زیرا بالاتر از ۰.۸ است.

```
--- KNN ---

Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.89      0.85        28
           1       0.90      0.82      0.86        33

    accuracy                           0.85        61
   macro avg       0.85      0.86      0.85        61
weighted avg       0.86      0.85      0.85        61
```





بر اساس گزارش طبقه بندی، مدل K-Nearest Neighbors یا نزدیک ترین همسایه به دقت کلی ۸۵٪ روی مجموعه آزمون دست یافته است که میتوان گفت دقت بالا و قابل توجهی است.

معیار recall در کلاس مثبت برابر ۰.۸۲ است که نشان می دهد مدل توانایی خوبی در شناسایی بیماران واقعی دارد و درصد کمی از بیماران واقعی را به عنوان افراد سالم طبقه بندی کرده است. این ویژگی در مسائل پزشکی اهمیت بالای دارد زیرا خطای منفی کاذب می تواند منجر به عدم تشخیص بیمار و پیامد های جدی برای فرد شود. معیار precision برای کلاس مثبت برابر ۰.۹۰ است که مقدار بالایی محسوب می شود و نشان می دهد که اکثر پیش بینی های مثبت مدل یعنی مواردی که مدل آنها را بیمار تشخیص داده صحیح بوده اند. بنابراین مدل در جلوگیری از تولید مثبت کاذب عملکرد بسیار خوبی دارد و افراد سالم را به ندرت به اشتباه بیمار تشخیص می دهد به عبارتی مدل تمایل بیشتری به تولید منفی کاذب دارد تا مثبت کاذب. مدل با وجود توانایی خوب در شناسایی بیماران واقعی، هنوز بخشی از بیماران را از دست می دهد. مدل رفتاری محافظه‌کار دارد و تنها در صورت اطمینان بالا اقدام به تشخیص بیماری می‌کند.

در مسائل پزشکی recall بالا اهمیت زیادی دارد زیرا تشخیص ندادن بیمار واقعی پیامد جدی تر و هزینه بیشتری نسبت به precision بالا دارد.

ماتریس سردرگمی نشان می دهد:

True Positive (TP) = 27

True Negative (TN) = 25

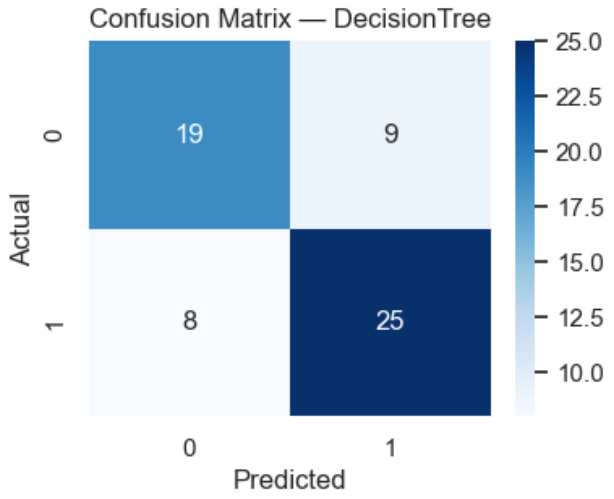
False Positive (FP) = 3

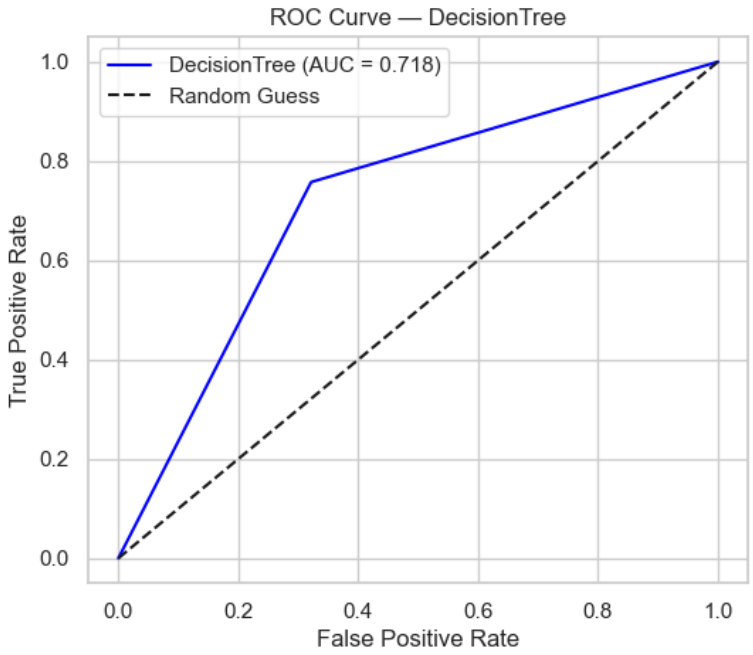
False Negative (FN) = 6

ماتریس سردرگمی نشان می دهد تعداد خطای منفی کاذب بیشتر از خطای مثبت کاذب است. این نشان می دهد مدل تمایل دارد افراد سالم را بهتر به درستی تشخیص دهد و در عین حال بیماران واقعی را نیز با دقت مناسبی شناسایی کند. این تعادل در مسائل پزشکی مطلوب است زیرا هم از دست دادن بیماران واقعی محدود شده و هم از تشخیص اشتباه افراد سالم جلوگیری شده است.

