مقایسه رگرسیون لوژستیک با برخی از الگوریتم‌های یادگیری ماشین در رده‌بندی داده‌ها

طیبه کرمی [۱]، محی‌الدین ایزدی [۲]، مهرداد نیاپرست [۳]

تاریخ دریافت: ۱۴۰۰/۰۲/۰۴
تاریخ پذیرش: ۱۴۰۰/۰۷/۱۶

چکیده:
یکی از مسائل مهم در علوم مختلف موضوع رده‌بندی است. رگرسیون لوژستیک یکی از روش‌های آماری برای رده‌بندی داده‌ها است که در آن توزیع داده‌ها معلوم فرض می‌شود. امروزه، محققان علاوه بر روش‌های آماری از روش‌های دیگری مانند الگوریتم‌های یادگیری ماشین که در آن نیاز به معلوم بودن توزیع داده‌ها نیست برای رده‌بندی داده‌ها استفاده می‌کنند. در این مقاله، رگرسیون لوژستیک و برخی از الگوریتم‌های یادگیری ماشین شامل CART، جنگل تصادفی، Bagging و تقویت در حوزه‌ی یادگیری با نظارت توضیح داده می‌شود. با استفاده از ۴ مجموعه داده واقعی و همچنین یک مثال شبیه‌سازی‌شده کارایی رگرسیون لوژستیک با الگوریتم‌های یادشده، بر اساس معیارهای دقت، حساسیت و صحت مورد مقایسه قرار می‌گیرند.

واژه‌های کلیدی: جنگل تصادفی، درخت تصمیم، یادگیری با نظارت، یادگیری گروهی

۱ مقدمه

یکی از مسائل مهم در علوم مختلف از جمله بیمه، اقتصاد، علوم پزشکی و ... رده‌بندی است. معمولاً در مسائل رده‌بندی، مشاهدات موجود شامل p متغیر پیشگو و یک متغیر پاسخ است که می‌تواند مقادیر ۱، ۲، ... ، k را اختیار کند. هدف در مسائل رده‌بندی این است که با استفاده از مشاهدات موجود، رده یک داده‌ی جدید پیش‌بینی شود. رگرسیون لوژستیک یکی از روش‌های آماری برای رده‌بندی داده‌ها است که در آن توزیع داده‌ها معلوم فرض می‌شود. پژوهشگران امروزه علاوه بر روش‌های آماری از روش‌های دیگری که در آن نیاز به معلوم بودن توزیع داده‌ها نیست؛ مانند یادگیری ماشین برای رده‌بندی داده‌ها استفاده می‌کنند. یکی از شاخه‌های وسیع و پرکاربرد در هوش مصنوعی یادگیری ماشین است. پژوهشگران در داده‌کاوی، کنترل روبات‌ها، تشخیص گفتار و بسیاری از مسائل دیگر یادگیری ماشین را به کار می‌گیرند. در یادگیری ماشین، کامپیوتر مجموعه داده را دریافت کرده و داده الگوریتم را مشخص کرده و به آن آموزش می‌دهد. یادگیری ماشین شامل یادگیری با نظارت، یادگیری بدون نظارت، یادگیری نیمه نظارتی و یادگیری تقویتی است. در این مقاله، به مطالعه برخی از الگوریتم‌های یادگیری با نظارت می‌پردازیم. در یادگیری با نظارت مجموعه داده‌ها دارای برچسب هستند. به‌عبارت‌دیگر، یک مجموعه داده شامل n مشاهده (نمونه) از ورودی‌ها (ویژگی‌ها، متغیرهای پیشگو) و خروجی (ویژگی هدف، متغیر پاسخ) است که ورودی‌ها با X_1، X_2، ...، X_p و خروجی با Y نشان داده می‌شود. در یک مجموعه داده اگر متغیر پاسخ، رسته‌ای باشد یادگیری با نظارت به مسئله رده‌بندی می‌پردازد. درخت تصمیم یکی از پرکاربردترین و قدیمی‌ترین الگوریتم‌های یادگیری با نظارت است که ساختاری شبیه فلوچارت دارد و جزء روش‌های ناپارامتری به‌حساب می‌آید ([۳]). هر درخت تصمیم شامل ریشه، گره‌های درونی، برگ و شاخه‌ها است. یک درخت تصمیم از ریشه به سمت پایین رشد می‌کند و در نهایت به برگ می‌رسد. تعریف اجزاء درخت تصمیم در زیر ارائه می‌شود.

ریشه: بالاترین گره در درخت تصمیم ریشه نام دارد و سایر گره‌ها زیر آن قرار می‌گیرند. ریشه مهم‌ترین جزء در ساختار درخت تصمیم است.

شاخه: در نمودار درخت تصمیم خطوطی که گره‌ها را به یکدیگر وصل می‌کنند شاخه نام دارند.

برگ: گره پایانی در ساختار درخت تصمیم برگ نامیده می‌شود. به‌عبارت‌دیگر برگ همان ویژگی هدف یا متغیر پاسخ است.

گره درونی: سایر گره‌ها به‌جز ریشه و برگ را گره درونی گویند.

[۱] دانش‌آموخته گروه آمار، دانشگاه رازی، کرمانشاه، (نویسنده مسئول: tayebeh.karami20@gmail.com)
[۲] هیئت‌علمی گروه آمار، دانشگاه رازی، کرمانشاه
[۳] هیئت‌علمی گروه آمار، دانشگاه رازی، کرمانشاه

درخت‌های تصمیم مختلفی از جمله درخت‌های ID3، C4.5، C5، CART[4]، CHAID و QUEST توسط پژوهشگران معرفی‌شده است ([۱۰]). الگوریتم CART یکی از پرکاربردترین درخت‌های تصمیم است که در بخش‌های بعد به‌صورت کامل توضیح داده می‌شود. از مزایای درخت تصمیم می‌توان به تفسیرپذیری بالا، فهم ساده، مقاومت در برابر نویز و سرعت مناسب اشاره کرد. زمانی که اندازه و عمق درخت زیاد باشد، بیش‌برازش اتفاق می‌افتد که می‌توان از روش هرس کردن برای رفع آن استفاده کرد. در هرس کردن درخت تصمیم، بعضی از شاخه‌ها و گره‌هایی که تأثیر ناچیزی بر رده‌بندی داده‌ها دارند حذف می‌شود. درخت تصمیم معایبی هم به شرح زیر دارد ([۱۰]).

۱ـ هرس درخت تصمیم به هزینه بالا و زمان بیشتری نیاز دارد.

۲ـ در مواردی که تعداد داده‌های آموزشی کم است ولی تعداد رده‌های متغیر پاسخ زیاد باشد، احتمال خطا در درخت تصمیم زیاد است.

۳ـ الگوریتم درخت تصمیم جز الگوریتم‌های یادگیری ناپایدار است؛ زیرا با یک تغییر کوچک در داده‌های آموزشی می‌تواند منجر به ایجاد یک درخت بسیار متفاوت می‌شود.

این معایب سبب می‌شود که درخت تصمیم در بعضی از مسائل رده‌بندی دارای دقت پایین باشد. به این منظور الگوریتم‌های یادگیری گروهی پیشنهاد می‌شود. الگوریتم‌های یادگیری گروهی متشکل از دو یا بیش از دو الگوریتم یادگیری ماشین است. مطالعات نشان می‌دهد که در بعضی مواقع عملکرد الگوریتم‌های یادگیری گروهی بهتر از استفاده از یک الگوریتم است ([۱]). یادگیری گروهی الگوریتم‌های مختلفی دارد که در این مقاله، سه الگوریتم Bagging[5]، جنگل تصادفی و تقویت[6]موردمطالعه قرار می‌گیرد. در این مطالعه، با استفاده از ۴ مجموعه داده مختلف، به مقایسه مدل رگرسیون لوژستیک با الگوریتم‌های CART، جنگل تصادفی، تقویت سازوار[7] CART و Bagging CART می‌پردازیم. ساختار ارائه مطالب در این مقاله به این صورت زیر است.

