

## یادگیری ماشین

40240112093 تمرین سری 2- گزارش مسئله 2

رویا شاهرودی

ابتدا کتابخانه های مورد نیاز pandas، numpy و kagglehub را به برنامه اضافه میکنیم.

Kagglehub برای دانلود داده از منابع Kaggle استفاده میشود.

سپس داده ها از Kaggle دانلود شده و به عنوان داده های آموزشی مورد استفاده قرار می گیرند. برای داده های تست نیز 4 نمونه تولید میکنیم و با تبدیل کردن آن به فرمت DataFrame آن را مانند داده آموزشی به آرایه دو بعدی تبدیل میکنیم.

تابع calculate\_entropy برای محاسبه ی انتروپی به کار می رود که یکی از شاخص های مهم در سافت درفت تصمیم است. بعنوان ورودی داده ها را که شامل ویژگی ها و ستون هدف است را میگیرد و با استفاده از توزیع احتمال کلاس ها، انتروپی کل ستون هدف را محاسبه می کند.

تابع subset\_entropy انتروپی یک زیرمجموعه از داده ها را محاسبه می کند که برای محاسبه ی انتروپی هر زیرمجموعه در هنگام تقسیم ویژگی ها استفاده می شود. این تابع بعنوان ورودی یک زیرمجموعه از داده ها که براساس یک ویژگی خاص تفکیک شده است را گرفته و در فروبی انتروپی زیر مجموعه را به ما میدهد.

تابع بعدی information\_gain است که برای محاسبه ی معیار کسب اطلاعات یا کاهش انتروپی برای یک ویژگی خاص به کار می رود که نشان می دهد چقدر از عدم قطعیت توسط تقسیم ویژگی کاهش یافته است. این تابع بعنوان ورودی مجموعه داده اصلی و نام ویژگی مورد نظر را گرفته و مقدار کسب اطلاعات بر اساس آن ویژگی محاسبه میکند.

تابع `choose_best_feature` برای انتخاب بهترین ویژگی جهت تقسیم گره‌ها در درخت تصمیم استفاده می‌شود. این تابع با استفاده از اطلاعات مناسبه‌شده از هر ویژگی، بهترین ویژگی را برای تقسیم و قرار گیری در گره بعدی انتخاب می‌کند.

تابع `build_decision_tree` درخت تصمیم را به صورت بازگشتی ایجاد می‌کند. این تابع از تقسیمات پی‌درپی برای هر ویژگی استفاده کرده و درخت را می‌سازد. این تابع برای هر ویژگی، گره‌ای ایجاد می‌کند و با سافت زیرمجموعه‌های داده، به صورت بازگشتی زیرشافه‌ها را ایجاد می‌کند.

تابع `predict` یک نمونه را از طریق درخت تصمیم عبور داده و نتیجه را پیش‌بینی می‌کند. این تابع برای هر نمونه‌ی جدید، ویژگی‌های مرتبط را در درخت جستجو کرده و نتیجه را برمی‌گرداند. بعنوان ورودی نیز درخت تصمیم به همراه نمونه تست را می‌گیرد و پیش‌بینی مدل را خروجی می‌دهد.

تابع `evaluate_model` مدل را روی داده‌های آزمایشی ارزیابی می‌کند. این تابع تعداد پیش‌بینی‌های درست را با مقدار واقعی مقایسه کرده و دقت مدل را مناسبه می‌کند. پس از سافت درخت، آن را چاپ کرده و سافتار درخت را نمایش می‌دهیم.

```
print("Decision Tree:", decision_tree)
Decision Tree: {'Outlook': {'Sunny': {'Humidity': {'High': 'No', 'Normal': 'Yes'}}, 'Overcast': 'Yes', 'Rain': {'Wind': {'Weak': 'Yes', 'Strong': 'No'}}}}
```

یک نمونه از داده تست را انتخاب کرده و به مدل می‌دهیم تا پیش‌بینی کند.

```
Test instance: {'Outlook': 'Overcast', 'Temperature': 'Mild', 'Humidity': 'Normal', 'Wind': 'Weak'}
Actual label: Yes
Predicted label: Yes
```

داده‌های تست را به همراه درخت تصمیم سافت‌شده به تابع `evaluate_model` می‌دهیم تا دقت مدل را گزارش دهد.

```
Model accuracy on test dataset: 1.0
```

در نهایت، از کتابخانه anytree برای نمایش سافتار درخت به صورت متنی استفاده میکنیم تا درخت را بصورت واضح تری نمایش دهیم.

