Drzewa decyzyjne

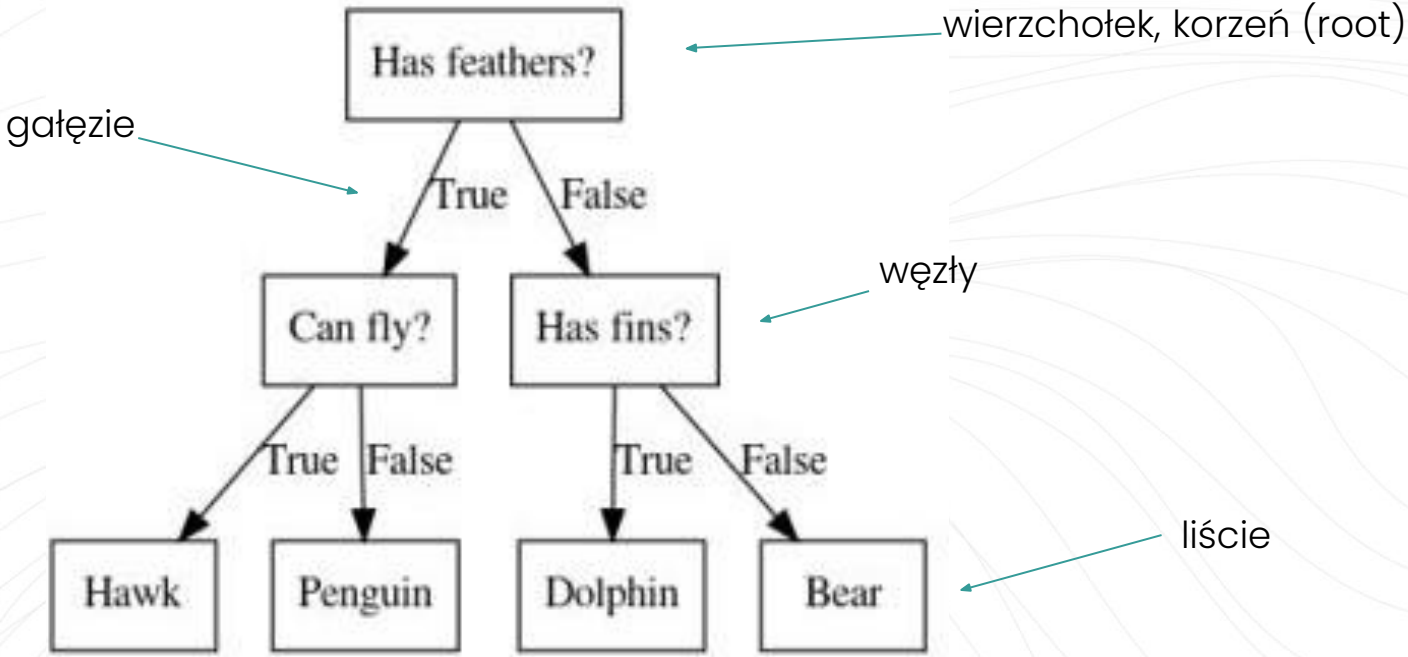
Roadmapa

- Co to są drzewa decyzyjne?
- Kryteria nieczystości
- Zastosowania praktyczne (API sklearn)
- Zalety i problemy drzew decyzyjnych

Drzewo decyzyjne

- Metoda wspomagania procesów decyzji
- Model używany do zadań regresji i klasyfikacji
- Intuicyjny model opierający się na podziale danych przez serię porównań

Wizualizacja algorytmu



Algorytm - CART (Classification and Regression Trees)

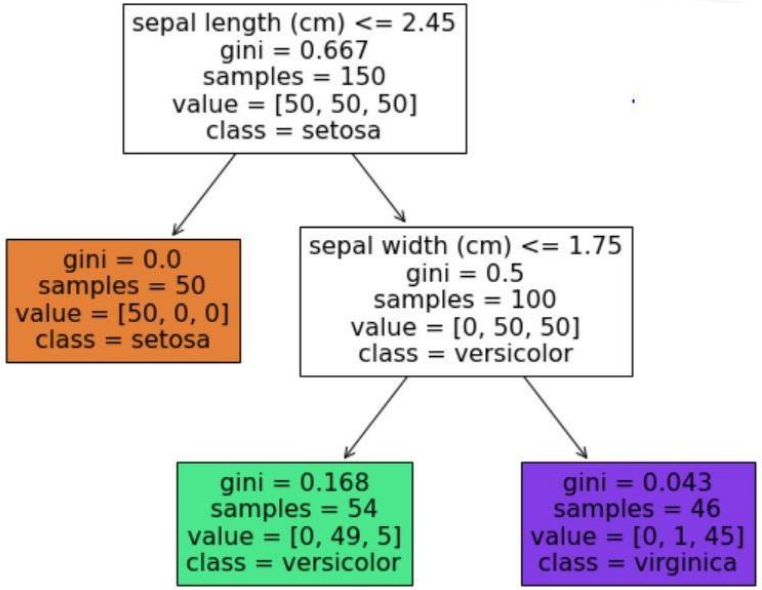
1. Zaczynamy od pierwszego węzła (root) - tu skupia się cała próba ucząca.

1. Wybieramy cechę i podział, które utworzą gałęzie. Kryterium podziału jest wspólne w każdym węźle - ma rozdzielać obserwacje na takie części by podzbiory były jak najczystsze.