در بخش ۲، رگرسیون لوژستیک ارائه می‌شود. در بخش ۳، الگوریتم CART بیان‌شده است. الگوریتم‌های یادگیری گروهی در بخش ۴ توضیح داده می‌شود. در بخش ۵، به توضیح معیار ارزیابی عملکرد الگوریتم پرداخته‌شده است و مقایسه الگوریتم‌ها در بخش ۶ ارائه می‌شود.

۲ رگرسیون لوژستیک

یک حالت از رگرسیون خطی تعمیم‌یافته، رگرسیون لوژستیک است که ارتباط بین متغیرهای پیشگو و متغیر پاسخ را بیان می‌کند. در رگرسیون لوژستیک متغیر پاسخ یک متغیر رسته‌ای با دو یا بیش از دو مقدار است و متغیرهای پیشگو می‌توانند متغیرهای پیوسته و رسته‌ای باشند.

۱.۲ رگرسیون لوژستیک دودویی

متغیر پاسخ در رگرسیون لوژستیک دودویی یک متغیر رسته‌ای است که شامل دو مقدار ۰ و ۱ است. برای مثال، دو رده بیماری یا سلامت، مرگ یا زندگی و... را می‌توان نام برد. فرض کنید Y متغیر پاسخ و همچنین X_1، X_2،...، X_p متغیرهای پیشگو با مشاهدات متناظر x_1، x_2، ...، x_p باشند. آنگاه در رگرسیون لوژستیک دودویی،

$$Y \sim Bin(1, \pi(\boldsymbol{\beta}, \boldsymbol{x}))$$

که در آن $\pi(\boldsymbol{\beta}, \boldsymbol{x}) = P(Y = 1)$ دلالت بر احتمال موفقیت (متعلق بودن به رده ۱) به ازای بردار مشاهدات $\boldsymbol{x} = (x_1, \ldots, x_p)$ است. تابع لوجیت به‌عنوان تابع ربط (تبدیل) در رگرسیون لوژستیک استفاده می‌شود و به‌صورت

$$\begin{aligned}\text{logit}(\pi(\boldsymbol{\beta}, \boldsymbol{x})) =& \log \frac{\pi(\boldsymbol{\beta}, \boldsymbol{x})}{1 - \pi(\boldsymbol{\beta}, \boldsymbol{x})} \\ =& \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p\end{aligned}$$

است. پارامترهای رگرسیون لوژستیک با استفاده از روش ماکسیمم درستنمایی برآورد می‌شوند. فرض کنید $\hat{\beta}_1$، $\hat{\beta}_2$،...، $\hat{\beta}_p$ برآورد پارامترهای رگرسیون لوژستیک باشند. آنگاه مدل برازش شده به‌صورت

$$\text{logit}(\hat{\pi}(\hat{\boldsymbol{\beta}}, \boldsymbol{x})) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p$$

و یا

$$\hat{\pi}(\hat{\boldsymbol{\beta}}, \boldsymbol{x}) = \frac{\exp(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p)}{1 + \exp(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p)}$$

است. برای مشاهده جدید اگر $\hat{\pi}(\hat{\boldsymbol{\beta}}, \boldsymbol{x}) > \frac{1}{2}$ باشد، آنگاه $\boldsymbol{x}$ متعلق به رده $y = 1$ است ([۲]).

[4]Classification And Regression Tree

[5]Bootstrap aggregating

[6]Boosting

[7]Adaptive boosting (Adaboosting)

۲.۲ رگرسیون لوژستیک چند رده‌ای

متغیر پاسخ در رگرسیون لوژستیک چند رده‌ای بیش از دو مقدار اختیار می‌کند. فرض کنید Y یک متغیر رسته‌ای با c مقدار ممکن (c رده) باشد و همچنین X_1، ...، X_p متغیرهای پیشگو باشند. آنگاه مدل رگرسیون لوژستیک بر پایه رابطه

$$\log\left(\frac{\pi_j(\boldsymbol{\beta},\boldsymbol{x})}{\pi_c(\boldsymbol{\beta},\boldsymbol{x})}\right)=\alpha_j+\beta_{j1}x_1+\beta_{j2}x_2+\cdots+\beta_{jp}x_p;$$

$$j=1,2,\ldots,c$$

است که در آن $\sum_{j=1}^{c}\pi_j(\boldsymbol{\beta},\boldsymbol{x})=1$ و $\pi_j(\boldsymbol{\beta},\boldsymbol{x})=P(Y=j)$. به‌طورکلی برای پیش‌بینی رده داده‌ی جدید در رگرسیون لوژستیک چند رده‌ای از شاخصی به نام نسبت بخت‌ها (OR[8])؛ یعنی

$$\mathrm{OR}_{ij}=\frac{\frac{\pi_i(\boldsymbol{\beta},\boldsymbol{x})}{1-\pi_i(\boldsymbol{\beta},\boldsymbol{x})}}{\frac{\pi_j(\boldsymbol{\beta},\boldsymbol{x})}{1-\pi_j(\boldsymbol{\beta},\boldsymbol{x})}}$$

استفاده می‌شود.

۱) اگر $\mathrm{OR}_{ij}>1$ باشد، آنگاه مشاهده $\boldsymbol{x}=(x_1,\ldots,x_p)$ متعلق به رده i است.

۲) اگر $\mathrm{OR}_{ij}<1$ باشد، آنگاه مشاهده $\boldsymbol{x}=(x_1,\ldots,x_p)$ متعلق به رده j است.

۳) اگر $\mathrm{OR}_{ij}=1$ باشد، آنگاه نمی‌توان پیش‌بینی کرد که مشاهده $\boldsymbol{x}=(x_1,\ldots,x_p)$ به کدام رده تعلق دارد ([۲]).

۳ الگوریتم CART

الگوریتم CART توسط بریمن[9] [۴] معرفی‌شده است. این درخت برای مسائل رده‌بندی و رگرسیون به‌کار برده می‌شود. در این مقاله، از الگوریتم CART در مسائل رده‌بندی استفاده می‌شود. خصوصیات الگوریتم CART به‌صورت زیر است:

۱) متغیرهای پیشگو در یک مجموعه داده برای این درخت می‌توانند به‌صورت متغیرهای رسته‌ای و کمی باشند.

۲) فقط دو شاخه از هر گره درونی این درخت خارج می‌شود.

۳) برای هرس کردن الگوریتم CART از معیار پیچیدگی هزینه هرس[10]، برای هرس کردن استفاده می‌شود.

۴) این درخت توانایی انجام رده‌بندی با داده گمشده در یک مجموعه داده را دارد.

۵) در این درخت برای انتخاب ریشه و گره‌ها از شاخص جینی و شاخص دوپاره‌سازی [11] استفاده می‌شود. در یک مجموعه داده هنگامی متغیر پاسخ دارای دو رده است از شاخص جینی[12] و هنگامی‌که متغیر پاسخ بیش از دو رده داشته باشد از شاخص دوپاره‌سازی استفاده می‌شود ([۱۰]). فرض کنید S یک مجموعه داده با n مشاهده است. شاخص جینی متغیر پاسخ Y به‌صورت

$$\mathrm{Gini}(Y,S)=1-\sum_{c_j\in\mathrm{dom}(Y)}\left(\frac{|\sigma_{Y=c_j}(S)|}{|S|}\right)^2 \qquad (1)$$

تعریف می‌شود که در آن

$\sigma_A(S)$: زیرمجموعه‌ای از S است که شرط A برای اعضای آن برقرار است.

$|S|$: تعداد اعضای مجموعه S است.

