

Poglavlje 7

Pascalov svijet

7.1 Vjerojatnost kao logika neizvjesnosti

U ovom poglavlju istražujemo **teoriju vjerojatnosti** kao prirodno proširenje logike sudova, inspirirani Pascalovim pionirskim radom o igrama na sreću i njegovim filozofskim razmatranjima o neizvjesnosti.

"Srce ima svoje razloge koje razum ne poznaje" (*Le cœur a ses raisons que la raison ne connaît point*) - Blaise Pascal

Pascal je, zajedno s Fermatom, postavio temelje teorije vjerojatnosti 1654. godine, ali njegov doprinos seže dalje - pokazao je kako matematički pristupiti neizvjesnosti i racionalnom odlučivanju.

7.1.1 Od logike sudova do vjerojatnosti

U klasičnoj logici, **elementarni sud** može biti samo istinit ($\top$) ili lažan ($\perp$). Vjerojatnost generalizira ovaj koncept dopuštajući **stupnjeve istinitosti** između 0 i 1:

- Klasična logika: $v(\varphi) \in \{\perp, \top\}$
- Vjerojatnost: $P(\varphi) \in [0, 1]$

gdje $P(\varphi) = 0$ odgovara $\perp$, a $P(\varphi) = 1$ odgovara $\top$.

```
1 class Sud:
2     """Predstavlja sud s klasičnom ili vjerojatnosnom vrijednošću."""
3
4     def __init__(self, naziv, klasicna_vrijednost=None, vjerojatnost=None):
```

```

5     self.naziv = naziv
6     self.klasicna = klasicna_vrijednost #  $\top$  ili  $\perp$ 
7     self.vjerojatnost = vjerojatnost    # [0, 1]
8
9     def __repr__(self):
10        if self.vjerojatnost is not None:
11            return f"P('{self.naziv}') = {self.vjerojatnost:.2f}"
12        elif self.klasicna is not None:
13            return f"Sud '{self.naziv}': {'T' if self.klasicna else '\perp'}"
14        return f"Sud '{self.naziv}': neodređen"
15
16 # Primjeri sudova
17 print("Usporedba logike i vjerojatnosti:")
18 print("="*34)
19
20 # Klasična logika
21 print("\nKlasična logika:")
22 kisa_klasicno = Sud("Pada kiša", klasicna_vrijednost=True)
23 sunce_klasicno = Sud("Sunčano je", klasicna_vrijednost=False)
24 print(f" {kisa_klasicno}")
25 print(f" {sunce_klasicno}")
26
27 # Vjerojatnosna logika
28 print("\nVjerojatnosna logika:")
29 kisa_vjerojatnost = Sud("Pada kiša", vjerojatnost=0.3)
30 sunce_vjerojatnost = Sud("Sunčano je", vjerojatnost=0.7)
31 oblacno_vjerojatnost = Sud("Oblačno", vjerojatnost=0.45)
32 print(f" {kisa_vjerojatnost}")
33 print(f" {sunce_vjerojatnost}")
34 print(f" {oblacno_vjerojatnost}")
35
36 # Interpretacija
37 print("\nInterpretacija:")
38 interpretacije = [
39     (0.0, " $\perp$  (sigurno lažno)"),
40     (0.3, "stupanj vjerovanja 30%"),
41     (0.5, "potpuna neizvjesnost"),
42     (0.7, "stupanj vjerovanja 70%"),
43     (1.0, " $\top$  (sigurno istinito)")
44 ]
45 for p, opis in interpretacije:
46     print(f" P = {p:.2f}  $\approx$  {opis}")

```

Output

```

Usporedba logike i vjerojatnosti:
=====

Klasična logika:
  Sud 'Pada kiša':  $\top$ 
  Sud 'Sunčano je':  $\perp$ 

```

Vjerojatnosna logika:

$P(\text{'Pada kiša'}) = 0.30$

$P(\text{'Sunčano je'}) = 0.70$

$P(\text{'Oblačno'}) = 0.45$

Interpretacija:

$P = 0.00 \approx \perp$ (sigurno lažno)

$P = 0.30 \approx$ stupanj vjerovanja 30%

$P = 0.50 \approx$ potpuna neizvjesnost

$P = 0.70 \approx$ stupanj vjerovanja 70%

$P = 1.00 \approx \top$ (sigurno istinito)

7.1.2 Kolmogorovljevi aksiomi vjerojatnosti

Andrej Kolmogorov je 1933. formalizirao teoriju vjerojatnosti kroz tri elegantna aksioma. Za skup svih mogućih sudova Ω i vjerojatnosnu mjeru P :

1. **Nenegativnost:** $P(\varphi) \geq 0$ za svaki sud φ
2. **Normalizacija:** $P(\Omega) = 1$ (sigurna istina ima vjerojatnost 1)
3. **Aditivnost:** Za međusobno isključive sudove $\varphi_1, \varphi_2, \dots$:

$$P(\varphi_1 \vee \varphi_2 \vee \dots) = P(\varphi_1) + P(\varphi_2) + \dots$$

```

1 class VjerojatnosniProstor:
2     """Predstavlja vjerojatnosni prostor s Kolmogorovljevim aksiomima."""
3
4     def __init__(self, omega, vjerojatnosti):
5         """
6         omega: skup elementarnih sudova
7         vjerojatnosti: rječnik {sud: vjerojatnost}
8         """
9         self.omega = omega
10        self.P = vjerojatnosti
11        self._provjeri_aksiome()
12
13    def _provjeri_aksiome(self):
14        """Provjerava zadovoljavaju li vjerojatnosti aksiome."""
15        # Aksiom 1: Negativnost
16        for p in self.P.values():
17            if p < 0:
18                raise ValueError(f"Vjerojatnost {p} krši aksiom nenegativnosti!")
19
20        # Aksiom 2: Normalizacija
21        suma = sum(self.P.values())
22        if abs(suma - 1.0) > 1e-10:

```

```

23         raise ValueError(f"Suma vjerojatnosti {suma}  $\neq$  1 (krši normalizaciju!)")
24
25     def vjerojatnost_suda(self, predikat):
26         """Računa vjerojatnost složenog suda."""
27         p = 0
28         for ishod, vjerojatnost in self.P.items():
29             if predikat(ishod):
30                 p += vjerojatnost
31         return p
32
33 # Primjer: Poštena kocka
34 kocka_omega = {1, 2, 3, 4, 5, 6}
35 kocka_P = {i: 1/6 for i in kocka_omega}
36 kocka = VjerojatnosniProstor(kocka_omega, kocka_P)
37
38 print("Demonstracija Kolmogorovljevih aksioma")
39 print("="*39)
40 print("\nProstor elementarnih sudova  $\Omega$ :")
41 print(f"   Kocka pokazuje: {kocka_omega}")
42
43 # Aksiom 1: Nenegativnost
44 print("\nAKSIOM 1 - Nenegativnost:")
45 testovi = [
46     ("Paran broj", lambda x: x % 2 == 0),
47     ("Broj > 4", lambda x: x > 4),
48     ("Broj = 7", lambda x: x == 7)
49 ]
50 for naziv, predikat in testovi:
51     p = kocka.vjerojatnost_suda(predikat)
52     print(f"   P('{naziv}') = {p:.3f}  $\geq$  0 ✓")
53
54 # Aksiom 2: Normalizacija
55 print("\nAKSIOM 2 - Normalizacija:")
56 p_omega = kocka.vjerojatnost_suda(lambda x: True)
57 print(f"   P( $\Omega$ ) = P('Bilo koji broj 1-6') = {p_omega:.3f} ✓")
58
59 # Aksiom 3: Aditivnost
60 print("\nAKSIOM 3 - Aditivnost:")
61 print("   Međusobno isključivi sudovi:")
62 p1 = kocka.vjerojatnost_suda(lambda x: x == 1)
63 p3 = kocka.vjerojatnost_suda(lambda x: x == 3)
64 p5 = kocka.vjerojatnost_suda(lambda x: x == 5)
65 print(f"   P('Broj = 1') = {p1:.3f}")
66 print(f"   P('Broj = 3') = {p3:.3f}")
67 print(f"   P('Broj = 5') = {p5:.3f}")
68
69 p_unija = kocka.vjerojatnost_suda(lambda x: x in {1, 3, 5})
70 p_suma = p1 + p3 + p5
71
72 print(f"   \n   P('Broj  $\in$  {{1,3,5}}') = {p_unija:.3f}")
73 print(f"   P('1') + P('3') + P('5') = {p_suma:.3f}")
74 print(f"   Aditivnost vrijedi: {p_unija:.3f} = {p_suma:.3f} ✓")
75

```

```

76 print("\nPosljedice aksioma:")
77 print(" P( $\neg\varphi$ ) = 1 - P( $\varphi$ )")
78 print(" P( $\emptyset$ ) = 0")
79 print(" P( $\varphi \vee \psi$ ) = P( $\varphi$ ) + P( $\psi$ ) - P( $\varphi \wedge \psi$ )")

```

Output

```

Demonstracija Kolmogorovljevih aksioma
=====

Prostor elementarnih sudova  $\Omega$ :
  Kocka pokazuje: {1, 2, 3, 4, 5, 6}

AKSIOM 1 - Nenegativnost:
  P('Paran broj') = 0.500  $\geq$  0 ✓
  P('Broj > 4') = 0.333  $\geq$  0 ✓
  P('Broj = 7') = 0.000  $\geq$  0 ✓

AKSIOM 2 - Normalizacija:
  P( $\Omega$ ) = P('Bilo koji broj 1-6') = 1.000 ✓

AKSIOM 3 - Aditivnost:
  Međusobno isključivi sudovi:
    P('Broj = 1') = 0.167
    P('Broj = 3') = 0.167
    P('Broj = 5') = 0.167

  P('Broj  $\in$  {1,3,5}') = 0.500
  P('1') + P('3') + P('5') = 0.500
  Aditivnost vrijedi: 0.500 = 0.500 ✓

Posljedice aksioma:
  P( $\neg\varphi$ ) = 1 - P( $\varphi$ )
  P( $\emptyset$ ) = 0
  P( $\varphi \vee \psi$ ) = P( $\varphi$ ) + P( $\psi$ ) - P( $\varphi \wedge \psi$ )

```

7.1.3 Vjerojatnost kao proširenje logičkih veznika

Vjerojatnosni račun generalizira logičke veznike:

- **Negacija:** $P(\neg\varphi) = 1 - P(\varphi)$
- **Disjunkcija:** $P(\varphi \vee \psi) = P(\varphi) + P(\psi) - P(\varphi \wedge \psi)$
- **Konjunkcija:** $P(\varphi \wedge \psi) = P(\varphi) \cdot P(\psi|\varphi)$

Kada su sudovi **nezavisni**: $P(\varphi \wedge \psi) = P(\varphi) \cdot P(\psi)$

```

1 class VjerojatnosnaLogika:
2     """Implementira vjerojatnosne verzije logičkih veznika."""
3
4     @staticmethod
5     def negacija(p_phi):
6         """ $P(\neg\varphi) = 1 - P(\varphi)$ """
7         return 1 - p_phi
8
9     @staticmethod
10    def disjunkcija(p_phi, p_psi, p_phi_i_psi):
11        """ $P(\varphi \vee \psi) = P(\varphi) + P(\psi) - P(\varphi \wedge \psi)$ """
12        return p_phi + p_psi - p_phi_i_psi
13
14    @staticmethod
15    def konjunkcija_nezavisna(p_phi, p_psi):
16        """ $P(\varphi \wedge \psi) = P(\varphi) \cdot P(\psi)$  za nezavisne sudove"""
17        return p_phi * p_psi
18
19    @staticmethod
20    def uvjetna_vjerojatnost(p_phi_i_psi, p_psi):
21        """ $P(\varphi|\psi) = P(\varphi \wedge \psi) / P(\psi)$ """
22        if p_psi == 0:
23            return None # Nedefinirana
24        return p_phi_i_psi / p_psi
25
26    @staticmethod
27    def implikacija(p_phi, p_psi, p_phi_i_psi):
28        """ $P(\varphi \rightarrow \psi) = P(\neg\varphi \vee \psi)$ """
29        p_neg_phi = 1 - p_phi
30        #  $P(\neg\varphi \wedge \psi) = P(\psi) - P(\varphi \wedge \psi)$ 
31        p_neg_phi_i_psi = p_psi - p_phi_i_psi
32        return p_neg_phi + p_psi - p_neg_phi_i_psi
33
34    # Primjer: Vremenske prilike
35    vl = VjerojatnosnaLogika()
36
37    # Definiraj vjerojatnosti
38    P_kisa = 0.6      # P(pada kiša)
39    P_hladno = 0.4    # P(hladno je)
40    P_kisa_i_hladno = 0.3 # P(pada kiša ∧ hladno je)
41
42    print("Vjerojatnosni logički veznici")
43    print("="*30)
44    print("\nOsnovni sudovi:")
45    print(f"  P(A) = {P_kisa:.2f}  ('Pada kiša')")
46    print(f"  P(B) = {P_hladno:.2f}  ('Hladno je')")
47    print(f"  P(A ∧ B) = {P_kisa_i_hladno:.2f}  ('Pada kiša i hladno je')")
48
49    # Negacija
50    print("\n1. NEGACIJA:")
51    P_ne_kisa = vl.negacija(P_kisa)
52    print(f"  P(¬A) = 1 - P(A) = 1 - {P_kisa:.2f} = {P_ne_kisa:.2f}")

