Języki programowania – Python
Drzewa decyzyjne

Marzena Nowakowska

Wydział Zarządzania i Modelowania Komputerowego
Politechnika Świętokrzyska

Budynek C, p. 3.21

spimn@tu.kielce.pl

Modelowanie nadzorowane - pojęcia podstawowe

$$\mathbf{Y} = f(X_1, X_2, \ldots, X_n)$$

decyzja
atrybut decyzyjny
etykieta
zmienna celu

atrybuty, atrybuty objaśniające

Przykład zbioru danych →

pclass	embarked	age	sex	survived
1	S	29	female	**1**
1	S	1	male	**1**
1	S	2	female	**0**
1	S	30	male	**0**

0 – nieuratowany, nie przeżył
1 – uratowany, przeżył

Drzewo decyzyjne

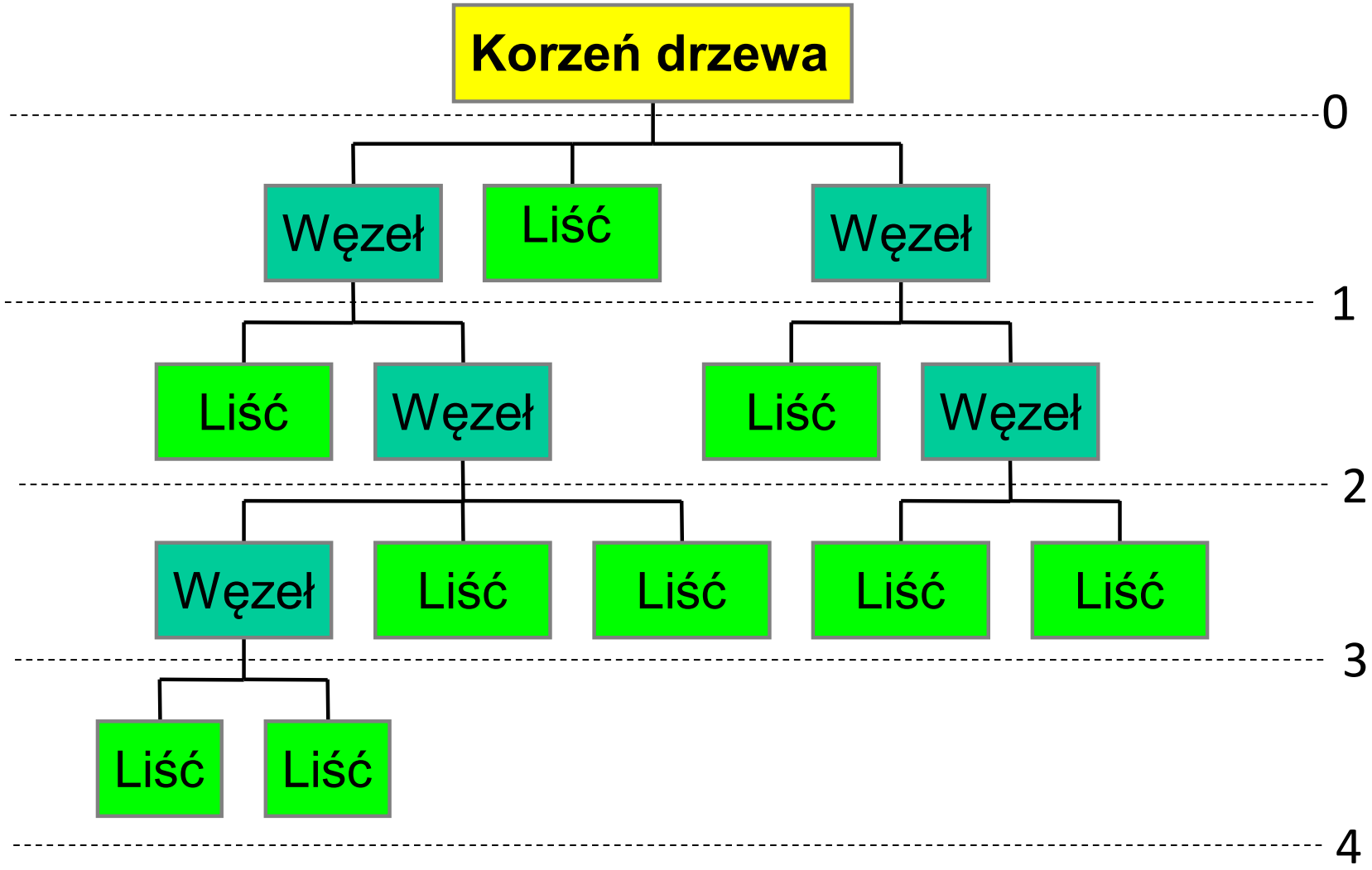
Drzewo decyzyjne jest metodą indukcyjnego uczenia. To technika analizy danych, za pomocą której, na podstawie wartości wybranego zbioru atrybutów objaśniających, prognozuje się wartość zmiennej celu (decyzji), która jest cechą jakościową (klasyfikacyjne drzewo decyzyjne) lub ilościową (regresyjne drzewo decyzyjne).