1. Uzyskujemy podzbiory zgodnie z powyższym warunkiem.

1. Powtarzamy 2. i 3. aż do momentu stopu.

1. W liściu przypisuje się taką samą klasę jak większość elementów, które dotarły do liścia.



Kryteria podziału

Podział polega na jak najlepszym rozdzieleniu podgrupy na części - tak aby w węzłach dzieci różnorodność była jak najmniejsza.

Miara różnorodności:

- 0 - wszystkie obserwacje należą do tej samej klasy,
- wartość maksymalna - rozkład przynależności do klas jest jednostajny.

Kryteria nieczystości (impurity criterion)

Indeks Giniego

$$I_G = 1 - \sum_{j=1}^{c} p_j^2$$

p_j: proportion of the samples that belongs to class c for a particular node

Entropia

$$I_H = -\sum_{j=1}^{c} p_j log_2(p_j)$$

p_j: proportion of the samples that belongs to class c for a particular node.

*This is the the definition of entropy for all non-empty classes ($p \neq 0$). The entropy is 0 if all samples at a node belong to the same class.

Funkcja kosztu dla algorytmu CART

$$J(k, t_k) = \frac{m_{\text{left}}}{m} G_{\text{left}} + \frac{m_{\text{right}}}{m} G_{\text{right}}$$

$$\text{where} \begin{cases} G_{\text{left/right}} \text{ measures the impurity of the left/right subset,} \\ m_{\text{left/right}} \text{ is the number of instances in the left/right subset.} \end{cases}$$

Information Gain - różnica miary nieczystości rodzica i funkcji J, mówi o przyroście informacji po dodaniu kolejnego węzła. Pozwala na dokonanie doboru warunków kolejnych podziałów.

Moment stopu

Gdy osiągniemy maksymalną głębokość drzewa (*max_depth*) albo nie można znaleźć podziału, który zlikwiduje nieczystość (impurity).

Regresja przy użyciu drzew decyzyjnych

Drzewa decyzyjne możemy również stosować dla problemów regresji. Wówczas jako kryterium stosujemy zwykle MSE (Mean square error).

Funkcja kosztu:

$$J(k, t_k) = \frac{m_{\text{left}}}{m}\text{MSE}_{\text{left}} + \frac{m_{\text{right}}}{m}\text{MSE}_{\text{right}} \quad \text{where} \begin{cases} \text{MSE}_{\text{node}} = \sum_{i \in \text{node}} \left(\hat{y}_{\text{node}} - y^{(i)}\right)^2 \\ \hat{y}_{\text{node}} = \frac{1}{m_{\text{node}}} \sum_{i \in \text{node}} y^{(i)} \end{cases}$$

Problemy drzew decyzyjnych

Overfitting

Drzewa dążą do tego, by rozrastać się aż do uzyskania czystych podzbiorów w liściach. Często wiąże się to z tym, że w praktyce takie drzewo “zapamiętuje” zbiór treningowy.

Aby zredukować efekty overfittingu możemy manipulować parametrami modelu.

API sklearn.tree

sklearn.tree.DecisionTreeClassifier

class sklearn.tree.**DecisionTreeClassifier**(*, *criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, ccp_alpha=0.0*) [source]

sklearn.tree.DecisionTreeRegressor

class sklearn.tree.**DecisionTreeRegressor**(*, *criterion='mse', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, ccp_alpha=0.0*) [source]

Parametry drzewa decyzyjnego

max_depth - głębokość drzewa

min_samples_leaf - minimalna liczba obserwacji w liściu

max_leaf_nodes - maksymalna liczba liści w drzewie

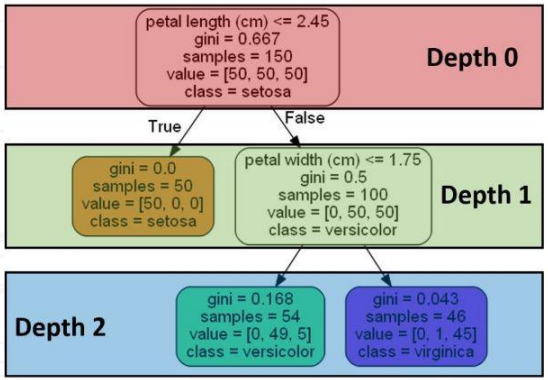
max_features - maksymalna liczba zmiennych rozważanych w podziale

ccp_alpha - jak mocno przycinane są drzewa

criterion - kryterium nieczystości `gini`, `entropy`

random_state - algorytm stochastyczny (!)

class_weight - ważenie klas przy niezbalansowaniu



Wady i zalety

Wady i zalety

Wady

- Niestabilność algorytmu
- Podatność na overfitting
- Regresja nie przewiduje danych spoza zakresów, które widziała
- Podziały ortogonalne (prostopadłe do osi)

Zalety

- Łatwa wizualizacja i prosta interpretacja
- Odpowiedni do problemów klasyfikacji i regresji
- Niewrażliwość na monotoniczne przekształcenia zmiennych
- Niewrażliwość na istnienie w algorytmie nieistotnych atrybutów
- Prosty w obsłudze - można używać cech kategorycznych i liczbowych (uwaga! Sklearn nie obsługuje kategorycznych)

Dzięki!