


# الگوریتم جنگل تصادفی (Random Forest)

آرزو حبیبی راد

دانشگاه فردوسی مشهد



«جنگل تصادفی» (Random Forest)، یک الگوریتم یادگیری ماشین با قابلیت استفاده آسان است که اغلب اوقات نتایج بسیار خوبی را حتی بدون تنظیم فرایپارامترهای آن، فراهم می‌کند. این الگوریتم به دلیل سادگی و قابلیت استفاده، هم برای «دسته‌بندی» Classification و هم «رگرسیون»، یکی از پرکاربردترین الگوریتم‌های یادگیری ماشین محسوب می‌شود.

## تاریخچه جنگل تصادفی

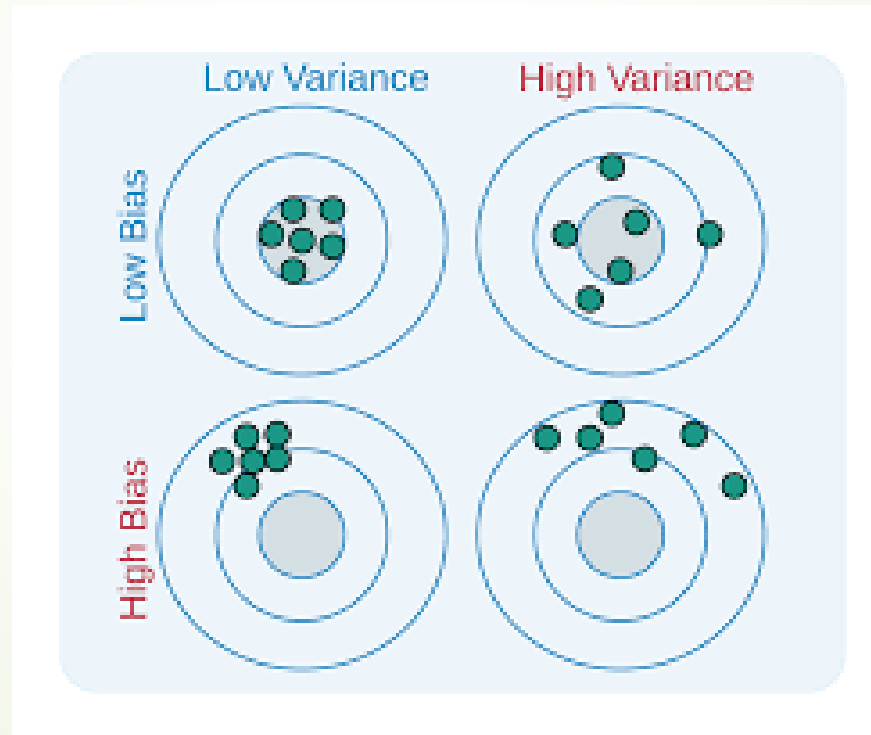
نخستین الگوریتم برای جنگل‌های تصمیم تصادفی را «تین کم هو» با بهره‌گیری از روش زیرفضاهای تصادفی پدید آورد. نسخه‌های بعدی آن توسط لیو بریمن ارتقا یافت. پژوهش‌های «بریمن» روی کار «امیت و گمن» اثر گذاشت، کسانی که پژوهش براساس دسته تصادفی که نود را تقسیم می‌کند (در مبحث بزرگ شدن تک درخت) ارائه کردند در این روش، پیش از این که هر درخت یا هر گره را جاسازی کنند، جنگلی از درختان بزرگ می‌شود و گزینش از بین گونه‌ای از درختان که برای گزینش تصادفی زیرفضاهایی از داده آموزش دیده‌اند، صورت می‌گیرد.

## تفاوت بین درخت تصمیم و جنگل تصادفی

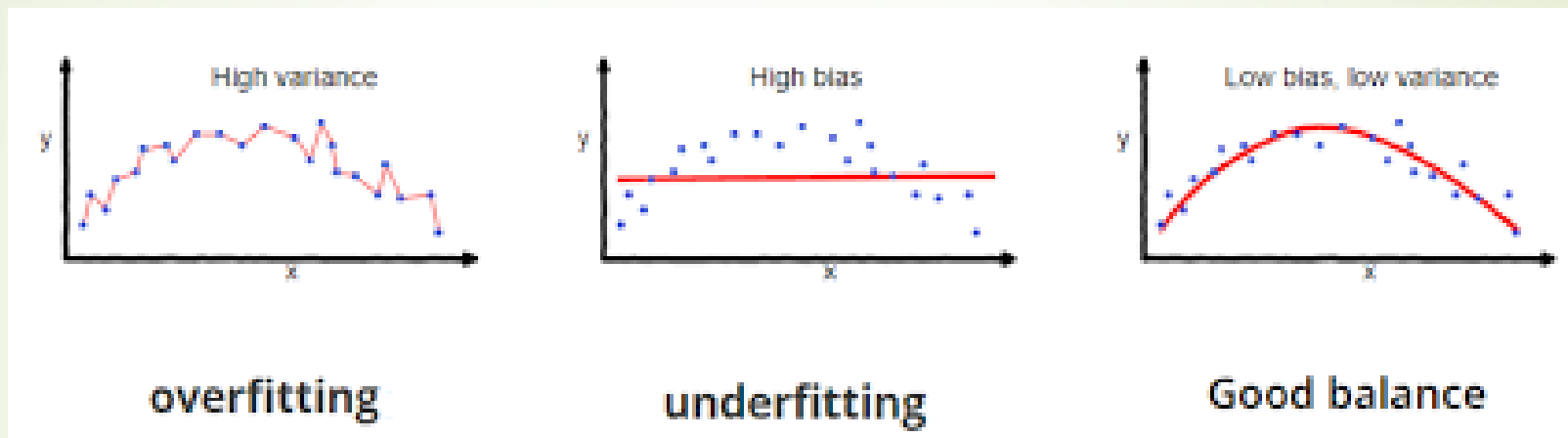
چنانکه پیش‌تر بیان شد، جنگل تصادفی مجموعه‌ای از درخت‌های تصمیم است. اما تفاوت‌هایی میان آن‌ها وجود دارد. اگر یک مجموعه داده ورودی با ویژگی‌ها و برچسب‌های آن به عنوان ورودی به الگوریتم داده شود، برخی از مجموعه قوانین را به گونه‌ای فرموله می‌کند که برای انجام پیش‌بینی مورد استفاده قرار می‌گیرند. برای مثال، اگر کاربر قصد داشته باشد پیش‌بینی کند که «آیا فرد روی یک تبلیغ آنلاین کلیک می‌کند یا نه»، می‌تواند تبلیغاتی که فرد در گذشته روی آن‌ها کلیک کرده و ویژگی‌هایی که تصمیمات او را توصیف می‌کنند گردآوری کند. سپس، با استفاده از آن‌ها می‌تواند پیش‌بینی کند که یک تبلیغ مشخص توسط یک فرد خاص کلیک می‌شود یا خیر.

یکی از روش‌ها برای به تصویر کشیدن داده‌ها، درخت تصمیم است.

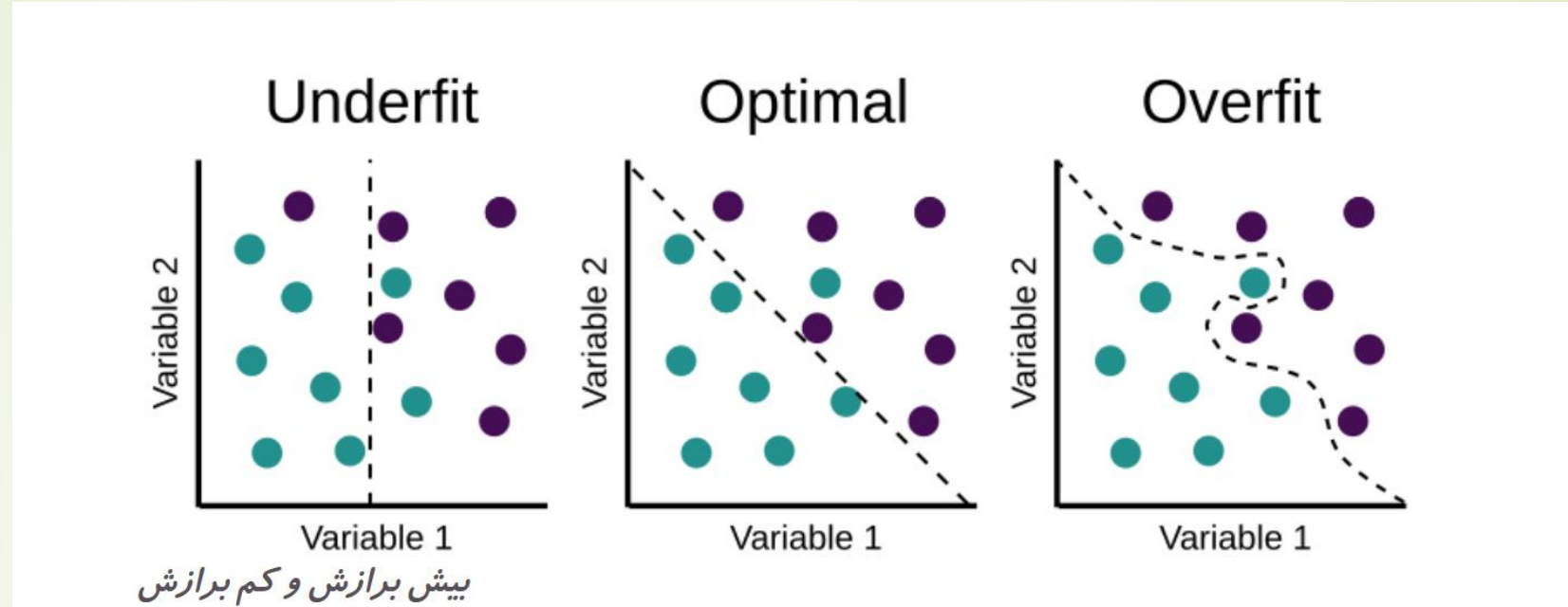
با افزایش تعداد ویژگی‌ها، تعداد شاخ و برگ درخت تصمیم بشدت افزایش پیدا می‌کند. در این روش، اریبی مدل بسیار پایین اما واریانس آن بسیار بالا است.



