
بنام خداوند جان و خرد

۱

پروژه درس داده‌کاوی

درخت تصمیم و جنگل تصادفی

استاد محترم درس

جناب آقای دکتر پاینده

دانشجو

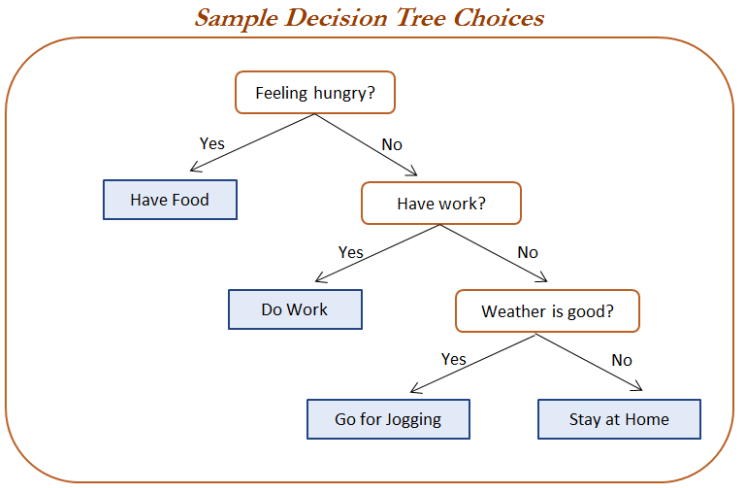
آیدا اعلابیکی

در ابتدا برخود واجب می‌دانم از زحمات استاد ارجمندم جناب آقای دکتر پاینده، کمال تشکر را به جا آورم.

فهرستت مطالب

درخت تصمیم

طبقه‌بندی‌کننده‌های درخت تصمیم، قوانینی را به صورت جملات ساده تولید می‌کنند که به راحتی می‌توان آن‌ها را تفسیر کرد. بر اساس ویژگی‌های موجود در داده‌های آموزشی، مدل‌های درخت تصمیم مجموعه‌ای از سؤالات را برای پیش‌بینیِ برچسبِ طبقات مربوط به نمونه‌های آزمایشی یاد می‌گیرند. برای توضیح نحوه کار کردن یک درخت تصمیم، شکل زیر را در نظر بگیرید که یک درخت تصمیم ساده را نشان می‌دهد. این درخت تصمیم می‌گیرد که یک شخص چه کاری را انجام دهد.



در روش درخت تصمیم، ابتدا با استفاده از داده‌های آموزشی یک چنین نموداری ساخته می‌شود. سپس برای یک مشاهده جدید، از این درخت تصمیم استفاده شده و برچسب (طبقه‌ی) مربوط به آن مشاهده مشخص می‌شود. در شکل بالا برچسب‌ها درون کادر با رنگ آبی قرار گرفته‌اند. در واقع متغیر پاسخ یک متغیر گسسته با چهار سطحِ «خوردن غذا»، «انجام دادن کار»، «رفتن به پیاده‌روی» و «ماندن در خانه» است. بنابراین اگر y نشان‌دهنده یا متغیر پاسخ باشد، سطوح آن به صورت زیر هستند

$$y \in \{Have\ Food, Do\ Work, Go\ for\ Jogging, Stay\ at\ home\}$$

درخت تصمیم براساس ویژگی‌های شخص تصمیم می‌گیرد که کدامیک از این برچسب‌ها را به او اختصاص بدهد. در این شکل همچنین متغیرهایی که درون کادر با رنگ قهوه‌ای قرار دارند، در واقع ویژگی‌ها یا متغیرهای پیش‌بین (مستقل) هستند. به عنوان مثال، یکی از ویژگی‌هایی که در این درخت

تصمیم در نظر گرفته شده این است که «آیا شخص احساس گرسنگی می‌کند؟» در صورتی که جواب مثبت باشد به او برچسب $Have\ Food$ داده شده و کار تمام می‌شود و سایر ویژگی‌های این شخص مورد بررسی قرار نمی‌گیرند. اما در صورتی که جواب منفی باشد، درخت تصمیم به سراغ بررسی یک ویژگیِ دیگر این شخص می‌رود. ویژگی بعدی که از این شخص مورد پرسش قرار می‌گیرد این است که «آیا این شخص کار دارد؟» در صورتی که پاسخ مثبت باشد به این شخص برچسب $Do\ Work$ تعلق گرفته و کار تمام می‌شود و سایر ویژگی‌های این شخص مورد بررسی قرار نمی‌گیرد. اما اگر پاسخ این سوال نیز منفی باشد، درخت تصمیم به سراغ ویژگی بعدی رفته و آن را مورد بررسی قرار می‌دهد. «آیا هوا خوب است؟» در صورتی که پاسخ به این سوال مثبت باشد، درخت تصمیم برچسب $Go\ for\ Jogging$ را به این شخص اختصاص می‌دهد اما اگر پاسخ این سوال نیز منفی باشد، درخت تصمیم برچسب $Stay\ at\ home$ را به این شخص اختصاص داده و کار پایان می‌پذیرد.

در این مثال تمامی ویژگی‌هایی که در نظر گرفته شده متغیرهایی کیفی هستند. اما در صورتی که در مجموعه ویژگی‌ها متغیرهای کمی و پیوسته نیز وجود داشته باشد، معمولا الگوریتم‌های درخت تصمیم این متغیرها را به صورت گسسته تبدیل می‌کنند. مثلا اگر $X \in \mathbb{R}$ یک متغیر پیوسته باشد که به عنوان یک ویژگی در نظر گرفته شده، در این صورت درخت تصمیم با طبقه‌بندی کردن دامنه X آن را به متغیری کیفی تبدیل می‌کند. به عنوان مثال، درخت تصمیم فقط دو حالت $X \geq 0$ و $X < 0$ را در نظر می‌گیرد. یا مثلا یک نوع گسسته‌سازی دیگر می‌تواند به صورت زیر باشد

$$X < -1, -1 \leq X < 0, 0 \leq X < 1, X \geq 1$$

درخت تصمیم یکی از مشهورترین روش‌های طبقه‌بندی است. در یک مسئله‌ی مربوط به درخت تصمیم، هدف اصلی به دست آوردن درخت بهینه با بهترین پیش‌بینی و کمترین پیچیدگی می‌باشد.

اصطلاحات مهمی که در درخت تصمیم مورد استفاده قرار می‌گیرند به شرح زیر هستند:

گره[1]: گره یک قسمت اساسی از یک درخت تصمیم است. گره‌ها انواع مختلف دارند که برخی از آنها عبارتند از: گره ریشه[2]، گره داخلی[3]، گره برگ[4] (یا گره انتهایی[5])، گره والد[6] و گره فرزند[7].

شاخه[8]: شاخه‌ها اتصال‌دهنده گره‌ها هستند و نتایج تصمیم‌گیری را نشان می‌دهند.

گره ریشه: این گره شامل تمام مجموعه داده‌های مسئله است که می‌توانند به دو یا چند زیرمجموعه‌ی همگن تقسیم شود. گره ریشه اولین و بالاترین گره در درخت تصمیم به حساب می‌آید.

گره‌ی برگ یا گره انتهایی: برگ‌ها در واقع گره‌های پایانی یک درخت تصمیم هستند و نتایج پیش‌بینی‌شده را نشان می‌دهند.

گره داخلی: هر گره بجز گره ریشه و گره برگ را گره داخلی می‌نامند.

گره والد و گره فرزند: دو گره پیاپی در یک درخت تصمیم را در نظر می‌گیریم به طوری که بین آنها هیچ گره دیگری وجود نداشته باشد. در این صورت گره‌ای که در درخت تصمیم بالاتر قرار گرفته را والد گره پایینی و گره‌ای که در درخت تصمیم پایین‌تر قرار گرفته را فرزند گره بالایی می‌گویند.

به عنوان مثال، در مثالی که در ابتدای این بخش ذکر شد، تمامی کادرهای مستطیلی شکل گره هستند. گره‌ای که در بالای درخت قرار دارد (در این مثال، Felling Hungry) گره‌ی ریشه است. Have Work یک گره‌ی داخلی است و تمامی گره‌هایی که در انتهای درخت قرار دارند (مثلا Have Food و Stay at Home) گره‌ی برگ هستند. گره‌ی Wheather is Good فرزند گره‌ی Have Work است و گره‌ی Have Work والد گره‌ی Wheather is Good است. همچنین تمامی خطوطی که روی شکل هستند شاخه (یا یال) می‌باشند. نتایج تصمیم‌گیری روی این شاخه‌ها مشخص شده است.

---

[1] Node
[2] Root Node
[3] Interior Node
[4] Leaf Node
[5] Terminal Node
[6] Parent Node
[7] Child Node
[8] Branch

لازم به ذکر است که گره‌ی والد همواره مهم‌تر از گره‌ی فرزند است. به بیان دیگر، ویژگی‌هایی که در قسمت‌های بالاتر درخت تصمیم قرار می‌گیرند همواره از ویژگی‌هایی که در قسمت‌های پایین‌تر قرار می‌گیرند دارای اهمیت بیشتری در تشخیص برچسب (طبقه) مشاهدات جدید دارند. بنابراین در یک درخت تصمیم، همواره **گره‌ی ریشه مهم‌ترین گره** است و در این گره آن ویژگی قرار می‌گیرد که بیشترین تاثیر را در تعیین برچسب یک مشاهده جدید دارد. اما گره ریشه چگونه انتخاب می‌شود؟ چه ویژگی را باید در گره‌ی ریشه قرار داد و کدام‌یک از ویژگی‌ها را باید در گره‌های میانی قرار داد؟ گره‌ی مربوط به کدام ویژگی باید والد و گره‌ی مربوط به کدام ویژگی باید فرزند باشد؟ برای پاسخ به این سوالات، الگوریتم‌های درخت تصمیم از معیارهایی استفاده می‌کنند که مهم‌ترین آنها بهره اطلاعاتی[1] است که در ادامه شرح داده می‌شود.

برای هر یک از ویژگی‌های موجود در مجموعه داده‌ها، بهره اطلاعاتی (IG) به وسیله سنجش تغییرات آنتروپی محاسبه می‌شود. در این روش بررسی می‌شود که هر کدام از ویژگی‌ها چه مقدار اطلاعات درباره یک برچسب می‌دهد. سپس ویژگی‌ها بر مبنای مقادیر بهره اطلاعاتی به دست آمده تقسیم می‌شوند و درخت تصمیم ساخته می‌شود. الگوریتم درخت تصمیم همیشه تا حد امکان سعی خود را در به حداکثر رساندن اطلاعات به دست آمده می‌کند، سپس گره ریشه را به آن ویژگی اختصاص می‌دهد که بیشترین اطلاعات را دارد. در مرحله بعدی این ویژگی حذف می‌شود و مجددا بررسی می‌شود که در بین ویژگی‌های باقیمانده، کدام ویژگی دارای بیشترین اطلاعات است و گره‌ی بعدی به همین ویژگی اختصاص داده می‌شود و همین روند تا کامل شدن درخت تصمیم ادامه پیدا می‌کند.

برای تعریف ریاضی بهره اطلاعاتی ابتدا لازم است توضیحاتی در مورد آنتروپی[2] یک متغیر تصادفی (گسسته) داده شود. فرض کنید X یک متغیر تصادفی گسسته با تابع احتمال $p_X(x)$ باشد، در این

[1] Information Gain

[2] Entropy

صورت آنتروپی این متغیر تصادفی که آن را با $H(X)$ نمایش می‌دهیم به صورت زیر تعریف می‌شود:

$$H(X) = -\sum_{x} p_X(x) \log_2[p_X(x)] = E[\log_2[p_X(X)]]$$

در برخی متون از $H(X)$ تحت عنوان معیار اطلاع شانون نیز یاد می‌شود.

آنتروپی معیاری عددی برای اندازه گرفتن اطلاعات، یا تصادفی بودن یک متغیر تصادفی است. به بیان دقیق‌تر، آنتروپیِ یک متغیر تصادفی، متوسط اطلاعات آن است.

اگر X, Y دو متغیر تصادفی (گسسته) با تابع احتمال توام $p_{X,Y}(x,y)$ باشند، در این صورت آنتروپی توام این دو متغیر تصادفی که آن را با $H(X,Y)$ نمایش می‌دهیم به صورت زیر تعریف می‌شود:

$$H(X,Y) = -\sum_{x}\sum_{y} p_{X,Y}(x,y) \log_2\big[p_{X,Y}(x,y)\big]$$

همچنین آنتروپی شرطی Y به شرط X که آن را با $H(Y|X)$ نمایش می‌دهیم به صورت زیر تعریف می‌شود:

$$H(Y|X) = -\sum_{x}\sum_{y} p_{X,Y}(x,y) \log_2\big[p_{Y|X}(y|x)\big]$$

که در آن

$$p_{Y|X}(y|x) = \frac{p(x,y)}{p(x)}$$

تابع احتمال شرطی Y به شرط X است. به راحتی می‌توان نشان داد که بین این سه معیار رابطه زیر برقرار است که به آن قاعده زنجیره‌ای گفته می‌شود

$$H(X,Y) = H(Y|X) + H(X)$$

اگر X, Y دو متغیر تصادفی مستقل باشند، رابطه بالا به صورت زیر تبدیل می‌شود

$$H(X,Y) = H(Y) + H(X)$$

لازم به ذکر است که نتایج فوق را می‌توان به حالت‌هایی که تعداد بیشتری متغیر تصادفی وجود دارد نیز تعمیم داد. همچنین با تبدیل سیگما به انتگرال، به راحتی می‌توان این تعاریف را برای متغیرهای تصادفی پیوسته نیز به کار گرفت.

اگر X, Y دو متغیر تصادفی (گسسته) با تابع احتمال توام $p_{X,Y}(x,y)$ و توابع احتمال حاشیه‌ای $p_X(x)$ و $p_Y(y)$ باشند، در این صورت آنتروپی نسبی بین این دو تابع احتمال، که به آن فاصله یا واگرایی کولبک-لیبلر نیز گفته می‌شود و معمولا با $D(p_X||p_Y)$ نشان داده می‌شود، به صورت زیر تعریف می‌شود:

$$D(p_X||p_Y) = \sum_x p_X(x) \log_2 \left[\frac{p_X(x)}{p_Y(x)}\right]$$

در آمار ریاضی از آنتروپی نسبی (یا واگرایی کولبک-لیبلر) به عنوان معیاری برای اندازه‌گیری واگرایی یک توزیع احتمال از یک توزیع احتمال ثانویه، یاد می‌شود. از جمله کاربردهای این مفهوم شامل توصیف سیستم‌های اطلاعاتی، میزان تصادفی‌بودن داده‌های سری‌های زمانی پیوسته و بهره اطلاعاتی (در زمانی که هدف مقایسه مدل‌های آماری است) می‌باشد.

بهره اطلاعاتی یا اطلاع متقابل بین دو متغیر تصادفی X, Y در واقع فاصله کولبک-لیبلر بین $p(x,y)$ و $p_X(x)p_Y(y)$ است. بهره اطلاعاتی که آن را با $I(X,Y)$ نشان می‌دهیم به صورت زیر تعریف می‌شود

$$I(X,Y) = \sum_x \sum_y p_{X,Y}(x,y) \log_2 \left[\frac{p(x,y)}{p_X(x)p_Y(y)}\right]$$

به راحتی می‌توان نشان داد که

$$I(X,Y) = H(Y) - H(Y|X) = H(X) - H(X|Y)$$

مثال: فرض کنید X, Y دو متغیر تصادفی با توزیع توام زیر باشند

$$p_{X,Y}(x,y) = \frac{x+y}{12};\ x = 1,2;\ \ y = 1,2$$

می‌خواهیم معیارهای داده شده در بالا را محاسبه کنیم. ابتدا داریم

$$p_X(x) = \sum_y p(x,y) = \sum_{y=1}^{2} \frac{x+y}{12} = \frac{2x+3}{12}; x = 1,2$$

$$p_Y(y) = \sum_x p(x,y) = \sum_{x=1}^{2} \frac{x+y}{12} = \frac{2y+3}{12}; y = 1,2$$

همچنین تابع احتمال شرطی Y به شرط X نیز به صورت زیر است

$$p_{Y|X}(y|x) = \frac{p_{X,Y}(x,y)}{p_X(x)} = \frac{\frac{x+y}{12}}{\frac{2x+3}{12}} = \frac{x+y}{2x+3};\ x = 1,2;\ \ y = 1,2$$

بنابراین

$$H(X) = -\sum_x p_X(x)\log_2[p_X(x)] = -p_X(1)\log_2[p_X(1)] - p_X(2)\log_2[p_X(2)]$$

$$= -\frac{5}{12}\log_2\left[\frac{5}{12}\right] - \frac{7}{12}\log_2\left[\frac{7}{12}\right] = 0.9799$$

واحد اندازه‌گیری اطلاع بیت است، بنابراین مقدار $H(X)$ برابر با 0.9799 بیت است. با توجه به توزیع یکسان دو متغیر تصادفی X و Y داریم

$$H(Y) = 0.9799$$

آنتروپی توام این دو متغیر تصادفی به صورت زیر است

$$H(X,Y) = -\sum_x\sum_y p_{X,Y}(x,y)\log_2[p_{X,Y}(x,y)] = -\sum_{x=1}^{2}\sum_{y=1}^{2}\frac{x+y}{12}\log_2\left[\frac{x+y}{12}\right]$$

$$= -\left[\frac{2}{12}\log_2\left(\frac{2}{12}\right) + \frac{3}{12}\log_2\left(\frac{3}{12}\right) + \frac{3}{12}\log_2\left(\frac{3}{12}\right) + \frac{4}{12}\log_2\left(\frac{4}{12}\right)\right] = 1.9591$$

همچنین آنتروپی شرطی Y به شرط X به صورت زیر است

$$H(Y|X) = -\sum_x\sum_y p_{X,Y}(x,y)\log_2[p_{Y|X}(y|x)] = -\sum_{x=1}^{2}\sum_{y=1}^{2}\frac{x+y}{12}\log_2\left[\frac{x+y}{2x+3}\right]$$

$$= -\left[\frac{2}{12}\log_2\left(\frac{2}{5}\right) + \frac{3}{12}\log_2\left(\frac{3}{5}\right) + \frac{3}{12}\log_2\left(\frac{3}{7}\right) + \frac{4}{12}\log_2\left(\frac{4}{7}\right)\right] = 0.9793$$

با توجه به محاسبات بالا داریم

$$H(Y|X) + H(X) = 0.9793 + 0.9799 = 1.9592 = H(X,Y)$$

دلیل تفاوت اندک بین $H(Y|X) + H(X)$ و $H(X,Y)$ خطای گرد کردن می‌باشد.