$\mathrm{dom}(X_i)$: مجموعه مقادیر ممکن متغیر X_i است؛ یعنی

$$\mathrm{dom}(X_i)=\{x_{i,1},x_{i,2},\ldots,x_{i,|\mathrm{dom}(X_i)|}\}.$$

$\mathrm{dom}\ (Y)$: مجموعه مقادیر ممکن متغیر پاسخ Y است؛ یعنی

$$\mathrm{dom}(Y)=\{c_1,c_2,\ldots,c_{|\mathrm{dom}(Y)|}\}.$$

همچنین شاخص دوپاره‌سازی متغیر پاسخ Y به‌صورت

$$\mathrm{Twoing}(Y,\mathrm{dom}_1(Y),\mathrm{dom}_2(Y))=0/25\times\frac{|\sigma_{Y\in\mathrm{dom}_1(Y)}(S)|}{|S|}\times\frac{|\sigma_{Y\in dom_2(Y)}(S)|}{|S|}\times\left(\sum_{c_j\in\mathrm{dom}(Y)}\left|\frac{|\sigma_{Y\in\mathrm{dom}_1(Y)\ AND\ y=c_i}(S)|}{|\sigma_{Y\in\mathrm{dom}_1(Y)}(S)|}-\frac{|\sigma_{Y\in\mathrm{dom}_2(Y)\ AND\ y=c_i}(S)|}{|\sigma_{Y\in\mathrm{dom}_2(Y)}(S)|}\right|\right)^2$$

تعریف می‌شود که در آن رده‌های متغیر پاسخ Y (دارای هرچند رده که باشد) به دو رده $\mathrm{dom}_1(Y)$ و $\mathrm{dom}_2(Y)$ تقسیم‌شده است.

لازم به ذکر است شاخص جینی متغیر X_i در این مجموعه به‌صورت

$$\mathrm{Gini}(X_i,S)=\frac{|\sigma_{X_{i,1}}(S)|}{|S|}\mathrm{Gini}(X_{i,1})+\cdots+\frac{|\sigma_{X_{i,|\mathrm{dom}(X_i)|}}(S)|}{|S|}\mathrm{Gini}(X_{i,|\mathrm{dom}(X_i)|})$$

تعریف می‌شود که در آن

$$\mathrm{Gini}(X_{i,j},S)=1-\sum_{x_{i,j}\in dom(X_i)}\left(\frac{|\sigma_{X_i=x_{i,j}}(S)|}{|S|}\right)^2$$

است.

[8] Odd ratio
[9] Breiman
[10] Cost Complexity Pruning
[11] Twoing index
[12] Gini index

۱.۳ نحوه هرس کردن الگوریتم CART

الگوریتم CART برای هرس کردن از معیار پیچیدگی هزینه هرس استفاده می‌کند. معیار پیچیدگی هزینه هرس در دو مرحله انجام می‌شود. در مرحله‌ی اول، دنباله‌ای از درخت‌های $T_k, \ldots, T_1, T_0$ براساس مجموعه داده S ساخته می‌شوند که T_0 درخت اصلی قبل از هرس کردن و T_k درخت ریشه است. درخت T_1 با هرس کردن T_0، یعنی با جایگزینی یک یا چند گره درونی از درخت T_0 با برگ ایجاد می‌شود. به همین ترتیب، درخت T_i با هرس کردن T_{i-1} به دست می‌آید. در مرحله‌ی دوم، با استفاده از برآورد خطای تعمیم‌یافته؛ یعنی

$$\varepsilon(T, D) = \sum_{(\boldsymbol{x}, y) \in S} D(\boldsymbol{x}, y) \times L(y, T(\boldsymbol{x})) \qquad (۲)$$

که در آن D توزیع مجموعه داده S و نامعلوم است، درختی که دارای کمترین مقدار خطای تعمیم‌یافته است به‌عنوان درخت هرس شده انتخاب می‌شود.

در مرحله اول، درخت T_{i+1} با جایگزینی یک یا چند گره در درخت T_i با برگ‌های مناسب به دست می‌آید. گره‌هایی که هرس می‌شوند گره‌هایی هستند که با هرس کردن آن‌ها نرخ خطای متناظر آن‌ها؛ یعنی

$$\alpha_i = \frac{\varepsilon(pruned(T_i, S), S) - \varepsilon(T, S)}{|leaves(T)| - |leaves(pruned(T_i, t))|} \qquad (۳)$$

کمترین مقدار باشد که در آن

$\varepsilon(T, S)$: نرخ خطا درخت T که بر اساس مجموعه داده S ساخته‌شده‌است.

$\varepsilon(pruned(T_i, S), S)$: نرخ خطا درخت هرس شده T_i که بر اساس مجموعه داده S ساخته‌شده است.

$leaves(T)$: تعداد برگ‌های درخت T است.

$leaves(pruned(T, t))$: تعداد برگ‌های درخت هرس شده T_i است.

برای جزئیات بیشتر در مورد الگوریتم CART و کاربردهای آن، می‌توان به [۴]، [۱۰]، [۱۲] و [۱۵] مراجعه کرد.

۱.۴ بازنمونه‌گیری خودگردان

روش بازنمونه‌گیری خودگردان[14] توسط افرون [15] [۷] معرفی‌شده است. فرض کنید Z_1، Z_2، ...، Z_n یک نمونه تصادفی از توزیع نامعلوم F باشد. براساس نمونه تصادفی در نظر گرفته‌شده، یکی از رایج‌ترین برآوردگرهای تابع توزیع تجمعی نامعلوم جامعه، تابع توزیع تجربی است که به‌صورت

$$\hat{F}_n(z) = \frac{1}{n} \sum_{i=1}^{n} I(Z_i \leqslant z)$$

تعریف می‌شود که در آن $I(A) = 1$ اگر پیشامد A رخ دهد و در غیر این صورت $I(A) = 0$.

اکنون آماره $T = T(\mathbf{Z})$ را در نظر بگیرید که در آن $\mathbf{Z} = (Z_1, \ldots, Z_n)$. بازنمونه‌گیری خودگردان روشی برای تقریب توزیع T است. در روش بازنمونه‌گیری خودگردان، با استخراج نمونه‌هایی به حجم n به روش باجایگذاری از نمونه اولیه Z_1، Z_2، ...، Z_n می‌توان نمونه‌ای تصادفی از T به دست آورد که براساس آن، ویژگی‌های توزیع T را موردبررسی قرار داد. بنابراین اگر B باز نمونه $\mathbf{Z}^*_{(i)}$، $i = 1, 2, \ldots, B$، از نمونه اولیه استخراج شود، آنگاه نمونه $T(\mathbf{Z}^*_{(1)})$، $T(\mathbf{Z}^*_{(2)})$، ...، $T(\mathbf{Z}^*_{(B)})$ از توزیع T خواهیم داشت. بنابراین می‌توان با روش‌های ناپارامتری موجود، ویژگی‌های توزیع T را موردمطالعه قرار دارد.

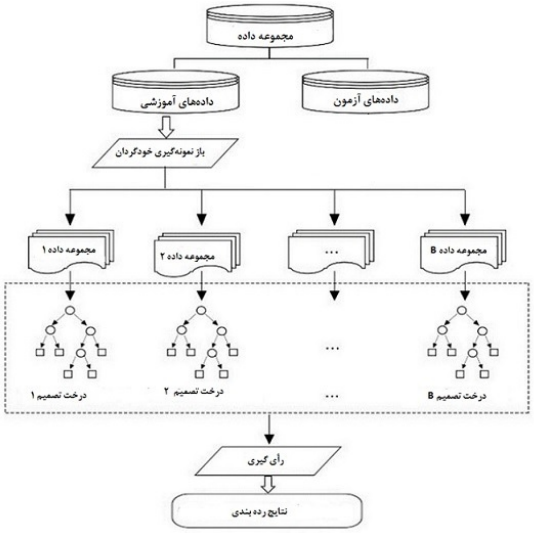
۲.۴ الگوریتم Bagging

یکی از ساده‌ترین و درعین‌حال موفق‌ترین الگوریتم‌های یادگیری گروهی الگوریتم Bagging در حوزه یادگیری با نظارت است که توسط بریمن [۵] معرفی‌شده است. این الگوریتم معمولاً برای الگوریتم‌های ناپایدار مانند درخت تصمیم بسیار کاراتر است و باعث کاهش واریانس، افزایش دقت و جلوگیری از بیش‌برازش می‌شود. الگوریتم Bagging به‌صورت زیر است. فرض کنید S یک مجموعه داده با n مشاهده است و G مجموعه داده آموزشی با m مشاهده است $(m < n)$. از مجموعه داده آموزشی G به روش بازنمونه‌گیری خودگردان B مجموعه داده آموزشی تولید می‌شود. یک الگوریتم یادگیری ماشین رده‌بندی یکسان (الگوریتم درخت تصمیم، الگوریتم شبکه عصبی و ...) روی هرکدام از این B مجموعه تولیدشده، پیاده‌سازی می‌شود و در نهایت نتایج این B الگوریتم باهم ترکیب می‌شوند.