Drzewo decyzyjne charakteryzuje się strukturą hierarchiczną, złożoną z węzłów, gałęzi i liści. W tej strukturze zbiór danych jest dzielony na podzbiory wg wartości wybranego atrybutu, tworzące węzły. W każdym węźle następuje kolejny podział tworząc węzły niższych poziomów. Proces postępuje tak długo, aż zostanie osiągnięte kryterium stopu. Każdy węzeł i wszystkie jego podwęzły tworzą gałąź drzewa.

W algorytmach konstrukcji drzew jednym z kluczowych elementów jest wybór cech, według których, na poszczególnych etapach, będzie dokonywany podział zbioru danych.

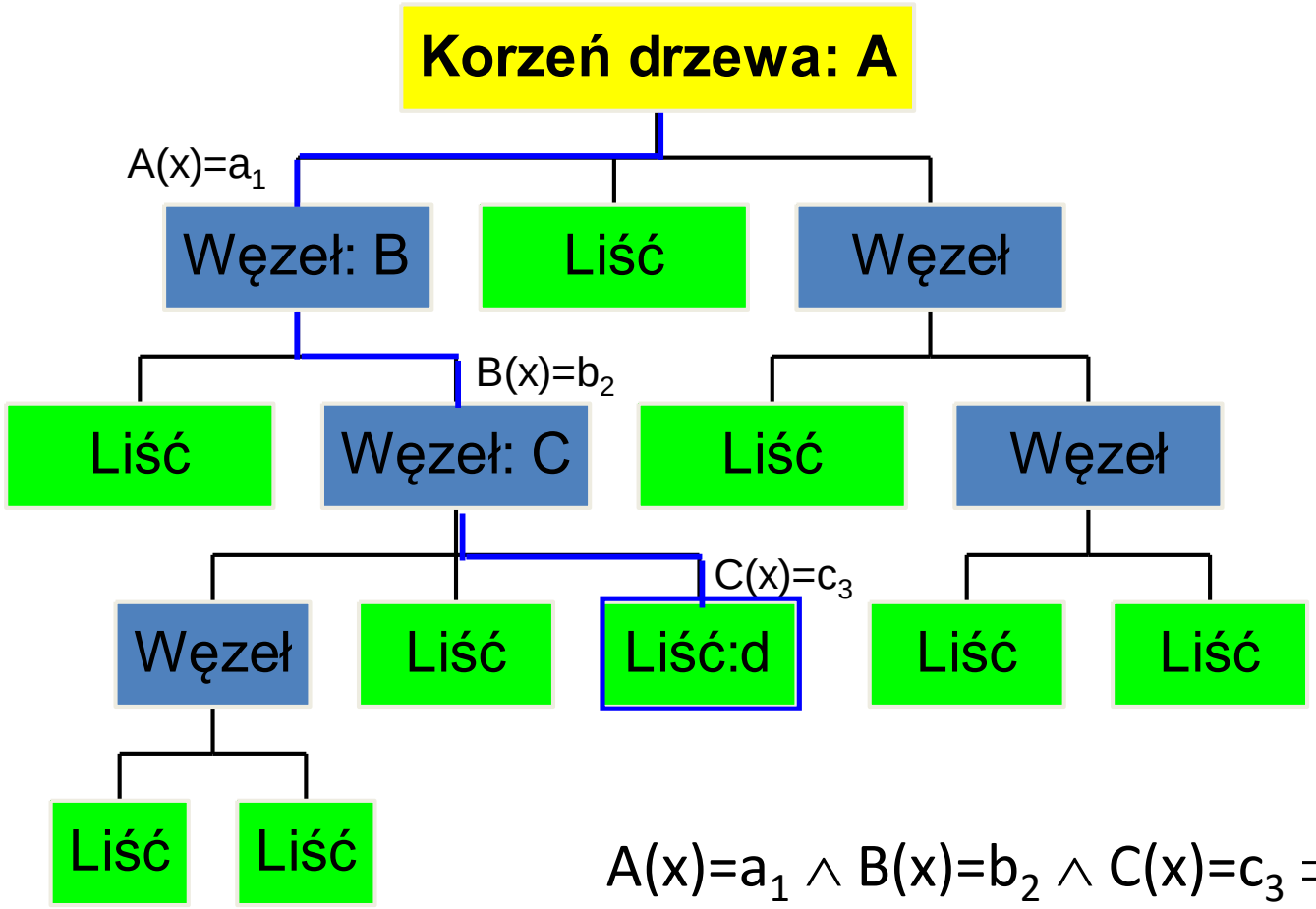
Węzeł, w którym nie ma już podziału zbioru jest nazywany węzłem terminalnym lub inaczej liściem. Liść decyduje jaka jest prognozowana wartość zmiennej celu, jeżeli zbiór zmiennych objaśniających ma określone wartości.

