استاد: فاطمه منصوری

نیمسال دوم ۱۴۰۲-۱۴۰۳

# درخت تصمیم

## یادگیری عمیق

گردآورنده: محمد مظاهری

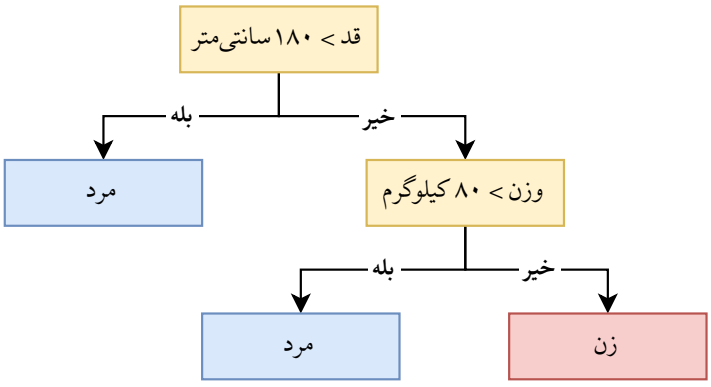
## ۱ مقدمه

درخت تصمیم [1]، به عنوان یکی از پرکاربردترین مدل‌های یادگیری ماشینی نظارت‌شده، برای حل مسائل گوناگون، تجزیه و تحلیل مسائل پیچیده، دسته‌بندی داده‌ها و پیش‌بینی نتایج بر اساس ویژگی‌ها و شرایط مختلف استفاده می‌شود. از ویژگی‌های این مدل می‌توان به سادگی و قابل فهم بودن، بدون_ پارامتر بودن [2] (برای آموزش نیاز به تنظیم پارامتر ندارد، مشابه KNN) و قطعی بودن اشاره داشت که موجب محبوبیت زیاد این مدل شده است.

یکی از ویژگی‌های مهم درخت تصمیم، قابلیت نمایش و مدل‌سازی تصمیمات و ارتباطات بین آن‌هاست. این ساختار از تعدادی گره و یال تشکیل شده که هر یال بیانگر ارتباط بین دو گره می‌باشد. این ارتباطات به ما کمک می‌کنند تا فرآیند تصمیم‌گیری را به صورت یکپارچه و قابل فهم مدل کنیم. در این مدل، داده‌ها به شکل یک درخت سلسله مراتبی از تصمیم‌ها و شرایط تقسیم‌بندی، مورد بررسی قرار می‌گیرند. این ساختار سلسله مراتبی به ما امکان می‌دهد تا مسیری از تصمیمات را از بالاترین سطح (ریشه) تا پایین‌ترین سطح درخت (برگ) پیمایش کرده و به تصمیم نهایی برسیم. با افزایش ارتفاع (عمق) درخت مدل قادر به استفاده از قوانین تصمیم‌گیری پیچیده‌تری برای پیش‌بینی اهداف می‌شود. این ساختار درختی به ما این امکان را می‌دهد تا داده‌های پیچیده را تحلیل کرده و تصمیماتی را به صورت قابل فهم و قابل تفسیر اتخاذ کنیم و الگوهای مهم در داده‌ها را شناسایی کنیم.

### ۱.۱ مثال

مجموعه داده‌ای با قد، وزن و جنسیت داریم. می‌خواهیم با استفاده از درخت تصمیم مدلی را بسازیم که قد و وزن را به عنوان ورودی مدل دریافت و جنسیت را پیش‌بینی کند. درخت دودویی شکل ۱ را می‌توان به عنوان مدل درخت تصمیم دودویی ساده در نظر گرفت.