۴ الگوریتم‌های یادگیری گروهی

الگوریتم‌های یادگیری گروهی[13] متشکل از دو یا بیش از دو الگوریتم یادگیری ماشین است. در استفاده برخی از الگوریتم‌های یادگیری گروهی از روش بازنمونه‌گیری خودگردان برای تولید چندین مجموعه داده آموزشی متفاوت استفاده می‌شود که در زیر به‌اختصار به بیان آن می‌پردازیم.

[13] Ensemble learning
[14] Bootstrap Resampling
[15] Efron



شکل ۱: فرایند ساخت جنگل تصادفی

به‌عنوان‌مثال، فرض کنید x_i یک مشاهده جدید است و هرکدام از این B الگوریتم یک رده خاص برای x_i در نظر گرفته باشند. بر این اساس، رده‌ای برای x_i در نظر گرفته می‌شود که در B نتیجه به‌دست‌آمده دارای بیشترین فراوانی باشد. در این مقاله از روش Bagging CART استفاده می‌شود؛ یعنی بر روی هرکدام از B مجموعه داده، الگوریتم CART اعمال می‌شود. برای توضیحات بیشتر و کاربرد این الگوریتم می‌توان به [۵]، [۱۳] و [۱۴] مراجعه نمود.

۳.۴ جنگل تصادفی

یکی دیگر از الگوریتم‌های یادگیری گروهی در حوزه یادگیری با نظارت جنگل تصادفی است که توسط بریمن [۶] معرفی‌شده است. جنگل تصادفی شامل مجموعه‌ای از درخت‌های هرس نشده است. در مواردی که مجموعه داده شامل داده‌های گمشده، دورافتاده و نویز باشد از جنگل تصادفی می‌توان استفاده کرد. شایان‌ذکر است الگوریتم جنگل تصادفی در بعضی از مسائل رده‌بندی عملکرد بهتری نسبت به درخت تصمیم دارد ([۱]). فرض کنید S یک مجموعه داده شامل n مشاهده و p متغیر پیشگو و G مجموعه داده آموزشی با m مشاهده باشد $(m < n)$. فرایند الگوریتم جنگل تصادفی به‌صورت زیر است:

۱) یک مجموعه داده آموزشی به روش بازنمونه‌گیری خودگردان از مجموعه داده آموزشی G تولید می‌شود.

۲) از مجموعه داده آموزشی تولیدشده (به روش بازنمونه‌گیری خودگردان) q متغیر پیشگو به روش نمونه‌گیری تصادفی بدون جایگذاری از p متغیر پیشگو انتخاب می‌شود $(q < p)$. به‌طورمعمول $q = \sqrt{p}$ برای ساخت جنگل تصادفی در نظر گرفته می‌شود.

۳) بر روی این مجموعه داده آموزشی تولیدشده با q متغیر پیشگو الگوریتم CART پیاده‌سازی می‌شود.

مراحل ۱ تا ۳، B بار تکرار می‌شود.

هر درخت تصمیم در جنگل تصادفی پیش‌بینی می‌کند که هر مشاهده به کدام رده تعلق دارد و در نهایت جنگل تصادفی رده‌ای که اکثریت آرا را به دست آورده باشد برای هر مشاهده انتخاب می‌کند.

الگوریتم جنگل تصادفی تا حدودی شبیه الگوریتم Bagging است، با این تفاوت که این الگوریتم q متغیر پیشگو به‌صورت تصادفی انتخاب می‌کند. اگر $q = p$، در این صورت جنگل تصادفی همان الگوریتم Bagging است. شکل ۱ فرایند تشکیل جنگل تصادفی بر روی یک مجموعه داده را نشان می‌دهد. برای توضیحات بیشتر در مورد جنگل تصادفی و کاربردهای آن به [۶]، [۱۵] و [۱۷] مراجعه نمایید.

۴.۴ الگوریتم تقویت

الگوریتم تقویت یکی دیگر از الگوریتم‌های یادگیری گروهی در حوزه یادگیری با نظارت است که توسط شاپیر [16] [۱۰] معرفی‌شده است. برای آموزش الگوریتم تقویت از کل داده‌های آموزشی استفاده می‌شود، اما در هر بار تکرار وزن داده‌های آموزشی تغییر می‌کند. برای مثال، فرض کنید یک مجموعه داده آموزشی دارای سه مشاهده x_1، x_2 و x_3 است که به دو رده تعلق دارند. در ابتدا الگوریتم یادگیری ماشین h_1 روی داده‌های آموزشی پیاده‌سازی می‌شود. فرض کنید نتایج این مرحله به این صورت است که رده‌های

[16] Schapire

x_1 و x_2 توسط الگوریتم h_1 به‌درستی پیش‌بینی‌شده است، اما رده x_3 به‌اشتباه پیش‌بینی‌شده است. در تکرار دوم، به دلیل اینکه رده مشاهده x_2 در مرحله قبل به‌اشتباه رده‌بندی‌شده است، بنابراین وزن بیشتری به x_2 اختصاص داده می‌شود. اکنون الگوریتم h_2 روی مشاهدات با وزن‌های مشخص‌شده پیاده‌سازی می‌شود. فرض کنید در این مرحله رده‌ی x_2 به‌اشتباه پیش‌بینی‌شده است. در تکرار سوم، وزن x_2 افزایش و وزن x_3 کاهش می‌یابد و مشاهده x_1 نیز کمترین وزن را به خود اختصاص می‌دهد. سپس، الگوریتم h_3 روی این داده‌های آموزشی که وزن‌های آن‌ها تغییر کرده است، پیاده‌سازی می‌شود. در نهایت نتایج الگوریتم‌های h_1، h_2 و h_3 باهم ترکیب می‌شوند. لازم به ذکر است الگوریتم‌های h_1، h_2 و h_3 همگی از یک نوع الگوریتم هستند. به‌عنوان‌مثال، همگی درخت تصمیم یا همگی شبکه عصبی و... می‌باشند. شایان‌ذکر است که الگوریتم تقویت شامل الگوریتم‌های متفاوت است. در ادامه، یکی از پرکاربردترین آن‌ها به نام الگوریتم تقویت سازوار را بیان خواهیم کرد.

۱.۴.۴ الگوریتم تقویت سازوار

الگوریتم تقویت سازوار توسط شاپیر و فروند[17][۸] معرفی شد. فرض کنید S یک مجموعه داده شامل n مشاهده از p متغیر پیشگو و یک متغیر پاسخ (ویژگی هدف) دو رده‌ای است که m مشاهده آن داده‌های آموزشی را تشکیل می‌دهند. برای انجام الگوریتم تقویت سازوار، در ابتدا از بین p متغیر پیشگو، یک متغیر که دارای کمترین مقدار شاخص جینی یا بیشترین مقدار معیار بهره اطلاع (IG[18]) باشد، انتخاب می‌شود. معیار بهره اطلاع متغیر پیشگو X_i در این مجموعه داده به‌صورت

$$\begin{aligned}\mathrm{IG}(X_i,S) =& E(Y,S) - \sum_{x_{i,j}\in dom(X_i)} \frac{|\sigma_{X_i=x_{i,j}}(S)|}{|S|} \\ &\times E(Y,\sigma_{X_i=x_{i,j}}(S))\end{aligned} \qquad (۴)$$

تعریف می‌شود که در آن

$$E(Y,S) = -\sum_{c_j\in dom(Y)} \frac{|\sigma_{Y=c_j}(S)|}{|S|}\log_2\frac{|\sigma_{Y=c_j}(S)|}{|S|}$$

$E(Y,B)$: آنتروپی متغیر پاسخ Y (ویژگی هدف) براساس مجموعه داده B است که با $E(Y|B)$ نیز نشان داده می‌شود.