واگرایی کولبک-لیبلر نیز به صورت زیر است

$$D(p_X||p_Y) = \sum_x p_X(x)\log_2\left[\frac{p_X(x)}{p_Y(x)}\right] = \sum_{x=1}^{2}\frac{2x+3}{12}\log_2\left(\frac{\frac{2x+3}{12}}{\frac{2x+3}{12}}\right)$$

$$= \sum_{x=1}^{2}\frac{2x+3}{12}\log_2(1) = 0$$

و بالاخره بهره اطلاعاتی یا اطلاع متقابل بین دو متغیر تصادفی X, Y به صورت زیر است

$$I(X,Y) = \sum_x\sum_y p_{X,Y}(x,y)\log_2\left[\frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)}\right] = \sum_{x=1}^{2}\sum_{y=1}^{2}\frac{x+y}{12}\log_2\left[\frac{\frac{x+y}{12}}{\frac{2x+3}{12}\frac{2y+3}{12}}\right]$$

$$= \sum_{x=1}^{2}\sum_{y=1}^{2}\frac{x+y}{12}\log_2\left[\frac{12(x+y)}{(2x+3)(2y+3)}\right]$$

$$= \frac{2}{12}\log_2\left(\frac{24}{25}\right) + \frac{3}{12}\log_2\left(\frac{36}{35}\right) + \frac{3}{12}\log_2\left(\frac{36}{35}\right) + \frac{4}{12}\log_2\left(\frac{48}{49}\right) = 0.0005896$$

همچنین داریم

$$H(Y) - H(Y|X) = 0.9799 - 0.9793 = 0.0006 = I(X,Y)$$

که درستی رابطه

$$I(X,Y) = H(Y) - H(Y|X) = H(X) - H(X|Y)$$

را نشان می‌دهد. تفاوت کوچک بین $I(X,Y)$ و $H(Y) - H(Y|X)$ ناشی از خطای گرد کردن است. ■

در مثال زیر نحوه استفاده از بهره اطلاعاتی برای تعیین گره‌ی ریشه و همچنین ترتیب قرارگیری گره‌ها در درخت تصمیم مورد بحث قرار می‌گیرد.

مثال: جدول زیر یک مجموعه داده آموزشی را نشان می‌دهد.

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

جهت راحتی فرض کنیم:

$$Y = \begin{cases} 1 & buys\ computer = yes \\ 2 & buys\ computer = no \end{cases}$$

$$X_1 = \begin{cases} 1 & age = youth \\ 2 & age = middle\_aged \\ 3 & age = senior \end{cases};\ X_2 = \begin{cases} 1 & income = high \\ 2 & income = medium \\ 3 & income = low \end{cases}$$

$$X_3 = \begin{cases} 1 & student = yes \\ 2 & student = no \end{cases};\ X_4 = \begin{cases} 1 & credit\_rating = fair \\ 2 & credit\_rating = excellent \end{cases}$$

دیدیم که برای محاسبه آنتروپی و بهره اطلاعاتی نیاز به توزیع‌های توام این متغیرهای تصادفی داریم. اما در اینجا و کلا در الگوریتم‌های مربوط به درخت تصمیم یک چنین توزیع‌هایی در دسترس نیستند. بنابراین احتمالات مربوط به این توزیع‌ها را با استفاده از داده‌های آموزشی برآورد می‌کنند. با توجه به داده‌های فوق، برآورد تابع احتمال‌های حاشیه‌ای، تابع احتمال توام Y و X_1، تابع احتمال توام Y و X_2، تابع احتمال توام Y و X_3 و تابع احتمال توام Y و X_4 به صورت زیر هستند:

$$\hat{p}_Y(y) = \begin{cases} \frac{9}{14} & y = 1 \\ \frac{5}{14} & y = 2 \end{cases}$$

$$\hat{p}_{X_1}(x) = \begin{cases} \frac{5}{14} & x = 1 \\ \frac{4}{14} & x = 2 \\ \frac{5}{14} & x = 2 \end{cases} ; \hat{p}_{X_2}(x) = \begin{cases} \frac{4}{14} & x = 1 \\ \frac{6}{14} & x = 2 \\ \frac{4}{14} & x = 2 \end{cases}$$

$$\hat{p}_{X_3}(x) = \begin{cases} \frac{7}{14} & x = 1 \\ \frac{7}{14} & x = 2 \end{cases} ; \hat{p}_{X_4}(x) = \begin{cases} \frac{8}{14} & x = 1 \\ \frac{6}{14} & x = 2 \end{cases}$$

$$\hat{p}_{X_1,Y}(x,y) = \begin{cases} \frac{2}{14} & x = 1, y = 1 \\ \frac{4}{14} & x = 2, y = 1 \\ \frac{3}{14} & x = 3, y = 1 \\ \frac{3}{14} & x = 1, y = 2 \\ 0 & x = 2, y = 2 \\ \frac{2}{14} & x = 3, y = 2 \end{cases} ; \hat{p}_{X_2,Y}(x,y) = \begin{cases} \frac{2}{14} & x = 1, y = 1 \\ \frac{4}{14} & x = 2, y = 1 \\ \frac{3}{14} & x = 3, y = 1 \\ \frac{2}{14} & x = 1, y = 2 \\ \frac{2}{14} & x = 2, y = 2 \\ \frac{1}{14} & x = 3, y = 2 \end{cases}$$

$$\hat{p}_{X_3,Y}(x,y) = \begin{cases} \frac{6}{14} & x = 1, y = 1 \\ \frac{3}{14} & x = 2, y = 1 \\ \frac{1}{14} & x = 1, y = 2 \\ \frac{4}{14} & x = 2, y = 2 \end{cases} ; \hat{p}_{X_4,Y}(x,y) = \begin{cases} \frac{6}{14} & x = 1, y = 1 \\ \frac{3}{14} & x = 2, y = 1 \\ \frac{2}{14} & x = 1, y = 2 \\ \frac{3}{14} & x = 2, y = 2 \end{cases}$$

برای تعیین اینکه کدام‌یک از متغیرها باید در گره‌ی ریشه قرار بگیرد بهره اطلاعاتی متغیر Y با تمامی متغیرهای دیگر را محاسبه می‌کنیم. هر متغیری که بهره اطلاعاتی آن با Y بیشتر از دیگران باشد در گره‌ی ریشه قرار می‌گیرد. داریم:

$$H(Y) = -\sum_{y} \hat{p}_Y(y) \log_2[\hat{p}_Y(y)] = -\hat{p}_Y(1) \log_2[\hat{p}_Y(1)] - \hat{p}_Y(2) \log_2[\hat{p}_Y(2)]$$

$$= -\frac{9}{14} \log_2\left[\frac{9}{14}\right] - \frac{5}{14} \log_2\left[\frac{5}{14}\right] = 0.940\ bits$$

$$H(X_1) = -\sum_x \hat{p}_{X_1}(x)\log_2[\hat{p}_{X_1}(x)] = -\frac{5}{14}\log_2\left[\frac{5}{14}\right] - \frac{4}{14}\log_2\left[\frac{4}{14}\right] - \frac{5}{14}\log_2\left[\frac{5}{14}\right]$$
$$= 1.577$$

$$H(X_2) = -\sum_x \hat{p}_{X_2}(x)\log_2[\hat{p}_{X_2}(x)] = -\frac{4}{14}\log_2\left[\frac{4}{14}\right] - \frac{6}{14}\log_2\left[\frac{6}{14}\right] - \frac{4}{14}\log_2\left[\frac{4}{14}\right]$$
$$= 1.557$$

$$H(X_3) = -\sum_x \hat{p}_{X_3}(x)\log_2[\hat{p}_{X_3}(x)] = -\frac{7}{14}\log_2\left[\frac{7}{14}\right] - \frac{7}{14}\log_2\left[\frac{7}{14}\right] = 1$$

$$H(X_4) = -\sum_x \hat{p}_{X_4}(x)\log_2[\hat{p}_{X_4}(x)] = -\frac{8}{14}\log_2\left[\frac{8}{14}\right] - \frac{6}{14}\log_2\left[\frac{6}{14}\right] = 0.985$$

آنتروپی‌های توام به صورت زیر هستند

$$H(X_1, Y) = -\sum_x \sum_y \hat{p}_{X_1,Y}(x,y)\log_2[\hat{p}_{X_1,Y}(x,y)]$$
$$= -\left[\frac{2}{14}\log_2\left(\frac{2}{14}\right) + \frac{4}{14}\log_2\left(\frac{4}{14}\right) + \frac{3}{14}\log_2\left(\frac{3}{14}\right) + \frac{3}{14}\log_2\left(\frac{3}{14}\right)\right.$$
$$\left. + \frac{2}{14}\log_2\left(\frac{2}{14}\right)\right] = 2.271$$

$$H(X_2, Y) = -\sum_x \sum_y \hat{p}_{X_2,Y}(x,y)\log_2[\hat{p}_{X_2,Y}(x,y)]$$
$$= -\left[\frac{2}{14}\log_2\left(\frac{2}{14}\right) + \frac{4}{14}\log_2\left(\frac{4}{14}\right) + \frac{3}{14}\log_2\left(\frac{3}{14}\right) + \frac{2}{14}\log_2\left(\frac{2}{14}\right)\right.$$
$$\left. + \frac{2}{14}\log_2\left(\frac{2}{14}\right) + \frac{1}{14}\log_2\left(\frac{1}{14}\right)\right] = 2.468$$

$$H(X_3, Y) = -\sum_x \sum_y \hat{p}_{X_3,Y}(x,y)\log_2[\hat{p}_{X_3,Y}(x,y)]$$
$$= -\left[\frac{6}{14}\log_2\left(\frac{6}{14}\right) + \frac{3}{14}\log_2\left(\frac{3}{14}\right) + \frac{1}{14}\log_2\left(\frac{1}{14}\right) + \frac{4}{14}\log_2\left(\frac{4}{14}\right)\right]$$
$$= 1.788$$

$$H(X_4, Y) = -\sum_x \sum_y \hat{p}_{X_4,Y}(x,y)\log_2[\hat{p}_{X_4,Y}(x,y)]$$
$$= -\left[\frac{6}{14}\log_2\left(\frac{6}{14}\right) + \frac{3}{14}\log_2\left(\frac{3}{14}\right) + \frac{2}{14}\log_2\left(\frac{2}{14}\right) + \frac{3}{14}\log_2\left(\frac{3}{14}\right)\right]$$
$$= 1.877$$

بنابراین آنتروپی‌های شرطی به صورت زیر هستند:

$$H(Y|X_1) = H(X_1, Y) - H(X_1) = 2.271 - 1.577 = 0.694$$
$$H(Y|X_2) = H(X_2, Y) - H(X_2) = 2.468 - 1.557 = 0.689$$
$$H(Y|X_3) = H(X_3, Y) - H(X_3) = 1.788 - 1 = 0.788$$
$$H(Y|X_4) = H(X_4, Y) - H(X_4) = 1.877 - 0.985 = 0.892$$

در نتیجه بهره اطلاعاتی بین متغیرها به صورت زیر است

$$I(X_1, Y) = H(Y) - H(Y|X_1) = 0.940 - 0.694 = 0.246$$
$$I(X_2, Y) = H(Y) - H(Y|X_2) = 0.940 - 0.689 = 0.251$$
$$I(X_3, Y) = H(Y) - H(Y|X_3) = 0.940 - 0.788 = 0.152$$
$$I(X_4, Y) = H(Y) - H(Y|X_4) = 0.940 - 0.892 = 0.048$$

با توجه به اینکه $I(X_2, Y)$ دارای بیشترین مقدار است، بنابراین متغیر X_2 یا به بیان دقیق‌تر متغیر *income* در گره‌ی ریشه قرار می‌گیرد. ■

لازم به ذکر است که به جای بهره اطلاعاتی، می‌توان از معیارهای دیگری نیز استفاده کرد که معروف‌ترین آنها ضریب جینی است. همچنین الگوریتم‌های مختلفی برای درخت تصمیم وجود دارد که در زیر به برخی از مهمترین‌های آنها اشاره می‌شود.

الگوریتم ID3: الگوریتم ID3 یکی از اولین الگوریتم‌هایی است که برای ساخت درخت تصمیم ارائه شده است. این الگوریتم از بهره اطلاعاتی برای انتخاب ویژگی‌ها جهت تقسیم داده‌ها استفاده می‌کند.

الگوریتم C4.5: این الگوریتم نسخه‌ای به روز شده از الگوریتم ID3 است. ID3 فقط با ویژگی‌های گسسته می‌تواند مورد استفاده قرار گیرد در حالی که الگوریتم C4.5 می‌تواند هم با ویژگی‌های گسسته و هم با ویژگی‌های پیوسته کار کند. علاوه بر این، الگوریتم C4.5 از معیار نسبت بهره‌وری[1] استفاده می‌کند که نسبت بهره اطلاعاتی یک ویژگی به آنتروپی آن ویژگی است.

الگوریتم CART: این الگوریتم نیز یک الگوریتم دیگر برای ساخت درخت تصمیم است که برای مسائل

[1] Gain Ratio

طبقه‌بندی و رگرسیون قابل استفاده است. این الگوریتم از ضریب جینی برای انتخاب ویژگی‌ها استفاده می‌کند و درخت‌های دو-دویی می‌سازد.

الگوریتم CHID: الگوریتمی است که از آزمون آماری کای-دو برای ارزیابی ویژگی‌ها و انتخاب بهترین ویژگی برای تقسیم داده‌ها استفاده می‌کند.

مثالی کاربردی از درخت تصمیم در نرم افزار R:

در این بخش در نرم‌افزار R با نحوه ترسیم چنین درختی آشنا خواهیم شد بنابراین این روش را بر روی مجموعه داده Heart پیاده‌سازی می‌نماییم.

شرح مجموعه داده

داده‌های این پروژه شامل سوابق پزشکی 299 بیمارِ مبتلا به نارسایی قلبی است که در آوریل و دسامبر سال 2015 در بیمارستان بستری بوده‌اند. برای هر بیمار 13 متغیر اندازه‌گیری شده است. 12 متغیر که به نظر می‌رسد در مرگ بیمارانی که نارسایی قلبی دارند موثر هستند و متغیر دیگری با نام «وقوع مرگ» که هدف از جمع آوری این داده‌ها بررسی اثر 12 متغیر دیگر بر روی این متغیر بوده است. در واقع متغیر پاسخ یا هدف ما متغیر DEATH می‌باشد.

متغیرهایی که در مجموعه داده وجود دارند به صورت زیر هستند:

شماره	متغیر	توضیحات متغیر	نوع متغیر
1	Age	سن بیمار بر حسب سال	کمی
2	Anaemia	کم خونی (1=دارد و 0=ندارد)	اسمی دارای دو سطح
3	Phosphokinase	کراتین فسفوکیناز نشان‌دهنده سطح آنزیم CPK در خون است و برحسب میکروگرم بر لیتر اندازه‌گیری می‌شود.	کمی
4	Diabetes	دیابت (1=دارد و 0=ندارد)	اسمی دارای دو سطح
5	Ejection	درصد خون خروجی از قلب در هر انقباض (به درصد)	کمی
6	Pressure	فشار خون بالا (1=دارد و 0=ندارد)	اسمی دارای دو سطح
7	Platelets	پلاکت‌های موجود در خون (کیلوپلاکت در میلی لیتر)	کمی
8	Creatinine	سطح کراتین در خون(mg/dL)	کمی
9	Sodium	سطح سدیم در خون(mEq/L)	کمی
10	Sex	جنسیت (1=مرد و 0=زن)	اسمی دارای دو سطح

11	Smoking	سیگار کشیدن (1=بله و 0=خیر)	اسمی دارای دو سطح
12	Time	تعداد روزهایی که بیمار تحت نظر بوده است	کمی گسسته
13	DEATH	فوت (1=بله و 0=خیر)	اسمی دارای دو سطح

برای این داده‌ها متغیر پاسخ وقوع مرگ (DEATH یا در مجموعه‌ی داده‌ها DEATH_EVENT) بوده که یک متغیر کیفی با دو سطح است.

داده‌ها از طریق لینک‌های زیر قابل دانلود هستند:

https://archive.ics.uci.edu/dataset/519/heart+failure+clinical+records

https://www.kaggle.com/datasets/andrewmvd/heart-failure-clinical-data

یکی از مراحل مهم در هر گونه مطالعه، به‌کارگیری و دسترسی به داده‌های اولیه خوب ومناسب است؛ که از آن به آماده‌سازی یا پیش پردازش داده‌ها یاد می شود. در واقع برای رسیدن به نتایج محتمل بایستی مقدماتی صورت گیرد؛ که مجموعه این اقدامات را آماده‌سازی داده‌ها گویند.