Struktura drzewa decyzyjnego



Jeżeli z węzła wychodzą dwa węzły, to jest drzewo binarne → Python

Interpretacja



$A(x)=a_1 \wedge B(x)=b_2 \wedge C(x)=c_3 \Rightarrow \mathbf{d}$

x – badany przykład, czyli obserwacja, dla której jest dokonywana klasyfikacja do kategorii *d* zmiennej celu czyli decyzji

Jakość klasyfikacji
Tabela klasyfikacji/Macierz pomyłek

	Prognozowane porażki (ujemne)	Prognozowane sukcesy (dodatnie)	Liczebności obserwowane
Obserwowane porażki (negatywne)	**A** (prawdziwie negatywne)	B (fałszywie pozytywne)	**A + B**
Obserwowane sukcesy (pozytywne)	C (fałszywie negtywne)	**D** (prawdziwie pozytywne)	**C + D**

Prawdziwie ujemne (True Negative Rate) TNR= A / (A + B)
TNR = (Liczba porażek sklasyfikowanych jako porażki)/(Ogólna liczba porażek)

Fałszywie dodatnie (False Positive Rate) FPR = B / (A + B)
FPR = (Liczba porażek sklasyfikowanych jako sukcesy)/(Ogólna liczba porażek)

Fałszywie ujemne (False Negative Rate) FNR = C / (C + D)
FNR = (Liczba sukcesów sklasyfikowanych jako porażki)/(Ogólna liczba sukcesów)

Prawdziwie dodatnie (True Positive Rate) TPR = D / (C + D)
PD = (Liczba sukcesów sklasyfikowanych jako sukcesy)/(Ogólna liczba sukcesów)

Odsetek poprawnych klasyfikacji: (A + D) / (A + B + C + D)
Odsetek błędnych klasyfikacji: (B + C) / (A + B + C + D)

Zasięg i precyzja – ocena jakości klasyfikacji

	Prognozowane porażki (ujemne)	Prognozowane sukcesy (dodatnie)	**Liczebności obserwowane**
Obserwowane porażki (negatywne)	**A** (prawdziwie negatywne)	B (fałszywie pozytywne)	**A + B**
Obserwowane sukcesy (pozytywne)	C (fałszywie negtywne)	**D** (prawdziwie pozytywne)	**C + D**
Liczebności prognozowane	**A + C**	**B + D**	

Miary zasięgu (pokrycia, dotarcia)

Specyficzność (Specificity) - Prawdziwie ujemne (True Negative Rate)

$$TNR = A / (A + B)$$

Czułość (Sensitivity) - Prawdziwie dodatnie (True Positive Rate)

$$TPR = D / (C + D)$$

Miary precyzji

Precyzja przewidywania negatywnego (Negative Predictive Value)

$$NPV = A / (A + C)$$

NPV = (Liczba poprawnie sklasyfikowanych porażek)/ (Liczba prognozowanych porażek)