فرایند الگوریتم تقویت سازوار بر روی یک متغیر انتخاب‌شده از بین p متغیر پیشگو به شرح زیر است. فرض کنید D یک مجموعه داده آموزشی شامل m مشاهده از یک متغیر پیشگو و یک متغیر پاسخ به‌صورت

$$D = \{(x_i,y_i)\}_{i=1}^m \qquad x_i\in\chi \qquad y_i \in= \{+1,-1\}$$

است. همان‌گونه که پیش‌تر گفته شد، در ابتدا وزن یکسان به‌صورت

$$w_0(i) = \frac{1}{m}$$

برای تمام داده‌های آموزشی در نظر گرفته می‌شود. وزن مرحله‌ی rام از الگوریتم مورداستفاده‌شده به‌صورت

$$\alpha_r = \frac{1}{2}\ln\frac{1-\varepsilon_r(h_r(x))}{\varepsilon_r(h_r(x))}$$

به دست می‌آید که در آن خطای وزنی الگوریتم در مرحله rام، $r = 1,2,\ldots,B$ به‌صورت

$$\varepsilon_r(h_r(x)) = \frac{\sum_{i=1}^m w_r(i)\times L(y,h_r(x))}{\sum_{i=1}^m w_r(i)}$$

محاسبه می‌شود و در آن

$$L(y,h_r(x)) = \begin{cases} 0 & if\ y = h_r(x) \\ 1 & if\ y \neq h_r(x)\end{cases}$$

است و $h_r(x)$ رده پیش‌بینی‌شده مشاهده x با استفاده از الگوریتم h_r، $w_r(i)$ وزن مشاهده‌ی iام در مرحله rام است. وزن نرمال شده هر داده آموزشی بعد از پیاده‌سازی مرحله rام الگوریتم به‌صورت

$$w_{r+1}(i) = \frac{w_r(i)}{Z_r}\times\begin{cases} e^{-\alpha_r} & if\ y = h_r(x_i) \\ e^{+\alpha_r} & if\ y \neq h_r(x_i)\end{cases}$$

به دست می‌آید که در آن

$$Z_r = \sum_{j=1}^m w_r(j)\times\begin{cases} e^{-\alpha_r} & if\ y = h_r(x_j) \\ e^{+\alpha_r} & if\ y \neq h_r(x_j).\end{cases}$$

در نهایت، الگوریتم نهایی

$$H(x) = \sum_{r=1}^B \alpha_r\, h_r(x)$$

که متوسط وزنی از الگوریتم‌های به‌دست‌آمده در B مرحله است مورداستفاده قرار می‌گیرد. رده مشاهده x براساس علامت $H(x)$ مشخص می‌شود. اگر علامت $H(x)$ مثبت باشد، مشاهده x به رده $(+)$ تعلق دارد، اما اگر علامت $H(x)$ باشد مشاهده به رده $(-)$ تعلق دارد.

لازم به ذکر است، در این مطالعه، الگوریتم CART بر روی B مجموعه داده آموزشی با وزن‌های متفاوت پیاده‌سازی شده است که به‌اصطلاح به آن تقویت سازوار CART گفته می‌شود. برای توضیحات بیشتر و کاربرد این الگوریتم، می‌توان به [۱۳] و [۱۴] مراجعه نمود.

[17] Freund

[18] Information Gain

۵ معیار ارزیابی عملکرد الگوریتم

در معیارهای ارزیابی یک الگوریتم از ماتریس درهم‌ریختگی استفاده می‌شود که در زیر به‌اختصار به تعریف آن می‌پردازیم.

ماتریس درهم‌ریختگی

ماتریس درهم‌ریختگی[19] یک ماتریس مربع $k \times k$ است که k تعداد رده‌های متغیر پاسخ در مسائل رده‌بندی است. به‌منظور سهولت کار، فرض کنید $k = ۲$؛ یعنی متغیر پاسخ شامل دو رده مثبت و منفی است. مانند جدول ۱، مؤلفه‌های روی قطر فرعی ماتریس درهم‌ریختگی تعداد مشاهداتی است که الگوریتم رده‌ی آن‌ها را به‌درستی پیش‌بینی کرده است؛ یعنی

TP[20]: تعداد مشاهداتی است که به رده مثبت تعلق دارند و الگوریتم به‌درستی رده مثبت برای آن‌ها پیش‌بینی کرده است.

TN[21]: تعداد مشاهداتی است که به منفی تعلق دارند و الگوریتم به‌درستی رده منفی برای آن‌ها پیش‌بینی کرده است.

قطر اصلی ماتریس درهم‌ریختگی تعداد مشاهداتی است که الگوریتم رده آن‌ها را به‌اشتباه پیش‌بینی کرده است؛ یعنی

FN[22]: تعداد مشاهداتی است که به رده مثبت تعلق دارند، اما الگوریتم به‌اشتباه رده منفی برای آن‌ها پیش‌بینی کرده است.

FP[23]: تعداد مشاهداتی است که به رده منفی تعلق دارند، اما الگوریتم به‌اشتباه رده مثبت برای آن‌ها پیش‌بینی کرده است.

جدول ۱. ماتریس درهم‌ریختگی

واقعی	پیش‌بینی	
	مثبت	منفی
مثبت	TP	FN
منفی	FP	TN

برای ارزیابی عملکرد الگوریتم‌ها معیارهای متفاوتی وجود دارد که در ادامه به توضیح معیارهای دقت، حساسیت و صحت می‌پردازیم.

۱.۵ دقت

دقت یکی از متداول‌ترین معیارهای ارزیابی الگوریتم‌ها در مسائل رده‌بندی است. این معیار بیانگر این است که رده چه درصدی از مشاهدات توسط الگوریتم به‌کاررفته به‌درستی پیش‌بینی‌شده است. با توجه به جدول ۱، دقت یک الگوریتم به‌صورت

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + TN + FP}$$

به دست می‌آید. با توجه به معیار دقت می‌توان نتیجه گرفت که یک الگوریتم به چه میزان به‌درستی آموزش داده‌شده است و یا کارایی الگوریتم به‌طورکلی چگونه است.

۲.۵ حساسیت

در برخی از مسائل رده‌بندی، پیش‌بینی درست مشاهدات مربوط به رده مثبت اهمیت بیشتری دارد. معیار حساسیت که به‌صورت زیر تعریف می‌شود، دقت پیش‌بینی در رده مثبت را اندازه‌گیری می‌کند.

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

۳.۵ صحت

معیار صحت، که مقدار پیشگویی مثبت [24] هم گفته می‌شود، میزان صحت رده پیش‌بینی‌شده مثبت توسط الگوریتم برای یک مشاهده را اندازه‌گیری می‌کند. معیار صحت به‌صورت زیر تعریف می‌شود.

$$\text{Percision} = \frac{TP}{TP + FP}$$

در حالت کلی، برای محاسبه معیار حساسیت یک الگوریتم، هر بار یکی از رده‌ها به‌عنوان رده مثبت در نظر گرفته‌شده و معیار حساسیت برای آن رده به دست می‌آید. میانگین مقدارهای به‌دست‌آمده برای تمام رده‌ها به‌عنوان حساسیت الگوریتم در نظر گرفته می‌شود. به شیوه مشابه، معیار صحت یک الگوریتم محاسبه می‌شود. برای جزئیات بیشتر در مورد معیارهای ارزیابی می‌توان به [۱۸] مراجعه کرد.