در ابتدا داده‌ها را فراخوانی می‌کنیم.

```
#read data
Heart=read.csv("C:/Users/acer/Desktop/heart.csv")
head(Heart)
```

```
  age anaemia creatinine_phosphokinase diabetes ejection_fraction high_blood_pressure platelets
1  75       0                     582        0                20                   1    265000
2  55       0                    7861        0                38                   0    263358
3  65       0                     146        0                20                   0    162000
4  50       1                     111        0                20                   0    210000
5  65       1                     160        1                20                   0    327000
6  90       1                      47        0                40                   1    204000
  serum_creatinine serum_sodium sex smoking time DEATH_EVENT
1              1.9          130   1       0    4           1
2              1.1          136   1       0    6           1
3              1.3          129   1       1    7           1
4              1.9          137   1       0    7           1
5              2.7          116   0       0    8           1
6              2.1          132   1       1    8           1
```

برای شناخت بیشتر داده‌ها از کدهای زیر استفاده می‌کنیم.

```
str(Heart)
```

```
> str(Heart)
'data.frame':   299 obs. of  13 variables:
 $ age                     : num  75 55 65 50 65 90 75 60 65 80 ...
 $ anaemia                 : int  0 0 0 1 1 1 1 1 0 1 ...
 $ creatinine_phosphokinase: int  582 7861 146 111 160 47 246 315 157 123 ...
 $ diabetes                : int  0 0 0 0 1 0 0 1 0 0 ...
 $ ejection_fraction       : int  20 38 20 20 20 40 15 60 65 35 ...
 $ high_blood_pressure     : int  1 0 0 0 0 1 0 0 0 1 ...
 $ platelets               : num  265000 263358 162000 210000 327000 ...
 $ serum_creatinine        : num  1.9 1.1 1.3 1.9 2.7 2.1 1.2 1.1 1.5 9.4 ...
 $ serum_sodium            : int  130 136 129 137 116 132 137 131 138 133 ...
 $ sex                     : int  1 1 1 1 0 1 1 1 0 1 ...
 $ smoking                 : int  0 0 1 0 0 1 0 1 0 1 ...
 $ time                    : int  4 6 7 7 8 8 10 10 10 10 ...
 $ DEATH_EVENT             : int  1 1 1 1 1 1 1 1 1 1 ...
```

تابع بالا تعداد متغیرها و مشاهدات داده‌ی مورد نظر را نمایش می‌دهد و همچنین مشخص می‌کند متغیرهای ما از چه نوعی هستند.

در بسیاری از تحقیقات با مواردی برخورد می‌کنیم که برای بعضی از متغیرها پاسخ یا جوابی وجود ندارد یا اینکه پاسخ نامعلوم بوده و قادر به استخراج آن نیستیم. به این مقادیر، مقادیر نامعلوم یا داده‌های گم‌شده می‌گوییم.

داده‌های گم‌شده مشکل‌های بسیاری را برای نتیجه‌گیری‌های صحیح پژوهش‌های آماری به وجود می‌آورند، زیرا می‌توانند قدرت استنباط آماری مطالعه را کاهش داده، باعث کاهش کارایی برآوردگرها و ایجاد اریبی شوند؛ بنابراین برای جلوگیری از مشکلاتی مانند کم شدن حجم نمونه و یا عدم حضور واحدهایی از نمونه در مجموعه ی داده‌ها باید با استفاده از روش‌های مناسب جایگذاری شوند.

برای اطلاعات راجب داده‌های گمشده و تعداد آن‌ها داریم:

```
colSums(is.na(Heart))
```

```
> colSums(is.na(Heart))
                      age                   anaemia creatinine_phosphokinase                  diabetes
                        0                         0                         0                         0
       ejection_fraction       high_blood_pressure                 platelets          serum_creatinine
                        0                         0                         0                         0
            serum_sodium                       sex                   smoking                      time
                        0                         0                         0                         0
             DEATH_EVENT
                        0
```

به‌دلیل عدم وجود مقادیر گم‌شده در مجموعه داده‌ها سراغ گام بعدی می‌رویم و برای آگاه شدن از شاخص‌های مرکزی در نرم افزار R از طریق تابع summary خلاصه‌ای از متغیرهای جامعه ارائه می‌دهیم.

```
summary(Heart)
```

خروجی به صورت زیر است:

```
> summary(Heart)
      age            anaemia       creatinine_phosphokinase    diabetes      ejection_fraction
 Min.   :40.00   Min.   :0.0000   Min.   :  23.0           Min.   :0.0000   Min.   :14.00
 1st Qu.:51.00   1st Qu.:0.0000   1st Qu.: 116.5           1st Qu.:0.0000   1st Qu.:30.00
 Median :60.00   Median :0.0000   Median : 250.0           Median :0.0000   Median :38.00
 Mean   :60.83   Mean   :0.4314   Mean   : 581.8           Mean   :0.4181   Mean   :38.08
 3rd Qu.:70.00   3rd Qu.:1.0000   3rd Qu.: 582.0           3rd Qu.:1.0000   3rd Qu.:45.00
 Max.   :95.00   Max.   :1.0000   Max.   :7861.0           Max.   :1.0000   Max.   :80.00
high_blood_pressure   platelets      serum_creatinine  serum_sodium        sex            smoking
 Min.   :0.0000      Min.   : 25100   Min.   :0.500    Min.   :113.0   Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.0000      1st Qu.:212500   1st Qu.:0.900    1st Qu.:134.0   1st Qu.:0.0000   1st Qu.:0.0000
 Median :0.0000      Median :262000   Median :1.100    Median :137.0   Median :1.0000   Median :0.0000
 Mean   :0.3512      Mean   :263358   Mean   :1.394    Mean   :136.6   Mean   :0.6488   Mean   :0.3211
 3rd Qu.:1.0000      3rd Qu.:303500   3rd Qu.:1.400    3rd Qu.:140.0   3rd Qu.:1.0000   3rd Qu.:1.0000
 Max.   :1.0000      Max.   :850000   Max.   :9.400    Max.   :148.0   Max.   :1.0000   Max.   :1.0000
      time        DEATH_EVENT
 Min.   :  4.0   Min.   :0.0000
 1st Qu.: 73.0   1st Qu.:0.0000
 Median :115.0   Median :0.0000
 Mean   :130.3   Mean   :0.3211
 3rd Qu.:203.0   3rd Qu.:1.0000
 Max.   :285.0   Max.   :1.0000
```

با توجه به خروجی حاصل از تابع()summary مقادیر مینیمم، چارک اول، میانه، میانگین، چارک سوم و ماکزیمم برای هر متغیر موجود در جامعه، قابل مشاهده می‌باشد.

اغلب در مجموعه‌ی بزرگی از داده‌ها نمونه‌هایی وجود دارند که رفتارشان با رفتار عمومی نمونه‌ها یکسان نیست. این رفتار یا کامل مختلف است و یا با دیگر نمونه‌ها ناسازگارند. به عبارتی دیگر همیشه داده‌های ما ناقص نیستند، می‌توانند وجود داشته باشند، اما با رفتاری متفاوت از اکثر نمونه‌های موجود.

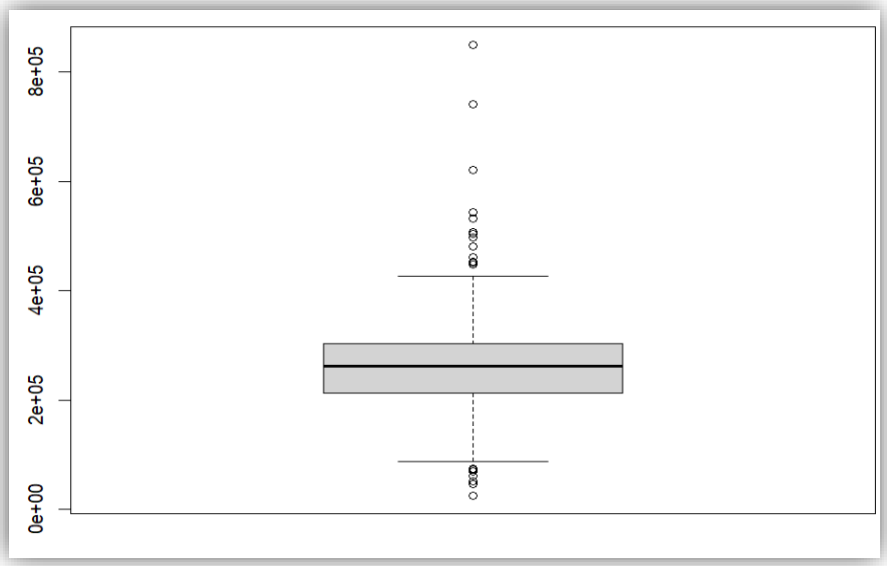
با توجه اینکه این داده‌ها می‌توانند نتایج تحلیل را تحت تاثیر قرار دهند، می‌بایست تمامی داده‌های پرت در ابتدا شناسایی و بررسی شوند. نویز با داده‌ی پرت متفاوت است، نویز خطای یا واریانس تصادفی در داده است که باید قبل از تشخیص داده پرت حذف شود

برای بررسی داده‌های پرت موجود در مشاهدات مربوط به هر متغیر به رسم نمودار جعبه‌ای آن‌ها می‌پردازیم.

نمودار جعبه‌ای مربوط به متغیر platelets:

```
Heart%<%
  ggplot(aes( y = platelets))+
  geom_boxplot()
out=boxplot(Heart$platelets)$out
head(out)
```

```
> head(out)
[1] 454000  47000 451000 461000 497000 621000
```

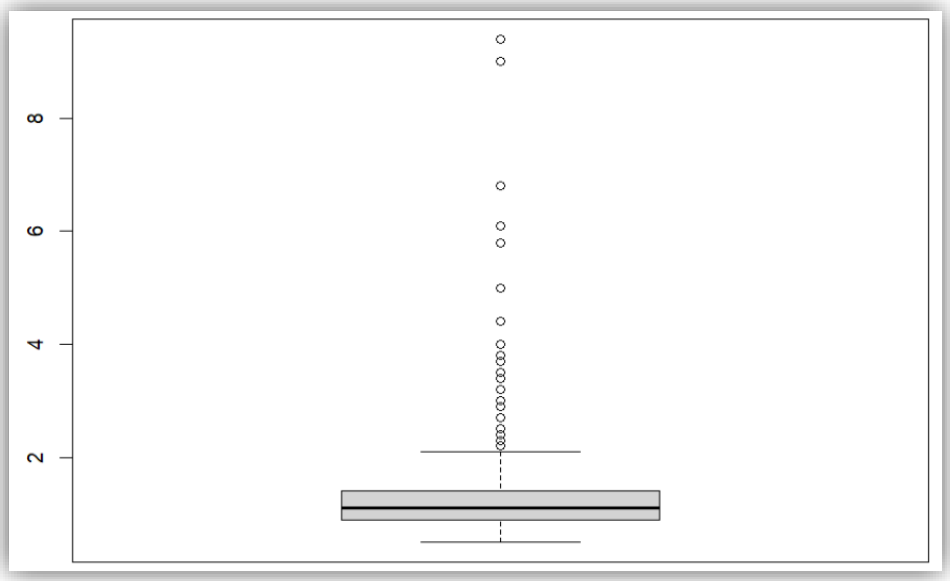


نمودار جعبه‌ای مربوط به متغیر serum_creatinine:

```
Heart%<%
  ggplot(aes( y = serum_creatinine))+
  geom_boxplot()
out=boxplot(Heart$serum_creatinine)$out
head(out)
length(out)
```

```
> head(out)
[1] 2.7 9.4 4.0 5.8 3.0 3.5
> length(out)
[1] 29
```

همانطور که از خروجی مشخص این متغیر شامل 29 مشاهده به عنوان داده پرت می‌باشد.

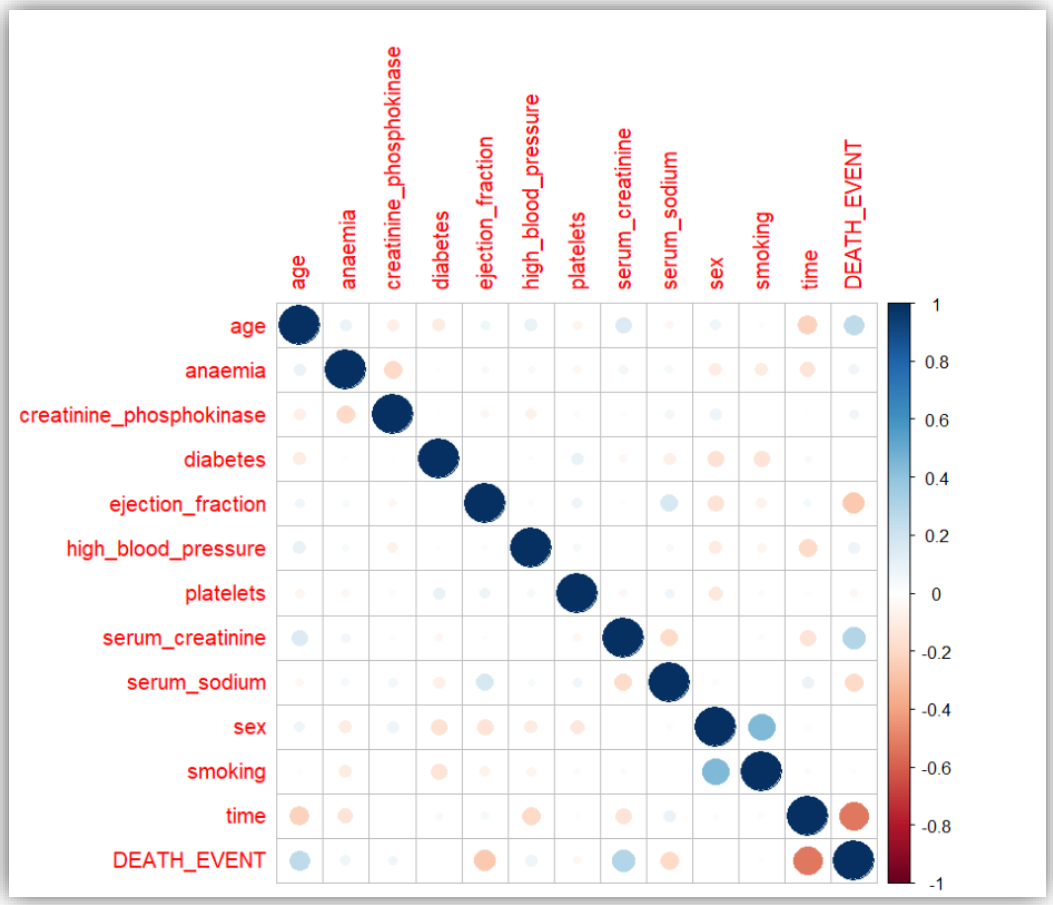


برای شناخت بهتر مشاهدات می‌توانیم از یکی دیگر از شاخص‌های پراکندگی، کوواریانس، و محاسبه‌ی ماتریس همبستگی متغیرهای کمی بهره ببریم.

```
# Select only numeric variables for correlation
numeric_data <- Heart[, sapply(Heart, is.numeric)]
if (ncol(numeric_data) > 1) {
  cor_matrix <- cor(numeric_data, method = "pearson")
  print("Correlation Matrix:")
  print(cor_matrix)
  library(corrplot)
  corrplot(cor_matrix, method = "circle")
} else {
  print("Not enough numeric variables for a correlation matrix.")
}
```

```
[1] "Correlation Matrix:"
                                age     anaemia creatinine_phosphokinase     diabetes ejection_fraction
age                     1.00000000  0.08800644             -0.081583900 -0.101012385        0.06009836
anaemia                 0.08800644  1.00000000             -0.190741030 -0.012729046        0.03155697
creatinine_phosphokinase -0.08158390 -0.19074103              1.000000000 -0.009638514       -0.04407955
diabetes                -0.10101239 -0.01272905             -0.009638514  1.000000000       -0.00485031
ejection_fraction        0.06009836  0.03155697             -0.044079554 -0.004850310        1.00000000
high_blood_pressure      0.09328868  0.03818200             -0.070589980 -0.012732382        0.02444473
platelets               -0.05235437 -0.04378555              0.024463389  0.092192828        0.07217747
serum_creatinine         0.15918713  0.05217360             -0.016408480 -0.046975315       -0.01130247
serum_sodium            -0.04596584  0.04188161              0.059550156 -0.089550619        0.17590228
sex                      0.06542952 -0.09476896              0.079790629 -0.157729504       -0.14838597
smoking                  0.01866787 -0.10728984              0.002421235 -0.147173413       -0.06731457
time                    -0.22406842 -0.14141398             -0.009345653  0.033725509        0.04172924
DEATH_EVENT              0.25372854  0.06627010              0.062728160 -0.001942883       -0.26860331
                        high_blood_pressure   platelets serum_creatinine serum_sodium          sex
age                          0.093288685 -0.05235437      0.159187133 -0.045965841  0.065429524
anaemia                      0.038182003 -0.04378555      0.052173604  0.041881610 -0.094768961
creatinine_phosphokinase    -0.070589980  0.02446339     -0.016408480  0.059550156  0.079790629
diabetes                    -0.012732382  0.09219283     -0.046975315 -0.089550619 -0.157729504
ejection_fraction            0.024444731  0.07217747     -0.011302475  0.175902282 -0.148385965
high_blood_pressure          1.000000000  0.04996348     -0.004934525  0.037109470 -0.104614629
platelets                    0.049963481  1.00000000     -0.041198077  0.062124619 -0.125120483
serum_creatinine            -0.004934525 -0.04119808      1.000000000 -0.189095210  0.006969778
serum_sodium                 0.037109470  0.06212462     -0.189095210  1.000000000 -0.027566123
sex                         -0.104614629 -0.12512048      0.006969778 -0.027566123  1.000000000
smoking                     -0.055711369  0.02823445     -0.027414135  0.004813195  0.445891712
time                        -0.196439479  0.01051391     -0.149315418  0.087640000 -0.015608220
DEATH_EVENT                  0.079351058 -0.04913887      0.294277561 -0.195203596 -0.004316376
```

	smoking	time	DEATH_EVENT
age	0.018667868	-0.224068420	0.253728543
anaemia	-0.107289838	-0.141413982	0.066270098
creatinine_phosphokinase	0.002421235	-0.009345653	0.062728160
diabetes	-0.147173413	0.033725509	-0.001942883
ejection_fraction	-0.067314567	0.041729235	-0.268603312
high_blood_pressure	-0.055711369	-0.196439479	0.079351058
platelets	0.028234448	0.010513909	-0.049138868
serum_creatinine	-0.027414135	-0.149315418	0.294277561
serum_sodium	0.004813195	0.087640000	-0.195203596
sex	0.445891712	-0.015608220	-0.004316376
smoking	1.000000000	-0.022838942	-0.012623153
time	-0.022838942	1.000000000	-0.526963779
DEATH_EVENT	-0.012623153	-0.526963779	1.000000000