دردخت تصمیم ما از مدل **Overfitting** استفاده می‌کنیم.



در زمانی که داده ما از مدل غیرخطی تبعیت میکنند، اگر مدل خطی بر داده‌ها برازش کنیم حالت **underfitting** ایجاد می‌شود.

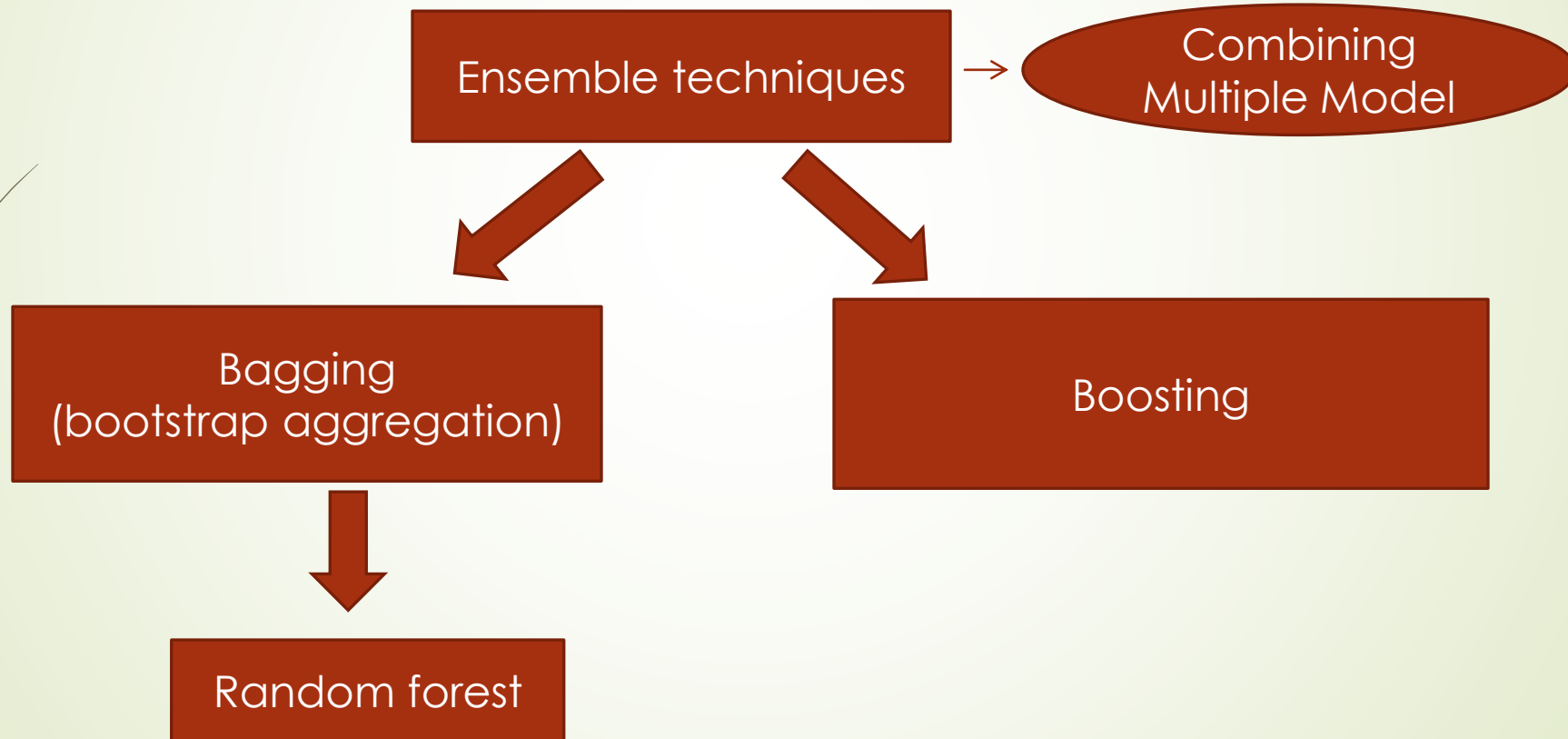


احتمال بیش برآزش به این دلیل وجود دارد که معیار برآزش مدل با معیاری که برای ارزیابی آن به کار می‌رود یکسان نیست. به این مفهوم که معمولاً برای برآزش مدل کارایی آن بر روی یک مجموعه نمونه‌های برآزش بیشینه می‌شود. در صورتی که برای سنجش مؤثر بودن مدل نه تنها کارایی آن بر روی نمونه‌های برآزش را می‌سنجند بلکه توانایی مدل بر روی نمونه‌هایی دیده نشده نیز در نظر گرفته می‌شود. بیش برآزش زمانی اتفاق می‌افتد که مدل در هنگام برآزش به جای "یادگیری" داده‌ها شروع به "حفظ کردن" آنها می‌کند.

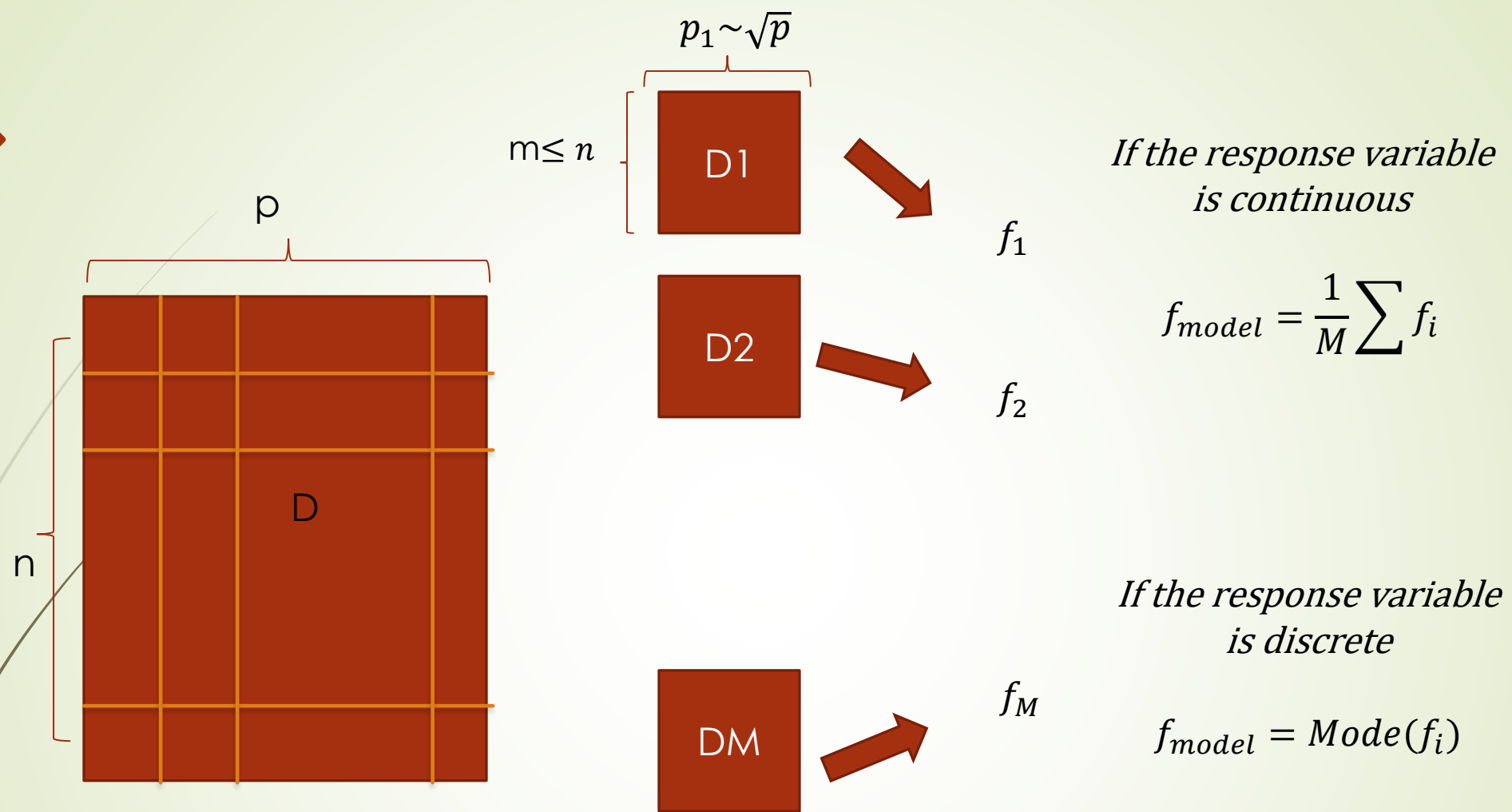
## جنگل تصادفی:

ترکیب چند مدل برای ایجاد یک پیش بینی دقیق (Ensembling)

روشی است برای کاهش واریانس در مدل درخت تصمیم

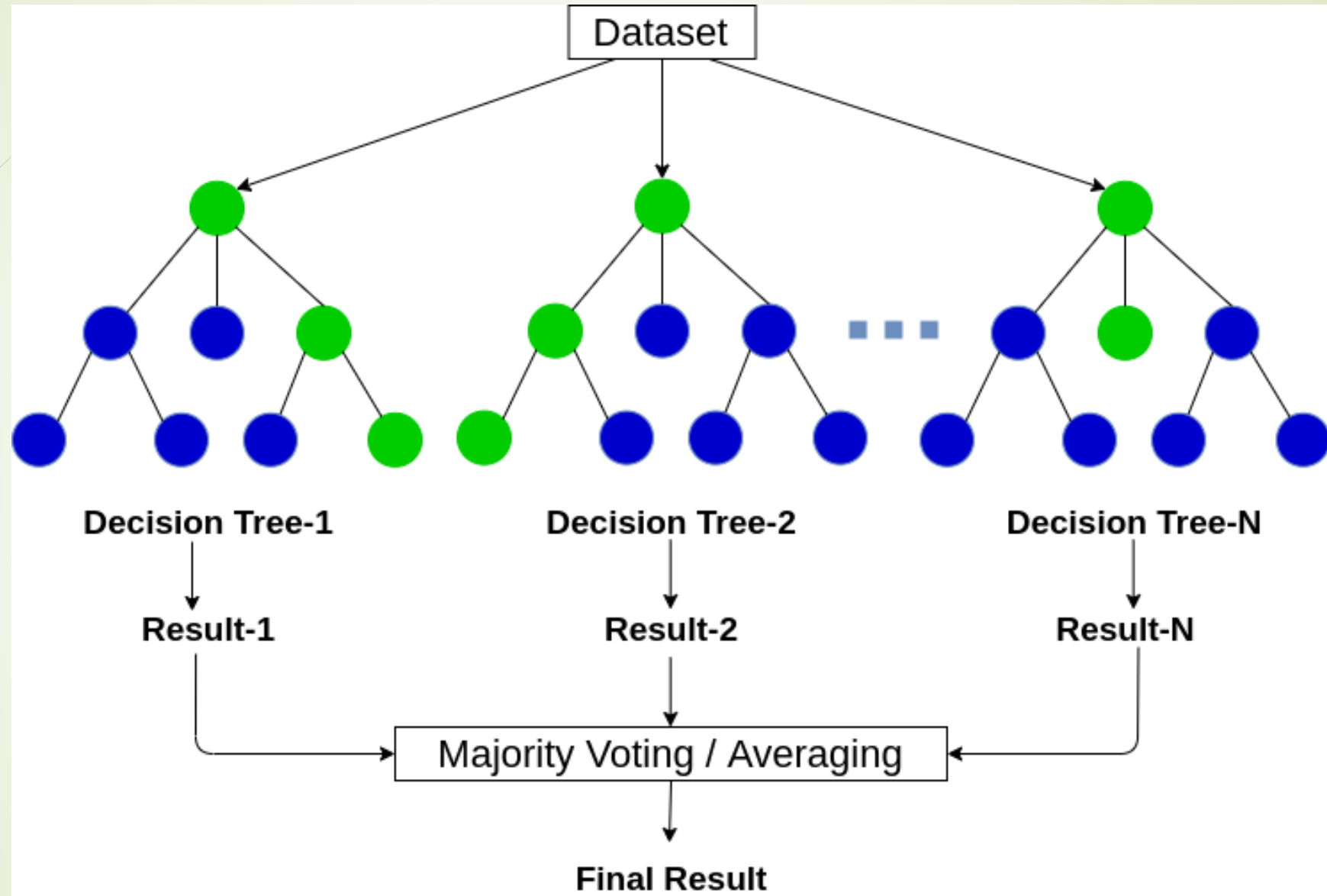


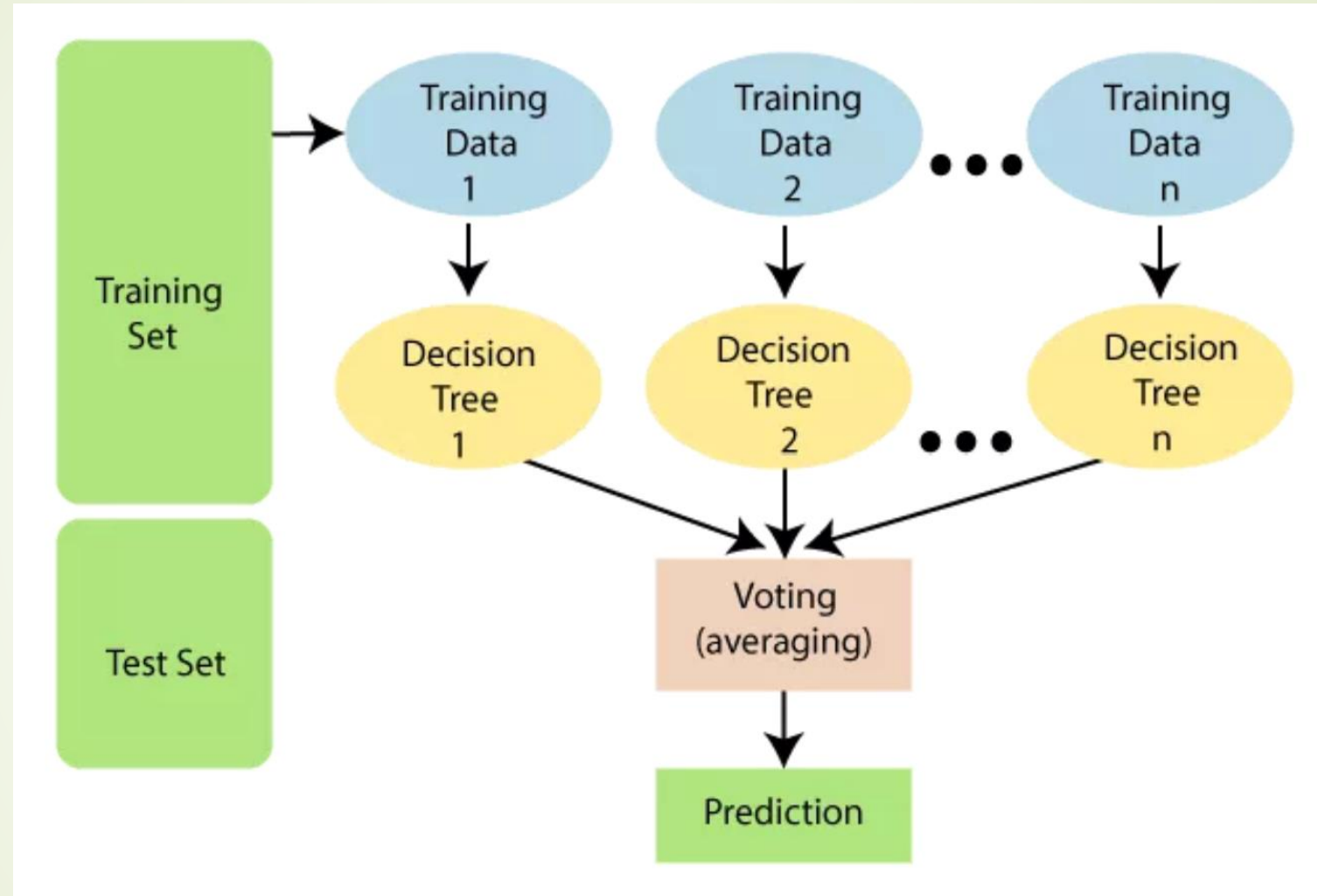


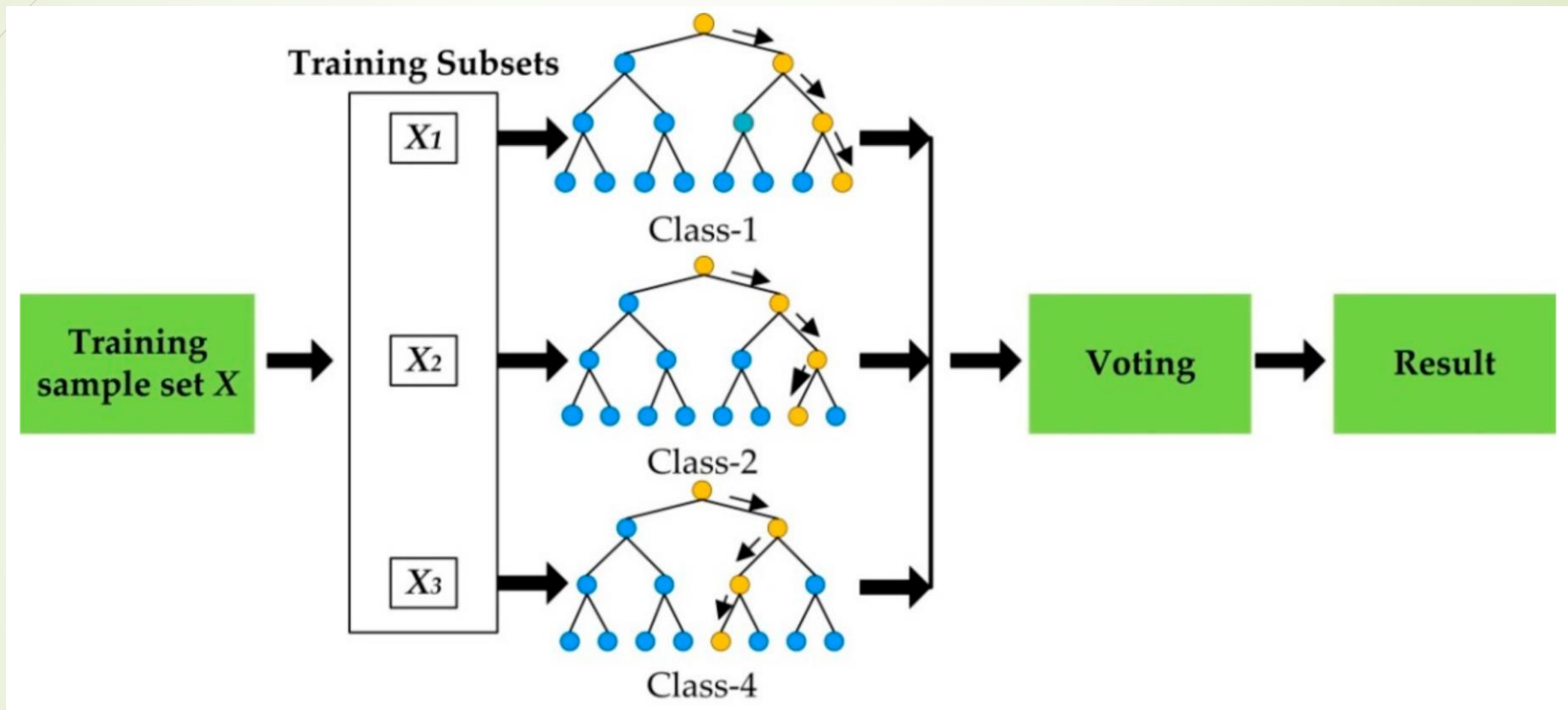


Bagging method → Random forest method







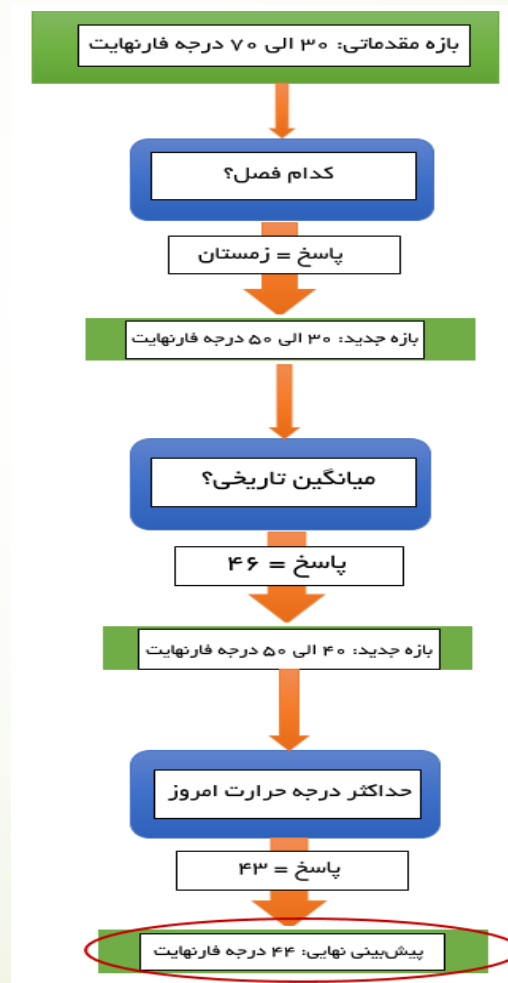


در این مثال، فصل زمستان است و بنابراین می‌توان رنج پیش‌بینی را بین ۵۰-۳۰ درجه محدود کرد، زیرا حداکثر درجه حرارت در شمال غربی اقیانوس آرام در طول زمستان در همین حدود است. اولین پرسش انتخاب خوبی بود، زیرا منجر شد که طیف عددی نصف شود. اگر پرسش نامرتبلی مانند «فردا کدام روز از هفته است؟» پرسیده می‌شد، امکان کاهش محدوده پیش‌بینی‌ها به طور کلی وجود نداشت و باز به نقطه اولی رسیده می‌شد که کار از آنجا آغاز شده است.


با این وجود، این سوال یکتا، برای محدود کردن بازه تخمین کافی نبوده و بنابراین نیاز به پرسیدن سوالات بیشتری هست. سوال خوب بعدی که می‌توان پرسید این است که «میانگین حداکثر درجه حرارت در این روز بر اساس اطلاعات تاریخی چقدر بوده؟». برای سیاتل در ۷ دسامبر، پاسخ ۴۶ درجه است. این کار امکان محدود کردن طیف به ۵۰-۴۰ را فراهم می‌کند. سوال مطرح شده، یک پرسش با ارزش به شمار می‌آید، زیرا توانسته دامنه تخمین را به شدت محدود کند.