۶ مقایسه الگوریتم‌ها

در این بخش، بر اساس معیارهای دقت، حساسیت و صحت و با استفاده از ۴ مجموعه داده واقعی و یک مثال شبیه‌سازی، به مقایسه کارایی الگوریتم‌های رگرسیون لوژستیک، CART، جنگل تصادفی، Bagging CART و تقویت سازوار CART می‌پردازیم. همچنین برای اعتبارسنجی آن‌ها از روش اعتبارسنجی ۱۰ گروهی استفاده می‌شود.

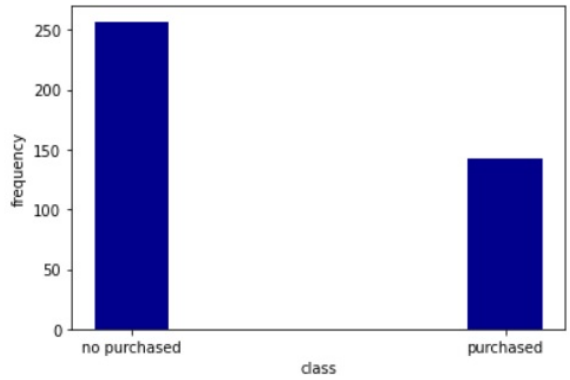
لازم به ذکر است برای اجرای الگوریتم‌های ذکرشده از نرم‌افزار Python 3.7 در محیط Anoconda 3 (2020.07) استفاده‌شده است.

[19] Confusion Matrix
[20] True Positive
[21] True Negative
[22] False Negative
[23] False Positive
[24] Positive predictive value

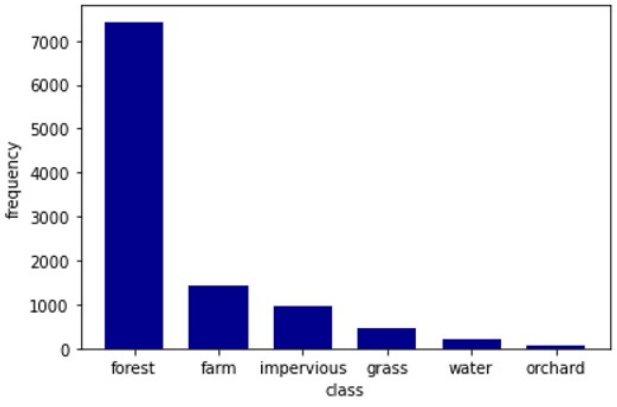
ویژگی‌های ۴ مجموعه داده مورداستفاده به‌طور اختصار در جدول ۲ آورده شده است.

جدول ۲. معرفی مجموعه داده‌ها

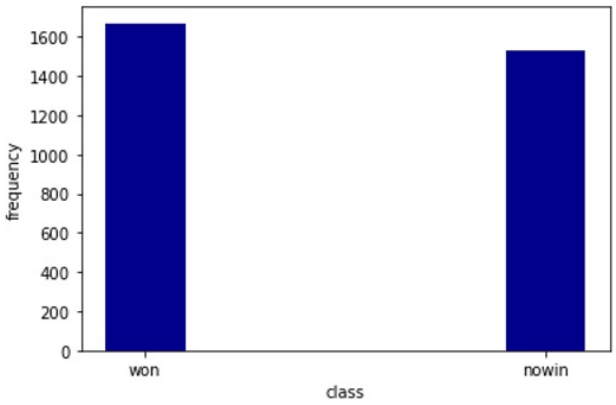
مجموعه داده	تعداد مشاهده	تعداد متغیرهای پیشگو	نوع متغیرها	تعداد رده متغیر پاسخ
تبلیغات شبکه اجتماعی	۴۰۰	۳	کمی و رسته‌ای	۲
تصاویر ماهواره‌ای	۱۰۵۴۵	۲۸	کمی و رسته‌ای	۶
بازی شطرنج	۳۱۹۶	۳۶	رسته‌ای	۲
ارزیابی اتومبیل	۱۷۲۸	۴	رسته‌ای	۴



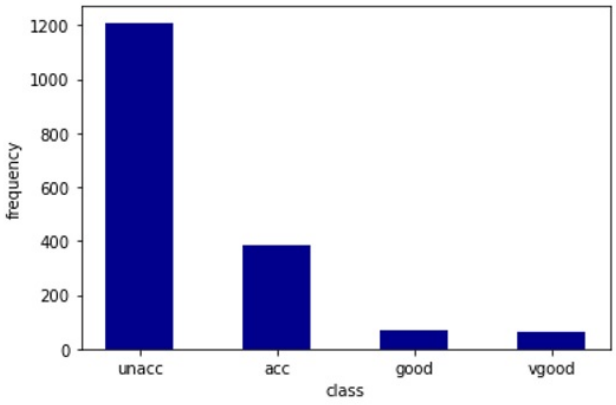
شکل ۱. نمودار میله‌ای متغیر پاسخ برای مجموعه داده تبلیغات شبکه اجتماعی



شکل ۲. نمودار میله‌ای متغیر پاسخ برای مجموعه داده تصاویر ماهواره‌ای



شکل ۳. نمودار میله‌ای متغیر پاسخ برای مجموعه داده بازی شطرنج



شکل ۴. نمودار میله‌ای متغیر پاسخ برای مجموعه داده ارزیابی اتومبیل

مجموعه داده تبلیغات شبکه اجتماعی [۲۵] مربوط به مشتری‌های سایت trused است که شامل ۴۰۰ مشاهده از ۳ متغیر پیشگو و یک متغیر پاسخ با دو رده خرید به‌صورت اینترنتی و عدم خرید است. مجموعه داده تصاویر ماهواره‌ای [۲۶] مربوط به سال‌های ۲۰۱۴ تا ۲۰۱۵ است و شامل ۱۰۵۴۵ مشاهده از ۲۸ متغیر پیشگو و یک متغیر پاسخ با ۶ رده جنگل، مزرعه، مناطق غیرقابل نفوذ، فضای سبز، آب و باغ است. مجموعه داده بازی شطرنج [۲۷] شامل ۳۱۹۶ مشاهده از ۳۶ متغیر پیشگو و یک متغیر پاسخ با دو رده برنده و بازنده است. مجموعه داده ارزیابی اتومبیل [۲۸] شامل ۱۷۲۸ مشاهده از ۴ متغیر پیشگو و یک متغیر پاسخ با ۴ رده غیرقابل‌قبول، قابل‌قبول، خوب و خیلی خوب است. نمودار میله‌ای متغیر پاسخ در مجموعه داده‌های در نظر گرفته‌شده در شکل‌های ۱، ۲، ۳ و ۴ رسم شده است. الگوریتم‌های رگرسیون لوژستیک، CART، جنگل تصادفی، Bagging CART و تقویت سازوار CART روی این چهار مجموعه

[۲۵] مجموعه داده Social Network Ads در سایت www.superdatascience.com/machine-learning موجود است.
[۲۶] مجموعه داده Crowdsourced Mapping در سایت archive.ics.uci.edu/ml/index.php موجود است.
[۲۷] مجموعه داده Chess (King-Rook -King-Pawn) در سایت archive.ics.uci.edu/ml/index.php موجود است.
[۲۸] مجموعه داده Car Evaluation در سایت archive.ics.uci.edu/ml/index.php موجود است.

داده اجرا و دقت پیش‌بینی آن‌ها براساس معیارهای دقت، حساسیت و صحت در جدول‌های ۳، ۴، ۵ و ۶ محاسبه‌شده است. با توجه به جدول ۳، برای مجموعه داده تبلیغات شبکه اجتماعی، الگوریتم تقویت سازوار CART دارای بیشترین دقت با مقدار ۰/۸۸۰ و جنگل تصادفی دارای بیشترین حساسیت و صحت به ترتیب با مقدارهای ۰/۸۹۷۵ و ۰/۹۰۰۸ است. رگرسیون لوژستیک دارای ضعیف‌ترین کارایی براساس معیارهای ارزیابی است. الگوریتم تقویت سازوار CART نسبت به بقیه الگوریتم‌ها به مدت‌زمان بیشتری برای اجرا نیاز دارد.