```

```

53 print(f" Interpretacija: Vjerojatnost da ne pada kiša je {P_ne_kisa*100:.0f}%")
54
55 # Disjunkcija
56 print("\n2. DISJUNKCIJA:")
57 P_kisa_ili_hladno = vl.disjunkcija(P_kisa, P_hladno, P_kisa_i_hladno)
58 print(f"  $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$ ")
59 print(f"  $P(A \vee B) = \{P\_kisa:.2f\} + \{P\_hladno:.2f\} - \{P\_kisa\_i\_hladno:.2f\} =$ 
    ↳  $\{P\_kisa\_ili\_hladno:.2f\}$ ")
60 print(f" Interpretacija: Vjerojatnost da pada kiša ili je hladno je
    ↳  $\{P\_kisa\_ili\_hladno*100:.0f\}%")$ ")
61
62 # Implikacija
63 print("\n3. IMPLIKACIJA:")
64 P_ako_kisa_onda_hladno = vl.implikacija(P_kisa, P_hladno, P_kisa_i_hladno)
65 print(f"  $P(A \rightarrow B) = P(\neg A \vee B) = P(\neg A) + P(B) - P(\neg A \wedge B)$ ")
66 P_ne_kisa_i_hladno = P_hladno - P_kisa_i_hladno
67 print(f"  $P(A \rightarrow B) = \{P\_ne\_kisa:.2f\} + \{P\_hladno:.2f\} - \{P\_ne\_kisa\_i\_hladno:.2f\} =$ 
    ↳  $\{P\_ako\_kisa\_onda\_hladno:.2f\}$ ")
68 print(f" Interpretacija: Vjerojatnost da 'ako pada kiša, onda je hladno' je
    ↳  $\{P\_ako\_kisa\_onda\_hladno*100:.0f\}%")$ ")
69
70 # Uvjetna vjerojatnost
71 print("\n4. UVJETNA VJEROJATNOST:")
72 P_hladno_ako_kisa = vl.uvjetna_vjerojatnost(P_kisa_i_hladno, P_kisa)
73 print(f"  $P(B|A) = P(A \wedge B) / P(A) = \{P\_kisa\_i\_hladno:.2f\} / \{P\_kisa:.2f\} =$ 
    ↳  $\{P\_hladno\_ako\_kisa:.2f\}$ ")
74 print(f" Interpretacija: Ako pada kiša, vjerojatnost da je hladno je
    ↳  $\{P\_hladno\_ako\_kisa*100:.0f\}%")$ ")
75
76 # Test nezavisnosti
77 print("\nTest nezavisnosti:")
78 P_nezavisno = vl.konjunkcija_nezavisna(P_kisa, P_hladno)
79 print(f"  $P(A) \cdot P(B) = \{P\_kisa:.2f\} \cdot \{P\_hladno:.2f\} = \{P\_nezavisno:.2f\}$ ")
80 print(f"  $P(A \wedge B) = \{P\_kisa\_i\_hladno:.2f\}$ ")
81 if abs(P_nezavisno - P_kisa_i_hladno) < 0.01:
82     print(f" Budući da  $\{P\_nezavisno:.2f\} \approx \{P\_kisa\_i\_hladno:.2f\}$ , sudovi su približno
        ↳ nezavisni")
83 else:
84     print(f" Budući da  $\{P\_nezavisno:.2f\} \neq \{P\_kisa\_i\_hladno:.2f\}$ , sudovi NISU nezavisni")
85     print(" (Kiša i hladnoća su povezani!)")

```

Output

Vjerojatnosni logički veznici

=====

Osnovni sudovi:

$P(A) = 0.60$ ('Pada kiša')

$P(B) = 0.40$ ('Hladno je')

$P(A \wedge B) = 0.30$ ('Pada kiša i hladno je')

1. NEGACIJA:

$$P(\neg A) = 1 - P(A) = 1 - 0.60 = 0.40$$

Interpretacija: Vjerojatnost da ne pada kiša je 40%

2. DISJUNKCIJA:

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

$$P(A \vee B) = 0.60 + 0.40 - 0.30 = 0.70$$

Interpretacija: Vjerojatnost da pada kiša
ili je hladno je 70%

3. IMPLIKACIJA:

$$P(A \rightarrow B) = P(\neg A \vee B) = P(\neg A) + P(B) - P(\neg A \wedge B)$$

$$P(A \rightarrow B) = 0.40 + 0.40 - 0.10 = 0.70$$

Interpretacija: $P(\text{'ako pada kiša, onda je hladno'}) = 70\%$

4. UVJETNA VJEROJATNOST:

$$P(B|A) = P(A \wedge B) / P(A) = 0.30 / 0.60 = 0.50$$

Interpretacija: Ako pada kiša, vjerojatnost da je hladno je 50%

Test nezavisnosti:

$$P(A) \cdot P(B) = 0.60 \cdot 0.40 = 0.24$$

$$P(A \wedge B) = 0.30$$

Budući da $0.24 \neq 0.30$, sudovi NISU nezavisni

(Kiša i hladnoća su povezani!)

7.1.4 Bayesov teorem - zaključivanje s neizvjesnošću

Bayesov teorem omogućava ažuriranje vjerovanja na temelju novih dokaza:

"Vjerojatnost je vodič života" - Ciceron, anticipirajući Bayesovu logiku

$$P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E)}$$

gdje je:

- $P(H)$ - **apriorna vjerojatnost** hipoteze
- $P(E|H)$ - **izglednost** dokaza ako je hipoteza istinita
- $P(H|E)$ - **aposteriorna vjerojatnost** nakon dokaza

```
1 class BayesovoZakljucivanje:
2     """Implementira Bayesovo ažuriranje vjerojatnosti."""
3
```



```

4  def __init__(self, apriorna, izglednosti):
5      """
6      apriorna: dict {hipoteza: P(hipoteza)}
7      izglednosti: dict {hipoteza: {dokaz: P(dokaz/hipoteza)}}
8      """
9      self.apriorna = apriorna
10     self.izglednosti = izglednosti
11
12     def azuriraj(self, dokaz):
13         """Ažurira vjerojatnosti na temelju novog dokaza."""
14         # Izračunaj P(dokaz)
15         p_dokaz = 0
16         for hipoteza, p_h in self.apriorna.items():
17             p_dokaz += self.izglednosti[hipoteza][dokaz] * p_h
18
19         # Bayesovo ažuriranje za svaku hipotezu
20         aposteriorna = {}
21         for hipoteza, p_h in self.apriorna.items():
22             p_dokaz_ako_h = self.izglednosti[hipoteza][dokaz]
23             aposteriorna[hipoteza] = (p_dokaz_ako_h * p_h) / p_dokaz
24
25         return aposteriorna, p_dokaz
26
27     # Primjer: Medicinska dijagnoza
28     print("Bayesovo zaključivanje - Medicinska dijagnoza")
29     print("="*46)
30     print("\nScenarij: Test za rijetku bolest\n")
31
32     # Definiraj vjerojatnosti
33     apriorna = {
34         "bolest": 0.001,      # 1 od 1000 ljudi ima bolest
35         "zdrav": 0.999
36     }
37
38     izglednosti = {
39         "bolest": {
40             "pozitivan": 0.99, # Osjetljivost testa (true positive rate)
41             "negativan": 0.01  # False negative rate
42         },
43         "zdrav": {
44             "pozitivan": 0.05, # False positive rate
45             "negativan": 0.95  # True negative rate
46         }
47     }
48
49     bayes = BayesovoZakljucivanje(apriorna, izglednosti)
50
51     print("Početne informacije:")
52     print(f" P(Bolest) = {apriorna['bolest']:.3f} (1 od 1000 ljudi ima bolest)")
53     print(f" P(Test+|Bolest) = {izglednosti['bolest']['pozitivan']:.3f} (osjetljivost testa)")
54     print(f" P(Test+|¬Bolest) = {izglednosti['zdrav']['pozitivan']:.3f} (lažno pozitivna
55         ↪ stopa)")