شکل ۱ : مثالی از درخت تصمیم

برای پیش‌بینی برچسب یک نمونه، باید درخت را از بالاترین تا پایین‌ترین سطح (از ریشه به برگ) پیمایش کرد. برای مثال می‌خواهیم برچسب یک نمونه با ویژگی‌هایش که در ادامه ذکر شده است را روی درخت شکل ۱ پیمایش کنیم. داریم:

[1] Decision tree
[2] non-Parametric

- ورودی: [ قد = ۱۷۰ سانتی متر، وزن = ۹۰ کیلوگرم ]
- آیا قد بیشتر از ۱۸۰ سانتی‌متر است؟ خیر
- آیا وزن بیشتر از ۸۰ کیلوگرم است؟ بله
- خروجی: [مرد]

### ۲.۱ ساختار و اجزاء درخت تصمیم

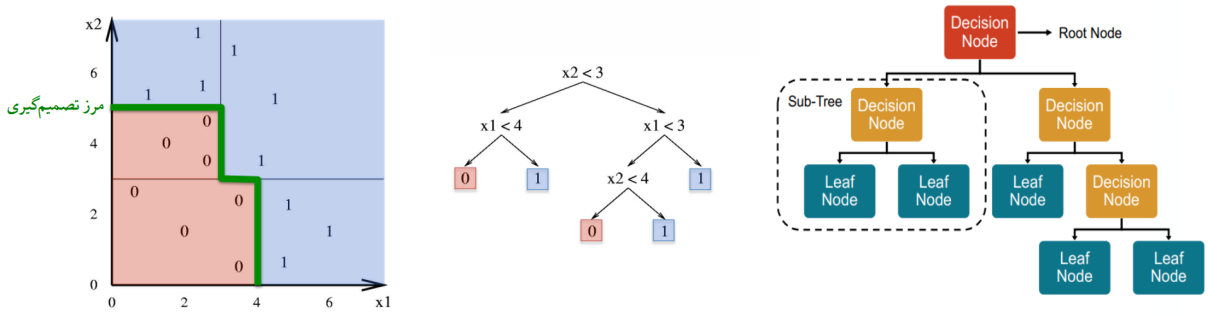
هر درخت تصمیم، از تعدادی گره و یال تشکیل شده، به طوری که هر گره ویژگی خاصی را به همراه شرط تصمیم مرتبط با آن ویژگی را نمایان می‌کند و هر یال یک انتخاب بین دو وضعیت یا شرط مختلف را برای این ویژگی یا گره بررسی می‌کند. مرز تصمیم [۳] نقطه‌ای است که در آن، فضای ویژگی به دو بخش یا زیر درخت تقسیم می‌شود. این تقسیم‌بندی بر اساس مقادیر خاص ویژگی‌ها اتفاق می‌افتد، به عبارت دیگر، مرز تصمیم نشان دهنده‌ی نقطه‌ای است که جدایی بین دو دسته مختلف را ایجاد می‌کند. انواع مختلفی از گره‌های درخت تصمیم وجود دارد که در ادامه به آن می‌پردازیم.

- گره ریشه [۴]: گره ریشه یا گره اولیه که در ابتدا و بالاترین سطح درخت تصمیم قرار دارد و توسط یک یا چند ویژگی مشخص می‌شود، گره‌ای است که مجموعه داده بر اساس ویژگی‌ها یا شرایط مختلف شروع به تقسیم‌بندی می‌کنند.
- گره داخلی [۵]: گره‌هایی هستند که فضای ویژگی را به دو یا چند زیر فضا تقسیم می‌کنند. هر گره داخلی با استفاده از یک شرطی یا مقدار ویژگی مشخص می‌شود و تصمیمی در مورد ادامه مسیر درخت اتخاذ می‌کند. این گره یک یال ورودی و دو یا چند یال خروجی دارد.
- گره برگ [۶]: گره‌هایی هستند که تصمیم‌گیری و برچسب‌گذاری در آن صورت می‌گیرد. اگر گره دارای شرط قطعی باشد یک برچسب متناسب با فضای برچسب مجموعه داده‌ها، به آن اختصاص داده می‌شود؛ در غیر این صورت، برچسب کلاس توسط رأی اکثریت تعیین می‌شود.
- گره والد و فرزند [۷]: هر گره در درخت تصمیم می‌تواند والد یک گره دیگر باشد و یک گره می‌تواند یک یا چند فرزند داشته باشد. به عبارت دیگر، ارتباط بین گره‌ها در یک درخت تصمیم به صورت والدین و فرزندان است. گره والد به گره‌هایی اطلاق می‌شود که به گره‌های دیگر تقسیم شده باشد و و گره‌های حاصل از این تقسیم‌بندی را گره فرزند گوییم.
- گره تصمیم [۸]: این نوع از گره، از جنس گره والد است و باید در آن تصمیمی گرفته که باعث تقسیم‌بندی فضای ویژگی می‌شود. گره‌های تصمیم مسیر فرآیند تصمیم‌گیری را ساخته و مجموعه تصمیمات را به گره‌های برگ ختم می‌کنند.