منحنی ROC فاصله قابل توجهی از خط حدس تصادفی دارد و نرخ تشخیص صحیح مناسبی در اکثر آستانه ها دارد. همچنین مقدار AUC برابر۰.۸۷۹ بیانگر قدرت تفکیک پذیری خوب میان بیماران و افراد سالم است.

```
--- DecisionTree ---

Classification Report:
              precision    recall  f1-score   support

           0       0.70      0.68      0.69        28
           1       0.74      0.76      0.75        33

    accuracy                           0.72        61
   macro avg       0.72      0.72      0.72        61
weighted avg       0.72      0.72      0.72        61
```





براساس گزارش طبقه بندی، مدل درخت تصمیم به دقت کلی ۷۲٪ بر روی مجموعه آزمون دست یافته است که از دقت مدل های قبلی که بررسی کردیم کمتر است.

معیار recall برای کلاس مثبت برابر ۰.۷۶ است که میزان حساسیت متوسط مدل را نشان می دهد و نشان می دهد مدل توانایی متوسطی در شناسایی بیماران واقعی دارد و نسبت به مدل هایی که بررسی کردیم درصد قابل توجهی از بیماران را به عنوان افراد سالم طبقه بندی کرده است. و این ویژگی در مسائل پزشکی اهمیت بالایی دارد و مقدار قابل توجهی از بیماران از دست داده است . در مقابل precision متوسط برای کلاس مثبت نشان می دهد که برخی از پیش بینی های مثبت مدل که مواردی است که مدل بیمار تشخیص داده است ممکن است اشتباه باشد. پس مدل تمایل بیشتری به تولید مثبت کاذب دارد.

ماتریس سردرگمی نشان می دهد:

True Positive (TP) = 25

True Negative (TN) = 19

False Positive (FP) = 9

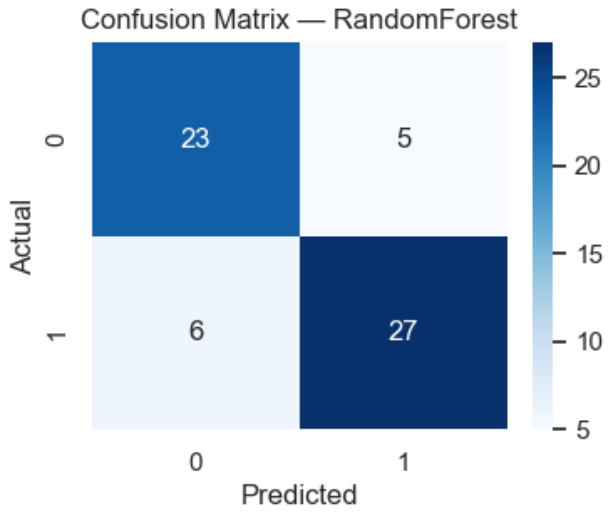
False Negative (FN) = 8

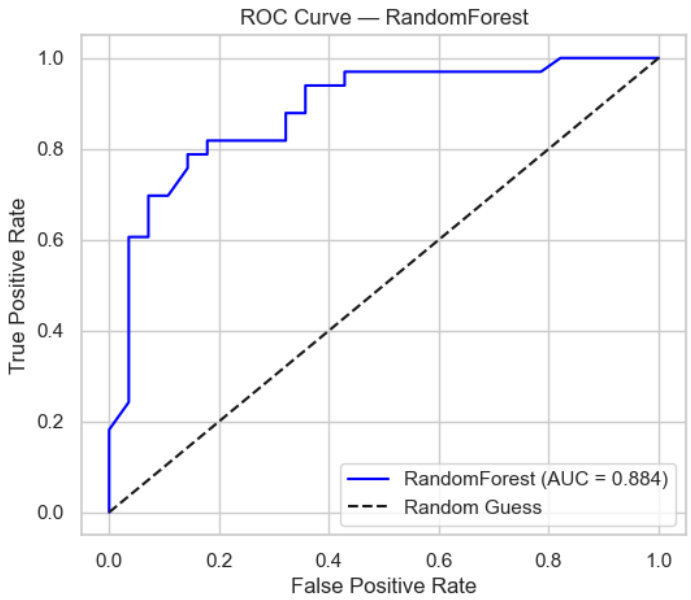
ماتریس سردرگمی نشان می دهد تعداد خطای منفی کاذب از خطای مثبت کاذب کمتر است که نشان می دهد این مدل تمایل دارد بیماران واقعی را بهتر شناسایی کند و تمایل دارد بیمار کمتری را از دست بدهد حتی اگر در برخی موارد افراد سالم را به اشتباه بیمار تشخیص دهد.