```

```

56 print("\nPitanje: Ako je test pozitivan, kolika je vjerojatnost bolesti?")
57
58 # Bayesovo ažuriranje
59 aposteriorna, p_pozitivan = bayes.azuriraj("pozitivan")
60
61 print("\nBayesov račun:")
62 print("1.  $P(\text{Test+}) = P(\text{Test+}|\text{Bolest}) \cdot P(\text{Bolest}) + P(\text{Test+}|\neg\text{Bolest}) \cdot P(\neg\text{Bolest})$ ")
63 p1 = izglednosti['bolest']['pozitivan'] * apriorna['bolest']
64 p2 = izglednosti['zdrav']['pozitivan'] * apriorna['zdrav']
65 print(f"     $P(\text{Test+}) = \{\text{izglednosti['bolest']['pozitivan']:.3f}\} \cdot \{\text{apriorna['bolest']:.3f}\} + "$ 
66        $f\{\text{izglednosti['zdrav']['pozitivan']:.3f}\} \cdot \{\text{apriorna['zdrav']:.3f}\})$ ")
67 print(f"     $P(\text{Test+}) = \{p1:.5f\} + \{p2:.5f\} = \{p\_pozitivan:.5f\}$ ")
68
69 print("\n2.  $P(\text{Bolest}|\text{Test+}) = P(\text{Test+}|\text{Bolest}) \cdot P(\text{Bolest}) / P(\text{Test+})$ ")
70 print(f"     $P(\text{Bolest}|\text{Test+}) = \{\text{izglednosti['bolest']['pozitivan']:.3f}\} \cdot "$ 
71        $f\{\text{apriorna['bolest']:.3f}\} / \{p\_pozitivan:.5f\}$ ")
72 print(f"     $P(\text{Bolest}|\text{Test+}) = \{\text{aposteriorna['bolest']:.5f}\}$ ")
73
74 print("\nZAKLJUČAK:")
75 print(f"    Apriorna vjerojatnost:  $\{\text{apriorna['bolest']} \cdot 100:.1f\}\%$ ")
76 print(f"    Aposteriorna vjerojatnost (nakon pozitivnog testa):  

    ↪  $\{\text{aposteriorna['bolest']} \cdot 100:.1f\}\%$ ")
77 print("    \n Iako je test vrlo pouzdan (99% osjetljivost),")
78 print(f"    pozitivan test povećava vjerojatnost bolesti samo na  

    ↪  $\{\text{aposteriorna['bolest']} \cdot 100:.1f\}\%$ !")
79 print("    \n Razlog: Bolest je vrlo rijetka, pa većina pozitivnih")
80 print("    testova su lažno pozitivni.")
81
82 # Vizualizacija
83 print("\nVizualizacija od 100,000 ljudi:")
84 n_ljudi = 100000
85 n_bolesnih = int(n_ljudi * apriorna['bolest'])
86 n_zdravih = n_ljudi - n_bolesnih
87
88 tp = int(n_bolesnih * izglednosti['bolest']['pozitivan']) # True positive
89 fn = n_bolesnih - tp # False negative
90 fp = int(n_zdravih * izglednosti['zdrav']['pozitivan']) # False positive
91 tn = n_zdravih - fp # True negative
92
93 print(f"    Bolesni:  $\{n\_bolesnih\}$  ljudi")
94 print(f"    → Pozitivan test:  $\{tp\}$  (istinito pozitivni)")
95 print(f"    → Negativan test:  $\{fn\}$  (lažno negativan)")
96 print(f"    \n Zdravi:  $\{n\_zdravih\}$  ljudi")
97 print(f"    → Pozitivan test:  $\{fp\}$  (lažno pozitivni)")
98 print(f"    → Negativan test:  $\{tn\}$  (istinito negativni)")
99 print(f"    \n Ukupno pozitivnih testova:  $\{tp + fp\}$ ")
100 print(f"    Od toga stvarno bolesnih:  $\{tp\}$ ")
101 print(f"    Vjerojatnost bolesti kod pozitivnog testa:  $\{tp\}/\{tp+fp\} = \{tp/(tp+fp) \cdot 100:.1f\}\%$ ")

```

Output

Bayesovo zaključivanje - Medicinska dijagnoza

=====

Scenarij: Test za rijetku bolest

Početne informacije:

$P(\text{Bolest}) = 0.001$ (1 od 1000 ljudi ima bolest)

$P(\text{Test+}|\text{Bolest}) = 0.990$ (osjetljivost testa)

$P(\text{Test+}|\neg\text{Bolest}) = 0.050$ (lažno pozitivna stopa)

Pitanje: Ako je test pozitivan, kolika je vjerojatnost bolesti?

Bayesov račun:

$$1. P(\text{Test+}) = P(\text{Test+}|\text{Bolest}) \cdot P(\text{Bolest}) + P(\text{Test+}|\neg\text{Bolest}) \cdot P(\neg\text{Bolest})$$

$$P(\text{Test+}) = 0.990 \cdot 0.001 + 0.050 \cdot 0.999$$

$$P(\text{Test+}) = 0.00099 + 0.04995 = 0.05094$$

$$2. P(\text{Bolest}|\text{Test+}) = P(\text{Test+}|\text{Bolest}) \cdot P(\text{Bolest}) / P(\text{Test+})$$

$$P(\text{Bolest}|\text{Test+}) = 0.990 \cdot 0.001 / 0.05094$$

$$P(\text{Bolest}|\text{Test+}) = 0.01943$$

ZAKLJUČAK:

Apriorna vjerojatnost: 0.1%

Aposteriorna vjerojatnost (nakon pozitivnog testa): 1.9%

Iako je test vrlo pouzdan (99% osjetljivost),
pozitivan test povećava vjerojatnost bolesti samo na 1.9%!

Razlog: Bolest je vrlo rijetka, pa većina pozitivnih
testova su lažno pozitivni.