جدول ۳. معیارهای ارزیابی الگوریتم‌ها برای مجموعه داده تبلیغات شبکه اجتماعی

معیار	رگرسیون لوژستیک	CART	*Bagging CART*	سازوار تقویت *CART*	جنگل تصادفی
دقت	۰/۶۹۵۰	۰/۸۴۷۴	۰/۷۸۷۵	۸۸۰۰	۰/۸۷۷
حساسیت	۰/۶۵	۰/۸۵۲۵	۰/۸۱۲۵	۰/۸۸۱۵	۰/۸۹۷۵
صحت	۰/۴۱۹۴	۰/۸۶۰۴	۰/۸۰۶۷	۰/۸۸۶۳	۰/۹۰۸۷
مدت‌زمان اجرا(ثانیه)	۰/۲۲۵۴	۰/۰۳۷۳	۰/۲۶۴۲	۲/۸۷۷۲	۰/۳۷۶۳

با توجه به جدول ۴، برای مجموعه داده تصاویر ماهواره‌ای الگوریتم جنگل تصادفی دارای بیشترین مقدار دقت، حساسیت و صحت به ترتیب با مقدارهای ۰/۸۳۸۷، ۰/۹۳۷۶ و ۰/۹۳۵۰ است. الگوریتم تقویت سازوار CART و سپس رگرسیون لوژستیک دارای ضعیف‌ترین کارایی براساس معیارهای ارزیابی هستند. الگوریتم رگرسیون لوژستیک نسبت به بقیه الگوریتم‌ها به مدت‌زمان بیشتری برای اجرا نیاز دارد.

جدول ۴. معیارهای ارزیابی الگوریتم‌ها برای مجموعه داده تصاویر ماهواره‌ای

معیار	رگرسیون لوژستیک	CART	*Bagging CART*	سازوار تقویت *CART*	جنگل تصادفی
دقت	۰/۷۵۴۳	۰/۷۷۶۵	۰/۸۲۸۴	۰/۵۵۹۱	۰/۸۳۸۷
حساسیت	۰/۸۸۰۵	۰/۸۸۹۰	۰/۹۲۶۲	۰/۶۶۶۷	۰/۹۳۷۶
صحت	۰/۸۷۴۶	۰/۸۹۲۶	۰/۹۲۳۵	۰/۷۹۹۸	۰/۹۳۵۰
مدت‌زمان اجرا(ثانیه)	۱۸۱/۳۳۱۹	۹/۸۸۳۵	۱۴/۰۷۵۸	۱۱۶/۶۱۲۸	۸/۷۱۸۸

با توجه به جدول ۵، برای مجموعه داده بازی شطرنج الگوریتم CART دارای بیشترین دقت، حساسیت و صحت به ترتیب با مقدارهای ۰/۹۴۹۳، ۰/۹۸۲۱ و ۰/۹۸۲۲ است. الگوریتم‌های جنگل تصادفی و تقویت سازوار CART دارای کارایی نزدیک (رقابت‌پذیر) به الگوریتم CART هستند. الگوریتم Bagging CART و سپس رگرسیون لوژستیک دارای ضعیف‌ترین کارایی هستند. رگرسیون لوژستیک نسبت به بقیه الگوریتم‌ها به مدت‌زمان بیشتری برای اجرا نیاز دارد.

جدول ۵. معیارهای ارزیابی الگوریتم‌ها برای مجموعه داده بازی شطرنج

معیار	رگرسیون لوژستیک	CART	*Bagging CART*	سازوار تقویت *CART*	جنگل تصادفی
دقت	۰/۸۸۱۴	۰/۹۴۹۳	۰/۷۶۶۰	۰/۹۰۲۱	۰/۸۹۴۲
حساسیت	۰/۹۵۶۱	۰/۹۸۲۱	۰/۹۱۹۵	۰/۹۶۲۰	۰/۹۷۹۵
صحت	۰/۹۵۶۳	۰/۹۸۲۲	۰/۹۳۸۴	۰/۹۶۳۷	۰/۹۷۵۶
مدت‌زمان اجرا(ثانیه)	۴/۱۶۹۰	۰/۳۳۰۶	۱/۱۷۶۳	۲۴/۱۴۹۰	۰/۸۶۳۵

با توجه به جدول ۶، برای مجموعه داده ارزیابی اتومبیل الگوریتم CART دارای بیشترین مقدار دقت، حساسیت و صحت به ترتیب با مقدارهای ۰/۷۴۱۷، ۰/۷۸۰۶ و ۰/۸۲۵۹ است. همچنین الگوریتم جنگل تصادفی دارایی کارایی نزدیک (رقابت‌پذیر) به الگوریتم CART است. برای این مجموعه داده، الگوریتم تقویت سازوار CART و سپس رگرسیون لوژستیک دارایی ضعیف‌ترین کارایی هستند. الگوریتم تقویت سازوار CART نسبت به بقیه الگوریتم‌ها به مدت‌زمان بیشتری برای اجرا نیاز دارد.

جدول ۶. معیارهای ارزیابی الگوریتم‌ها برای مجموعه داده ارزیابی اتومبیل

معیار	رگرسیون لوژستیک	CART	*Bagging CART*	سازوار تقویت *CART*	جنگل تصادفی
دقت	۰/۶۹۱۹	۰/۷۴۱۷	۰/۷۰۰۰	۰/۲۴۳۷	۰/۷۳۳۰
حساسیت	۰/۷۳۳۸	۰/۷۸۰۶	۰/۷۰۰۲	۰/۲۸۴۰	۰/۷۷۵۴
صحت	۰/۷۱۳۲	۰/۸۲۵۹	۰/۴۹۰۸	۰/۳۲۵۳	۰/۸۱۸۹
مدت‌زمان اجرا(ثانیه)	۲/۰۶۹۳	۰/۰۹۰۱	۰/۳۹۱۱	۷/۵۶۴۰	۰/۴۷۰۲

مثال ۱.۶ (شبیه‌سازی).

مدل رگرسیون لوژستیک دودویی به‌صورت

$$\log\frac{\pi(\mathbf{X})}{1-\pi(\mathbf{X})} = 1 + 2X_1 + 3X_2$$

را در نظر بگیرید که در آن X_1 و X_2 متغیرهای تصادفی نرمال استاندارد هستند. از این مدل ۱۰۰۰ مشاهده تولید و الگوریتم‌های رگرسیون لوژستیک، CART، جنگل تصادفی، Bagging CART و تقویت سازوار CART روی داده‌ها اجرا می‌شود. نتایج کارایی الگوریتم‌ها براساس معیارهای دقت، صحت و حساسیت در جدول ۷ آورده شده است. از نتایج به‌دست‌آمده، مشاهده می‌شود باوجوداینکه داده‌ها از مدل رگرسیون لوژستیک تولیدشده است، اما الگوریتم‌های یادگیری ماشین به‌ویژه جنگل تصادفی و تقویت سازوار CART دارای کارایی نزدیک به رگرسیون لوژستیک می‌باشند.

جدول ۷. معیارهای ارزیابی الگوریتم‌ها برای مجموعه

داده شبیه‌سازی‌شده

معیار	رگرسیون لوژستیک	CART	Bagging CART	سازوار تقویت CART	جنگل تصادفی
دقت	۰/۸۶۱۰	۰/۸۰۸	۰/۷۷۰	۰/۸۴۲	۰/۸۳۰۰
حساسیت	۰/۸۸۶۲	۰/۸۳۸۳	۰/۷۵۲۲	۰/۸۸۰۸	۰/۸۵۰۷
صحت	۰/۸۸۱۷	۰/۸۴۰۰	۰/۷۶۴۳	۰/۸۵۶۳	۰/۸۶۱۷
مدت‌زمان اجرا(ثانیه)	۰/۱۲۹۱	۰/۰۶۳۳	۰/۷۴۷۳	۳/۶۹۶۳	۰/۴۱۲۸

با توجه به نتایج به‌دست‌آمده، از مقایسه مدل رگرسیون لوژستیک با الگوریتم‌های یادگیری ماشین با استفاده از داده‌های واقعی می‌توان گفت که الگوریتم‌های جنگل تصادفی و CART دارای عملکرد بهتر از رگرسیون لوژستیک هستند.