منحنی ROC مدل درخت تصمیم فاصله کمی از خط حدس تصادفی دارد و نرخ تشخیص صحیح متوسط رو به پایینی برای طیف وسیع آستانه ها دارد.

مقدار AUC برابر ۰.۷۱۸ بیانگر قدرت تفکیک پذیری قابل قبول مدل برای تمایز میان بیماران و افراد سالم است. که تا به اینجا از همه مدل های مورد بررسی کمتر بوده است.

```
-- RandomForest ---

Classification Report:
              precision    recall  f1-score   support

           0       0.79      0.82      0.81        28
           1       0.84      0.82      0.83        33

    accuracy                           0.82        61
   macro avg       0.82      0.82      0.82        61
weighted avg       0.82      0.82      0.82        61
```





بر اساس گزارش طبقه بندی، مدل جنگل تصادفی به دقت کلی ۸۲٪ بر روی مجموعه آزمون دست یافته است.

معیار recall بالا رای کلاس مثبت نشان می دهد که مدل توانایی خوبی در شناسایی بیماران واقعی دارد و در مقابل معیار precision برای کلاس مثبت کمی از recall بالاتر است و نشان می دهد که اکثر پیش بینی های مثبت مدل یعنی مواردی که مدل آنها را بیمار تشخیص داده است صحیح بوده است بنابراین مدل در جلوگیری از تولید مثبت کاذب عملکر خوبی دارد به عبارتی مدل تمایل بیشتری به تولید منفی کاذب دارد تا مثبت کاذب مدل با وجو توانایی خوب در شناسایی بیماران واقعی هنوز بخشی از بیماران را از دست می دهد.

با توجه به نزدیک بودن مقادیر recall و precision می توان گفت مدل جنگل تصادفی رفتاری متعادل دارد و نه بیش از حد تهاجمی است و نه بیش از حد محافظه کار است. این ویژگی باعث می شود مدل برای کاربردهای پزشکی گزینه ای قابل اعتماد باشد.

مقدار F1-score برابر با ۰.۸۳ نشان‌دهنده تعادل مطلوب بین این دو معیار است.

ماتریس سردرگمی نشان می دهد:

True Positive (TP) = 27

True Negative (TN) = 13

False Positive (FP) = 5

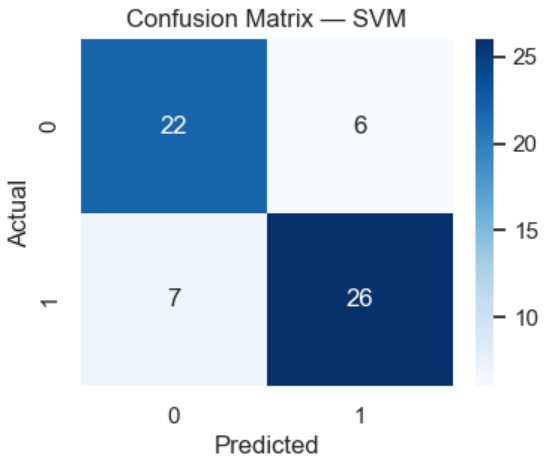
False Negative (FN) = 6

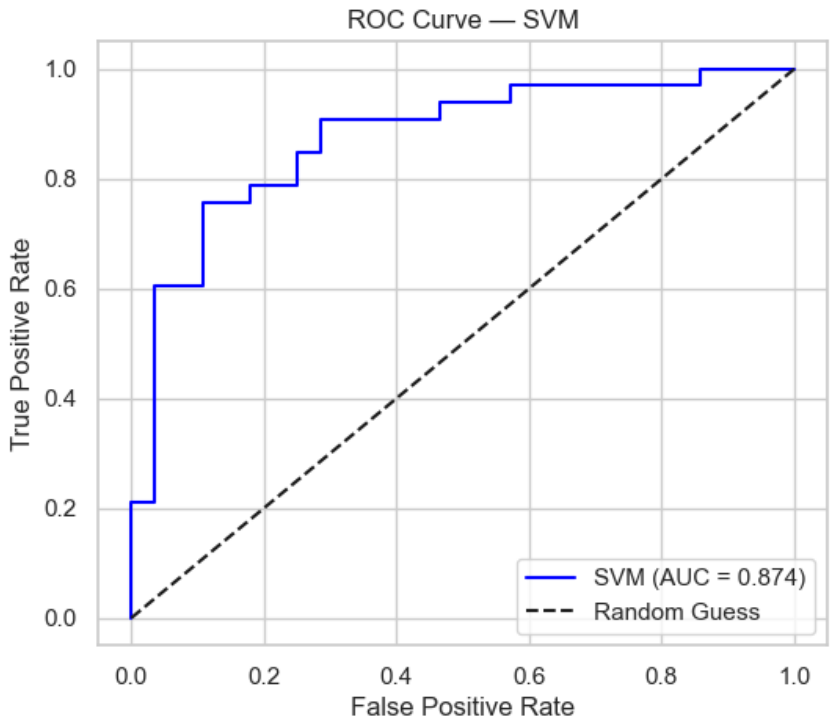
ماتریس سردرگمی نشان می دهد که مدل جنگل تصادفی تنها ۶ مورد از بیماران واقعی را به‌اشتباه سالم تشخیص داده است، در حالی که ۵ فرد سالم به‌اشتباه بیمار تشخیص داده شده‌اند. تعداد خطای منفی کاذب اندکی بیشتر از تعداد خطای مثبت کاذب است. این توزیع خطا نشان می‌دهد که مدل رفتار متعادلی دارد و از هر دو نوع خطای بحرانی تا حد قابل قبولی اجتناب کرده است.

منحنی ROC نشان می دهد که مدل جنگل تصادفی در تمام آستانه های تصمیم گیری عملکردی به مراتب بهتر از حد تصادفی دارد.

مقدار AUC برابر با ۰.۸۸۴ بیانگر قدرت بالای مدل در تفکیک بیماران از افراد سالم است و نشان می دهد که مدل مستقل از آستانه تصمیم توان تمایز مناسبی بین دو کلاس دارد.

```
--- SVM ---

Classification Report:
              precision    recall  f1-score   support

           0       0.76      0.79      0.77        28
           1       0.81      0.79      0.80        33

    accuracy                           0.79        61
   macro avg       0.79      0.79      0.79        61
weighted avg       0.79      0.79      0.79        61
```





بر اساس گزارش طبقه بندی، مدل SVM به دقت کلی ۷۹٪ بر روی مجموعه آزمون دست یافته است که نشان می‌دهد حدود ۷۹ درصد نمونه‌های مجموعه آزمون به‌درستی طبقه‌بندی شده‌اند.

مقدار Precision برای کلاس مثبت برابر با ۰.۸۱ نشان می‌دهد که بیشتر افرادی که توسط مدل به‌عنوان بیمار تشخیص داده شده‌اند، واقعاً بیمار بوده‌اند. Recall برابر با ۰.۷۹ بیانگر آن است که مدل توانسته است بخش قابل قبولی از بیماران واقعی را شناسایی کند. مقدار F1-score برابر با ۰.۸۰ نشان‌دهنده تعادل مناسب بین این دو معیار است.

مدل SVM رفتاری نسبتاً محافظه‌کار دارد به این معنا که مدل تمایل دارد تنها زمانی تشخیص بیماری بدهد که از پیش‌بینی خود اطمینان بیشتری داشته باشد که منجر به کاهش تشخیص‌های اشتباه اما افزایش اندک خطاهای منفی کاذب می‌شود یعنی در برخی موارد بیماران واقعی را از دست می دهد.

ماتریس سردرگمی نشان می دهد:

True Positive (TP) = 26

True Negative (TN) = 22

False Positive (FP) = 6

False Negative (FN) = 7

ماتریس سردرگمی نشان می دهد تعداد خطای منفی کاذب از خطای مثبت کاذب بیشتر است یعنی مدل در برخی موارد بیماران واقعی را از دست می دهد. با این حال اختلاف بین این دو خطا زیاد نیست و نشان دهنده رفتار نسبتا متعادل مدل است.

منحنی ROC این مدل فاصله مناسبی از خط حدس تصادفی دارد و نرخ تشخیص صحیح مناسبی در آستانه های مختلف تصمیم گیری دارد.

مقدار AUC برابر۰.۸۷۴ بیانگر قدرت تفکیک پذیری مدل بین بیماران و افراد سالم است.

به‌منظور بررسی پایداری و قابلیت تعمیم‌پذیری مدل‌ها، از روش اعتبارسنجی متقاطع ۱۰ بخشی استفاده شد. این روش با کاهش وابستگی نتایج به یک تقسیم‌بندی خاص از داده‌ها، ارزیابی قابل اعتمادتری از عملکرد مدل‌ها ارائه می‌دهد. برای هر مدل، میانگین دقت و انحراف معیار دقت محاسبه شد تا علاوه بر عملکرد متوسط، میزان نوسان و پایداری مدل نیز ارزیابی شود.