Vizualizacija od 100,000 ljudi:

Bolesni: 100 ljudi

→ Pozitivan test: 99 (istinito pozitivni)

→ Negativan test: 1 (lažno negativan)

Zdravi: 99,900 ljudi

→ Pozitivan test: 4,995 (lažno pozitivni)

→ Negativan test: 94,905 (istinito negativni)

Ukupno pozitivnih testova: 5,094

Od toga stvarno bolesnih: 99

Vjerojatnost bolesti kod pozitivnog testa: $99/5094 = 1.9\%$

7.1.5 Pascalova oklada - filozofija vjerojatnosti

Pascal je primijenio vjerojatnosno razmišljanje na ultimativno filozofsko pitanje - postojanje Boga:

"Morate se kladiti. To nije dobrovoljno, primorani ste na to" - Pascal, *Pensées*

Pascalova matrica odlučivanja:

		Bog postoji		Bog ne postoji							Vjerujem		$+\infty$
(vječna sreća)		$-c$ (mali gubitak)			Ne vjerujem		$-\infty$ (vječna kazna)		$+c$ (mali dobitak)				

Čak i ako je $P(\text{Bog postoji}) = \epsilon$ vrlo mala, očekivana korisnost vjerovanja je beskonačna!

```

1 import math
2
3 def fmt_prob(x: float) -> str:
4     return f"{x:.2f}" if x >= 0.01 else f"{x:.4f}"
5
6 class PascalovaOklada:
7     """Simulacija Pascalove oklade kroz teoriju odlučivanja."""
8
9     def __init__(self):
10         # Matrica korisnosti (payoff matrix)
11         # Koristimo velike brojeve umjesto beskonačnosti za demonstraciju
12         self.korisnost = {
13             ("vjeruj", "postoji"): float('inf'),      # Vječna sreća
14             ("vjeruj", "ne_postoji"): -10,            # Mali gubitak (trud vjerovanja)
15             ("ne_vjeruj", "postoji"): float('-inf'),   # Vječna kazna
16             ("ne_vjeruj", "ne_postoji"): 10           # Mali dobitak (sloboda)
17         }
18
19     def ocekivana_korisnost(self, akcija, p_bog):
20         """Računa očekivanu korisnost akcije."""
21         k_postoji = self.korisnost[(akcija, "postoji")]
22         k_ne_postoji = self.korisnost[(akcija, "ne_postoji")]
23
24         return p_bog * k_postoji + (1 - p_bog) * k_ne_postoji
25
26     def optimalna_odluka(self, p_bog):
27         """Vraća optimalnu odluku za danu vjerojatnost."""
28         eu_vjeruj = self.ocekivana_korisnost("vjeruj", p_bog)
29         eu_ne_vjeruj = self.ocekivana_korisnost("ne_vjeruj", p_bog)
30
31         if eu_vjeruj > eu_ne_vjeruj:
32             return "vjeruj", eu_vjeruj, eu_ne_vjeruj
33         elif eu_ne_vjeruj > eu_vjeruj:
34             return "ne_vjeruj", eu_vjeruj, eu_ne_vjeruj
35         else:
36             return "svejedno", eu_vjeruj, eu_ne_vjeruj
37
38 # Demonstracija
39 oklada = PascalovaOklada()
40
41 print("Pascalova oklada - Analiza odlučivanja")

```

```

42 print("="*39)
43 print("\nMatrica korisnosti:")
44 print("          Bog postoji    Bog ne postoji")
45 print("  Vjerujem:      +∞          -10          ")
46 print("  Ne vjerujem:    -∞          +10          ")
47
48 print("\nAnaliza za različite vjerojatnosti postojanja Boga:\n")
49
50 vjerojatnosti = [0.5, 0.1, 0.01, 0.0001]
51 for p in vjerojatnosti:
52     odluka, eu_v, eu_nv = oklada.optimalna_odluka(p)
53
54     # Example usage (replace your print block with this structure)
55     eu_v_str = f"{eu_v:.2f}" if (eu_v not in (float('inf'), float('-inf')))) else ("∞" if
    ↪ eu_v > 0 else "-∞")
56     eu_nv_str = f"{eu_nv:.2f}" if (eu_nv not in (float('inf'), float('-inf')))) else ("∞" if
    ↪ eu_nv > 0 else "-∞")
57
58     p_str = fmt_prob(p)
59     one_minus_p_str = fmt_prob(1 - p)
60
61     print(f"  E[vjerujem] = {p_str} · ∞ + {one_minus_p_str} · (-10) = {eu_v_str}")
62     print(f"  E[ne vjerujem] = {p_str} · (-∞) + {one_minus_p_str} · 10 = {eu_nv_str}")
63     print(f"  → Racionalna odluka: {'VJERUJ' if odluka == 'vjeruj' else 'NE VJERUJ'}\n")
64
65 print("Pascalov zaključak:")
66 print("  Čak i uz infinitezimalnu vjerojatnost postojanja Boga,")
67 print("  racionalno je vjerovati zbog beskonačne nagrade/kazne.")
68
69 print("\nFilozofske kritike:")
70 kritike = [
71     "Problem mnoštva religija (koja vjera?)",
72     "Autentičnost vjerovanja (može li se 'odlučiti' vjerovati?)",
73     "Beskonačnost kao korisnost (ima li smisla?)",
74     "Moralni prigovor (je li to pravo vjerovanje?)"
75 ]
76 for i, kritika in enumerate(kritike, 1):
77     print(f"  {i}. {kritika}")
78
79 # Konačna verzija
80 print("\nKonačna aproksimacija (bez beskonačnosti):")
81 print("  Korisnosti: vjeruj(Bog da)=1000, vjeruj(Bog ne)=-10")
82 print("             ne vjeruj(Bog da)=-1000, ne vjeruj(Bog ne)=10")
83
84 # Kritična vjerojatnost
85 #  $1000p - 10(1-p) = -1000p + 10(1-p)$ 
86 #  $1000p - 10 + 10p = -1000p + 10 - 10p$ 
87 #  $2020p = 20$ 
88 p_kritična = 20 / 2020
89 print(f"  \n Kritična vjerojatnost: P* = {p_kritična:.3f}")
90 print(f"  Ako P(Bog) > {p_kritična:.3f}, vjeruj; inače ne vjeruj.")