۷ بحث و نتیجه‌گیری

در این مقاله، در بحث رده‌بندی، مدل رگرسیون لوژستیک با الگوریتم‌های یادگیری ماشین CART، جنگل تصادفی، Bag-ging CART و تقویت سازوار CART مورد مقایسه قرار گرفت. برای انجام مقایسه، ۴ مجموعه داده واقعی و همچنین یک مثال شبیه‌سازی‌شده مورداستفاده قرار گرفت. براساس نتایج به‌دست‌آمده از معیارهای ارزیابی (دقت، حساسیت و صحت)، مشاهده گردید که برای هر مجموعه داده حداقل یک الگوریتم یادگیری ماشین وجود دارد که دارایی کارایی بهتر از مدل رگرسیون لوژستیک است. همچنین ضمن اینکه انتخاب بهترین الگوریتم بستگی به نوع داده از نظر تعداد مشاهدات، تعداد رده‌های متغیر پاسخ دارد، اما می‌توان گفت الگوریتم‌های جنگل تصادفی و CART دارای کارایی قابل‌قبول‌تر با توجه به انواع داده‌ها هستند.

مراجع

[1] Aggarwal, C. C. (2014). *Data classification: algorithms and applications*. CRC press, New York, USA.

[2] Agresti, A. (2003). *Categorical data analysis*. John Wiley and Sons, Florida, United States.

[3] Alpaydin, E. (2014). *Introduction to machine learning*. MIT press, London, England.

[4] Breiman, L., Friedman, J., Stone, C. J. and Olshen, R. A. (1984). *Classification and regression trees.* CRC press, London, New York ,Washington, D.C.

[5] Breiman, L. (1996). Bagging predictors. *Machine learning*, **24(2)**, 123-140.

[6] Breiman, L. (2001). Random forests. *Machine learning*, **45(1)**, 5-32.

[7] Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, **7(1)**, 1-26.

[8] Freund, Y., and R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boost-ing.*journal of Computer and System Sciences*, **55(1)**, 119-139.

[9] Kass, G. V. (1980). An exploratory technique for investigating large quantities of categorical data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **29(2)**, 119-127.

[10] Maimon, O. Z., Rokach, L. (2014). *Data mining with decision trees: theory and applications*. World scientific. Singa-pore.

[11] Schapire, R. E. (1990). The strength of weak learn ability. *Machine learning*, **5(2)**, 197-227.

[12] Soleimanpour, S. M., Mesbah, S. H., and Hedayati, B. (2018). Application of CART decision tree data mining to determine the most effective drinking water quality factors (case study: Kazeroon plain, Fars province). *Iranian Journal of Health and Environment*, **11**(1), 1-14.

[13] Syarif, I., Zaluska, E., Prugel-Bennett, A. and Wills, G. (2012) Application of bagging, boosting and stacking to intru-sion detection . In *International Workshop on Machine Learning and Data Mining in Pattern Recognition* (pp. 593-602). Springer, Berlin, Heidelberg.

[14] Taser, P. Y. (2021). Application of bagging and boosting approaches using decision tree-based algorithms in diabetes risk prediction. *In Multidisciplinary Digital Publishing Institute Proceedings* **74**, 6.

[15] Magidi, J., Nhamo, L., Mpandeli, S., and Mabhaudhi, T. (2021). Application of the random forest classifier to map irrigated areas using google earth engine. *Remote Sensing*, **13(5)**, 876.

[16] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Ithaca.

[17] You, J., van der Klein, S. A., Lou, E. and Zuidhof, M. J. (2020). Application of random forest classification to predict daily oviposition events in broiler breeders fed by precision feeding system. *Computers and Electronics in Agriculture*, **175**, 105526.

[18] Zhu, W., Zeng, N. and Wang, N. (2010). Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS implementations. *NESUG proceedings: health care and life sciences, Baltimore, Maryland*, **19**, 67.

پیوست

```
# code of computed accuracy of Logastic Regression for Social Network Ads data set
with a python 3.7 (anoconda 3) Software:
from pandas import read_csv
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
df=read_csv(r"C:Desktop\data\Social_Network_Ads.csv")
array=df.values
x=array[:,0:3]
y=arry[:,3]
kfold=KFold(n_splits=10, random_state=7)
model=LogisticRegression(max_iter=5000,solver='lbfgs')
results=cross_val_score(model, x, y, cv=10,scoring='accuracy')
results.mean()
-------------------------------------------------------------------------------

# code of computed accuracy of CART for Social Network Ads data set
from pandas import read_csv
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
a=read_csv(r"C:Desktop\data\Social_Network_Ads.csv")
array=a.values
x=array[:,0:3]
y=arry[:,3]
kfold=KFold(n_splits=10, random_state=7)
from sklearn.model_selection import cross_val_score
from sklearn.datasets import make_blobs
from sklearn.tree import DecisionTreeClassifier
cl1 = DecisionTreeClassifier(max_depth=None, min_samples_split=2, random_state=7)
results=cross_val_score(model, x, y, cv=10,scoring='accuracy')
results.mean()

# code of computed accuracy of Bagging CART for Social Network Ads data set
from pandas import read_csv
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import BaggingClassifier
df=read_csv(r"C:Desktop\data\Social_Network_Ads.csv")
array=df.values
```

```
x=array[:,0:3]
y=array[:,3]
kfold=KFold(n_splits=10, random_state=7)
bagging = BaggingClassifier( DecisionTreeClassifier(), max_samples=0.5, max_features=0.5)
results=cross_val_score(bagging, x, y, cv=10,scoring='accuracy')
results.mean()
-------------------------------------------------------------------------------

# code of computed accuracy of Adaboosting CART for Social Network Ads data set
from pandas import read_csv
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import AdaBoostClassifier
df=read_csv(r"C:Desktop\data\Social_Network_Ads.csv")
array=df.values
x=array[:,0:3]
y=arry[:,3]
kfold=KFold(n_splits=10, random_state=7)
boosting = AdaBoostClassifier(DecisionTreeClassifier(),n_estimators=100)
results=cross_val_score(boosing, x, y, cv=10,scoring='accuracy')
results.mean()
-------------------------------------------------------------------------------

# code of computed accuracy of Random Forest for Social Network Ads data set
from pandas import read_csv
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
a=read_csv(r"C:Desktop\data\Social_Network_Ads.csv")
array=a.values
x=array[:,0:3]
y=arry[:,3]
kfold=KFold(n_splits=10, random_state=7)
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import RandomForestClassifier
cl1 = RandomForestClassifier(n_estimators=10, max_depth=None,
min_samples_split=2, random_state=7)
results=cross_val_score(cl1, x, y, cv=10,scoring='accuracy')
results.mean()
-------------------------------------------------------------------------------
```

Comparison of logistic regression with some machine learning methods in classifying data

Tayebeh Karami[1], Muhyiddin Izadi[1], Mehrdad Niaparast[1]

Abstract:

One of the most important issues in various sciences is classification. Logistic regression is one of the statistical methods for data classification in which the distribution of data is supposed to be known. In addition to statistical methods, researchers are now using other methods to classify data such as machine learning algorithms that do not require the data distribution to be known. In this paper, logistic regression and some machine learning algorithms including CART, random forest, Bagging and Boosting of supervising learning are discussed. Using four real data sets and a simulation example, we compare the performance of the logistic regression with machine learning algorithms in terms of accuracy, sensitivity and precision measures.

Keywords: Decision Tree, Ensemble Learning, Random Forest and Supervised Learning.

[1]**Department of Statistics, Razi University**