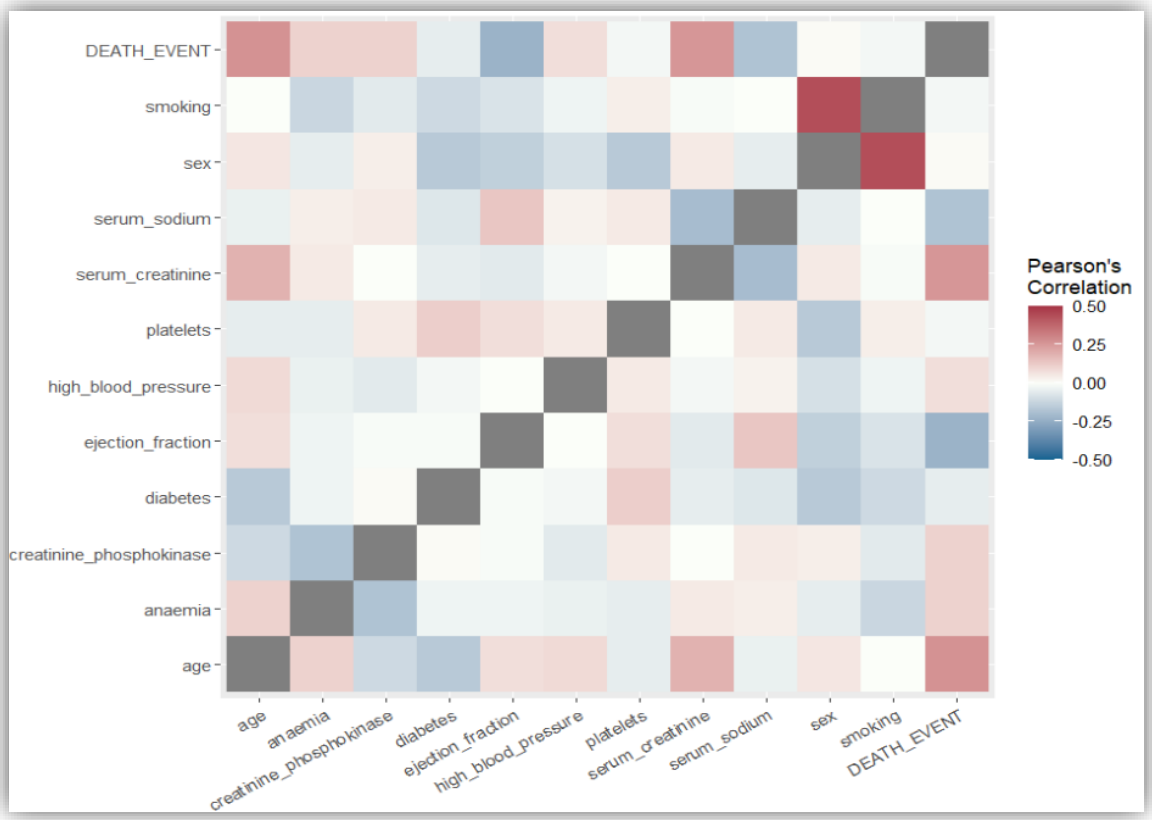


با توجه به خروجی به دست آمده توسط محاسبه‌ی همبستگی متغیرها با یکدیگر، ضریب همبستگی هر متغیر با خودش برابر با مقدار 1 است، و مقادیر منفی بیانگر وجود رابطه‌ی معکوس بین دو متغیر و مقادیر مثبت بیانگر رابطه‌ی هم‌جهت مابین آن دو می‌باشد.

هر چه این مقادیر نزدیک یا برابر مقدار 1 باشند، رابطه شدید و هم‌جهت بین دو متغیر موجود است، در این حالت می‌توان گفت که جهت تغییرات هر دو متغیر مانند یکدیگر است و بین دو متغیر رابطه مستقیم وجود دارد. بالعکس، اگر ضریب همبستگی، مقداری نزدیک یا برابر با ۱- باشد، رابطه شدید ولی در جهت عکس بین متغیرها وجود دارد. بنابراین با افزایش یکی، دیگری کاهش خواهد.

از طریق نمودار حرارتی نیز می‌توان همبستگی بین متغیرها را مشاهده نمود.

```
library(reshape2)
train_num <- data_num[train_index,]
cormat <- round(cor(train_num),2)
melted_cormat <- melt(cormat)
ggplot(data = melted_cormat, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile() +
  labs(x = NULL, y = NULL, fill = "Pearson's\nCorrelation") +
  scale_fill_gradient2(mid="#FBFEF9",low="#0C6291",high="#A63446", limits=c(-
0.5,0.5)) +
  scale_x_discrete(guide = guide_axis(angle = 30))
```



در نمودار فوق هر چقدر از رنگ آبی به سمت رنگ قرمز مایل می‌شود، شاهد افزایش همبستگی در جهت مثبت بین متغیرها می‌شویم به بیانی دیگر می‌توان گفت دارای رابطه مستقیم می‌باشند و بالعکس رنگ آبی موجود در ماتریس حاصل دلالت بر همبستگی در جهت عکس متغیرها دارد. شایان ذکر است هرچه درجه این رنگ‌ها بیشتر باشد، نشان همبستگی شدید بین متغیرها می‌باشد.

شایان ذکر است هرچه درجه این رنگ‌ها بیشتر باشد، نشان همبستگی شدید بین متغیرها می‌باشد.

```
install.packages("GGally")
install.packages("ggstats")
install.packages("hms")
library(GGally)
library(ggstats)
library(hms)
Heart$DEATH_EVENT <- factor(Heart$DEATH_EVENT)
Heart %>%
  ggpairs(columns = c("age", "creatinine_phosphokinase", "ejection_fraction", "platelets",
"serum_creatinine", "serum_sodium"),
          mapping = ggplot2::aes(color = DEATH_EVENT)) +
  ggplot2::theme_light()
}
```

می‌توان این نتیجه را از طریق رسم نمودارهای زیر نیز به‌دست آورد و پراکنش هر متغیر را مشاهده نمود.

در نمودار فوق، روی قطر اصلی نمودارهای توابع چگالی هر متغیر را شاهد هستیم.در درایه‌هایی که اندیس سطح از اندیس ستون کمتر است، همبستگی دو متغیر را نظاره‌گر هستیم.

در این مرحله به یکی از ارکان‌های مهم در علم آمار تحت عنوان مصورسازی می‌پردازیم.

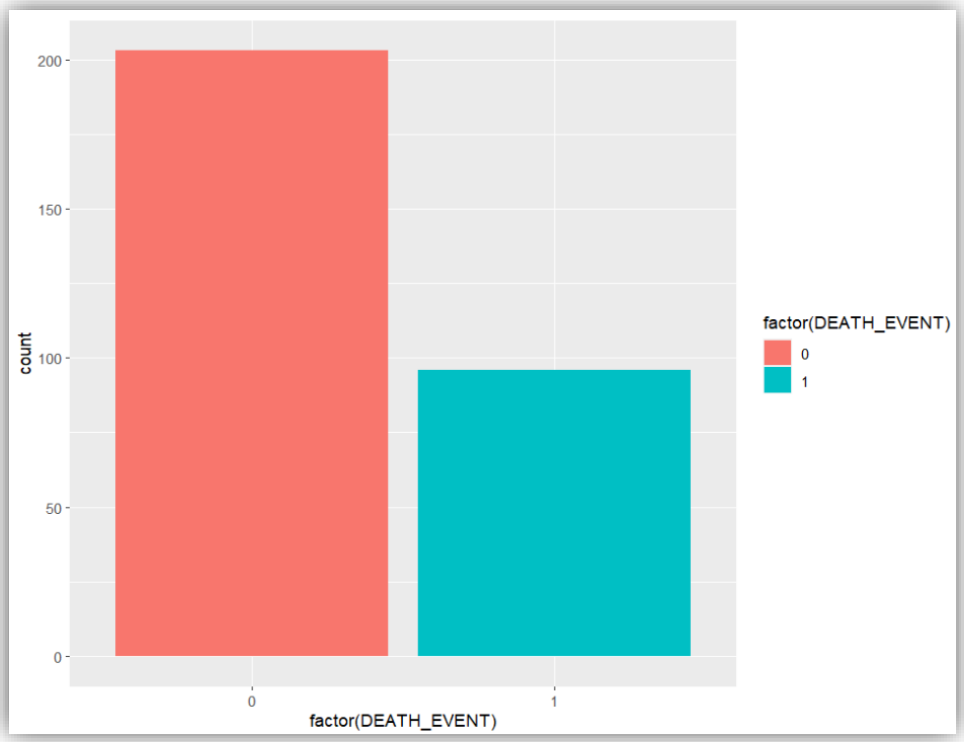
ارائه یک روش موثر برای مصورسازی داده، به ما در شناخت و کشف روابط بین پدیده‌ها و داده‌های حاصل از اندازه‌گیری آن‌ها، کمک می‌کند. به این ترتیب مجموعه داده‌های پیچیده، به شکلی ساده و موثر، نمایش داده می‌شوند و امر استخراج قواعد یا تحلیل آن‌ها آسان می‌گردد. در قرن حاضر، با توجه به حجم بسیار زیاد اطلاعات و لزوم به شناخت رفتار آن‌ها در زمان کوتاه، مصورسازی داده به یک حوزه فعال تحقیق، تدریس و توسعه تبدیل شده است، بطوری که این تکنیک، تجسم علمی و اطلاعاتی را متحد کرده است.

مصور سازی داده‌ها اساساً داده‌های تجزیه و تحلیل شده را در قالب تصاویری یعنی نمودارها و تصاویر قرار می‌دهد. این مصورسازی ها باعث می‌شود تا برای ما درک روند تجزیه و تحلیل از طریق تصاویر آسان شود. از نمودارهای میله‌ای و میله‌ای تجمعی و نمودارهای جعبه‌ای در این بخش برای مصورسازی داده‌ها بهره می‌گیریم.

در ابتدا فراوانی مشاهدات در دو سطح متغیر پاسخ را بررسی می‌کنیم که دریابیم آیا تعادلی بین دو کلاس موجود است یا خیر.

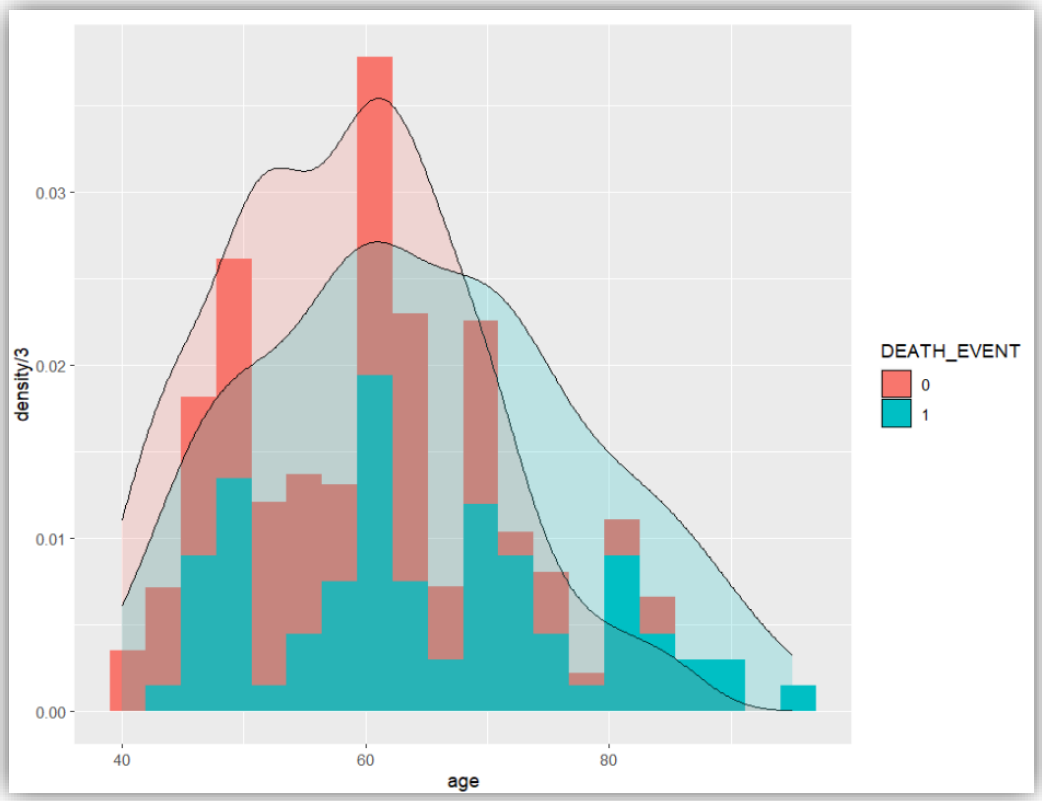
```
ggplot(Heart, aes(factor(DEATH_EVENT), fill = factor(DEATH_EVENT))) + geom_bar()
```

از نمودار پایین مشخص است که فراوانی طبقات با یکدیگر متفاوت است به‌طوری که فراوانی طبقه مربوط به سطح صفر متغیر پاسخ چند برابر فراوانی طبقه دیگر متغیر پاسخ می‌باشد.



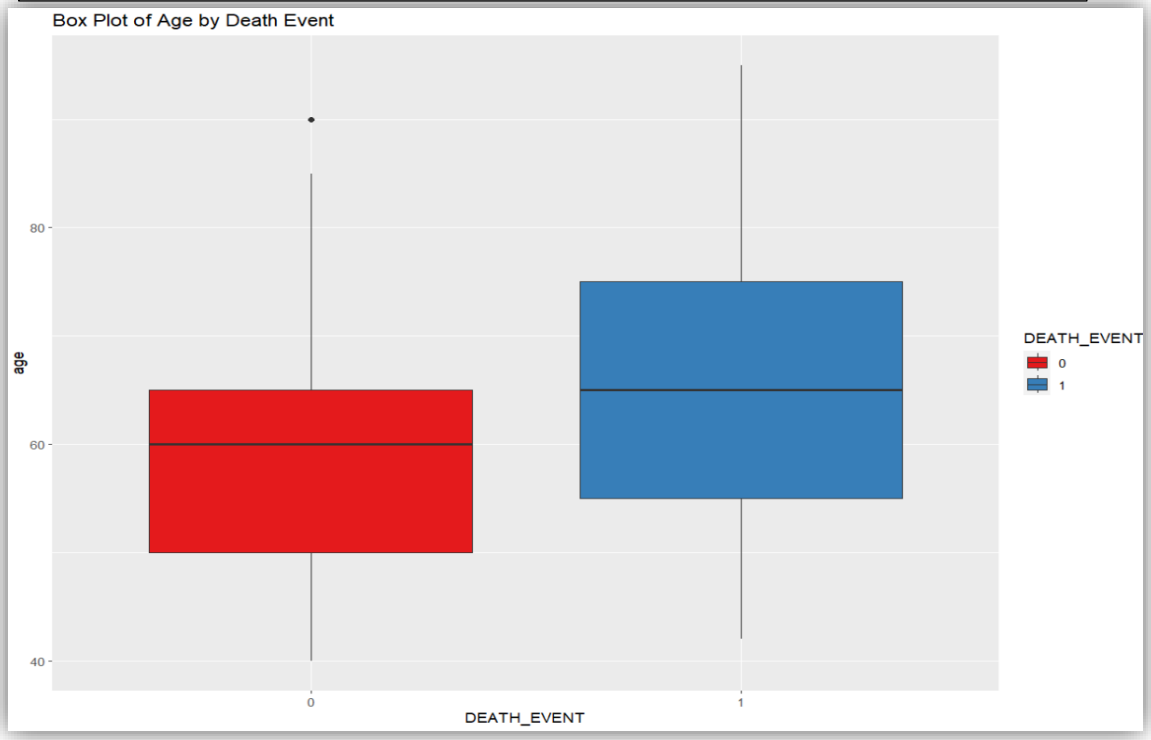
انتظار می‌رود سن بالاتر با مرگ و میر بیشتر همراه باشد. این فرض را می‌توان در هیستوگرام زیر با منحنی‌های چگالی همپوشانی تایید کرد یعنی بیماران مسن‌تر بیشتر از افراد جوان فوت می‌کردند.

```
Heart$DEATH_EVENT <- factor(Heart$DEATH_EVENT)
set.seed(2)
train_index <- createDataPartition(Heart$DEATH_EVENT, p = 0.8, list = FALSE)
train_not_sc <- train <- Heart[train_index,]
train_not_sc%<%

  ggplot(aes(age, fill = DEATH_EVENT))+

  theme_gray+ ()

  geom_histogram(aes(y = ..density../3), bins = 20)+

  geom_density(alpha = 0.2)
```