## ۲ ساخت درخت تصمیم

بهترین درخت تصمیم،کوچک‌ترین درختی است که تمام نمونه‌های آموزشی را به درستی دسته‌بندی می‌کند. یافتن کوچک‌ترین درخت تصمیم یک مساله‌ی NP-Hard است. اما به جای ساختن کوچک‌ترین درخت با ویژگی قید شده (همان جواب بهینه مسئله)، می‌توانیم به یک درخت کوچک دست یابیم که نمونه‌های آموزشی را به درستی دسته‌بندی کند.

[3] Decision Boundary
[4] Root Node
[5] Internal Node
[6] Leaf Node
[7] Parent and Child Node
[8] Decision Node




(ب) مثالی از مرز تصمیم (آ) اجزاء درخت تصمیم

ساخت یک درخت تصمیم دارای دو مرحله اصلی است. در مرحله اول، از میان ویژگی‌های ورودی، مهم‌ترین و تأثیرگذارترین ویژگی بر روی متغیر هدف انتخاب و سپس این ویژگی به عنوان مرز تصمیم برای تقسیم داده‌ها استفاده می‌شود. در مرحله دوم، بر اساس این مرز تصمیم، داده‌ها به دو یا چند زیرمجموعه تقسیم‌بندی می‌شوند. هدف از این تقسیم‌بندی، ایجاد گروه‌هایی از داده‌های مشابه است که فرآیند تصمیم‌گیری را برای پیش‌بینی متغیر هدف ساده‌تر می‌کند. این دو مرحله تا زمانی ادامه پیدا می‌کند که داده‌ها به صورت کامل تفکیک شده باشند و یا شرط پایانی مانند عمق درخت بر آن تاثیر بگذارد.

## ۱.۲ انتخاب ویژگی

در مرحله انتخاب ویژگی‌ها، از بین ویژگی‌های ورودی، ویژگی با بیشترین تأثیر بر متغیر هدف که بهترین تفکیک و تمایز بین دسته‌های مختلف و پیش‌بینی داده‌ها را ایجاد می‌کند، انتخاب می‌شود. این فرایند می‌تواند بر اساس شرایط خاص زیر انجام شود.

- تصادفی (Random): در این روش، ویژگی‌ای به طور تصادفی برای استفاده به عنوان مرز تصمیم انتخاب می‌شود. از این روش معمولا برای ایجاد درخت‌های متنوع استفاده می‌شود.
- کمترین مقادیر (Least Values): در این روش، ویژگی که کمترین تعداد مقادیر یا اندازه‌های مختلف را در داده‌ها دارد، انتخاب می‌شود.
- بیشترین مقادیر (Most Values): برعکس روش قبلی، در این روش ویژگی با بیشترین تعداد مقادیر مختلف انتخاب می‌شود.
- بیشترین مقدار Gain: میزان تفکیک بین دسته‌های مختلف داده‌ها را بر اساس روشی مانند Information Gain و Gini impurity اندازه‌گیری و ویژگی مناسب را برای تقسیم انتخاب می‌کند.