```

Output

Pascalova oklada - Analiza odlučivanja

=====

Matrica korisnosti:

	Bog postoji	Bog ne postoji
Vjerujem:	$+\infty$	-10
Ne vjerujem:	$-\infty$	+10

Analiza za različite vjerojatnosti postojanja Boga:

$$E[\text{vjerujem}] = 0.50 \cdot \infty + 0.50 \cdot (-10) = \infty$$

$$E[\text{ne vjerujem}] = 0.50 \cdot (-\infty) + 0.50 \cdot 10 = -\infty$$

→ Racionalna odluka: VJERUJ

$$E[\text{vjerujem}] = 0.10 \cdot \infty + 0.90 \cdot (-10) = \infty$$

$$E[\text{ne vjerujem}] = 0.10 \cdot (-\infty) + 0.90 \cdot 10 = -\infty$$

→ Racionalna odluka: VJERUJ

$$E[\text{vjerujem}] = 0.01 \cdot \infty + 0.99 \cdot (-10) = \infty$$

$$E[\text{ne vjerujem}] = 0.01 \cdot (-\infty) + 0.99 \cdot 10 = -\infty$$

→ Racionalna odluka: VJERUJ

$$E[\text{vjerujem}] = 0.0001 \cdot \infty + 1.00 \cdot (-10) = \infty$$

$$E[\text{ne vjerujem}] = 0.0001 \cdot (-\infty) + 1.00 \cdot 10 = -\infty$$

→ Racionalna odluka: VJERUJ

Pascalov zaključak:

Čak i uz infinitezimalnu vjerojatnost postojanja Boga, racionalno je vjerovati zbog beskonačne nagrade/kazne.

Filozofske kritike:

1. Problem mnoštva religija (koja vjera?)
2. Autentičnost vjerovanja (može li se 'odlučiti' vjerovati?)
3. Beskonačnost kao korisnost (ima li smisla?)
4. Moralni prigovor (je li to pravo vjerovanje?)

Konačna aproksimacija (bez beskonačnosti):

Korisnosti: vjeruj(Bog da)=1000, vjeruj(Bog ne)=-10

ne vjeruj(Bog da)=-1000, ne vjeruj(Bog ne)=10

Kritična vjerojatnost: $P^* = 0.010$

Ako $P(\text{Bog}) > 0.010$, vjeruj; inače ne vjeruj.

7.1.6 Problem indukcije i vjerojatnost

David Hume je ukazao na **problem indukcije** - nemogućnost logičkog opravdanja generalizacije iz konačnog broja opažanja. Vjerojatnost nudi djelomično rješenje:

"Navika je veliki vodič ljudskog života" - Hume

Umjesto traženja sigurnosti, vjerojatnost kvantificira stupanj racionalnog vjerovanja.

```

1 class InduktivnoZakljucivanje:
2     """Modelira induktivno zaključivanje kroz vjerojatnost."""
3
4     def __init__(self):
5         self.povijest = []
6
7     def laplace_sukcesija(self, uspjesi, pokusaji):
8         """Laplace-ov zakon sukcesije.
9
10        P(sljedeći uspjeh) = (k + 1) / (n + 2)
11        gdje je k broj uspjeha, n broj pokušaja
12        """
13        return (uspjesi + 1) / (pokusaji + 2)
14
15     def azuriraj_vjerovanje(self, novi_dokaz):
16         """Ažurira vjerovanje na temelju novog dokaza."""
17         self.povijest.append(novi_dokaz)
18         uspjesi = sum(self.povijest)
19         pokusaji = len(self.povijest)
20         return self.laplace_sukcesija(uspjesi, pokusaji)
21
22 # Primjer: Sunce izlazi svaki dan
23 print("Problem indukcije - Sunce izlazi")
24 print("="*33)
25
26 indukcija = InduktivnoZakljucivanje()
27
28 print("\nLaplace-ov zakon sukcesije:")
29 print("  P(uspjeh) = (k + 1) / (n + 2)")
30 print("  gdje je k = broj dosadašnjih uspjeha, n = ukupan broj pokušaja")
31
32 print("\nDana\tP(sunce izađe)\tPromjena")
33 print("-"*4 + "\t" + "-"*14 + "\t" + "-"*8)
34
35 # Simuliraj promatranje sunca
36 dani = [0, 1, 2, 3, 4, 5, 10, 30, 100, 365, 1000, 10000]
37 prethodna_p = 0.5
38
39 for dan in dani:
40     p = indukcija.laplace_sukcesija(dan, dan)
41     promjena = p - prethodna_p if dan > 0 else 0
42
43     print(f"{dan}\t{p:.4f}\t\t", end="")
44     if dan > 0:
45         print(f"+{promjena:.4f}" if promjena > 0 else f"{promjena:.4f}")
46     else:
47         print("-")

```

```

48
49     if dan in [1, 5, 30, 365, 10000]:
50         prethodna_p = p
51
52 print("\nHumeov problem:")
53 print("   Koliko god puta sunce izašlo, P(sunce izađe sutra) < 1")
54 print("   Nikad ne možemo biti potpuno sigurni!")
55
56 # Crni labud
57 print("\nCrni labud scenarij:")
58 bijeli_labudovi = 1000
59 p_bijeli = indukcija.laplace_sukcesija(bijeli_labudovi, bijeli_labudovi)
60 print(f"Nakon {bijeli_labudovi} bijelih labudova, P(sljedeći bijel) = {p_bijeli:.4f}")
61 print("Ali jedan crni labud mijenja sve!")
62
63 # Nakon crnog labuda
64 ukupno = bijeli_labudovi + 1
65 p_bijeli_novi = indukcija.laplace_sukcesija(bijeli_labudovi, ukupno)
66 p_crni = indukcija.laplace_sukcesija(1, ukupno)
67
68 print("\nNakon crnog labuda:")
69 print(f"   Vidjeli: {bijeli_labudovi} bijelih, 1 crni")
70 print(f"   P(sljedeći bijel) = {p_bijeli_novi:.4f}")
71 print(f"   P(sljedeći crni) = {p_crni:.4f}")
72
73 print("\nFilozofska pouka:")
74 print("   Indukcija ne daje sigurnost, već racionalne stupnjeve vjerovanja.")
75 print("   Vjerojatnost kvantificira našu neizvjesnost o budućnosti.")

```

Output

Problem indukcije - Sunce izlazi

=====

Laplace-ov zakon sukcesije:

$P(\text{uspjeh}) = (k + 1) / (n + 2)$

gdje je k = broj dosadašnjih uspjeha, n = ukupan broj pokušaja

Dana P(sunce izađe) Promjena

---- -

0	0.5000	-
1	0.6667	+0.1667
2	0.7500	+0.0833
3	0.8000	+0.1333
4	0.8333	+0.1667
5	0.8571	+0.1905
10	0.9167	+0.0595
30	0.9688	+0.1116
100	0.9902	+0.0214
365	0.9973	+0.0285
1000	0.9990	+0.0017

10000 0.9999 +0.0026

Humeov problem:

Koliko god puta sunce izašlo, $P(\text{sunce izađe sutra}) < 1$
 Nikad ne možemo biti potpuno sigurni!

Crni labud scenarij:

Nakon 1000 bijelih labudova, $P(\text{sljedeći bijel}) = 0.9990$

Ali jedan crni labud mijenja sve!