اما همچنان دو سوال برای انجام پیش‌بینی کافی نیست، زیرا امسال امکان دارد نسبت به میانگین سال‌های پیشین گرم‌تر یا سردتر باشد. بنابراین، نگاهی به حداکثر درجه حرارت امروز انداخته می‌شود تا مشخص شود که آیا امسال هوا نسبت به سال‌های پیش سردتر است یا گرم‌تر. اگر درجه حرارت امروز ۴۳ درجه باشد، نسبت به سال گذشته اندکی سردتر بوده و این یعنی فردا نیز امکان دارد درجه حرارت اندکی از میانگین تاریخی کمتر باشد.

بنابراین، برای رسیدن به تخمین مناسب، از یک مجموعه پرسش استفاده می‌شود و هر پرسش دامنه مقادیر پاسخ ممکن را محدودتر می‌کند تا اطمینان لازم برای انجام پیش‌بینی فراهم شود. این فرآیند تصمیم‌گیری هر روز در زندگی انسان‌ها تکرار می‌شود و تنها پرسش‌ها هستند که متناسب با نوع مساله از فردی به فردی دیگر تغییر می‌کنند. در این لحظه، آمادگی لازم برای ایجاد ارتباط با درخت تصمیم تقریباً وجود دارد، اما بهتر است دقایقی زمان برای بررسی ارائه گرافیکی گام‌های برداشته شده به منظور انجام پیش‌بینی اختصاص داده شود.