برای انتخاب بهترین ویژگی در هر گره، از دو روش معمول Information Gain و Gini impurity برای تقسیم مدل‌های درخت تصمیم استفاده می‌شود. این روش‌ها به ارزیابی کیفیت هر شرط آزمون کمک می‌کنند و و میزان دقت و کیفیت دسته‌بندی نمونه‌ها به یک کلاس را نشان می‌دهند. برای توضیح Information Gain ، نیاز است ابتدا به آنتروپی [۹] پرداخته شود. آنتروپی انحراف نمونه‌ها را اندازه‌گیری می‌کند. آنتروپی برای مجموعه داده S به صورت زیر محاسبه می‌شود:

$$Entropy(S) = -\sum_{c \in C} P(c) \log_2 P(c) \quad (1)$$

در فرمول ۱ ، $S$ نشان‌دهنده مجموعه داده است، $c$ کلاس‌ها را در مجموعه $S$ نشان می‌دهد و $p(c)$ نسبت نقاط داده‌ای است که به کلاس $c$ تعلق دارند به تعداد کل نقاط داده در مجموعه $S$ است. آنتروپی مقادیری بین ۰ و ۱ می‌تواند داشته باشد. اگر تمام نمونه‌ها در مجموعه داده S به یک کلاس تعلق داشته باشند، آنتروپی برابر با صفر خواهد بود. اما اگر نیمی از نمونه‌ها به یک کلاس و نیم دیگر به کلاس دیگر تعلق داشته باشند، آنتروپی بیشینه خود را با مقدار ۱ خواهد داشت.

[9] Entropy

Information Gain نشان‌دهنده تفاوت آنتروپی قبل و بعد از یک تقسیم بر اساس ویژگی داده شده است. ویژگی با بیشترین In-formation Gain بهترین تقسیم را ایجاد می‌کند زیرا بهترین کار را برای دسته‌بندی داده‌های آموزشی به تفاوت دسته‌بندی هدف خود انجام می‌دهد. Information Gain معمولا با فرمول زیر نمایش داده می‌شود:

$$\text{information gain}(S, a) = Entropy(S) - \sum_{v \in Values(a)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{2}$$

در این رابطه، متغیر $a$ به عنوان یک ویژگی خاص یا برچسب کلاس استفاده می‌شود تا داده‌ها را تقسیم کند. عبارت $Entropy(S)$ به معنای آنتروپی مجموعه داده $S$ است، که میزان تنوع و توزیع داده‌ها را در مجموعه نشان می‌دهد. نسبت $\frac{|S_v|}{|S|}$ نسبت تعداد مقادیر در مجموعه $S_v$ به کل مقادیر موجود در مجموعه داده $S$ است. $Entropy(S_v)$ آنتروپی مجموعه داده $S_v$ را نشان می‌دهد که میزان تنوع و توزیع داده‌ها در زیرمجموعه‌هایی است که از تقسیم مجموعه اصلی $S$ بر اساس ویژگی $a$ به دست می‌آید.

معیار Gini impurity برای اندازه‌گیری میزان ناخالصی یا بی‌نظمی داده‌ها به کار می‌رود. به طور دقیق‌تر، این معیار نشان‌دهنده احتمال اشتباه در دسته‌بندی تصادفی یک نمونه در آن گره است، در صورتی که برچسب‌گذاری بر اساس توزیع کلاس‌ها در کل مجموعه داده صورت گیرد. همانند آنتروپی، اگر یک مجموعه (به عنوان مثال مجموعه S کاملاً خالص باشد، به این معنی که همه داده‌ها به یک دسته تعلق داشته باشند، مقدار Gini impurity برای آن مجموعه صفر خواهد بود. هر چه مقدار Gini impurity به صفر نزدیک‌تر باشد، نشان‌دهنده خالصی و یکنواختی بیشتر گره از نظر توزیع کلاس است. با افزایش ناخالصی کلاس‌ها در آن گره، مقدار Gini impurity افزایش می‌یابد. این مفهوم با استفاده از فرمول زیر نمایش داده می‌شود:

$$\text{Gini Impurity}(S) = 1 - \sum_i (P_i)^2 \tag{3}$$

در این فرمول، $P_i$ نشان‌دهنده احتمال تخصیص به کلاس $i$ در مجموعه داده است. در فرآیند ساخت درخت تصمیم، الگوریتم به دنبال تقسیم‌بندی‌هایی می‌گردد که Gini impurity را در گره‌های فرزند به حداقل برساند. با این کار، در هر مرحله از ساخت درخت، خلوص داده‌ها افزایش می‌یابد و در نهایت منجر به یک درخت تصمیم با توانایی تفکیک بهتر می‌شود.

## ۲.۲ روش‌های ساخت درخت تصمیم

برای ساخت درخت تصمیم، از روش‌ها و الگوریتم‌های مختلفی مانند ID3، C4.5 و CART استفاده می‌شود. این الگوریتم‌ها از داده‌های ورودی و ویژگی‌های مختلف استفاده کرده و قوانین تصمیم‌گیری را ایجاد می‌کنند که در نهایت به ساختار یک درخت تصمیم منجر می‌شود. هر یک از این الگوریتم‌ها ویژگی‌ها، مزایا و معایب خاص خود را دارند که در فرآیند ساخت درخت تصمیم تأثیرگذار هستند. این سه الگوریتم از جمله مهم‌ترین روش‌های مورد استفاده در زمینه‌ی یادگیری ماشین برای ساخت درخت تصمیم هستند و هر یک با ویژگی‌ها و قابلیت‌های خاص خود، مناسب برای مسائل مختلفی از جمله دسته‌بندی و رگرسیون می‌باشند. در ادامه، به بررسی عمیق‌تر این الگوریتم‌ها و کاربردهای آن‌ها در مسائل مختلف خواهیم پرداخت.

- الگوریتم ID3 [10] توسط Ross Quinlan در سال ۱۹۸۶ توسعه یافت. این الگوریتم به صورت حریصانه و بازگشتی، برای هر گره یک درخت چند مسیری ایجاد می‌کند. ID3 برای انتخاب ویژگی‌ها از بیشترین Gini impurity یا کمترین Entropy استفاده می‌کند. درخت حاصل از این الگوریتم می‌تواند به حداکثر ارتفاع خود برسد و در نتیجه درختی کوتاه و پهن ایجاد کند. این الگوریتم این الگوریتم از داده‌های گسسته و غیر عددی پشتیبانی کرده و مسائل چندـ دسته‌ای را مدل‌سازی می‌کند. هرچند که درخت ساخته‌شده توسط این الگوریتم برای تصمیم‌گیری‌های قائم بر قوانین ساده و قابل فهم است، اما به دلیل عدم هرس برخی از شاخه‌ها، در معرض خطر بیش برازش [11] قرار دارد.