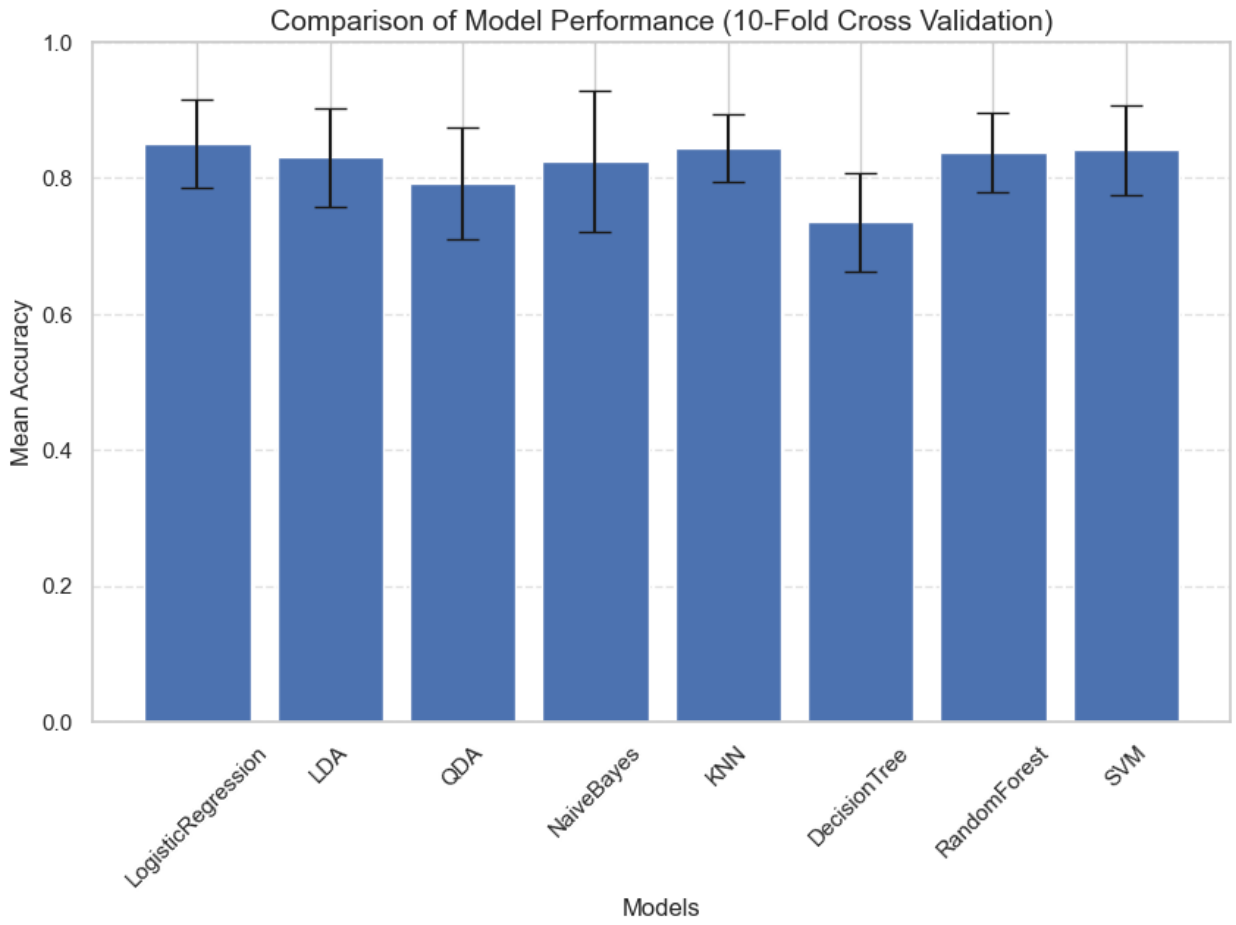
```
--- 10-Fold Cross-Validation (Robustness Check) ---

Model: LogisticRegression
  All 10 scores: [0.871 0.903 0.867 0.833 0.833 1.    0.8   0.767 0.767 0.867]
  Mean Accuracy: 0.851 (±0.066)
------------------------------
Model: LDA
  All 10 scores: [0.839 0.903 0.833 0.833 0.833 1.    0.767 0.733 0.767 0.8  ]
  Mean Accuracy: 0.831 (±0.073)
------------------------------
Model: QDA
  All 10 scores: [0.806 0.839 0.733 0.867 0.8   0.933 0.7   0.633 0.767 0.833]
  Mean Accuracy: 0.791 (±0.082)
------------------------------
Model: NaiveBayes
  All 10 scores: [0.677 0.903 0.8   0.933 0.9   1.    0.767 0.667 0.767 0.833]
  Mean Accuracy: 0.825 (±0.104)
------------------------------
Model: KNN
  All 10 scores: [0.806 0.903 0.867 0.867 0.8   0.933 0.767 0.8   0.867 0.833]
  Mean Accuracy: 0.844 (±0.049)
------------------------------
Model: DecisionTree
  All 10 scores: [0.806 0.645 0.8   0.667 0.767 0.833 0.733 0.6   0.733 0.767]
  Mean Accuracy: 0.735 (±0.072)
------------------------------
Model: RandomForest
  All 10 scores: [0.871 0.903 0.767 0.8   0.833 0.967 0.767 0.8   0.833 0.833]
  Mean Accuracy: 0.837 (±0.059)
------------------------------
Model: SVM
  All 10 scores: [0.839 0.903 0.867 0.8   0.833 1.    0.767 0.833 0.767 0.8  ]
  Mean Accuracy: 0.841 (±0.067)
------------------------------

Evaluation complete. You can now compare these results to the basic models.

--- BEST MODEL SELECTED ---
Best Model: LogisticRegression
Mean Accuracy: 0.8508
Std: 0.0658
```

