

درخت تصمیم (Decision Tree)

درخت تصمیم یک روش ساده و قدرتمند برای تصمیم‌گیری و دسته‌بندی داده‌ها است.

- مانند یک درخت واقعی، شامل شاخه‌هایی است که به تصمیم‌های مختلف منتهی می‌شوند.
- در هر مرحله، یک سوال ساده پرسیده می‌شود و داده‌ها به دسته‌های مختلف تقسیم می‌شوند.
- در انتهای مسیر (برگ درخت) نتیجه نهایی مشخص می‌شود.

مثال ساده

فرض کنید می‌خواهیم تصمیم بگیریم که یک نفر بیمه ماشین دریافت کند یا نه؟

بر اساس شرایطی مانند سن، تجربه رانندگی و سابقه تصادف، درخت تصمیم به ما کمک می‌کند که فرد واجد شرایط هست یا نه.



کاربردهای مهم درخت تصمیم

- دسته‌بندی (Classification): مثلاً تشخیص اینکه یک ایمیل اسپم هست یا نه.
- پیش‌بینی (Regression): مثلاً پیش‌بینی قیمت خانه بر اساس ویژگی‌های آن.
- سیستم‌های توصیه‌گر (Recommendation Systems): مثلاً پیشنهاد فیلم در یوتیوب بر اساس سلیقه کاربر.



کد پایتون درخت تصمیم

در کد زیر ما می خواهیم افراد را به دو دسته قبول و رد براساس سن و درآمد طبقه بندی کنیم.

سن کمتر از ۳۰ سال و درآمد بالا → احتمال قبول شدن بیشتر است.

سن بالای ۳۰ سال و درآمد پایین → احتمال رد شدن بیشتر است.

```
from sklearn.tree import DecisionTreeClassifier
X = [[25, 50000], [30, 60000], [35, 40000], [40, 30000],
     [20, 70000]]
y = ["قبول", "قبول", "رد", "رد", "قبول"]
model = DecisionTreeClassifier()
model.fit(X, y)
prediction = model.predict([[32, 55000]])

print("وضعیت فرد جدید:", prediction[0])
```

[برای مشاهده کد کامل و اجرای خروجی کلیک کنید.](#)



توضیح کد برای درخت تصمیم

وارد کردن کتابخانه های لازم

```
from sklearn.tree import DecisionTreeClassifier
```

- `DecisionTreeClassifier` کلاس برای تعریف مدل درخت تصمیم در کتابخانه `scikit-learn` است.

توضیح کد برای درخت تصمیم

تعریف داده‌های ورودی (X) و خروجی (y)

```
X = [[25, 50000], [30, 60000], [35, 40000], [40, 30000],  
      [20, 70000]]  
y = ["قبول", "رد", "رد", "قبول", "قبول"]
```

- `DecisionTreeClassifier` مدل درخت تصمیم را فراهم می‌کند.
- X شامل ویژگی‌های افراد است. در اینجا (سن، درآمد) داریم.
- y نتیجه را نشان می‌دهد: "قبول" یا "رد".
- برای مثال، داده‌ی $[25, 50000]$ → "قبول" یعنی فرد ۲۵ ساله با درآمد ۵۰ هزار پذیرفته شده است.

توضیح کد برای درخت تصمیم

ساخت مدل درخت تصمیم

```
model = DecisionTreeClassifier()
```

- این خط یک مدل درخت تصمیم می‌سازد که بعداً آموزش می‌بیند.



توضیح کد برای درخت تصمیم

آموزش مدل با داده‌های ورودی و خروجی

```
model.fit(X, y)
```

- مدل یاد می‌گیرد که چگونه سن و درآمد روی قبول یا رد شدن تأثیر می‌گذارند.



توضیح کد برای درخت تصمیم

پیش‌بینی وضعیت یک فرد جدید

```
prediction = model.predict([[32, 55000]])
```

- از مدل می‌خواهیم ببیند فردی ۳۲ ساله با درآمد ۵۵ هزار قبول می‌شود یا نه؟



توضیح کد برای درخت تصمیم

نمایش نتیجه پیش‌بینی

```
print("وضعیت فرد جدید:", prediction[0])
```

- مدل پاسخ را نمایش می‌دهد. اگر خروجی "قبول" باشد یعنی فرد پذیرفته شده است، اگر "رد" باشد یعنی رد شده است.

نتیجه اجرای برنامه

نمایش نتیجه پیش بینی

وضعیت فرد جدید: قبول

- نتیجه: مدل پیش بینی می کند که فرد ۳۲ ساله با درآمد ۵۵ هزار پذیرفته می شود.

چرا از درخت تصمیم استفاده کنیم؟

- تفسیرپذیر است: نتیجه را می‌توان به راحتی درک کرد.
- نیازی به تنظیمات پیچیده ندارد: مدل به‌طور خودکار تصمیم‌گیری می‌کند.
- مناسب برای داده‌های کوچک و دسته‌بندی‌های ساده است.

K نزدیک‌ترین همسایه (K-Nearest Neighbors - KNN)

الگوریتم K نزدیک‌ترین همسایه (KNN) یکی از ساده‌ترین و در عین حال قدرتمندترین الگوریتم‌های یادگیری ماشین است که برای دسته‌بندی (Classification) و پیش‌بینی مقدار (Regression) استفاده می‌شود.

KNN چیست و چگونه کار می‌کند؟

ایده اصلی: برای پیش‌بینی مقدار یا دسته‌ی یک داده جدید، به داده‌های اطرافش (همسایه‌ها) نگاه می‌کنیم.
روش کار:

- فاصله بین داده جدید و تمام داده‌های قبلی را محاسبه می‌کنیم.
- K تا از نزدیک‌ترین داده‌ها را پیدا می‌کنیم.
- اگر مسئله دسته‌بندی باشد: اکثریت همسایه‌ها تعیین می‌کنند که داده جدید به کدام دسته تعلق دارد.
- اگر مسئله پیش‌بینی مقدار باشد: میانگین مقادیر K همسایه‌ی نزدیک را محاسبه می‌کنیم.