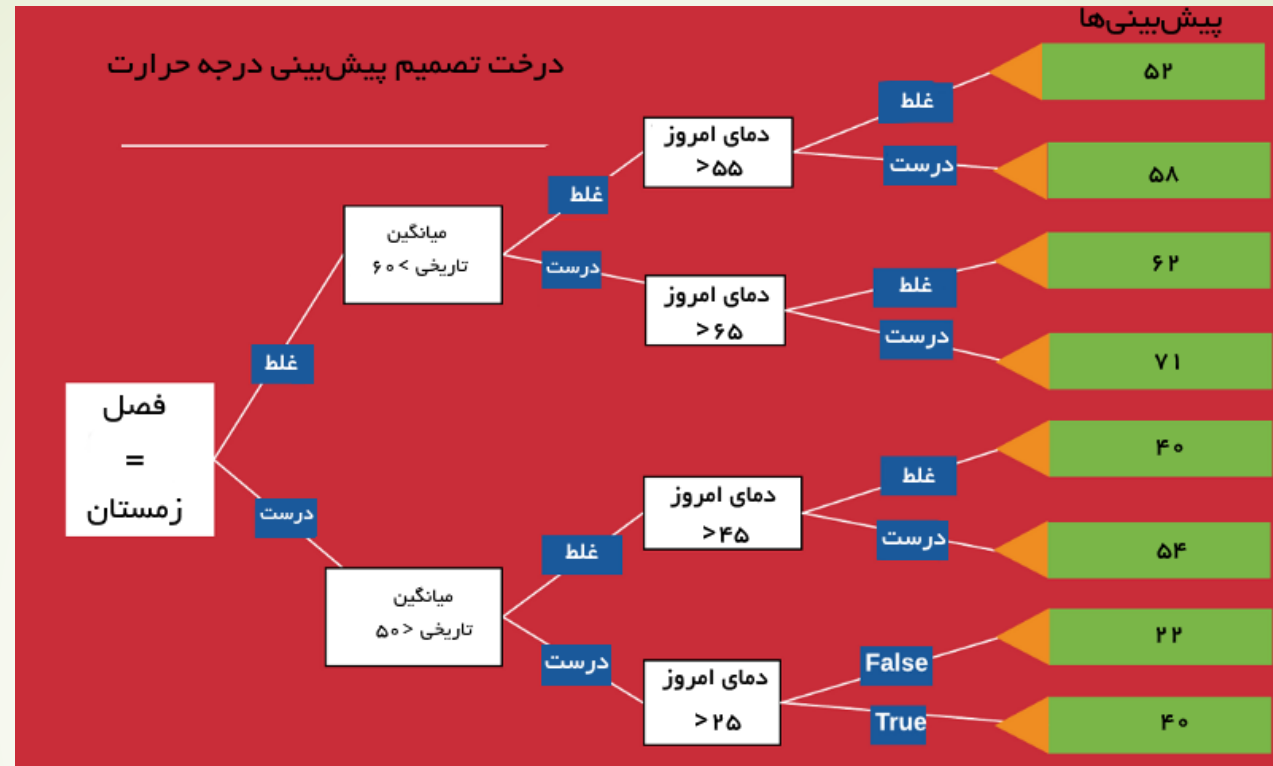




همانطور که مشاهده شد، در یک درخت تصمیم، کار با یک حدس اولیه بر مبنای دانش فرد از جهان آغاز و با دریافت اطلاعات بیشتر، این حدس پالایش می‌شود. طی این فرایند و به تدریج، به گردآوری داده‌ها پایان داده و تصمیمی اتخاذ می‌شود که در اینجا پیش‌بینی بیشینه درجه حرارت است. رویکرد طبیعی که انسان برای حل چنین مساله‌ای مورد استفاده قرار می‌دهد مطابق تصویر ارائه شده در بالا است و می‌توان به آن «روندنمای (Flowchart)» پرسش و پاسخ گفت. در حقیقت، این فلوچارت یک مدل مقدماتی از درخت تصمیم است. اگرچه، در اینجا یک درخت تصمیم کامل ساخته نشده زیرا انسان‌ها از میانبرهایی استفاده می‌کنند که برای ماشین فاقد معنا محسوب می‌شود.


دو تفاوت اصلی میان فرآیند تصمیم‌گیری به تصویر کشیده شده در اینجا و یک درخت تصمیم واقعی وجود دارد. اولاً، در اینجا از لیست کردن شاخه‌های جایگزین غفلت شده و این یعنی از پیش‌بینی‌هایی که در صورت متفاوت بودن پاسخ سوالات باید انجام می‌شدند، صرف‌نظر شده است. برای مثال، اگر فصل به جای زمستان، تابستان بود، بازه پیش‌بینی به مقادیر بیشتری تغییر می‌کرد. علاوه بر آن، پرسش‌ها به شیوه‌ای مطرح شده‌اند که می‌توانند هر تعدادی پاسخ داشته باشند.

هنگامی که پرسیده می‌شود «حداکثر درجه حرارت امروز چقدر است؟» پاسخ می‌تواند هر مقدار حقیقی باشد. به طور بالعکس، درخت تصمیم پیاده‌سازی شده در یادگیری ماشین، همه جایگزین‌های احتمالی برای هر پرسش را لیست کرده و همه سوالات را به شکل «درست» (True) / «غلط» (False) می‌پرسد. درک این موضوع کمی دشوار است، زیرا شیوه فکر کردن انسان‌ها در حالت طبیعی اینچنین نیست و بنابراین، بهترین راه برای نمایش این تفاوت ساخت یک درخت تصمیم واقعی از فرآیند پیش‌بینی است.




در تصویر بالا، می‌توان مشاهده کرد که هر پرسش (بلوک‌های سفید) تنها دارای دو پاسخ است: درست یا غلط. علاوه بر آن، برای هر پاسخ درست و غلط انشعاب‌های جدایی وجود دارد. صرف نظر از پاسخ به پرسش‌ها، به تدریج به یک پیش‌بینی دست یافته می‌شود. این نسخه «کامپیوترپسند» از درخت تصمیم ممکن است از روش حسی انسان متفاوت باشد، اما دقیقاً به همان شکل کار می‌کند. در واقع، با شروع از گره سمت چپ و از طریق پاسخگویی به پرسش‌های درخت در طول مسیر، پیش‌روی صورت می‌پذیرد.





در مثالی که در مطلب پیش رو بیان شده، فصل زمستان است، بنابراین برای اولین پرسش شاخه «True» انتخاب می‌شود. همانطور که پیش‌تر بیان شد، میانگین درجه حرارت بر اساس اطلاعات تاریخی ۴۶ بوده، بنابراین برای پرسش دوم نیز پاسخ «درست» گزینش می‌شود. در نهایت، پاسخ سوم نیز «درست» است، زیرا بیشینه درجه حرارت امروز ۴۳ درجه بوده. پیش‌بینی نهایی برای بیشینه درجه حرارت فردا برابر با ۴۰ است که مقدار نزدیکی به حدس زده شده در بالا یعنی ۴۴ درجه دارد.

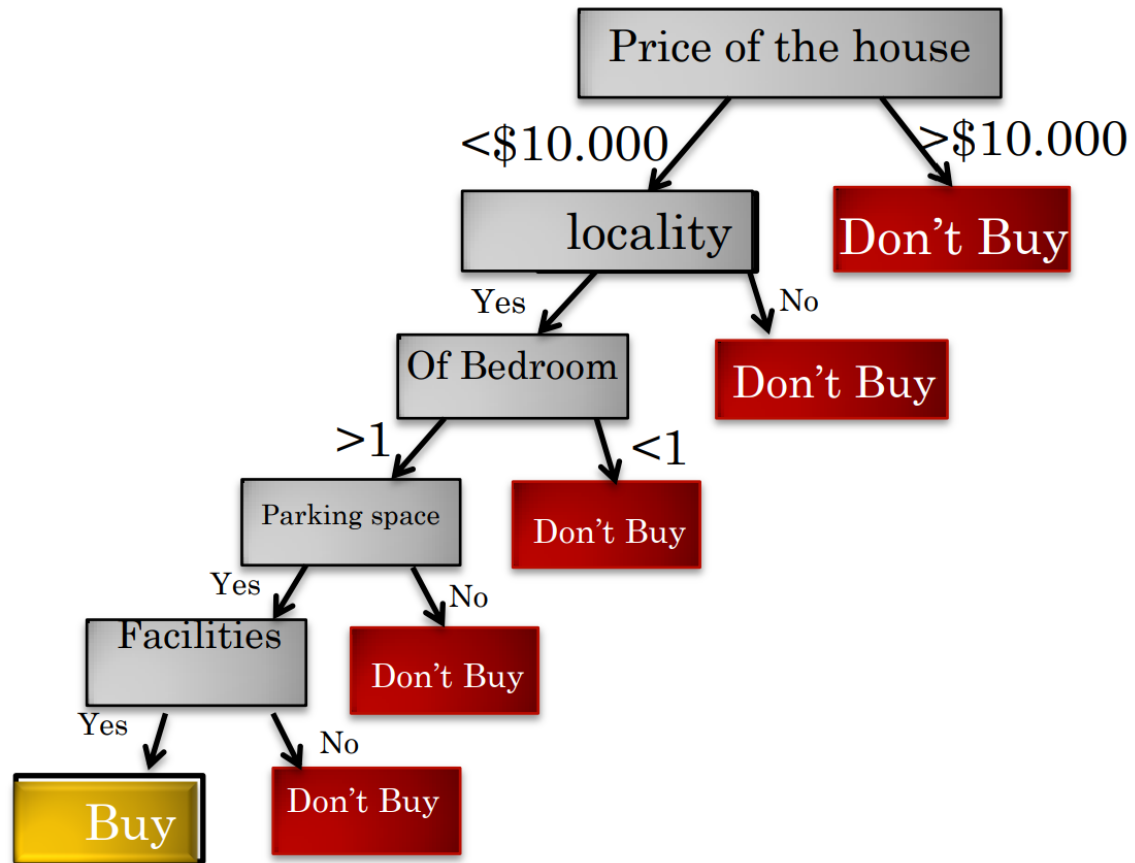
این مدل، شامل همه ویژگی‌های کیفی پایه‌ای یک درخت تصمیم می‌شود. در این مطلب، تعمداً از بیان جزئیات فنی مربوط به الگوریتم مانند اینکه پرسش‌ها چگونه شکل داده و آستانه چطور تنظیم می‌شود صرف‌نظر شده، و البته این موارد واقعا برای درک مفهومی مدل یا حتی پیاده‌سازی با کد پایتون نیاز نیستند. یک جنبه از درخت تصمیم که در اینجا باید به آن پرداخته شود، چگونگی «یادگیری» (Learning) «این مدل است. در اینجا، بازه تخمین زده شده بر پایه پاسخ هر پرسش پالایش می‌شود. اگر فصل زمستان است، مقدار تخمینی کمتر از زمانی خواهد بود که فصل تابستان باشد.