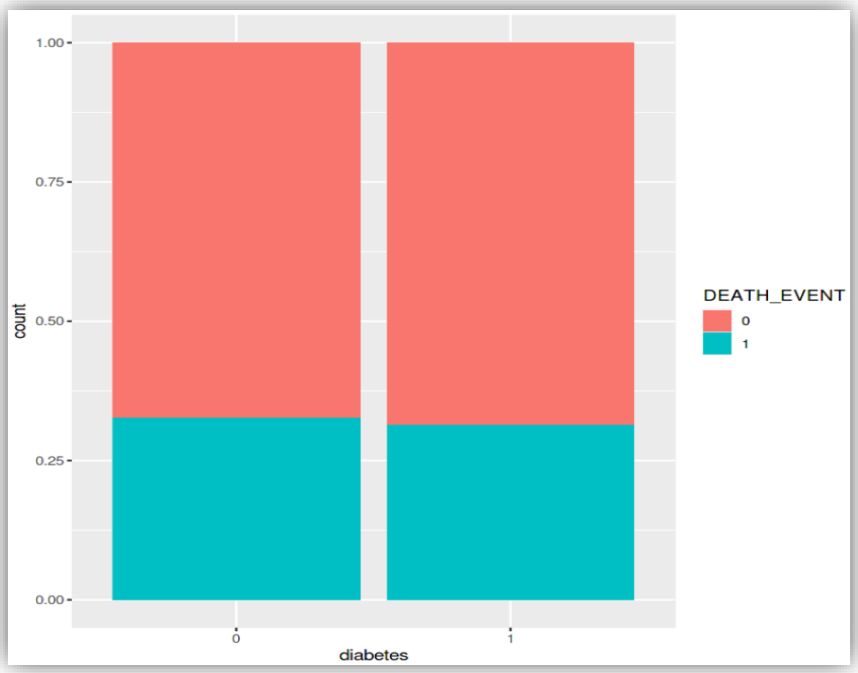
نمودارهای جعبه زیر بیشتر از این فرض حمایت می‌کنند.

```
ggplot(Heart, aes(x = DEATH_EVENT, y = age, fill = DEATH_EVENT)) +
  geom_boxplot() +
  labs(title = "Box Plot of Age by Death Event") +
  scale_fill_brewer(palette = "Set1")
```



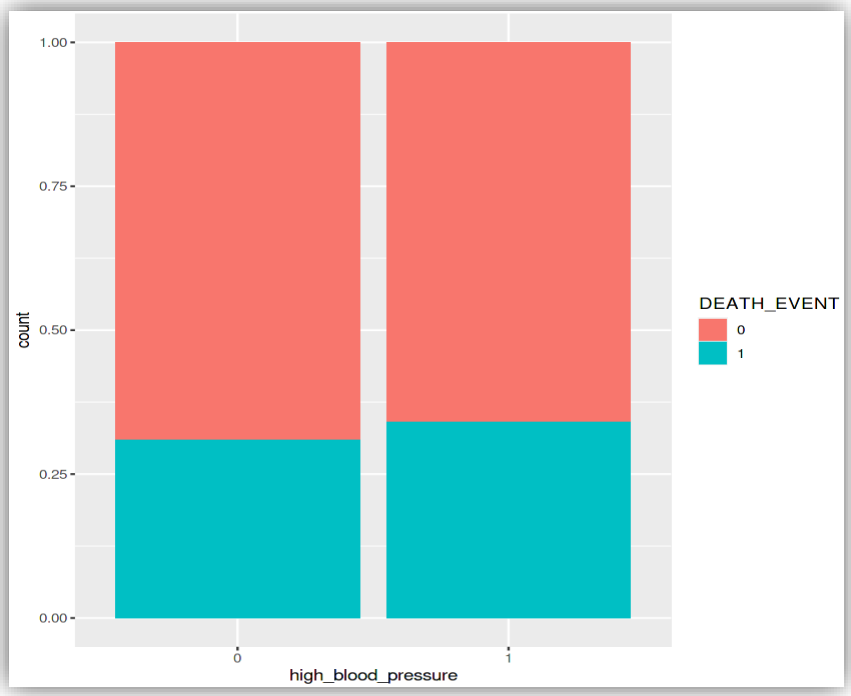
به نظر می‌رسد بیمارانی که می‌میرند بیشتر از بیمارانی که از بیمارستان مرخص می‌شوند، کم‌خونی دارند، همانطور که بر اساس نقشه حرارتی همبستگی انتظار می‌رود، تفاوت نسبتاً کوچک است.

```
train_not_sc %>%
  ggplot(aes(diabetes, fill = DEATH_EVENT)) +
  theme_gray() +
  geom_bar(position = "fill")
```

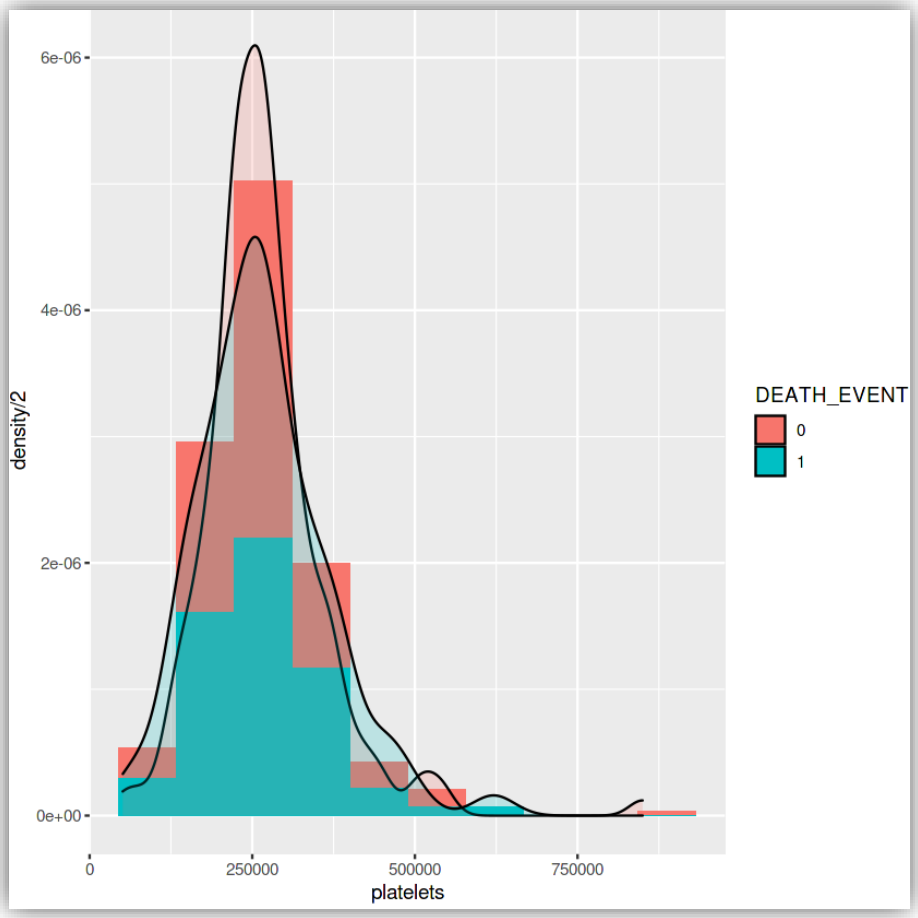


به نظر می‌رسد بیمارانی که می‌میرند بیشتر از بیمارانی که از بیمارستان مرخص می‌شوند فشار خون بالایی دارند.

```
train_not_sc %>%
  ggplot(aes(high_blood_pressure, fill = DEATH_EVENT)) +
  theme_gray() +
  geom_bar(position = "fill")
```

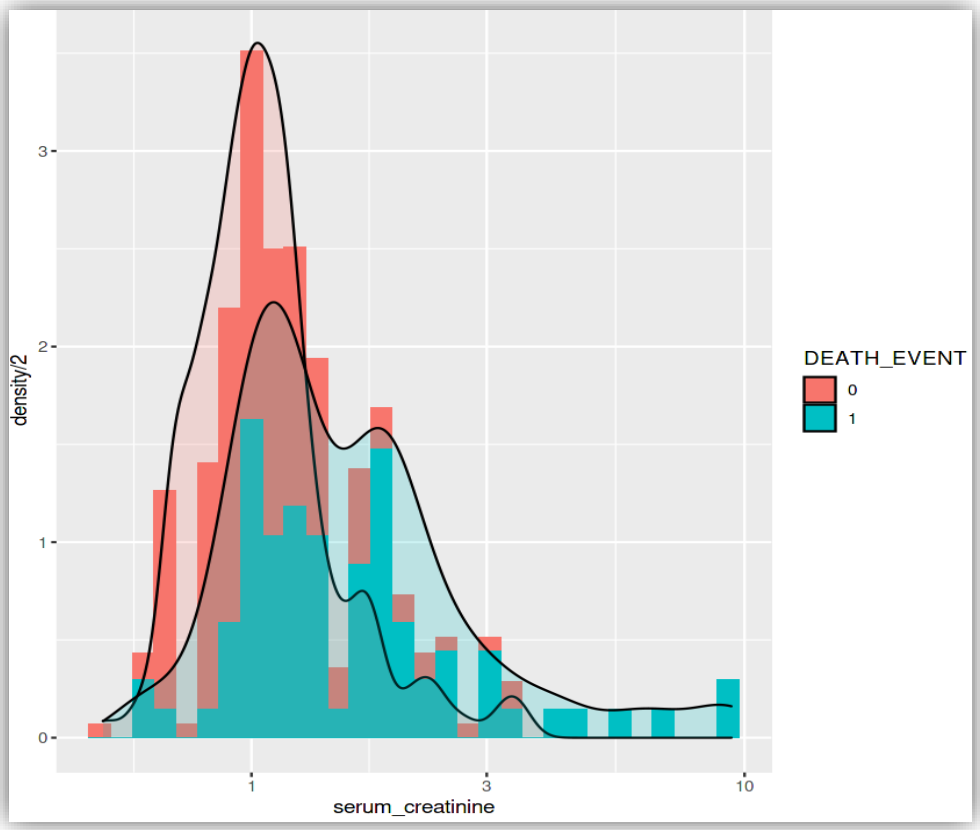


هیچ تفاوتی در سطوح پلاکتی برای هر دو گروه بیمار وجود ندارد، همانطور که در هیستوگرام زیر با منحنی‌های چگالی پوشانده نشان داده شده است.



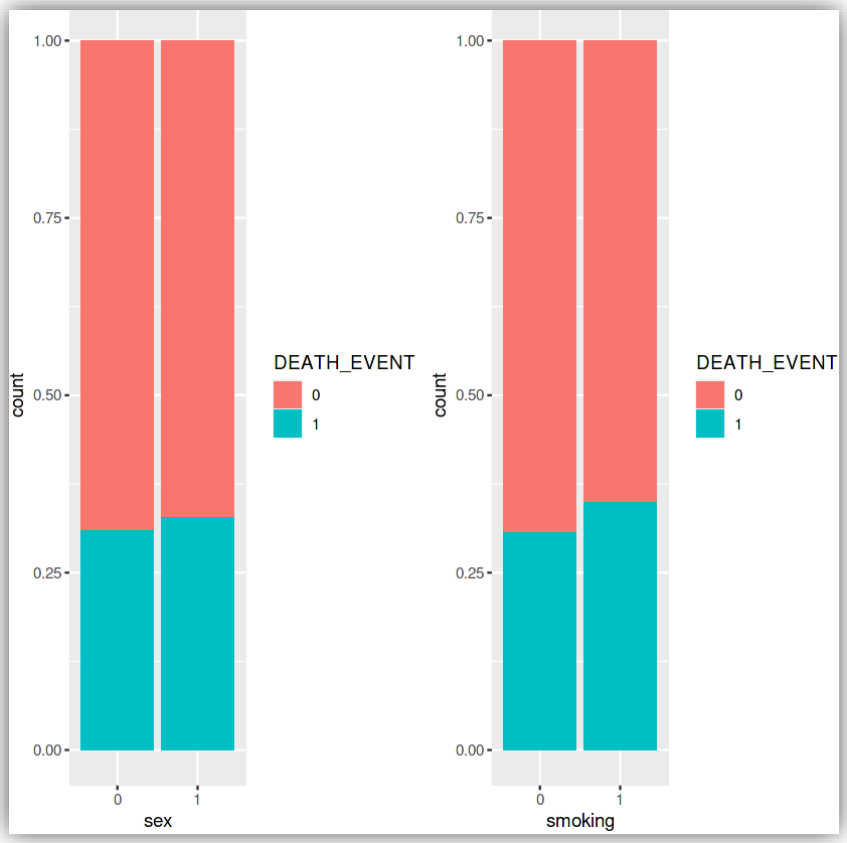
برای کراتینین سرم و سدیم سرم همبستگی قابل توجهی در نقشه حرارتی نشان داده شد. هیستوگرام‌های زیر با منحنی‌های چگالی پوشانده شده تایید می کنند که سطوح بالاتر کراتینین در سرم و سطوح پایین‌تر سدیم در سرم با موارد بیشتر مرگ و میر بیماران مطابقت دارد.

```
train_not_sc %>%
  ggplot(aes(serum_creatinine, fill = DEATH_EVENT)) +
  theme_gray() +
  geom_histogram(aes(y = ..density../2)) +
  geom_density(alpha = 0.2) +
  scale_x_log10()
```



جنسیت و سیگار کشیدن با مرگ و میر بیماران ارتباط کمی دارد. همانطور که مشاهده می شود، تفاوت معنی‌داری بین مرد و زن برای مرگ ناشی از نارسایی قلبی وجود ندارد، در حالی که در بین بیماران سیگاری موارد مرگ و میر بیشتر از بیماران غیر سیگاری بود که به طور کلی می‌توان انتظار داشت.

```
library(gridExtra)
p1 <- train_not_sc %>%
  ggplot(aes(sex, fill = DEATH_EVENT)) +
  theme_gray() +
  geom_bar(position = "fill")
p2 <- train_not_sc %>%
  ggplot(aes(smoking, fill = DEATH_EVENT)) +
  theme_gray() +
  geom_bar(position = "fill")
grid.arrange(p1, p2, nrow = 1)
```



مراحل پیاده سازی الگوریتم‌های طبقه‌بندی دارای شش بخش اصلی به صورت زیر است:

1. حجم نمونه مورد نیاز را به وسیله روابط و فرمول‌هایی که در ادامه توضیح داده خواهد شد؛ بدست می‌آوریم.
2. با استفاده از حجم نمونه تعیین شده داده‌ها را به دو دسته آزمایشی و آموزشی تقسیم می‌کنیم.
3. مدل پیشنهادی را روی داده‌های آزمایشی برازش داده و از طریق آن فرضیه‌ای برای روابط موجود بین متغیرهای ورودی و خروجی تعیین می‌شود.
4. بعد از تشخیص روابط موجود در صورتی که برخی متغیرهای ورودی در ایجاد روابط و رساندن ما به هدف مورد نظر بی تاثیر شناخته شوند، برای افزایش دقت و کارایی مدل پیشنهادی ، از مدل حذف شده و مدل اصلاح شده مجددا روی داده‌های آموزشی برازش داده می‌شود.
5. از طریق مدل اصلاح شده در گام چهار، پیش‌بینی متغیر پاسخ برای داده‌های آموزشی انجام می‌شود.