Precyzja przewidywania pozytywnego (Positive Predictive Value)

$$PPV = D / (B + D)$$

PPV = (Liczba poprawnie sklasyfikowanych sukcesów)/(Liczba prognozowanych sukcesów)

Ważność cechy

Wskaźnik ważności cechy (*feature importance*) informuje o znaczeniu cechy w budowaniu drzewa decyzyjnego. Informuje, jak bardzo ta cecha jest używana w tworzeniu drzewa.
Wyższa wartość wskaźnika oznacza, że konkretna cecha będzie miała większy wpływ na postać modelu używanego do przewidywania zmiennej celu.

Formalnie jest ona obliczana jako (znormalizowana) całkowita redukcja zanieczyszczenia wniesiona przez tę cechę. Przyjmuje wartości z przedziału <0, 1>.

Im bliżej jedności wartość wskaźnika tym cecha ma większe znaczenie.

Wyznaczenie ważności cechy

Dla drzewa decyzyjnego w Scikit-learn jest obliczana ważność węzłów za pomocą wskaźnika Gini Importance, zakładając drzewo binarne (dwa węzły podrzędne z węzła drzewa):

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)}$$

ni_j -	ważność węzła j
w_j -	ważona liczba próbek docierających do węzła j
C_j -	wartość zanieczyszczenia węzła j
$left(j)$ -	węzeł podrzędny lewy podziału w węźle j
$right(j)$ -	węzeł podrzędny prawy podziału w węźle j

Znaczenie każdej cechy *i* w drzewie decyzyjnym jest obliczane jako FI_i:

$$\mathrm{FI}_i = \frac{\sum_{j;j\ węzeł,\ w\ którym\ następuje\ podziałe\ wg\ cechy\ i} ni_j}{\sum_{k;k-kolejny\ węzeł\ drzewai} ni_k}$$

Źródło: https://towardsdatascience.com/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3

Drzewo decyzyjne w Pythonie

Zad_07_TitanicTree_DEMO.py – wybrane fragmenty

```
import pandas as pdfrom sklearn import tree
#  Utworzenie klasyfikatora drzewiastego i wykres drzewa decyzyjnego
clf = tree.DecisionTreeClassifier(random_state=1, criterion='entropy',  min_impurity_decrease=0.003, max_depth=4)
clf = clf.fit(X, Y)
# Zmienna celu jest dwuwartościowa i jest identyfikowana jako y.
# Dla tej dwuwartościowej zmiennej celu:
#    y[0] wartość mniejsza (dla uratowanych = -1 -> dead),
#    y[1] wartość większa (dla uratowanych = 1 -> alive)
#  Lewa gałąź reprezentuje odpowiedź True a prawa odpowiedź False dla pytanie definiujące podział zbioru

plt.figure(figsize = (45, 25))
tree.plot_tree(clf, max_depth=4, feature_names=X.columns.values, class_names=["dead :-(", "alive :-)"], rounded=True, filled=True,
proportion = False, fontsize=30)
plt.show()

# Prognoza przeżycia katastrofy dla poszczególnych rekordów (osób)
# Wyznaczenie miar oceny jakości klasyfikacji
from sklearn import metrics                    # Import modułu metrics z scikit-learn w celu obliczenia jakości klasyfikacji
Y_pred = clf.predict(X)
print("Dopasowanie: ", metrics.accuracy_score(Y, Y_pred))

# Ocena jakosci klasyfikacji na podstawie macierzy pomylek
Cm = metrics.confusion_matrix(Y, Y_pred)
disp = metrics.ConfusionMatrixDisplay(confusion_matrix=cm,display_labels=["dead :-(", "alive :-)"])
disp.plot()
plt.show()

# Waznosc cech w tworzeniu klasyfikatora drzewiastego
Importances = clf.feature_importances_
importances_frame = pd.DataFrame({'Feature': ["embarked_kod", "pclass", "sex_kod", "age_kod", "fare_kod"],  'Importance':
importances})
print(importances_frame)
importances_frame.plot(kind = 'bar', y = 'Importance', x="Feature", legend=False, title="Ważność zmiennych")
```