اگرچه، یک مدل کامپیوتری درخت تصمیم، هیچ دانش پیشینی ندارد و هرگز قادر به برقراری ارتباط میان «winter = colder» به خودی خود نیست. مدل باید همه چیز را پیرامون مساله بر اساس داده‌هایی که برای آن فراهم شده بیاموزد. انسان‌ها با توجه به تجربیات روزانه خود، می‌دانند که چگونه پاسخ‌ها را از یک «روندنا (flow-chart)» به یک پیش‌بینی معقول تبدیل کنند. در حالیکه، مدل باید هر یک از این روابط را بیاموزد، مثلاً اینکه اگر امروز هوا گرم‌تر از میانگین تاریخی است، این احتمال وجود دارد که فردا نیز حداکثر درجه حرارات بیشتر از میانگین سال گذشته باشد.

جنگل تصادفی، به عنوان یک مدل یادگیری ماشین نظارت شده، یاد می‌گیرد که در فاز آموزش (training) یا برآزش مدل، داده‌ها را (درجه حرارت امروز، میانگین تاریخی و دیگر موارد) به خروجی‌ها (حداکثر درجه حرارت فردا) نگاشت کند. در طول آموزش، داده‌های تاریخی به مدل داده می‌شوند که مرتبط با دامنه مساله هستند (درجه حرارت روز قبل، فصل سال و میانگین تاریخی) و مقدار صحیحی (که در این مثال بیشینه درجه حرارت فردا است) که مدل باید بیاموزد تا بتواند پیش‌بینی کند. مدل، روابط میان داده‌ها را (که با عنوان ویژگی‌ها در یادگیری ماشین شناخته شده‌اند) و مقادیری که کاربر می‌خواهد آن‌ها را پیش‌بینی کند (به آن‌ها هدف گفته می‌شود) می‌آموزد.

## ❖ تفاوت بین جنگل تصادفی و درخت تصمیم گیری



معیار و پارامترهای شما جهت خرید خانه :

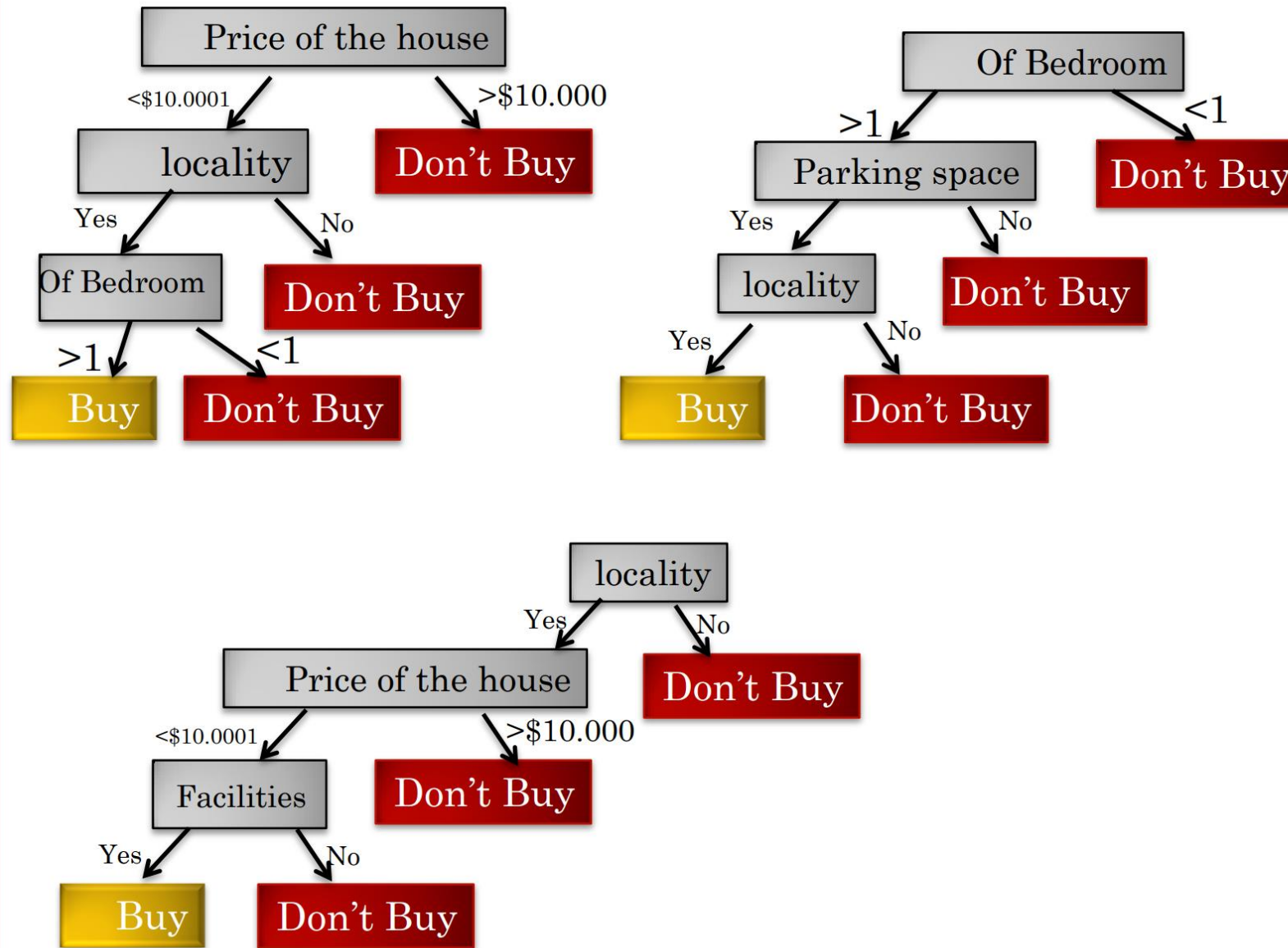
- قیمت خانه

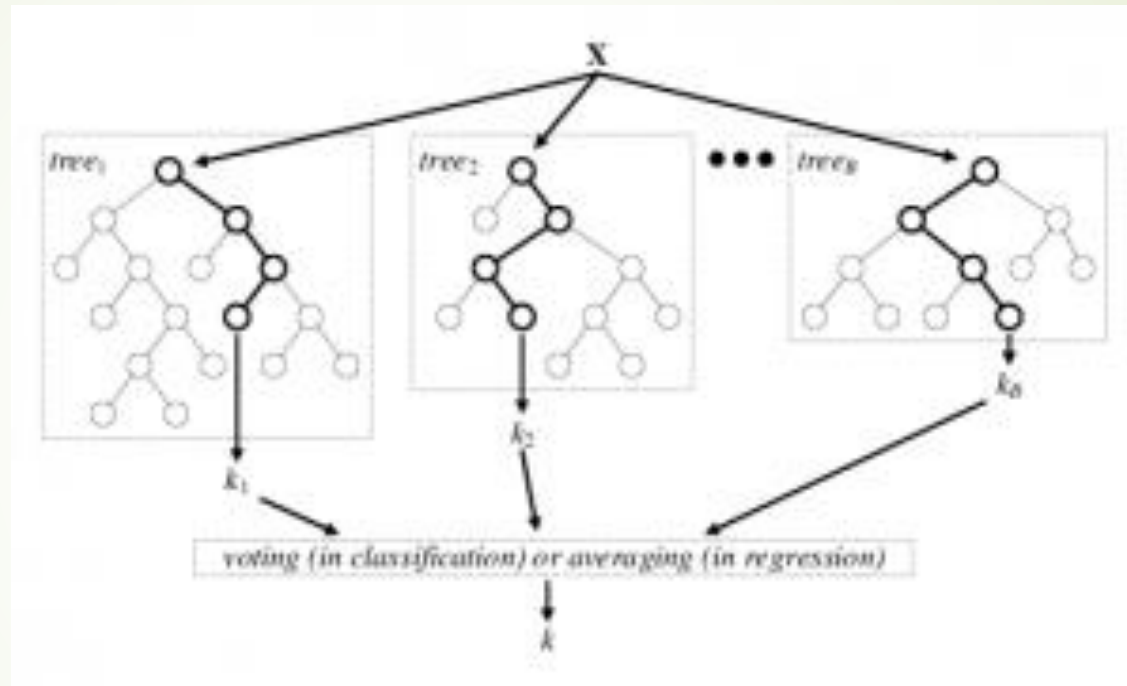
- محل

- تعداد اتاق خواب ها

- فضای پارکینگ

- امکانات موجود








## چگونگی عملکرد جنگل تصادفی

جنگل تصادفی یک الگوریتم یادگیری نظارت شده محسوب می‌شود. همانطور که از نام آن مشهود است، این الگوریتم جنگلی را به طور تصادفی می‌سازد. «جنگل» ساخته شده، در واقع گروهی از «درخت‌های تصمیم (Decision Trees) است. کار ساخت جنگل با استفاده از درخت‌ها اغلب اوقات به روش «کیسه‌گذاری (Bagging)» انجام می‌شود. ایده اصلی روش کیسه‌گذاری آن است که ترکیبی از مدل‌های یادگیری، نتایج کلی مدل را افزایش می‌دهد. به بیان ساده، جنگل تصادفی چندین درخت تصمیم ساخته و آن‌ها را با یکدیگر ادغام می‌کند تا پیش‌بینی‌های صحیح‌تر و پایدارتری حاصل شوند.