Nakon crnog labuda:

Vidjeli: 1000 bijelih, 1 crni

$P(\text{sljedeći bijel}) = 0.9980$

$P(\text{sljedeći crni}) = 0.0020$

Filozofska pouka:

Indukcija ne daje sigurnost, već racionalne stupnjeve vjerovanja.
 Vjerojatnost kvantificira našu neizvjesnost o budućnosti.

7.1.7 Monty Hall problem - paradoksi uvjetne vjerojatnosti

Monty Hall problem ilustrira kako naša intuicija često griješi kod uvjetnih vjerojatnosti:

```

1 import random
2
3 class MontyHall:
4     """Simulacija Monty Hall problema."""
5
6     def igray(self, strategija="promijeni"):
7         """Igra jednu rundu Monty Hall igre.
8
9         strategija: 'ostani' ili 'promijeni'
10        """
11        # Postavi igru
12        vrata = [1, 2, 3]
13        auto = random.choice(vrata)
14
15        # Igrač bira
16        izbor = random.choice(vrata)
17
18        # Voditelj otvara vrata s kozom
19        moguca_vrata = [v for v in vrata if v != izbor and v != auto]
20        voditelj_otvara = random.choice(moguca_vrata) if moguca_vrata else \
21            [v for v in vrata if v != izbor][0]
22
23        # Primijeni strategiju
24        if strategija == "promijeni":
25            preostala = [v for v in vrata if v != izbor and v != voditelj_otvara]
```

```
26     konacni_izbor = preostala[0]
27     else: # ostani
28         konacni_izbor = izbor
29
30     return konacni_izbor == auto
31
32     def simuliraj(self, n_igara=10000):
33         """Simulira više igara s obje strategije."""
34         rezultati = {
35             "ostani": 0,
36             "promijeni": 0
37         }
38
39         for _ in range(n_igara):
40             if self.igraj("ostani"):
41                 rezultati["ostani"] += 1
42             if self.igraj("promijeni"):
43                 rezultati["promijeni"] += 1
44
45         return rezultati, n_igara
46
47 # Simulacija
48 mh = MontyHall()
49 rezultati, n = mh.simuliraj(10000)
50
51 print("Monty Hall Problem - Simulacija")
52 print("="*32)
53 print("\nPostavka:")
54 print(" 3 vrata: iza jednih je automobil, iza drugih dvoje koze")
55 print(" 1. Birate vrata")
56 print(" 2. Voditelj otvara druga vrata s kozom")
57 print(" 3. Možete promijeniti izbor ili ostati")
58
59 print(f"\nSimulacija {n:,} igara:\n")
60
61 for strategija, pobjede in rezultati.items():
62     postotak = (pobjede / n) * 100
63     print(f"Strategija {strategija.upper()}:")
64     print(f"  Pobjede: {pobjede:,} / {n:,}")
65     print(f"  Postotak pobjede: {postotak:.1f}%\n")
66
67 # Bayesova analiza
68 print("Bayesova analiza:")
69 print("="*18)
70 print("\nPočetne vjerojatnosti:")
71 print(" P(auto na vratima 1) = 1/3")
72 print(" P(auto na vratima 2) = 1/3")
73 print(" P(auto na vratima 3) = 1/3")
74
75 print("\nBirate vrata 1. Voditelj otvara vrata 3 (koza).")
76
77 print("\nAžurirane vjerojatnosti:")
78 print(" P(auto na 1 | voditelj otvorio 3) = 1/3")
```

```

79 print(" P(auto na 2 | voditelj otvorio 3) = 2/3")
80 print(" P(auto na 3 | voditelj otvorio 3) = 0")
81
82 print("\nObjašnjenje:")
83 print(" Voditelj ZNA gdje je auto i MORA otvoriti kozu.")
84 print(" Njegovo otvaranje daje informaciju!")
85 print(" ")
86 print(" Ako je auto na vratima 1: voditelj bira između 2 i 3 (P=1/2)")
87 print(" Ako je auto na vratima 2: voditelj MORA otvoriti 3 (P=1)")
88 print(" Ako je auto na vratima 3: voditelj MORA otvoriti 2 (P=1)")
89 print(" ")
90 print(" Činjenica da je otvorio 3 favorizira vrata 2!")
91
92 print("\nZaključak: UVIJEK se isplati promijeniti izbor!")

```

Output

Monty Hall Problem - Simulacija
=====

Postavka:

- 3 vrata: iza jednih je automobil, iza drugih dvoje koze
- 1. Birate vrata
- 2. Voditelj otvara druga vrata s kozom
- 3. Možete promijeniti izbor ili ostati

Simulacija 10,000 igara:

Strategija OSTANI:

Pobjede: 3,328 / 10,000
Postotak pobjede: 33.3%

Strategija PROMIJENI:

Pobjede: 6,718 / 10,000
Postotak pobjede: 67.2%

Bayesova analiza:

=====

Početne vjerojatnosti:

P(auto na vratima 1) = 1/3
P(auto na vratima 2) = 1/3
P(auto na vratima 3) = 1/3

Birate vrata 1. Voditelj otvara vrata 3 (koza).

Ažurirane vjerojatnosti:

P(auto na 1 | voditelj otvorio 3) = 1/3
P(auto na 2 | voditelj otvorio 3) = 2/3
P(auto na 3 | voditelj otvorio 3) = 0

Objašnjenje:

Voditelj ZNA gdje je auto i MORA otvoriti kozu.
Njegovo otvaranje daje informaciju!

Ako je auto na vratima 1: vođitelj bira između 2 i 3 ($P=1/2$)

Ako je auto na vratima 2: vođitelj MORA otvoriti 3 ($P=1$)

Ako je auto na vratima 3: vođitelj MORA otvoriti 2 ($P=1$)

Činjenica da je otvorio 3 favorizira vrata 2!

Zaključak: UVIJEK se isplati promijeniti izbor!

7.1.8 Filozofske interpretacije vjerojatnosti

Postoje različite filozofske interpretacije što vjerojatnost zapravo znači:

1. **Klasična** (Laplace): Omjer povoljnih i mogućih ishoda
2. **Frekventistička** (von Mises): Granična relativna frekvencija
3. **Propenzitetna** (Popper): Objektivna tendencija
4. **Subjektivna/Bayesova** (de Finetti): Stupanj racionalnog vjerovanja

Svaka interpretacija ima filozofske implikacije za prirodu slučajnosti, determinizma i znanja.

```

1 def filozofske_interpretacije():
2     """Demonstrira različite filozofske interpretacije vjerojatnosti."""
3
4     print("Filozofske interpretacije vjerojatnosti")
5     print("="*40)
6
7     # 1. Klasična
8     print("\n1. KLASIČNA INTERPRETACIJA (Laplace)")
9     print("    'Vjerojatnost je omjer povoljnih i jednakvjerojatnih ishoda'")
10    print("    \n    Primjer: Kocka")
11    povoljni = 3 # parni brojevi: 2, 4, 6
12    mogući = 6
13    p_klasična = povoljni / mogući
14    print(f"    P(paran broj) = {povoljni}/{mogući} = {p_klasična:.3f}")
15    print("    \n    Problem: Pretpostavlja jednakvjerojatnost (cirkularnost)")
16
17    # 2. Frekventistička
18    print("\n2. FREKVENTISTIČKA INTERPRETACIJA (von Mises)")
19    print("    'Vjerojatnost je granična relativna frekvencija'")
20    print("    ")
21