بر اساس نتایجCross-Validation ، مدل رگرسیون لوژستیک با میانگین دقت ۰.۸۵۱ بالاترین عملکرد را در میان مدل‌های بررسی‌شده داشته است.

نتایج نشان داد که برخی مدل‌ها با وجود دقت متوسط مناسب، دارای نوسانات بیشتری بودند، در حالی که مدل‌هایی مانند رگرسیون لوژستیک علاوه بر دقت بالاتر، انحراف معیار کمتری نیز داشتند که نشان‌دهنده پایداری بیشتر آن‌ها در برابر تغییرات داده است.

مدل KNN با انحراف معیار ۰.۰۴۹ پایدارترین عملکرد را نشان داده است، در حالی که مدل Naive Bayes با انحراف معیار ۰.۱۰۴ بیشترین نوسان عملکرد را داشته است. این موضوع نشان می‌دهد که اگرچه Naive Bayes در برخی تقسیم‌بندی‌ها عملکرد بسیار خوبی دارد، اما قابلیت تعمیم آن کمتر است. و همچنین مشاهده می کنیم مدل‌های ساده‌تر مانند Logistic Regression و SVM عملکردی قابل رقابت و حتی بهتر از مدل‌های پیچیده‌تری مانند Random Forest داشته‌اند. این موضوع نشان می‌دهد که با پیش‌پردازش مناسب داده‌ها، مدل‌های خطی نیز می‌توانند عملکرد بسیار خوبی ارائه دهند.

این تحلیل نشان می‌دهد که استفاده از اعتبارسنجی متقاطع برای انتخاب مدل نهایی ضروری است، زیرا ارزیابی تک‌مرحله‌ای مبتنی بر یک تقسیم‌بندی تصادفی داده‌ها ممکن است منجر به برآورد خوش‌بینانه یا بدبینانه عملکرد مدل شود.



این یک نمودار میله‌ای مقایسه عملکرد مدل‌ها حاصل از 10-Fold Cross-Validation است.

میله ها عملکرد متوسط هر مدل و خط های خطا انحراف معیار دقت است که نشان دهنده پایداری مدل است.

این نمودار همزمان قدرت مدل و پایداری مدل را نشان می دهد.

همان‌طور که مشاهده می‌شود مدل رگرسیون لوژستیک علاوه بر دستیابی به دقت بالاتر، دارای کمترین میزان نوسان نیز می‌باشد که بیانگر قابلیت تعمیم مناسب آن نسبت به سایر مدل‌ها است. این نتایج نشان می‌دهد که انتخاب مدل نهایی صرفاً بر اساس دقت کافی نیست و پایداری مدل نیز نقش مهمی در ارزیابی عملکرد دارد.

```
--- Hyperparameter Tuning: Logistic Regression ---

Best Parameters: {'model__C': 0.1, 'model__penalty': 'l2', 'model__solver':
'lbfgs'}

Best CV Accuracy: 0.8423333333333334



--- Hyperparameter Tuning: Random Forest ---

Best Parameters: {'model__max_depth': 3, 'model__min_samples_leaf': 2,
'model__min_samples_split': 2, 'model__n_estimators': 200}

Best CV Accuracy: 0.8546666666666667



--- Hyperparameter Tuning: KNN ---

Best Parameters: {'model__metric': 'manhattan', 'model__n_neighbors': 8,
'model__weights': 'distance'}

Best CV Accuracy: 0.8630000000000001
```

به منظور بهبود عملکرد و افزایش قابلیت تعمیم مدل‌ها، بهینه‌سازی ابرپارامترها با استفاده از اعتبارسنجی متقاطع انجام شد.