یکی از مزایای جنگل تصادفی قابل استفاده بودن آن، هم برای مسائل دسته‌بندی و هم رگرسیون است که غالب سیستم‌های یادگیری ماشین کنونی را تشکیل می‌دهند. در اینجا، عملکرد جنگل تصادفی برای انجام «دسته‌بندی» (Classification) تشریح خواهد شد، زیرا گاهی دسته‌بندی را به عنوان بلوک سازنده یادگیری ماشین در نظر می‌گیرند. در تصویر زیر، می‌توان دو جنگل تصادفی ساخته شده از دو درخت را مشاهده کرد.



جنگل تصادفی دارای فرایرامترهایی مشابه درخت تصمیم یا «دسته‌بند کیسه‌گذاری (Bagging Classifier)» است. خوشبختانه، نیازی به ترکیب یک درخت تصمیم با یک دسته‌بند کیسه‌گذاری نیست و می‌توان از «کلاس دسته‌بندی (Classifier-Class)» جنگل تصادفی استفاده کرد. چنانکه پیش‌تر بیان شد، با جنگل تصادفی، و در واقع «رگرسور جنگل تصادفی (Random Forest Regressor)» می‌توان به حل مسائل رگرسیون نیز پرداخت.

جنگل تصادفی، تصادفی بودن افزوده‌ای را ضمن رشد درختان به مدل اضافه می‌کند. این الگوریتم، به جای جست‌وجو به دنبال مهم‌ترین ویژگی‌ها هنگام تقسیم کردن یک «گره (Node)»، به دنبال بهترین ویژگی‌ها در میان مجموعه تصادفی از ویژگی‌ها می‌گردد. این امر منجر به تنوع زیاد و در نهایت مدل بهتر می‌شود. بنابراین، در جنگل تصادفی، تنها یک زیر مجموعه از ویژگی‌ها توسط الگوریتم برای تقسیم یک گره در نظر گرفته می‌شود. با استفاده افزوده از آستانه تصادفی برای هر ویژگی به جای جست‌وجو برای بهترین آستانه ممکن، حتی می‌توان درخت‌ها را تصادفی‌تر نیز کرد (مانند کاری که درخت تصمیم نرمال انجام می‌دهد).



## مثال جهان واقعی از جنگل تصادفی

فرض می‌شود پسری به نام «آندرو» (Andrew) «»، می‌خواهد تصمیم‌گیری کند که برای یک سفر تفریحی یکساله به کدام مکان‌ها سفر کند. او از مردمی که او را می‌شناسند درخواست می‌کند که پیشنهادات خود را بگویند. از این رو، ابتدا به سراغ دوست قدیمی خود می‌رود، دوست اندرو از او می‌پرسد که در گذشته به کجا سفر کرده و آیا آن مکان را دوست داشته یا خیر. در نهایت بر اساس این پرسش و پاسخ، چند مکان را به اندرو پیشنهاد می‌دهد. این رویکرد متداولی است که الگوریتم درخت تصمیم دنبال می‌کند.

دوست اندرو با استفاده از پاسخ‌های او، قوانینی را برای هدایت تصمیم خود مبنی بر اینکه چه چیزی را پیشنهاد کند می‌سازد. پس از آن، اندرو شروع به پرسیدن سوالات بیشتر و بیشتری از دوست خود برای دریافت پیشنهاد از او می‌کند، بنابراین دوست اندرو نیز پرسش‌های متفاوتی را می‌پرسد که می‌تواند بر اساس آن‌ها توصیه‌هایی را به او ارائه کند. در نهایت، اندرو مکان‌هایی که بیشتر به او توصیه شده‌اند را انتخاب می‌کند. این رویکرد کلی الگوریتم جنگل تصادفی است.

### اهمیت ویژگی‌ها

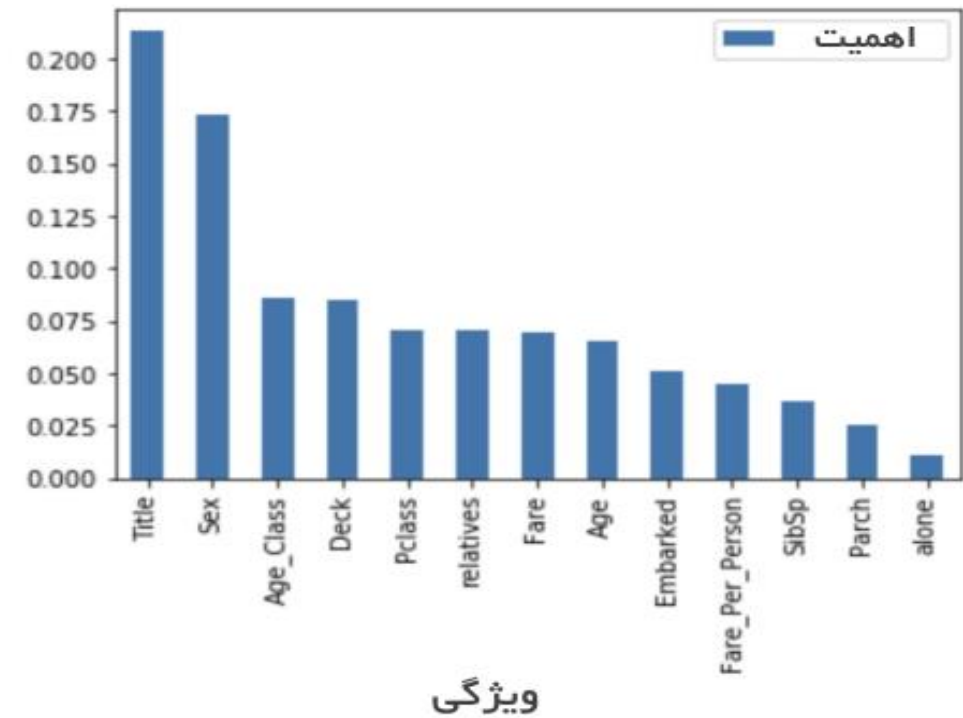
دیگر خصوصیت عالی الگوریتم جنگل تصادفی این است که اندازه‌گیری اهمیت نسبی هر ویژگی روی پیش‌بینی در آن آسان است. کتابخانه پایتون «سایکیت لرن» (Sklearn) «ابزار خوبی را برای این کار فراهم می‌کند. این ابزار، اهمیت یک ویژگی را با نگاه کردن به تعداد گره‌های درخت که از آن ویژگی استفاده می‌کنند، اندازه‌گیری کرده و ناخالصی را در سرتاسر درخت‌های جنگل کاهش می‌دهد. ابزار مذکور، این امتیاز را به صورت خودکار برای هر ویژگی پس از آموزش دادن محاسبه و نتایج را مقیاس می‌کند، بنابراین مجموع همه اهمیت‌ها برابر با ۱ است.

در ادامه توضیحی کوتاه پیرامون چستی «گره (Node)» و «برگ (Leaf)» به منظور یادآوری ارائه می‌شود:

در درخت تصمیم، هر گره داخلی یک «تست» را روی ویژگی‌ها نمایش می‌دهد (مثلاً، سکه در یک پرتاب شیر می‌آید یا خط)، هر شاخه نشانگر خروجی تست و هر گره برگ نشانگر برچسب دسته (کلاس) است (تصمیم پس از محاسبه همه این گره‌ها آن‌ها اتخاذ می‌شود). گره‌ای که هیچ فرزندی ندارد، «برگ (Leaf)» محسوب می‌شود.

از طریق بررسی اهمیت ویژگی‌ها، کاربر می‌تواند تصمیم بگیرد که کدام ویژگی‌ها را امکان دارد بخواهد با توجه به اینکه به طور کلی و یا به اندازه کافی در فرآیند تصمیم‌گیری نقش ندارند، حذف کند. این مساله حائز اهمیت است زیرا، یک قانون کلیدی در یادگیری ماشین آن است که هرچه ویژگی‌ها بیشتر باشند، احتمال آنکه مدل دچار «بیش‌برازش» (OverFitting) یا «کم‌برازش» (UnderFitting) شود وجود دارد. در ادامه می‌توان جدول و نموداری را مشاهده کرد که اهمیت ۱۳ ویژگی را که در طول یک پروژه دسته‌بندی (نظارت شده) مورد استفاده قرار گرفته‌اند و متعلق به مجموعه داده معروف Titanic (+) هستند که در سایت kaggle موجود است، مشاهده کرد.