[10] Iterative Dichotomiser 3
[11] overfitting

- الگوریتم C4.5 به عنوان یکی از توسعه‌های بعدی الگوریتم ID3 در سال **۱۹۹۳** توسط Ross Quinlan طراحی شده است تا با محدودیت‌های آن مقابله کند. این الگوریتم از ویژگی‌های پیوسته و گسسته پشتیبانی می‌کند و قادر است با ویژگی‌های گم‌شده [۱۲] نیز برخورد کند. C4.5 برای ارزیابی ترتیب قوانین، از مجموعه قوانین اگر_آنگاه استفاده می‌کند. همچنین این الگوریتم دارای مرحله‌ای برای هرس شاخه‌های اضافی درخت است تا به کاهش اثرات بیش برازش کمک کند.
- الگوریتم CART [۱۳] در سال **۱۹۸۴** توسط Leo Breiman معرفی شد. این الگوریتم معمولا از معیار Gini impurity برای انتخاب ویژگی‌ها و تقسیم داده‌ها استفاده می‌کند. این الگوریتم علاوه بر متغیرهای هدف گسسته و غیر عددی از متغیرهای پیوسته و عددی (رگرسیون) نیز پشتیبانی می‌کند. این الگوریتم درخت‌های دودویی را با استفاده از ویژگی و آستانه‌ای [۱۴] ساخته و بیشترین بهره اطلاعاتی را در هر گره بدست می‌آورد. این الگوریتم از انشعاب‌های دوتایی استفاده می‌کند که می‌تواند درخت‌های بهتری نسبت به C4.5 تولید کند، اما معمولا درخت‌های حاصل از آن بزرگ و تفسیر آنها دشوار است. پیچیدگی و هرس این درخت نیز هزینه‌بر و زمان‌بر است.

## ۳ درخت تصمیم برای مسائل رگرسیون

الگوریتم CART در مسائل رگرسیون نیز همانند مسائل دسته‌بندی با تغییرات اندک عمل می‌کند. تفاوت اساسی آن در این است که به جای پیش‌بینی یک کلاس در هر گره، یک مقدار را پیش‌بینی می‌کند. برای هر گره، مقدار هدف میانگین تمام نمونه‌های آموزشی مربوط به این گره برگ است. در روش کار این الگوریتم، سعی می‌شود مجموعه آموزش را به گونه‌ای تقسیم کند که میانگین مربعات خطا [۱۵] را به حداقل برساند. میانگین مربعات خطا در یک گره داده شده نیز معمولا به عنوان واریانس درون گره [۱۶] شناخته می‌شود، بنابراین معیار تقسیم به عنوان کاهش واریانس [۱۷] شناخته می‌شود.

## ۴ مشکل بیش برازش

درخت‌های تصمیم معمولاً دارای فرضیات بسیار کمی هستند و اگر محدود نشوند، ساختار خود را با داده‌های آموزشی بسیار نزدیک می‌کنند و به احتمال زیاد دچار بیش برازش می‌شوند. برای پیشگیری از این موضوع، اهمیت دارد که در طول آموزش، آزادی درخت تصمیم را محدود کنیم. بدین منظور، می‌توانیم پارامترهای زیر را برای جلوگیری از این اتفاق تنظیم کنیم.

- Max Tree Depth: این پارامتر حداکثر عمقی را که درخت می‌تواند داشته باشد مشخص می‌کند. درختی با عمق بیشتر از این حد، ممکن است دچار بیش برازش شود.
- Max Leaf Nodes: این پارامتر حداکثر تعداد برگ‌هایی را که درخت می‌تواند داشته باشد محدود می‌کند. این کمک می‌کند تا درخت از پیچیدگی جلوگیری کرده و از تقسیم بیش از حد داده‌ها اجتناب کند.
- Min Sample Leaf: این پارامتر حداقل تعداد نمونه‌هایی را که برای گسترش یک برگ لازم است تعیین می‌کند. اگر تعداد نمونه‌ها در یک برگ کمتر از این حد باشد، رشد در آن برگ متوقف می‌شود.
- Min Sample Split: این پارامتر تعیین می‌کند که چه تعداد نمونه در یک گره باید وجود داشته باشد تا قبل از انجام تقسیم، امکان تقسیم آن گره باشد. اگر تعداد نمونه‌ها در یک گره کمتر از این مقدار باشد، امکان تقسیم آن گره وجود نخواهد داشت.

