# Decision Trees

## مقدمه :

درختان تصمیم مدل یادگیری ماشین همه کاره ای هستند که قادر به طبقه بندی و رگرسیون هستند.

## پروژه iris :

می خواهیم با استفاده از دیتاست iris و با استفاده از الگوریتم درخت تصمیم برنامه ای بنویسیم که ویژگی های مربوط به گل ها را به عنوان داده جدید دریافت کند و آن ها را طبقه بندی کند.

داریم :

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import ShuffleSplit
from sklearn.model_selection import GridSearchCV
from sklearn.tree import plot_tree
from sklearn.metrics import classification_report
```

```
def load_data():
    DataSet = pd.read_csv("iris.csv", header = None,
                          names = ["sepal length",
                                   "sepal width",
                                   "petal length",
                                   "petal width",
                                   "label"])

    data = DataSet.iloc[:, :4]
    label = DataSet.iloc[:, 4]

    return data, label
```

```
data, label = load_data()
tree_clf = DecisionTreeClassifier(max_depth=3)
cv_set = ShuffleSplit(n_splits=5, test_size=0.2, random_state=42)
```

```
tree_clf = DecisionTreeClassifier()
tree_clf.get_params()
```

```
parameters = {"criterion":["gini", "entropy"], "max_depth":[3, 4, 5, 6, 7, 8]}
clf = GridSearchCV(estimator=tree_clf, param_grid=parameters, scoring="accuracy", cv=cv_set, verbose=10, return_train_score=True)
clf_result = clf.fit(data, label)
```

```
print(f"best score: {clf_result.best_score_}")
print(f"best params: {clf_result.best_params_}")
```

```
tree_clf_final = DecisionTreeClassifier(criterion="gini", max_depth=3)
tree_clf_final.fit(data, label)
```

```
DecisionTreeClassifier(max_depth=3)
```

```
fig = plt.figure(figsize=(20, 20))
x = plot_tree(tree_clf_final, class_names=["Iris-setosa", "Iris-versicolor", "Iris-virginica"], filled=True)
```

```
label_pred = tree_clf_final.predict(data)
print(classification_report(label, label_pred))
```

## پارامترهای درخت تصمیم :

می خواهیم با پارامترها و اصطلاح های مختلفی که در الگوریتم درخت تصمیم وجود دارد آشنا شویم.

### ❖ گره ( Node ) :

به هر باکس که در درخت تصمیم وجود دارد و در آن شرطی است که بر اساس درست یا غلط بودن شرط نتیجه مختلفی بدست می آید گره می گویند.

### ❖ Root Node :

به اولین گره Root Node می گویند.

### ❖ Leaf Node :

به آخرین گره ها Leaf Node می گویند.

### ❖ Decision Node :

به مابقی گره ها Decision Node می گویند.

**Written by Hossein Kariznoei**

❖ **gini** :

الگوریتم درخت تصمیم بر اساس احتمال کار می کند. با استفاده از علم احتمال ، معیار gini که میزان ناخالصی یک گره را اندازه گیری می کند ، به شکل زیر تعریف می کنند.

$$gini = \sum p_i * (1 - p_i)$$

$p_i$ نسبت تعداد نمونه های کلاس i به کل نمونه های یک Node است.

**نکته !** اگر gini برابر با صفر باشد بدین معنی است که کل نمونه های این Node فقط متعلق به یک کلاس است و گره ناخالصی ندارد یا به عبارت دیگر خالص است.

❖ **entropy** :

در مقابل معیار gini ، معیار دیگری به اسم entropy که میزان عدم قطعیت یا Randomness را نشان می دهد ، به شکل زیر تعریف می شود.

$$entropy = -\sum p_i * \log_2 p_i$$

**نکته !** اگر entropy برابر با صفر باشد بدین معنی است که کل نمونه های این Node فقط متعلق به یک کلاس است.

**نکته !** هر دو معیار gini و entropy تقریبا مشابه هم هستند اما معیار gini کمی سریع تر است زیرا در entropy باید لگاریتم محاسبه شود که کمی پیچیده تر خواهد بود.

❖ **Information Gain** :

برای تعیین آن که در هر Node کدام یک از ویژگی ها را برای بررسی انتخاب کنیم از Information Gain استفاده می کنیم.

$$Information\ Gain = entropy_{parent} - Avg(entropy_{children})$$

**نکته !** پس از محاسبه Information Gain برای حالات مختلف ، حالتی که Information Gain بیشتری داشته باشد بهتر است.

**نکته !** از جمله دلایل مهم برای انتخاب مدل درخت تصمیم در پروژه ها می توان به فهم و تفسیر ساده استفاده آسان و قدرتمند بودن اشاره کرد.