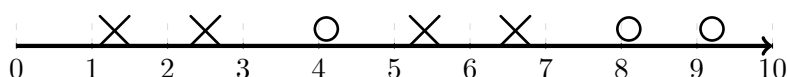


1. Jakie są zalety nauczenia się funkcji $f(x)$ w stosunku do zaimplementowania jej samodzielnie np. w Python. Dla jakich problemów/funkcji uczenie się funkcji jest korzystne, a dla jakich nie?
2. Jakie są wady i zalety rozwiązań korzystających z uczenia maszynowego?
3. Zaproponuj co najmniej 5 cech dla klasyfikatora filtrującego wiadomości e-mail (SPAM/–SPAM).
4. Czy potrafisz podać kilka własności dobrej cechy? Spróbuj odnieść się do poniższych przykładów cech zaprojektowanych dla klasyfikatora kandydatów na studia:
 - `student_id` = 299616
 - `specjalizacja` = SI
 - wiek studenta zakodowany jako ciąg 8 zmiennych binarnych, systemem dwójkowym
 - wynik z rozmowy kwalifikacyjnej = -1 (jeśli do niej nie przystąpiono, w przeciwnym wypadku wynik)
 - L.p. na liście kandydatów = 12

5. Odpowiedz na poniższe pytania wykorzystując poniższy zbiór danych do klasyfikacji binarnej (klasy: \times , \circ) z jedną cechą x oraz zakładając następującą klasę hipotez

$$\mathcal{H} = \{\times \text{ if } x < t \text{ else } \circ : t \in \mathbb{R}\} \cup \{\times \text{ if } x > t \text{ else } \circ : t \in \mathbb{R}\}$$

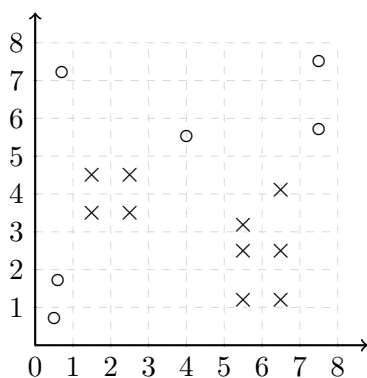


- (a) Który klasyfikator zostanie wybrany zgodnie z zasadą ERM i błędu zero-jedynkowego?
 - (b) W jaki sposób można zaimplementować algorytm wybierający klasyfikator zgodny z ERM dla tej klasy hipotez?
 - (c) Jak wyglądałoby rozszerzenie tego problemu i algorytmu dla zbioru z dwoma cechami?
 - (d) Oszacuj złożoność obliczeniową zaproponowanego rozwiązania.
6. Rozważmy problem klasyfikacji binarnej ze skończoną klasą hipotez \mathcal{H} i założeniem że funkcja f , której chcemy się nauczyć należy do tej klasy hipotez $f \in \mathcal{H}$. W takim wypadku jest możliwe użycie następującego algorytmu uczącego:
 - (i) Odczytaj kolejny przykład uczący (x_i, y_i)
 - (ii) Wyeliminuj z \mathcal{H} wszystkie hipotezy h takie że $h(x_i) \neq y_i$
 - (iii) Powtarzaj powyższe dwa punkty, a na końcu wybierz dowolną hipotezę która pozostała w \mathcal{H} .

Odpowiedz na poniższe pytania:

- (a) Zakładając n -elementowy zbiór niezależnych danych uczących, podaj wzór na prawdopodobieństwo że w ostatecznym zbiorze \mathcal{H} będzie hipoteza z błędem klasyfikacji większym niż $\epsilon \in [0, 1]$. (Innymi słowy: jak jest szansa że ten algorytm dostarczy klasyfikator z błędem większym niż ϵ .)
 - (b) Co najmniej ilu przykładów uczących n potrzebujesz, abyś miał 90% pewność, że algorytm nie zwróci klasyfikatora z błędem większym niż ϵ ?
7. Zbuduj drzewo decyzyjne dla poniższego zbioru danych.





8. Skonstruuj drzewo decyzyjne na podstawie poniższych danych.

Outlook	Windy	Play?
słonecznie	false	o
słonecznie	true	o
pochmurnie	false	x
deszcz	false	x
deszcz	false	x
deszcz	true	o
pochmurnie	true	x
słonecznie	false	o
słonecznie	false	x
deszcz	false	x
słonecznie	true	x
pochmurnie	true	x
pochmurnie	false	x
deszcz	true	o

9. Rozważmy tworzenie podziału zbioru w drzewie decyzyjnym (klasyfikacja binarna) w którym prawdopodobieństwo klasy + wynosi $p = 0.4$, a po podziale binarnym uzyskano podzbiory z prawdopodobieństwami klasy + odpowiednio: $p_1 = 0.1$ i $p_2 = 0.7$.

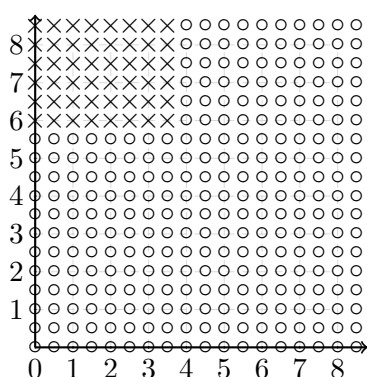
Wskazówka: pomocne mogą być następujące oznaczenia:

$$p = \frac{n_+}{n} \quad p_1 = \frac{n_{1,+}}{n_1} \quad p_2 = \frac{n_{2,+}}{n_2} \quad x = \frac{n_1}{n}$$

gdzie n to liczba przykładów przed podziałem, n_1 to liczba przykładów w pierwszym podzbiorze, a $n_{1,+}$ to liczba przykładów w pierwszym podzbiorze należących do klasy +.

- Ile procent przykładów znalazło się w pierwszym podzbiorze (x)?
- Ile wynosi błąd klasyfikacji w tym zbiorze przed i po wykonaniu tego podziału?

10. Zbuduj drzewo decyzyjne dla poniższego zbioru danych.



Fundusze Europejskie
Polska Cyfrowa



**Rzeczpospolita
Polska**

Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego



11. Wymień wady i zalety stosowania drzew decyzyjnych.
12. Korzystając z poniższego wzorca zaimplementuj drzewo decyzyjne zgodnie z zasadą ERM dla problemu regresji.

```
class Node(object):
    def __init__(self):
        #Zmienne przechowujące odpowiedni lewe i prawe dziecko wierzchołka
        self.left = None
        self.right = None

    def perform_split(self, data):
        # Znajdź najlepszy podział danych (data)
        if # uzyskano poprawę funkcji celu (lub inny, zaproponowany przez Ciebie w
            #podziel dane na dwie części d1 i d2, zgodnie z warunkiem
            self.left = Node()
            self.right = Node()
            self.left.perform_split(d1)
            self.right.perform_split(d2)
        else:
            #obecny Node jest liściem, zapisz jego wartość

    def predict(self, example):
        if # Node nie jest liściem:
            if warunek podziału jest spełniony:
                return self.right.predict(example)
            else:
                return self.left.predict(example)
        return zwroc wartosc liscia
```