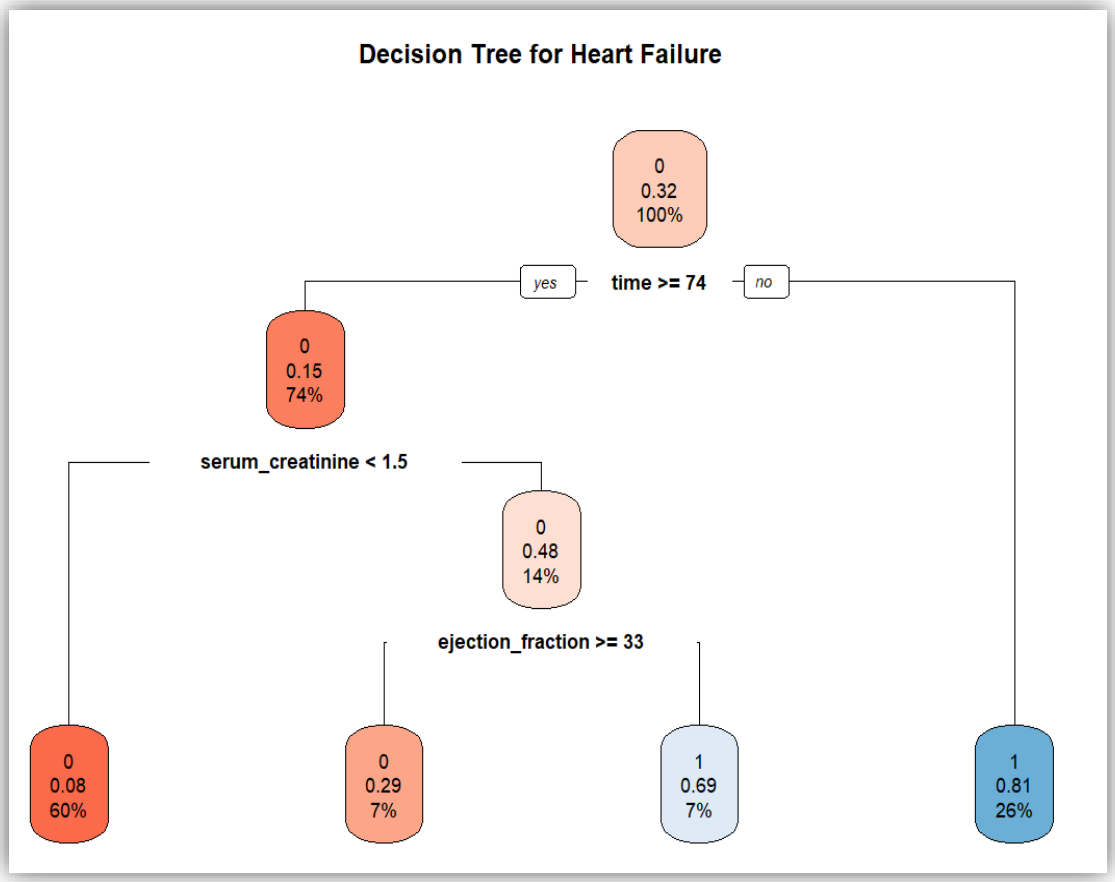
برای ارزیابی کارایی و دقت مدل پیشنهادی و پی بردن به این موضوع که آیا این مدل متناسب با داده‌های مورد مطالعه هست یا خیر مقادیر پیش بینی شده در گام پنج با مقادیر اصلی متغیر پاسخ مقایسه شده و دقت مدل محاسبه می‌شود. از طریق محاسبه این دقت و عدد حالت محاسبه شده برای مدل می‌توان تشخیص داد پیش‌بینی متغیر پاسخ برای ورودی‌های جدید با چه دقتی انجام می‌شود و این مدل چه میزان توانایی در تشخیص مقادیر صحیح متغیر پاسخ را دارا می‌باشد.

داده‌های آموزشی و آزمایشی را ایجاد می‌کنیم (داده‌هایی که برای ساخت مدل مورد استفاده قرار خواهد گرفت و سپس داده‌هایی که برای آزمون مدل استفاده می‌شوند.)

```
set.seed(123)
train_index <- createDataPartition(Heart$DEATH_EVENT, p = 0.8, list = FALSE)
train_data <- Heart[train_index, ]
test_data <- Heart[-train_index, ]
```

درخت تصمیم را با استفاده از دستور زیر تشکیل می‌دهیم:

```
tree_model <- rpart(DEATH_EVENT ~ ., data = train_data, method = "class")
rpart.plot(tree_model, main = "Decision Tree for Heart Failure", box.palette =
"RdBu")
```



باتوجه به درخت رسم شده و پیش‌رفتن از ریشه تا رسیدن به گره‌های برگ درمی‌یابیم که متغیر مدت

زمانی که بیمار تحت درمان بوده باتوجه به این که در ریشه قرار گرفته است؛ از اطلاعات بیشتری درباره‌ی متغیر پاسخ ما (فوت) برخوردار است و بی‌نظمی کمتری دارد.

در گره تصمیم اول اگر زمان بستری بیماران از 74 روز بیشتر باشد؛ سطح سدیم خون بیمار نیز از 1.5 کمتر باشد به احتمال 8 درصد به فوت دچار می‌شوند.

در این مرحله متغیر هدف را با استفاده از درخت تصمیم پیش‌بینی می‌کنیم. برای این منظور ابتدا ماتریس درهم‌ریختگی را تشکیل می‌دهیم و دقت مدل را محاسبه می‌کنیم:

```
pred <- predict(tree_model, newdata = test_data, type = "class")
confusionMatrix(pred, test_data$DEATH_EVENT)
```

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 36  3
         1  4 16

               Accuracy : 0.8814
                 95% CI : (0.7707, 0.9509)
    No Information Rate : 0.678
    P-Value [Acc > NIR] : 0.0002779

                  Kappa : 0.732

 Mcnemar's Test P-Value : 1.0000000

            Sensitivity : 0.9000
            Specificity : 0.8421
         Pos Pred Value : 0.9231
         Neg Pred Value : 0.8000
             Prevalence : 0.6780
         Detection Rate : 0.6102
   Detection Prevalence : 0.6610
      Balanced Accuracy : 0.8711

       'Positive' Class : 0
```

بنابراین در روش درخت تصمیم، از 59 داده آزمایشی، برای 36 تا از داده‌ها، مقدار متغیر پاسخ (یعنی DEATH_EVENT) واقعا صفر است و روش درخت تصمیم نیز این 36 داده را به درستی به رده صفر طبقه‌بندی کرده است. همچنین برای 16 تا از داده‌ها، مقدار متغیر پاسخ واقعا یک است و روش درخت تصمیم نیز این 16 داده را به درستی به رده یک طبقه‌بندی کرده است. اما برای 3 تا از داده‌ها، مقدار متغیر پاسخ واقعا یک است اما روش درخت تصمیم این 3 داده را به غلط به رده صفر طبقه‌بندی کرده است. در نهایت برای 4 تا از داده‌ها، مقدار متغیر پاسخ واقعا صفر است اما روش درخت تصمیم این 4 مشاهده را به غلط به رده یک طبقه‌بندی کرده است.

که با توجه به آن مقادیر TP , TN , FP, FN به ترتیب: 36، 16، 3 و4 هستند.

دقت مدل از رابطه زیر نیز قابل محاسبه است :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{36 + 16}{36 + 16 + 3 + 4} = 0.88$$

حساسیت نرخ مثبت را نمایش می‌دهد برای این روش داریم:

$$Sensitivity(TPR) = \frac{TP}{TP + FN} = \frac{36}{36 + 4} = 0.9$$

خصوصیت (ویژگی) نرخ منفی را نشان می‌دهد؛ برای این روش داریم:

$$Specificity(TNR) = \frac{TN}{TN + FP} = \frac{16}{16 + 3} = 0.842$$

معیار صحت به دست آمده برای این روش:

$$precision = \frac{TP}{TP+FP} = \frac{36}{36+3} = 0.92$$

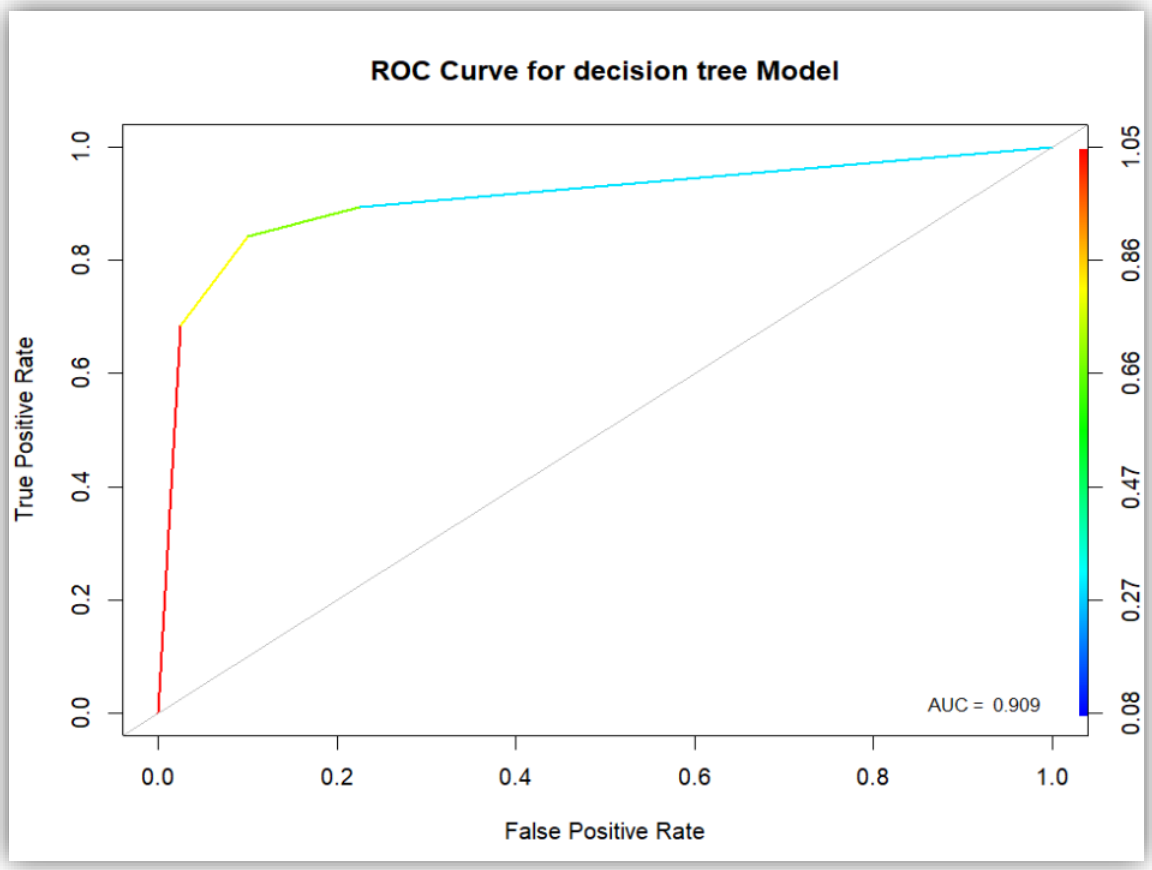
معیار امتیاز اف برای این کلاس عبارت است از :

$$F_1 = \frac{2TP}{2TP+FP+FN} = 0.91$$

حال به رسم نمودار راک می‌پردازیم.

دو پارامتر (حساسیت و خاصیت) نیز مشابه معیار صحت، معمولا به صورت درصد بیان می‌شوند. واضح است که پیش‌بینی عالی، پیش‌بینی است که مقادیر نرخ پاسخ‌های منفی درست و نرخ پاسخ‌های مثبت درست مربوط به آن، هر دو صد درصد باشند؛ اما احتمال وقوع این اتفاق در واقعیت بسیار کم است و همیشه یک حداقل خطایی وجود دارد. پارامترهای حساسیت و خاصیت، بنابر ماهیتی که دارند همواره در رقابت با یکدیگر هستند. یعنی افزایش یکی با کاهش دیگری همراه است و برعکس. به عنوان مثال در مسئله بیان شده، با تغییر حد آستانه در دسته‌بندی، تغییرات این دو معیار را با هم می‌سنجیم.برای این‌که بهتر بتوانیم از این نمودار استفاده کنیم و مقادیر هر دو محور با هم رشد یا کاهش پیدا کنند به جای معیارحساسیت از حساسیت − 1 استفاده می‌کنیم، به این ترتیب، نموداری حاصل می‌شود که به آن نمودار ROC می‌گوییم. هرچه قدر سطح زیر نمودار بیشتر باشد، مدل ما بهتر می‌باشد.

```
acc_glm <- confusionMatrix(pred,
as.factor(test_data$DEATH_EVENT))$overall["Accuracy"]
tpr_glm <- confusionMatrix(pred,
as.factor(test_data$DEATH_EVENT))$byClass["Specificity"]
pred <- predict(tree_model, newdata = test_data, type = "prob")
pred_obj <- prediction(pred[, 2], test_data$DEATH_EVENT)
perf_obj <- ROCR::performance(pred_obj, measure = "tpr", x.measure = "fpr")
plot(perf_obj, main = "ROC Curve for decision tree Model ,"

    xlab = "False Positive Rate", ylab = "True Positive Rate", colorize=T, lwd = 2(

abline(a = 0, b = 1, lwd = 1.5, col = "gray")
auc <- ROCR::performance(pred_obj, "auc")
auc <- auc@y.values[[1]]

legend("bottomright", paste("AUC = ", round(auc, 3)), bty = "n", cex = 0.8)
```


ROC Curve for decision tree Model
True Positive Rate
False Positive Rate
AUC = 0.909
0.0
0.2
0.4
0.6
0.8
1.0
1.05
0.86
0.66
0.47
0.27
0.08

جنگل تصادفی

برای ورود به مبحث روش یادگیری جنگل تصادفی، ابتدا محدودیت‌های طبقه‌بندی‌کننده درخت تصمیم را بررسی می‌کنیم. از جمله‌ی این محدودیت‌ها می‌توان به موارد زیر اشاره کرد.

بیش برازش[1]: درخت‌های تصمیم تمایل دارند تا داده‌های آموزشی را بیش از حد برازش دهند، به خصوص زمانی که درخت عمیق و پیچیده می‌شود. بیش برازش زمانی اتفاق می‌افتد که درخت نویز یا الگوهای نامربوط در داده‌های آموزشی را تشخیص می‌دهد، که منجر می‌شود درخت توانایی رویارویی با داده‌های دیده نشده را نداشته باشد. به زبان ساده‌تر، درخت بیش از حد خوب به داده‌های آموزشی برازش داده می‌شود.

ناپایداری[2]: درخت‌های تصمیم نسبت به تغییرات کوچک در داده‌های آموزشی حساس هستند. به عبارتی با تغییری کوچک در داده‌های آموزشی، درخت‌هایی با ساختارهای متفاوت و در نتیجه با پیش‌گویی‌های متفاوت نتیجه می‌شود. این ناپایداری را می‌توان با استفاده از روش‌های گروهی مانند جنگل‌های تصادفی یا روش تقویتی کاهش داد.

ناهمواری[3]: درخت‌های تصمیم، پیش‌گویی‌های ثابت تکه‌ای تولید می‌کنند. به عبارت صریح‌تر، بین منطقه‌ها با پیش‌گویی‌های متفاوت، مرز ایجاد می‌کنند و باعث می‌شوند مرزهایی بین مناطق با پیش‌بینی‌های مختلف ایجاد شود. بنابراین برای مسائلی که نیاز به یک مرز هموار برای تصمیم‌گیری است، مناسب نیستند.

درخت‌های تصمیم، بلوک‌های سازنده‌ی یک جنگل تصادفی هستند. همانطور که از نام آن مشهود است، این الگوریتم، جنگلی را به طور تصادفی می‌سازد. جنگل ساخته شده، در واقع گروهی از درخت‌های تصمیم است. کار ساخت جنگل با استفاده از درخت‌ها، اغلب اوقات به روش کیسه‌گذاری[4] یا بسته‌بندی

[1] Overfitting
[2] Instability
[3] Lack of smoothness
[4]

انجام می‌شود. ایده اصلی روش کیسه‌گذاری این است که در حالت کلی، ترکیبی از مدل‌های یادگیری، می‌تواند منجر به نتیجه‌ای بهتر از حالت‌های تکی شود. به بیان ساده، جنگل تصادفی چندین درخت تصمیم ساخته و آنها را با یکدیگر ادغام می‌کند تا پیشگویی‌های صحیح‌تر و پایدارتری حاصل شود.

با توجه به توضیحات بالا، جنگل تصادفی یک روش طبقه‌بندی محبوب است که به صورت ترکیبی از درخت‌های تصمیم چندگانه بوده که در آن هر درخت از طریق مکانیزم رای‌گیری یا میانگین‌گیری در پیش‌گویی نهایی نقش دارد.

جنگل تصادفی با انتخاب تصادفی زیر مجموعه‌ای از داده‌های آموزشی و ویژگیها (متغیرهای مستقل)، مجموعه‌ای از درخت‌های تصمیم را می‌سازد. انتخاب تصادفی داده‌ها با استفاده از روش بوت‌استرپ انجام می‌شود و هر درخت بر روی یک زیرمجموعه متفاوت از داده‌های آموزشی برازش داده می‌شود. با این کار تنوع زیاد می‌شود و خطر بیش برازش کاهش پیدا می‌کند. در هر گره‌ی درخت تصمیم، یک زیرمجموعه تصادفی از ویژگی‌ها برای تعیین بهترین تقسیم در نظر گرفته می‌شود. این تصادفی بودن تضمین می‌کند که هر درخت ساختار منحصر به خود را داشته باشد و از ویژگی یا متغیر خاصی حمایت نکند. با ترکیب پیشگویی همه درخت‌ها، جنگل تصادفی به نتایج قوی‌تر و دقیق‌تری دست می‌یابد.

در طول مرحله پیش‌گویی، هر درخت در جنگل تصادفی به طور مستقل نقطه داده ورودی را طبقه‌بندی می‌کند. در مورد طبقه‌بندی، طبقه‌ای که اکثریت آرا را از درختان کسب کند به عنوان پیشگویی نهایی انتخاب می‌شود. برای رگرسیون، میانگین پیش‌گویی‌های انجام شده توسط همه درخت‌ها در نظر گرفته می شود. به این روش بگینگ[1] یا بسته‌بندی می‌گویند.

از مزایای جنگل تصادفی می‌توان به موارد زیر اشاره کرد:

استواری[2]: جنگل تصادفی نسبت به درخت‌های تصمیم تکی، کمتر موجب بیش برازش می‌شود. ترکیب

[1] Bagging
[2] Robustness

چندین درخت تأثیر ویژگی‌های نویزدار و نامناسب را کاهش می‌دهد و منجر به بهبود عملکرد و فهم بهتر داده‌ها می‌شود.

اهمیت ویژگی[1]: جنگل تصادفی معیاری برای تشخیص ویژگی‌های مهم ارائه می‌دهد که سهم نسبی هر ویژگی را در طبقه‌بندی نشان می‌دهد. این اطلاعات از رفتار جمعی درخت‌ها مشتق می‌شود و امکان درک و تفسیر بهتر داده‌ها را فراهم می‌کند.

مدیریت داده های گمشده[2]: جنگل تصادفی می‌تواند داده‌های گمشده را با استفاده از تقسیم‌های جایگزین[3] کنترل کند. به عبارتی اگر برخی ویژگی‌ها داری مقادیر گمشده باشند، جنگل تصادفی می‌تواند از ویژگی‌های موجود برای پیشگویی دقیق استفاده کند.

غیرخطی بودن[4]: جنگل تصادفی می‌تواند روابط پیچیده غیرخطی را در داده‌ها کشف کند. با ترکیب چندین درخت تصمیم، می‌تواند مرزهای تصمیم‌گیری پیچیده و اثرات متقابل بین ویژگی‌ها را مدل کند.

از محدودیت‌های جنگل تصادفی نیز می‌توان به موارد زیر اشاره کرد:

تفسیرپذیری[5]: اگرچه جنگل تصادفی اطلاعاتی در مورد اهمیت ویژگی‌ها ارائه دهد، اما در عین حال مدل نهایی به راحتی قابل تفسیر نیست. به دلیل ماهیت الگوریتم، درک منطق دقیق و استدلالی که در پشت پیش‌گویی‌های فردی قرار دارد چالش برانگیز است.

از نظر محاسباتی گران است: جنگل تصادفی می‌تواند از نظر محاسباتی گران باشد، به خصوص زمانی که با تعداد زیادی درخت یا داده‌ها با ابعاد بالا سروکار داشته باشیم، ارزیابی یک مجموعه بزرگ از درخت‌ها ممکن است به زمان و منابع محاسباتی بیشتری نیاز داشته باشد.

اریبی در مجموعه داده‌های نامتعادل: جنگل تصادفی می‌تواند نسبت به طبقه‌ی اکثریت در مجموعه

[1] Feature importance
[2] Handing of missing data
[3] Surrogate split
[4] Non-linearity
[5] Interpretability

داده‌های نامتعادل نتایجی اریب ارائه دهد. از آنجایی که هر درخت به طور مستقل آموزش داده می‌شود، طبقه اکثریت (در مسائل طبقه‌بندی) تأثیر قوی‌تری بر پیش‌گویی‌های نهایی دارد. ممکن است برای مدیریت موثر داده‌های نامتعادل، تکنیک‌های تعادل یا اصلاحات تخصصی مورد نیاز باشد.

مثال کاربردی از الگوریتم جنگل تصادفی در نرم افزار R

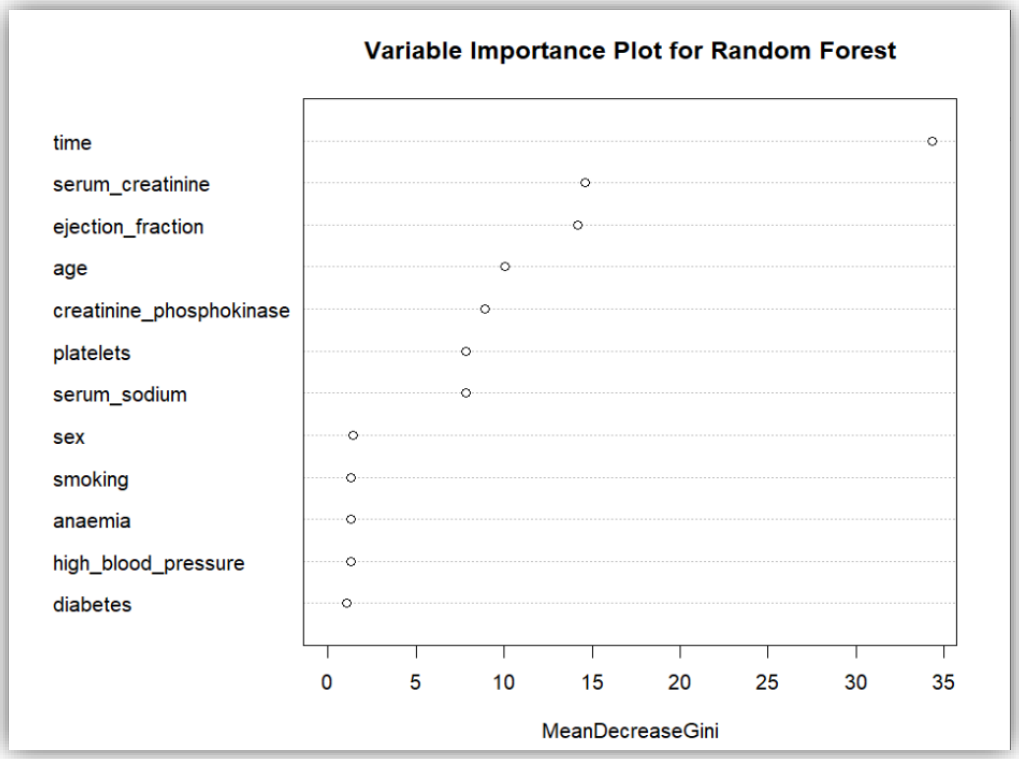
با استفاده از کدهای زیر می‌توان یک جنگل تصادفی به داده‌ها برازش داد.