ویژگی	اهمیت
Title	0.213
Sex	0.173
Age_Class	0.086
Deck	0.085
Pclass	0.071
relatives	0.070
Fare	0.069
Age	0.065
Embarked	0.051
Fare_Per_Person	0.045
SibSp	0.037
Parch	0.025
alone	0.011



## تفاوت بین درخت تصمیم و جنگل تصادفی

چنانکه پیش‌تر بیان شد، جنگل تصادفی مجموعه‌ای از درخت‌های تصمیم است. اما تفاوت‌هایی میان آن‌ها وجود دارد. اگر یک مجموعه داده ورودی با ویژگی‌ها و برچسب‌های آن به عنوان ورودی به الگوریتم داده شود، برخی از مجموعه قوانین را به گونه‌ای فرموله می‌کند که برای انجام پیش‌بینی مورد استفاده قرار می‌گیرند. برای مثال، اگر کاربر قصد داشته باشد پیش‌بینی کند که «آیا فرد روی یک تبلیغ آنلاین کلیک می‌کند یا نه»، می‌تواند تبلیغاتی که فرد در گذشته روی آن‌ها کلیک کرده و ویژگی‌هایی که تصمیمات او را توصیف می‌کنند گردآوری کند. سپس، با استفاده از آن‌ها می‌تواند پیش‌بینی کند که یک تبلیغ مشخص توسط یک فرد خاص کلیک می‌شود یا خیر.

در مقایسه، الگوریتم درخت تصمیم مشاهدات را به صورت تصادفی انتخاب می‌کند، برای ویژگی‌های ساخت چندین درخت تصمیم می‌گیرد و سپس از محاسبه میانگین نتایج استفاده می‌کند. تفاوت دیگر آن است که درخت تصمیم «عمیق (Deep)» ممکن است دچار «بیش‌برازش (Overfitting)» شود. جنگل تصادفی اغلب اوقات با ساخت زیردرخت تصادفی از ویژگی‌ها و ساخت درخت کوچک‌تر با استفاده از این زیردرخت، از بیش‌برازش جلوگیری می‌کند. پس از آن، زیردرخت‌های تصادفی را با یکدیگر ترکیب می‌کند. شایان توجه است که این راهکار همیشه جوابگو نیست و فرآیند محاسبات را بسته به تعداد جنگل‌های تصادفی که ساخته می‌شوند کندتر می‌کند.

تعریف: یک جنگل تصادفی یک روش طبقه‌بندی متشکل از مجموعه‌ای از درخت‌های تصمیم است به طوریکه

$$\{h(x, \Theta_k), k = 1, \dots\}$$

که در آن  $\{\Theta_k\}$  بردار تصادفی iid بوده و هر درخت یک رای برای کلاس مورد انتخاب از ورودی  $x$  را دریافت می‌کند.



هایپرپارامترها در جنگل تصادفی برای افزایش قدرت پیش‌بینی مدل و یا سریع‌تر کردن آن مورد استفاده قرار می‌گیرند. در ادامه، پیرامون هایپرپارامترهای تابع جنگل تصادفی توکار کتابخانه `sklearn` صحبت می‌شود.

### ۱. افزایش قدرت پیش‌بینی

اولاً، یک هایپرپارامتر «`n_estimators`» وجود دارد که در واقع تعداد درختانی است که الگوریتم پیش از دریافت آرای بیشینه یا دریافت میانگین پیش‌بینی‌ها می‌سازد. به طور کلی، تعداد بیشتر درخت‌ها، کارایی را افزایش می‌دهند و پیش‌بینی‌ها را پایدارتر می‌سازند، اما محاسبات را کندتر می‌کنند. دیگر هایپرپارامتر مهمی که پیرامون آن صحبت خواهد شد، «`min_sample_leaf`» است. این هایپرپارامتر همانطور که از نام آن مشخص است، حداقل تعداد برگ‌هایی که برای تقسیم یک نود خارجی مورد نیاز هستند را مشخص می‌کند.

### ۲. افزایش سرعت مدل

هایپرپارامتر «`n_jobs`» به موتور می‌گوید که اجازه استفاده از چه تعداد پردازنده را دارد. اگر مقدار این هایپرپارامتر برابر با ۱ باشد، می‌تواند تنها از یک پردازنده استفاده کند. مقدار «۱» بدین معنا است که هیچ محدودیتی وجود ندارد «`random_state`». خروجی مدل را تکرارپذیر می‌کند. مدل هنگامی که مقدار قطعی برای `random_state` دارد و اگر هایپرپارامترها و داده‌های آموزش مشابهی به آن داده شود، همیشه نتایج مشابهی را تولید می‌کند.

در نهایت، یک هایپرپارامتر «`oob_score`» وجود دارد (به آن `oob sampling` نیز گفته می‌شود)، که روشی برای «اعتبارسنجی متقابل (Random Forest)»، جنگل تصادفی است. در این نمونه‌برداری، حدود یک سوم از داده‌ها برای آموزش مدل استفاده نمی‌شوند و برای ارزیابی کارایی آن مورد استفاده قرار می‌گیرند. به این نمونه‌ها «نمونه‌های کیسه (Bag Samples)» گفته می‌شود. این راهکار، شباهت زیادی به روش «اعتبارسنجی یک‌طرفه (leave-one-out)» دارد، اما تقریباً هیچ بار محاسباتی برای آن وجود ندارد.

## مزایا و معایب

همانطور که پیش از این بیان شد، یکی از مزایای جنگل تصادفی آن است که هم برای رگرسیون و هم برای دسته‌بندی قابل استفاده است و راهکاری مناسب برای مشاهده اهمیت نسبی که به ویژگی‌های ورودی تخصیص داده می‌شود است. جنگل تصادفی الگوریتمی بسیار مفید و با استفاده آسان محسوب می‌شود، زیرا هایپیرامترهای پیش‌فرض آن اغلب نتایج پیش‌بینی خوبی را تولید می‌کنند. همچنین، تعداد هایپیرامترهای آن بالا نیست و درک آن‌ها آسان است.

یکی از بزرگ‌ترین مشکلات در یادگیری ماشین، بیش‌برازش است، اما اغلب اوقات این مساله به آن آسانی که برای دسته‌بند جنگل تصادفی به وقوع می‌پیوندد، اتفاق نمی‌افتد. محدودیت اصلی جنگل تصادفی آن است که تعداد زیاد درخت‌ها می‌توانند الگوریتم را برای پیش‌بینی‌های جهان واقعی کند و غیر موثر کنند.

به طور کلی، آموزش دادن این الگوریتم‌ها سریع انجام می‌شود، اما پیش‌بینی کردن پس از آنکه مدل آموزش دید، اندکی کند به وقوع می‌پیوندد. یک پیش‌بینی صحیح‌تر نیازمند درختان بیشتری است که منجر به کندتر شدن مدل نیز می‌شود. در اغلب کاربردهای جهان واقعی، الگوریتم جنگل تصادفی به اندازه کافی سریع عمل می‌کند، اما امکان دارد شرایط‌هایی نیز وجود داشته باشد که در آن کارایی زمان اجرا حائز اهمیت است و دیگر رویکردها ترجیح داده می‌شوند. البته، جنگل تصادفی یک ابزار مدل‌سازی پیش‌بین و نه یک ابزار توصیفی است. این یعنی، اگر کاربر به دنبال ارائه توصیفی از داده‌های خود است، استفاده از رویکردهای دیگر ترجیح داده می‌شوند.

## برخی از زمینه‌های کاربرد

الگوریتم جنگل تصادفی در زمینه‌های گوناگونی مانند بانکداری، بازار بورس، پزشکی و تجارت الکترونیک مورد استفاده قرار می‌گیرد. در بانکداری، از این الگوریتم برای شناسایی مشتریانی که بیشتر از سایرین از خدمات بانکی استفاده می‌کنند و بدهی خود را به موقع باز می‌گردانند استفاده می‌شود. این الگوریتم برای شناسایی مشتریان کلاهبرداری که قصد کلاهبرداری از بانک را دارند نیز مورد بهره‌برداری قرار می‌گیرد.

در امور مالی، از جنگل تصادفی برای شناسایی رفتار بورس در آینده استفاده می‌شود. در حوزه پزشکی، این الگوریتم برای شناسایی ترکیب صحیحی از مولفه‌ها و تحلیل تاریخچه پزشکی بیمار، برای شناسایی بیماری او مورد استفاده قرار می‌گیرد. در نهایت، در تجارت الکترونیک (E-commerce) «، جنگل تصادفی برای شناسایی اینکه مشتریان یک محصول را دوست داشته‌اند یا خیر، استفاده می‌شود.