[12] Missing Attributes
[13] Classification and Regression Trees
[14] Threshold
[15] Mean squared error
[16] intra-node variance
[17] variance reduction

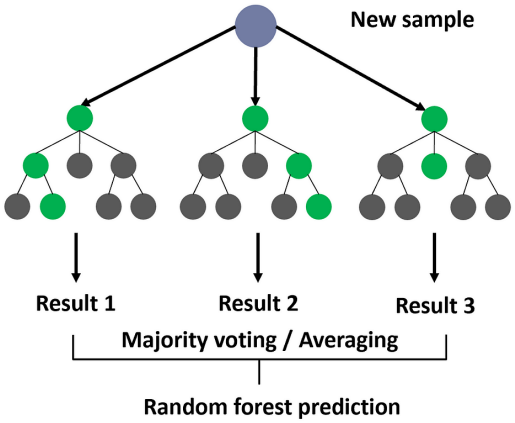
## ۵ مزایا و معایب

درخت‌های تصمیم به عنوان یک مدل یادگیرنده، چندین مزیت دارند. یکی از مهم‌ترین مزایای آنها، سادگی در تفسیر و ارتباط است که آن را برای کاربران با سطوح مختلف تخصص قابل دسترس می‌سازد. همچنین، درخت‌های تصمیم مستقل از مقیاس ویژگی‌ها هستند، به این معنی که قبل از آموزش مدل، نیازی به استانداردسازی یا نرمال سازی ویژگی‌های ورودی نیست، که این امر مراحل پیش‌پردازش را ساده می‌کند.

با این حال، درخت‌های تصمیم همچنین دارای محدودیت‌ها و نقاط ضعفی نیز هستند. یکی از چالش‌های اصلی این است که تمایل به بیش برازش داده‌های آموزش را دارند، به ویژه زمانی که درخت به سطح عمق یا پیچیدگی زیادی می‌رسد. این موضوع می‌تواند منجر به عملکرد نه چندان خوبی بر روی داده‌های جدید شود. به علاوه، برای کاهش بیش برازش، اغلب نیاز به هرس جزئی درخت است که زمان‌بر و نیازمند تنظیمات دقیق پارامترها باشد. علاوه بر این، درخت‌های تصمیم ممکن است برای مسائلی که خروجی آنها می‌تواند به خوبی با یک مدل خطی ساده تقریب شود، محاسباتی گران‌قیمت داشته باشند، زیرا درخت‌ها حتی برای ارتباط‌های ساده معمولاً ساختارهای پیچیده‌ای ایجاد می‌کنند. در مسائل رگرسیون، دامنه خروجی درخت‌های تصمیم محدود است و به داده‌های آموزشی وابسته است، که این موضوع انعطاف‌پذیری آنها را نسبت به مدل‌های رگرسیون دیگر محدود می‌کند.

## ۶ جنگل‌های تصادفی

جنگل تصادفی، یکی از روش‌های پیشرفته در حوزه پیش‌بینی و یادگیری ماشین است که بر پایه مجموعه‌ای از درختان تصمیم ساخته شده است. در این روش، هر درخت تصمیم به صورت مستقل و موازی ایجاد و سپس نتایج آن‌ها ترکیب می‌شود (شکل ۳ ). با استفاده از جنگل تصادفی، می‌توانیم از تنوع در مدل‌ها بهره ببریم و از این طریق، دقت پیش‌بینی را افزایش داده و مشکلات مربوط به بیش برازش را کاهش دهیم. این روش از جمله روش‌های موثری است که در حوزه یادگیری ماشین برای مقابله با چالش‌های پیش‌آمده مورد استفاده قرار می‌گیرد.




شکل ۳: یک مثال از فرایند پیش‌بینی در جنگل‌های تصادفی، که نتایج درختان تصمیم به طور یکجا گردآوری می‌شوند و رای اکثریت یا میانگین آراء، مقدار پیش‌بینی‌شده را تشکیل می‌دهد.

## مراجع:


- Decision Trees and IBM
- Decision Trees in scikit-learn

- STAT 451: Machine Learning
- COMP 642 - Machine Learning