```
#random forest
rf_model <- randomForest(DEATH_EVENT ~ ., data = train_data, ntree = 500, mtry = 3)
pred <- predict(rf_model, newdata = test_data)
confusionMatrix(pred, test_data$DEATH_EVENT)
```

خروجی به صورت زیر است:

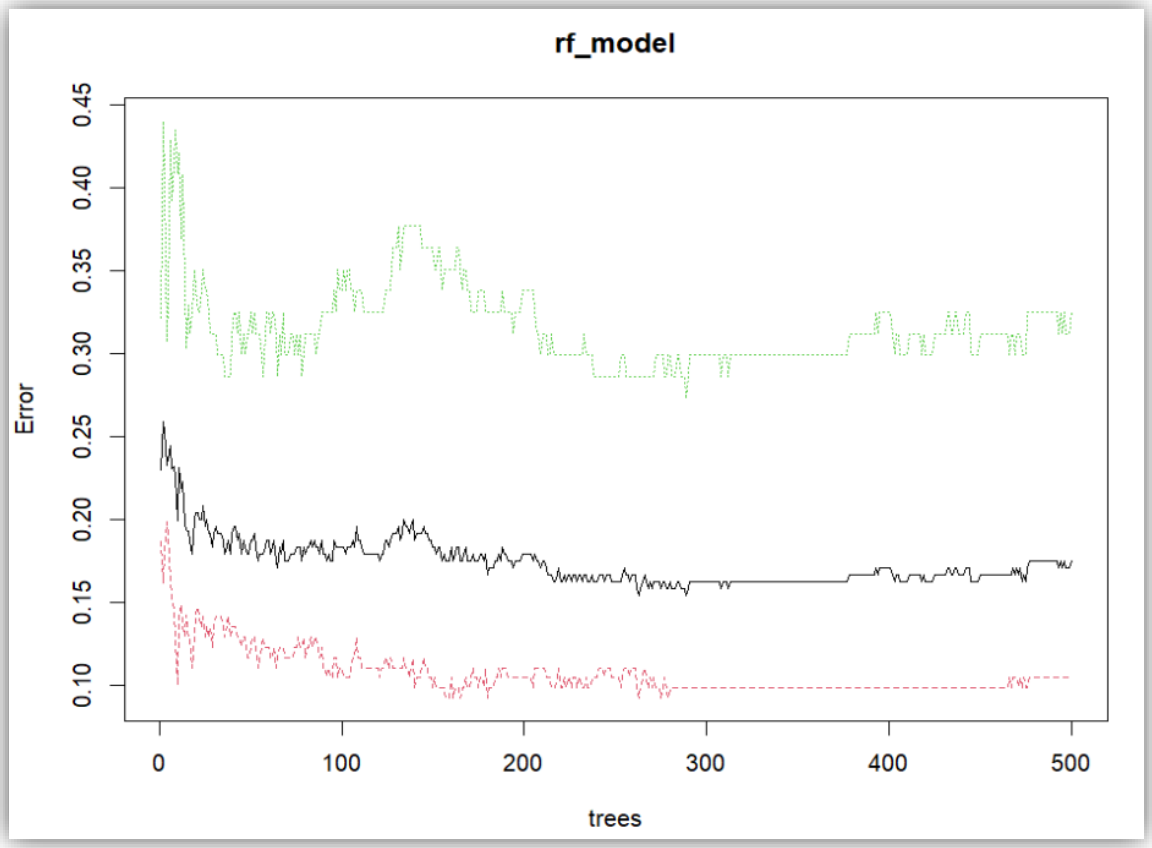
```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 37  4
         1  3 15

               Accuracy : 0.8814
                 95% CI : (0.7707, 0.9509)
    No Information Rate : 0.678
    P-Value [Acc > NIR] : 0.0002779

                  Kappa : 0.7245

 Mcnemar's Test P-Value : 1.0000000

            Sensitivity : 0.9250
            Specificity : 0.7895
         Pos Pred Value : 0.9024
         Neg Pred Value : 0.8333
             Prevalence : 0.6780
         Detection Rate : 0.6271
   Detection Prevalence : 0.6949
      Balanced Accuracy : 0.8572

       'Positive' Class : 0
```

```
acc_rf <- confusionMatrix(pred,
as.factor(test_data$DEATH_EVENT))$overall["Accuracy"]
tpr_rf <- confusionMatrix(pred,
as.factor(test_data$DEATH_EVENT))$byClass["Specificity"]
varImpPlot(rf_model, type = 2, main = "Variable Importance Plot for Random
Forest")
```



با رسم نمودار اولویت و اهمیت متغیرها متوجه شدیم برای بهبود دقت جنگل تصادفی باید با متغیرهای time، serum، ejection، age و creatinine کار را ادامه بدهیم.

```
plot(rf_model)
rf_model <- randomForest(DEATH_EVENT ~ time +
                         ejection_fraction + serum_creatinine + age
+creatinine_phosphokinase,data = train_data ,ntree = 300 ,mtry = 2)
```




```
pred <- predict(rf_model, newdata = test_data)
confusionMatrix(pred, test_data$DEATH_EVENT)
```

```
          Reference
Prediction  0  1
         0 38  4
         1  2 15

               Accuracy : 0.8983
                 95% CI : (0.7917, 0.9618)
    No Information Rate : 0.678
    P-Value [Acc > NIR] : 7.32e-05

                  Kappa : 0.7605

 Mcnemar's Test P-Value : 0.6831
```

دقت مدل از رابطه زیر نیز قابل محاسبه است :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{38 + 15}{38 + 15 + 4 + 2} = 0.898$$

حساسیت نرخ مثبت را نمایش می‌دهد برای این روش داریم:

$$Sensitivity(TPR) = \frac{TP}{TP + FN} = \frac{38}{38 + 2} = 0.95$$

خصوصیت (ویژگی) نرخ منفی را نشان می‌دهد؛ برای این روش داریم:

$$Specificity(TNR) = \frac{TN}{TN + FP} = \frac{15}{15 + 4} = 0.79$$

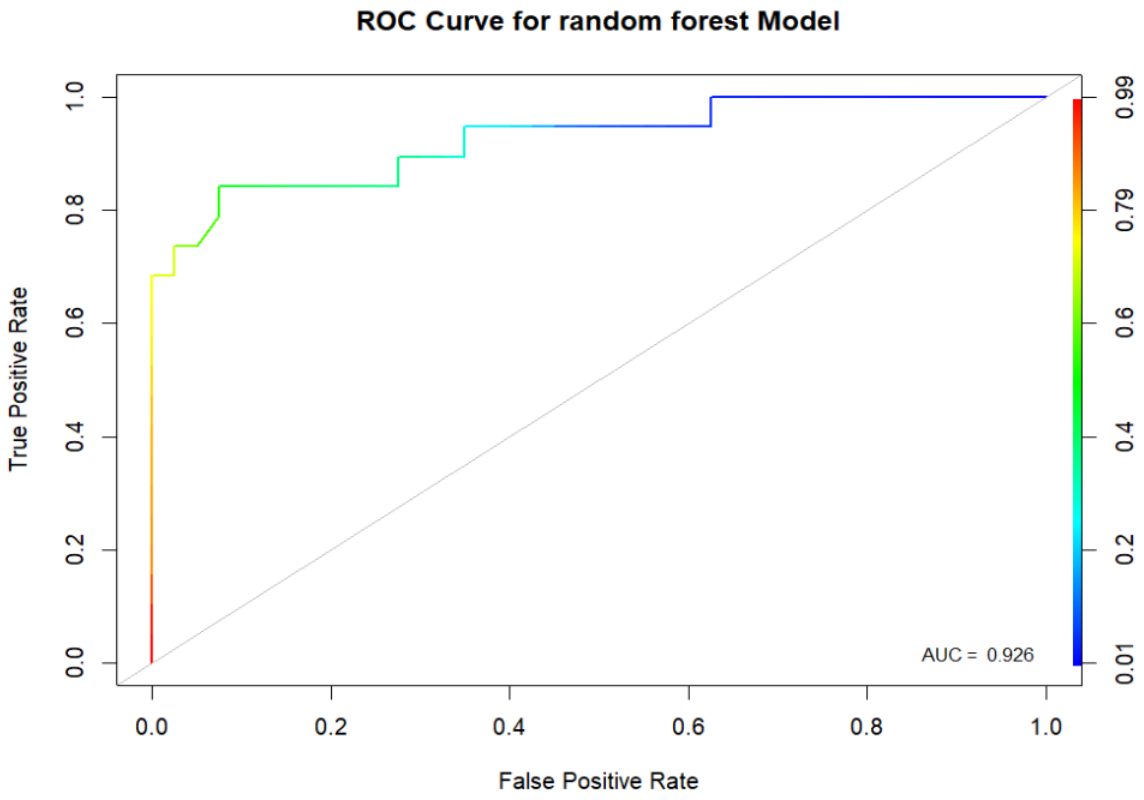
معیار صحت به دست آمده برای این روش:

$$precision = \frac{TP}{TP+FP} = \frac{38}{38+4} = 0.9$$

معیار امتیاز اف برای این کلاس عبارت است از :

$$F_1 = \frac{2TP}{2TP+FP+FN} = 0.93$$

```
acc_glm <- confusionMatrix(pred,
as.factor(test_data$DEATH_EVENT))$overall["Accuracy"]
tpr_glm <- confusionMatrix(pred,
as.factor(test_data$DEATH_EVENT))$byClass["Specificity"]
pred <- predict(rf_model, newdata = test_data, type = "prob")
pred_obj <- prediction(pred[, 2], test_data$DEATH_EVENT)
perf_obj <- ROCR::performance(pred_obj, measure = "tpr", x.measure = "fpr")
plot(perf_obj, main = "ROC Curve for random forest Model ,"
    xlab = "False Positive Rate", ylab = "True Positive Rate", colorize=T, lwd = 2(
abline(a = 0, b = 1, lwd = 1.5, col = "gray")
auc <- ROCR::performance(pred_obj, "auc")
auc <- auc@y.values[[1]]
legend("bottomright", paste("AUC = ", round(auc, 3)), bty = "n", cex = 0.8)
```

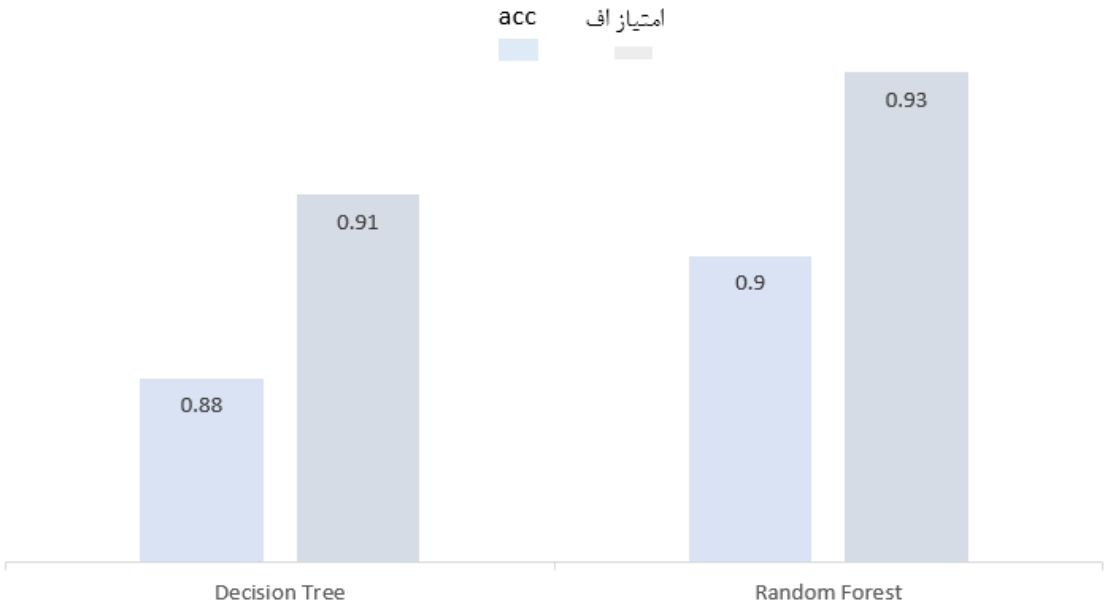


2 بحث و نتیجه‌گیری

در این بخش دو روش درخت تصمیم و جنگل تصادفی که برای طبقه‌بندی داده‌های Heart مورد استفاده قرار گرفت را با هم مقایسه می‌کنیم. معیارهای مختلفی را می‌توان برای انجام مقایسه در نظر گرفت که در اینجا معیاری که برای مقایسه روش‌ها مورد استفاده قرار می‌دهیم، درصد پیش‌گویی صحیح برای پیش‌بینی طبقه مربوط به داده‌های آزمایشی است. در جدول زیر درصد پیش‌بینی درست برای هر دو روش درخت تصمیم و جنگل تصادفی آورده شده است:

روش	درخت تصمیم	جنگل تصادفی
درصد پیش‌بینی درست	%88	%90

با توجه به جدول فوق مشخص می‌شود که روش جنگل تصادفی داری درصد پیش‌بینی درست بالاتری نسبت به درخت تصمیم است. بنابراین برای این مجموعه داده، روش جنگل تصادفی برتر از روش درخت تصمیم است.



در نمودار فوق، بخش سمت راست مربوط به جنگل تصادفی می‌باشد. همانطور که مشاهده می‌کنیم این

روش هم از نظر امتیاز اف هم از نظر دقت پیش‌بینی کمی از درخت تصمیم بالاتر می‌باشد. به همین منظور برای این مجموعه دیتا جنگل تصادفی کاراتر از درخت تصمیم عمل می‌کند و بایستی از این روش برای پیش‌بینی متغیر پاسخ و رده‌بندی میزان فوتی‌های بیماران استفاده نمود.

واژه نامه

درخت تصمیم.....Decision tree

داده آموزشی.....Training Data

طبقه بندی.....Classification

گره.....Node

ریشه.....Root Node

گره داخلی.....Interior Rode

گره والد.....Parent Node

گره فرزند.....Child Node

شاخه.....Branch

داده پرت.....Outlier Data

داده گم شده.....Missing Dara

پیوست

کدهای R