منظم سازی قوی تر:

```
C = 0.1
```

جلوگیری از بیش برازش با توزیع وزن ها:

```
penalty = l2
```

پایدار و مناسب برای دیتاست های کوچک – متوسط:

```
solver = lbfgs
```

تحلیل نتیجه:

```
Best CV Accuracy = 0.842
```

دقت CV کمی کمتر از حالت بدون تنظیم است، اما این انتخاب مدل ساده‌تر و تعمیم‌پذیرتر ایجاد می‌کند. نتایج بهینه‌سازی نشان داد که مقدار کوچک‌تر پارامتر C منجر به عملکرد پایدارتر مدل شده است. این موضوع بیانگر آن است که کاهش پیچیدگی مدل و اعمال منظم‌سازی قوی‌تر از بیش‌برازش جلوگیری کرده و قابلیت تعمیم مدل را بهبود می‌بخشد.

کاهش واریانس مدل :

```
n_estimators = 200
```

درخت های کم عمق (کنترل بیش برازش):

```
max_depth = 3
```

جلوگیری از یادگیری نویز:

```
min_samples_leaf = 2

min_samples_split = 2
```

تحلیل نتیجه:

```
Best CV Accuracy = 0.855
```

مدل جنگل تصادفی زمانی بهترین عملکرد را دارد که پیچیدگی آن کنترل شود، نه زمانی که بیش‌ازحد عمیق شود. نتایج نشان داد که محدود کردن عمق درخت‌ها نقش مهمی در بهبود عملکرد مدل جنگل تصادفی داشته است. این موضوع نشان می‌دهد که کنترل پیچیدگی مدل برای جلوگیری از بیش‌برازش در این دیتاست ضروری است.

تعادل بین بایاس و واریانس:

`n_neighbors = 8 , K = 8`

مناسب تر برای داده های مقیاس بندی شده:

`metric = Manhattan`

همسایه های نزدیک تر تاثیر بیشتری دارند.

`weights = distance`

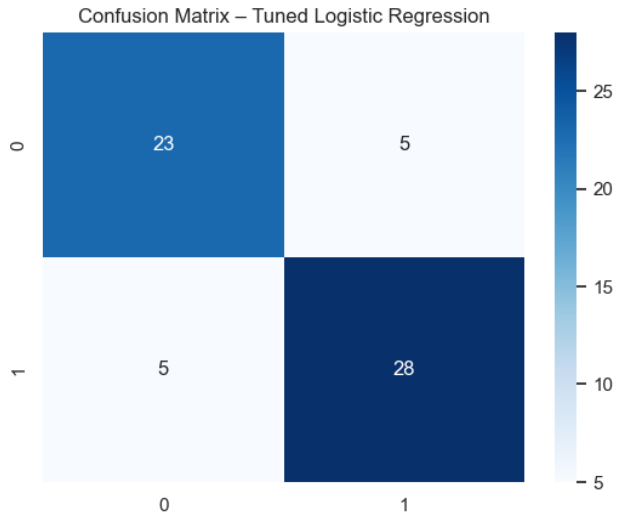
تحلیل نتیجه:

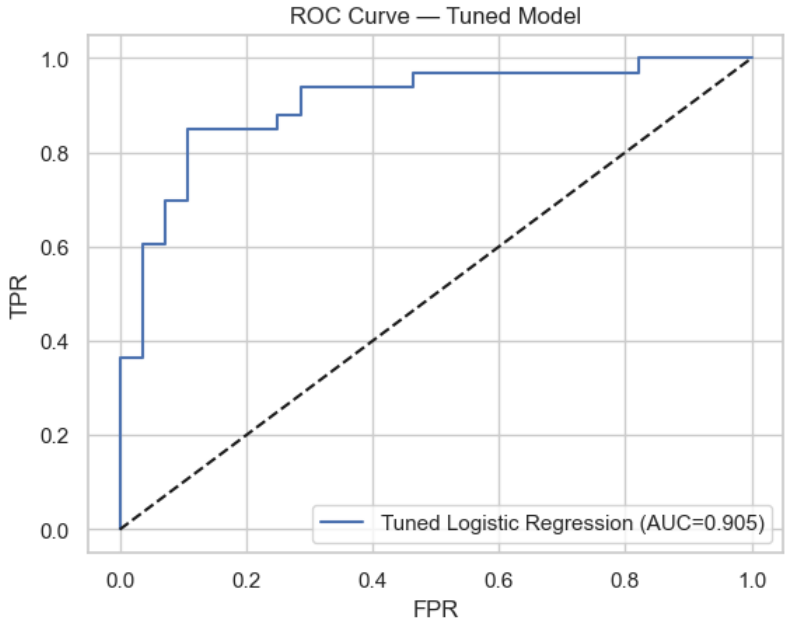
`Best CV Accuracy = 0.863`

KNN پس از تنظیم پارامترها به بهترین عملکرد در میان مدل‌های بررسی‌شده دست یافت، که نشان‌دهنده اهمیت انتخاب صحیح فاصله و تعداد همسایه‌ها در این الگوریتم است. استفاده از وزن‌دهی مبتنی بر فاصله باعث شد که همسایه‌های نزدیک‌تر نقش بیشتری در تصمیم‌گیری داشته باشند که منجر به بهبود عملکرد مدل شد.