```
library(tidyr)
library(dplyr)
library(caret)
library(rpart)
library(rpart.plot)
library(randomForest)
library(Boruta)
library(ROCR)
library(gbm)
library(corrplot)
library(stringr)
#read data
Heart=read.csv("C:/Users/acer/Desktop/heart.csv")
head(Heart)
str(Heart)
#missing values
colSums(is.na(Heart))
sum(duplicated(Heart))
glimpse(Heart)
summary(Heart)
Heart[,<-]

  ggplot(aes( y = platelets))+

  geom_boxplot()

out=boxplot(Heart$platelets)$out
head(out)
```

```
Heart%<%

  ggplot(aes( y = serum_creatinine))+

  geom_boxplot()

out=boxplot(Heart$serum_creatinine)$out
head(out)
length(out)
#########################

 #Select only numeric variables for correlation

numeric_data <- Heart[, sapply(Heart, is.numeric)]

if (ncol(numeric_data) > 1) }

  cor_matrix <- cor(numeric_data, method = "pearson")
  print("Correlation Matrix:")
  print(cor_matrix)
  library(corrplot)
  corrplot(cor_matrix, method = "circle")
 {else}

  print("Not enough numeric variables for a correlation matrix.")
{

#######################################

#pearson

data <- dplyr::select(Heart, -time)
data_num <- data

factors <- c(12 ,11 ,10 ,6 ,4 ,2)

data[factors] <- lapply(data[factors], factor)
data_sc <- data
data_sc[-factors] <- sapply(data[-factors], scale)
levels(data_sc$DEATH_EVENT) = c("No","Yes")
```

```
set.seed(2)
train_index <- createDataPartition(Heart$DEATH_EVENT, p = 0.8, list = FALSE)
train_not_sc <- train <- data[train_index,]
train <- data_sc[train_index,]
test <- data_sc[-train_index,]
prop.table(table(train$DEATH_EVENT))
prop.table(table(test$DEATH_EVENT))
library(reshape2)
train_num <- data_num[train_index,]
cormat <- round(cor(train_num),2)
melted_cormat <- melt(cormat)
ggplot(data = melted_cormat, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile+ ()
  labs(x = NULL, y = NULL, fill = "Pearson's\nCorrelation")+
  scale_fill_gradient2(mid="#FBFEF9",low="#0C6291",high="#A63446",      limits=c(-
0.5,0.5))+
  scale_x_discrete(guide = guide_axis(angle = 30))
##############################
ggplot(Heart,    aes(factor(DEATH_EVENT),    fill    =    factor(DEATH_EVENT)))    +
geom_bar()
Heart$DEATH_EVENT <- factor(Heart$DEATH_EVENT)
set.seed(2)
train_index <- createDataPartition(Heart$DEATH_EVENT, p = 0.8, list = FALSE)
train_not_sc <- train <- Heart[train_index,]
train_not_sc'/.<'/.
  ggplot(aes(age, fill = DEATH_EVENT))+
  theme_gray+ ()
```

```
  geom_histogram(aes(y = ..density../3), bins = 20)+
  geom_density(alpha = 0.2)



ggplot(Heart, aes(x = DEATH_EVENT, y = age, fill = DEATH_EVENT))+
  geom_boxplot+ ()
  labs(title = "Box Plot of Age by Death Event")+
  scale_fill_brewer(palette = "Set1")

################################3
library(GGally)
library(ggstats)
library(hms)
Heart$DEATH_EVENT <- factor(Heart$DEATH_EVENT)


Heart˙/.<˙/.
  ggpairs(columns = c("age", "creatinine_phosphokinase", "ejection_fraction",
"platelets", "serum_creatinine", "serum_sodium",("
        mapping = ggplot2::aes(color = DEATH_EVENT)+ (
  ggplot2::theme_light()


Heart˙/.<˙/.
  ggplot(aes(x = age)) +
  geom_histogram(binwidth = 5 ,
```

```
                color = "white ,"
                alpha = 0.5+ (
  labs(title = "Age Distribution")+
  scale_x_continuous(breaks = seq(40,100,10))
####
Heart ˙/.<˙/.
  ggplot(aes(x = age, fill = DEATH_EVENT))+
  geom_histogram(binwidth = 5 ,
                position = "identity,"
                alpha = 0.5,color = "white+ ("
  scale_fill_manual(values = c("#999999", "#1F77B4"))+
  labs(title = "Age Distribution with Death Event")+
  scale_x_continuous(breaks = seq(40,100,10))
##############
ggplot(Heart, aes(x = DEATH_EVENT, y = age, fill = DEATH_EVENT))+
  geom_boxplot+ ()
  labs(title = "Box Plot of Age by Death Event")+
  scale_fill_brewer(palette = "Set1")
####################
Heart˙/.<˙/.
  ggplot(aes(x = as.factor(sex), y = age)) + geom_boxplot(aes(fill = as.factor(sex)))+
  labs(title = "Boxplot of Age by Sex") + xlab("Sex") + ylab("Age")
############################
```

```
Heart$agegp <- ifelse( Heart$age<65, "Age <65", "Age >=65")
Heart$agegpn <- ifelse( Heart$age<65, 0, 1)
#
Heart$sexc <-ifelse(Heart$sex==1, "Male", "Female")
Heart$smoke <-ifelse(Heart$smoking==1, "Yes", "No")
Heart$hbp <- ifelse(Heart$high_blood_pressure==1, "Yes","No")
Heart$dia <-ifelse(Heart$diabetes==1, "Yes", "No")
Heart$anaemiac <- ifelse(Heart$anaemia==1 ,"Yes", "No")
###############
 #1age group
p1<-ggplot(Heart, aes(x=agegp))+geom_bar(fill="lightblue")+ labs(x="Age Group")+
theme_minimal(base_size=10)

 #2Sex
p2<-ggplot(Heart, aes(x=sexc))+geom_bar(fill="indianred3")+ labs(x="Sex")+
theme_minimal(base_size=10)

 #3Smoking
p3<-ggplot(Heart, aes(x=smoke))+geom_bar(fill="seagreen2")+ labs(x="Smoking")+
theme_minimal(base_size=10)

 #4Diabetes
p4<-ggplot(Heart, aes(x=dia))+geom_bar(fill="orange2")+
  labs(x="Diabetes Status")+ theme_minimal(base_size=10)
)p1+p2+p3 +p4+(
  plot_annotation(title="Demographic and Histology Distribution")
```

```
train_not_sc%>%
  ggplot(aes(diabetes, fill = DEATH_EVENT))+
  theme_gray+ ()
  geom_bar(position = "fill")

train_not_sc%>%
  ggplot(aes(high_blood_pressure, fill = DEATH_EVENT))+
  theme_gray+ ()
  geom_bar(position = "fill")

train_not_sc%>%
  ggplot(aes(platelets, fill = DEATH_EVENT))+
  theme_gray+ ()
  geom_histogram(aes(y = ..density../2), bins = 10)+
  geom_density(alpha = 0.2)

train_not_sc%>%
  ggplot(aes(serum_creatinine, fill = DEATH_EVENT))+
  theme_gray+ ()
  geom_histogram(aes(y = ..density../2))+
```

```
  geom_density(alpha = 0.2)+
  scale_x_log10()


library(gridExtra)
p1 <- train_not_sc%>%
  ggplot(aes(sex, fill = DEATH_EVENT))+
  theme_gray+ ()
  geom_bar(position = "fill")
p2 <- train_not_sc%>%
  ggplot(aes(smoking, fill = DEATH_EVENT))+
  theme_gray+ ()
  geom_bar(position = "fill")
grid.arrange(p1, p2, nrow = 1)
#################
c1<- ggplot(Heart, aes(x=age))+ geom_histogram(binwidth=5, colour="white", fill="darkseagreen2", alpha=0.8)+
  geom_density(eval(bquote(aes(y=..count..*5))),colour="darkgreen", fill="darkgreen", alpha=0.3)+ scale_x_continuous(breaks=seq(40,100,10))+geom_vline(xintercept = 65, linetype="dashed")+ annotate("text", x=50, y=45, label="Age <65", size=2.5, color="dark green") + annotate("text", x=80, y=45, label="Age >= 65", size=2.5, color="dark red") +labs(title="Age Distribution") + theme_minimal(base_size = 8)
c1
Heart$time
#########################3
set.seed(123)
train_index <- createDataPartition(Heart$DEATH_EVENT, p = 0.8, list = FALSE)
```

```
train_data <- Heart[train_index, ]
test_data <- Heart[-train_index, ]
head(train_data)
################################
#decision tree
tree_model <- rpart(DEATH_EVENT ~ ., data = train_data, method = "class")
rpart.plot(tree_model, main = "Decision Tree for Heart Failure", box.palette = "RdBu")
pred <- predict(tree_model, newdata = test_data, type = "class")
confusionMatrix(pred, test_data$DEATH_EVENT)
acc_glm <- confusionMatrix(pred, as.factor(test_data$DEATH_EVENT))$overall["Accuracy"]
tpr_glm <- confusionMatrix(pred, as.factor(test_data$DEATH_EVENT))$byClass["Specificity"]
pred <- predict(tree_model, newdata = test_data, type = "prob")
pred_obj <- prediction(pred[, 2], test_data$DEATH_EVENT)
perf_obj <- ROCR::performance(pred_obj, measure = "tpr", x.measure = "fpr")
plot(perf_obj, main = "ROC Curve for decision tree Model ,"
    xlab = "False Positive Rate", ylab = "True Positive Rate", colorize=T, lwd = 2(
abline(a = 0, b = 1, lwd = 1.5, col = "gray")
auc <- ROCR::performance(pred_obj, "auc")
auc <- auc@y.values[[1]]
legend("bottomright", paste("AUC = ", round(auc, 3)), bty = "n", cex = 0.8)


plotROC(test_data$DEATH_EVENT,pred[,2])
#######################
#random forest
rf_model <- randomForest(DEATH_EVENT ~ ., data = train_data, ntree = 500, mtry = 3)
pred <- predict(rf_model, newdata = test_data)
```

```
confusionMatrix(pred, test_data$DEATH_EVENT)



acc_glm <- confusionMatrix(pred, as.factor(test_data$DEATH_EVENT))$overall["Accuracy"]
tpr_glm <- confusionMatrix(pred, as.factor(test_data$DEATH_EVENT))$byClass["Specificity"]
pred <- predict(rf_model, newdata = test_data, type = "prob")
pred_obj <- prediction(pred[, 2], test_data$DEATH_EVENT)
perf_obj <- ROCR::performance(pred_obj, measure = "tpr", x.measure = "fpr")
plot(perf_obj, main = "ROC Curve for random forest Model ,"
    xlab = "False Positive Rate", ylab = "True Positive Rate", colorize=T, lwd = 2(
abline(a = 0, b = 1, lwd = 1.5, col = "gray")
auc <- ROCR::performance(pred_obj, "auc")
auc <- auc@y.values[[1]]
legend("bottomright", paste("AUC = ", round(auc, 3)), bty = "n", cex = 0.8)
##############################
acc_rf <- confusionMatrix(pred, as.factor(test_data$DEATH_EVENT))$overall["Accuracy"]
tpr_rf <- confusionMatrix(pred, as.factor(test_data$DEATH_EVENT))$byClass["Specificity"]
varImpPlot(rf_model, type = 2, main = "Variable Importance Plot for Random Forest")
plot(rf_model)
rf_model <- randomForest(DEATH_EVENT ~ time +
                         ejection_fraction +
                         serum_creatinine +
```

```
                  age+
                  creatinine_phosphokinase ,
                data = train_data ,
                ntree = 300 ,
                mtry = 2(
pred <- predict(rf_model, newdata = test_data)
confusionMatrix(pred, test_data$DEATH_EVENT)
rpredict<- predict(rf_model, test_data, type="class")
cm2<-confusionMatrix(rpredict, test_data$DEATH_EVENT1)
draw_confusion_matrix(cm2)
########################
#regression
glm_model <- caret::train(DEATH_EVENT ,. ~
                data = train_data ,
                method = "glm ,"
                trControl = trainControl(method = "cv", number = 10)(
pred <- predict(glm_model, newdata = test_data)
confusionMatrix(pred, test_data$DEATH_EVENT)
acc_glm <- confusionMatrix(pred,
as.factor(test_data$DEATH_EVENT))$overall["Accuracy"]
tpr_glm <- confusionMatrix(pred,
as.factor(test_data$DEATH_EVENT))$byClass["Specificity"]
pred <- predict(tree_model, newdata = test_data, type = "prob")
pred_obj <- prediction(pred[, 2], test_data$DEATH_EVENT)
perf_obj <- ROCR::performance(pred_obj, measure = "tpr", x.measure = "fpr")
plot(perf_obj, main = "ROC Curve for decision tree Model ,"
```

```
    xlab = "False Positive Rate", ylab = "True Positive Rate", colorize=T, lwd = 2(
abline(a = 0, b = 1, lwd = 1.5, col = "gray")
auc <- ROCR::performance(pred_obj, "auc")
auc <- auc@y.values[[1]]
legend("bottomright", paste("AUC = ", round(auc, 3)), bty = "n", cex = 0.8)
########################################
normalize <- function(x)}
  return ((x - min(x)) / (max(x) - min(x)))(
Heart$age=normalize(Heart$age)
Heart$creatinine_phosphokinase=normalize(Heart$creatinine_phosphokinase)
Heart$ejection_fraction=normalize(Heart$ejection_fraction)
Heart$platelets=normalize(Heart$platelets)
Heart$serum_creatinine =normalize(Heart$serum_creatinine )
Heart$serum_sodium =normalize(Heart$serum_sodium )
Heart$time=normalize(Heart$time)
#KNN
prop.table(table(Heart$DEATH_EVENT))
heart_train_x <- train_data %>%
  select_if(is.numeric)
heart_test_x <- test_data %>%
  select_if(is.numeric)
heart_train_y <- train_data[,"DEATH_EVENT"]
heart_test_y <- test_data[,"DEATH_EVENT"]


heart_train_xs <- scale(heart_train_x)
heart_test_xs <- scale(heart_test_x ,
                       center = attr(heart_train_xs, "scaled:center") ,
```

```
                scale = attr(heart_train_xs, "scaled:scale") (


sqrt(nrow(train_data))
i=1                    # declaration to initiate for loop
k.optm=1               # declaration to initiate for loop
for (i in 1:16) }
  knn.mod <-  knn(train=train_data, test=test_data, cl=train_data$DEATH_EVENT, k=i)
  k.optm[i]      <-      100      *      sum(test_data$DEATH_EVENT      ==
knn.mod)/NROW(test_data$DEATH_EVENT)
  k=i
  cat(k,'=',k.optm[i],'\n')       # to print % accuracy
{
plot(k.optm, type="b", xlab="K- Value",ylab="Accuracy level")  # to plot % accuracy
wrt to k-value

###################################
data.frame(algorithm = c("decision\ntree", "random\nforest,("
          accuracy = c(acc_dt, acc_rf)*100,
          recall = c(tpr_glm, tpr_dt, tpr_rf, tpr_gbm)*100⁄.<⁄. (
  pivot_longer(col = -algorithm, names_to = "metrics", values_to = "percent")⁄.<⁄.
  ggplot(aes(x = reorder(algorithm, X = percent,(
            y = percent,
            fill = metrics+ ((
  geom_bar(stat = "identity,"
          position = "dodge,"
```

```
        alpha=0.9+ (
geom_text(aes(group = metrics, label = str_c(sprintf("%2.1f", percent ,(("٪" ,(
         position = position_dodge(width = 0.9), vjust = -0.2+ (
scale_fill_manual(values = c("#1F77B4", "#999999"))+
labs(x = "algorithm", title = "Metrics of different classifier models")+
theme_minimal(base_size = 12)
```