نتایج نشان می‌دهد که بهینه‌سازی ابرپارامترها تأثیر قابل‌توجهی بر عملکرد مدل‌ها داشته است، به‌ویژه در مدل KNNکه به بهترین عملکرد کلی دست یافته است. با توجه به نتایج بهینه‌سازی ابرپارامترها، مدل KNN بالاترین دقت را ارائه داد، در حالی که مدل Logistic Regression به دلیل سادگی و تفسیرپذیری، همچنان گزینه‌ای قابل اعتماد برای کاربردهای پزشکی محسوب می‌شود.

```
--- Final Evaluation on Test Set (After Tuning) ---

Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.82      0.82        28
           1       0.85      0.85      0.85        33

    accuracy                           0.84        61
   macro avg       0.83      0.83      0.83        61
weighted avg       0.84      0.84      0.84        61
```





مرحله‌ی ارزیابی نهایی مدل بعد از تنظیم هایپرپارامترها روی مجموعه‌ی **تست** است. پس از انجام تنظیم هایپرپارامترها با استفاده ازGridSearchCV ، بهترین مدل به‌دست‌آمده بر روی مجموعه آزمون مورد ارزیابی نهایی قرار گرفت. این مجموعه در هیچ‌یک از مراحل آموزش و تنظیم پارامترها استفاده نشده بود و بنابراین معیار مناسبی برای سنجش توان تعمیم مدل محسوب می‌شود.

نتایج ارزیابی نهایی نشان می‌دهد که مدل عملکرد متعادلی بین دقت و حساسیت ارائه داده است و نه رویکردی بیش‌ازحد تهاجمی و نه محافظه‌کارانه اتخاذ کرده است. این تعادل باعث می‌شود که مدل هم از تشخیص‌های اشتباه غیرضروری جلوگیری کند و هم موارد واقعی مثبت را تا حد قابل قبولی شناسایی نماید.

۶. نتیجه گیری

در این پروژه، مسئله‌ی پیش‌بینی وجود بیماری قلبی به‌عنوان یک مسئله‌ی طبقه‌بندی دودویی مورد بررسی قرار گرفت. با توجه به ماهیت پزشکی داده‌ها و حساسیت تشخیص، تلاش شد رویکردی سیستماتیک شامل پیش‌پردازش دقیق داده‌ها، مقایسه‌ی چندین الگوریتم یادگیری ماشین و ارزیابی چندمعیاره اتخاذ شود تا مدلی با قابلیت تعمیم مناسب حاصل گردد. نتایج نشان داد که اگرچه چندین الگوریتم کلاسیک عملکرد قابل قبولی داشتند، اما مدل رگرسیون لوژستیک پس از تنظیم هایپرپارامترها، به تعادل بهتری میان دقت (Precision) و بازخوانی (Recall) دست یافت. این موضوع بیانگر آن است که در این دیتاست، روابط خطی بین ویژگی‌ها و متغیر هدف نقش پررنگ‌تری نسبت به الگوهای بسیار پیچیده ایفا می‌کنند. استفاده از اعتبارسنجی متقاطع ۱۰-بخشی نشان داد که عملکرد مدل انتخاب‌شده تنها به یک تقسیم خاص از داده‌ها وابسته نیست و نوسان عملکرد در فولدهای مختلف در بازه‌ی قابل قبولی قرار دارد. این موضوع بیانگر پایداری مدل و کاهش احتمال بیش‌برازش است. با وجود نتایج قابل قبول، این مطالعه با محدودیت‌هایی نیز همراه است از جمله حجم نسبتا محدود داده‌ها و عدم درنظرگرفتن متغیرهای بالینی تکمیلی. در مطالعات آینده می‌توان با استفاده از داده‌های بزرگ‌تر، روش‌های انتخاب ویژگی پیشرفته‌تر و مدل‌های غیرخطی پیچیده‌تر، عملکرد مدل را بهبود بخشید. همچنین تنظیم آستانه تصمیم بر اساس هزینه خطاهای پزشکی می‌تواند منجر به مدل‌های کاربردی‌تر در محیط‌های بالینی شود.

در مجموع، این پروژه نشان داد که با ترکیب پیش‌پردازش هدفمند، ارزیابی چندمعیاره و انتخاب آگاهانه مدل، می‌توان به راه‌حل‌هایی قابل اعتماد برای مسائل واقعی در حوزه سلامت دست یافت.