```

```

22 random.seed(42)
23 for n in [100, 1000, 10000, 100000]:
24     parni = sum(1 for _ in range(n) if random.randint(1, 6) % 2 == 0)
25     frekvencija = parni / n
26     print(f"    Bacanja: {n}, Parni: {parni}, Frekvencija: {frekvencija:.3f}")
27 print("    → Konvergira prema 0.500")
28 print("    \n    Problem: Što s jedinstvenim događajima?")
29
30 # 3. Propenzitetna
31 print("\n3. PROPENZITETNA INTERPRETACIJA (Popper)")
32 print("    'Vjerojatnost je objektivna tendencija sustava'")
33 print("    ")
34 print("    Kocka ima propenzitet 0.5 za paran broj")
35 print("    zbog svoje fizičke strukture.")
36 print("    \n    Problem: Kako mjeriti propenzitet?")
37
38 # 4. Subjektivna
39 print("\n4. SUBJEKTIVNA INTERPRETACIJA (de Finetti)")
40 print("    'Vjerojatnost je stupanj racionalnog vjerovanja'")
41 print("    ")
42 print("    Moje vjerovanje:")
43 print("    P(kiša sutra) = 0.30 (na temelju prognoze)")
44 print("    ")
45 print("    Koherentnost: Moja vjerovanja moraju zadovoljiti")
46 print("    aksiome vjerojatnosti inače mogu biti 'Dutch booked'")
47 print("    \n    Problem: Subjektivnost vs. objektivnost")
48
49 # Filozofske implikacije
50 print("\nFilozofske implikacije:")
51 print("=*24)
52
53 print("\nDETERMINIZAM vs. INDETERMINIZAM:")
54 print("    - Je li vjerojatnost samo naša neznanja (epistemička)?")
55 print("    - Ili je svijet inherentno slučajan (ontološka)?")
56
57 print("\nPROBLEM REFERENTNE KLASJE:")
58 print("    - Jesam li '35-godišnjak', 'Hrvat', 'filozof'?")
59 print("    - Različite klase daju različite vjerojatnosti!")
60
61 print("\nPRINCIP INDIFERENCIJE:")
62 print("    - Bez informacija, dodjeli jednake vjerojatnosti?")
63 print("    - Bertrandov paradoks pokazuje probleme")
64
65 print("\n> \"Bog ne igra kocke\" - Einstein")
66 print("> \"Einstein, prestani govoriti Bogu što da radi\" - Bohr")
67
68 filozofske_interpretacije()

```

Output

Filozofske interpretacije vjerojatnosti

=====

1. KLASIČNA INTERPRETACIJA (Laplace)

'Vjerojatnost je omjer povoljnih i jednakovjerojatnih ishoda'

Primjer: Kocka

$P(\text{paran broj}) = 3/6 = 0.500$

Problem: Pretpostavlja jednakovjerojatnost (cirkularnost)

2. FREKVENTISTIČKA INTERPRETACIJA (von Mises)

'Vjerojatnost je granična relativna frekvencija'

Bacanja: 100, Parni: 49, Frekvencija: 0.490

Bacanja: 1000, Parni: 510, Frekvencija: 0.510

Bacanja: 10000, Parni: 5057, Frekvencija: 0.506

Bacanja: 100000, Parni: 50215, Frekvencija: 0.502

→ Konvergira prema 0.500

Problem: Što s jedinstvenim događajima?

3. PROPENZITETNA INTERPRETACIJA (Popper)

'Vjerojatnost je objektivna tendencija sustava'

Kocka ima propenzitet 0.5 za paran broj
zbog svoje fizičke strukture.

Problem: Kako mjeriti propenzitet?

4. SUBJEKTIVNA INTERPRETACIJA (de Finetti)

'Vjerojatnost je stupanj racionalnog vjerovanja'

Moje vjerovanje:

$P(\text{kiša sutra}) = 0.30$ (na temelju prognoze)

Koherentnost: Moja vjerovanja moraju zadovoljiti
aksiome vjerojatnosti inače mogu biti 'Dutch booked'

Problem: Subjektivnost vs. objektivnost

Filozofske implikacije:

=====

DETERMINIZAM vs. INDETERMINIZAM:

- Je li vjerojatnost samo naša neznanja (epistemička)?
- Ili je svijet inherentno slučajan (ontološka)?

PROBLEM REFERENTNE KLAZE:

- Jesam li '35-godišnjak', 'Hrvat', 'filozof'?
- Različite klase daju različite vjerojatnosti!

PRINCIP INDIFERENCIJE:

- Bez informacija, dodjeli jednake vjerojatnosti?
 - Bertrandov paradoks pokazuje probleme
- > "Bog ne igra kocke" - Einstein
- > "Einstein, prestani govoriti Bogu što da radi" - Bohr

7.1.9 Prijedlozi za daljnje istraživanje

Za produbljivanje razumijevanja vjerojatnosti kao proširenja logike kroz praktične Python zadatke:

Fuzzy logika

Implementirajte fuzzy logičke operatore gdje istinite vrijednosti mogu biti bilo koji broj između 0 i 1. Definirajte Łukasiewicz, Gödel i product t-norme. Pokažite kako se klasična logika dobiva kao granični slučaj.

Cox-Jaynesovi teoremi

Dokažite da svaki sustav koji zadovoljava razumne uvjete za stupnjeve vjerovanja mora biti izomorfan s teorijom vjerojatnosti. Implementirajte alternativni sustav i pokažite kako se svodi na vjerojatnost.

Dutch book argument

Simulirajte kladenje gdje agent s nekoherentnim vjerovanjima (koja krše aksiome vjerojatnosti) uvijek gubi novac. Vizualizirajte kako koherentnost štiti od sigurnog gubitka.

Dempster-Shafer teorija

Implementirajte teoriju dokaza koja generalizira vjerojatnost dopuštajući "ne znam" odgovore. Pokažite kombiniranje dokaza i rad s intervalima vjerojatnosti.

Paradoks spavajuće ljepotice

Simulirajte ovaj paradoks samolociranja u vremenu. Analizirajte sukob između "halfer" i "thirder" pozicija. Povežite s antropijskim principom.

Kvantna vjerojatnost

Implementirajte Born rule i pokažite kako kvantne amplitude daju vjerojatnosti. Demonstrirajte narušavanje Bell-ovih nejednakosti klasičnom vjerojatnošću.

Algoritamska vjerojatnost

Implementirajte Solomonoff-ovu indukciju koristeći Kolmogorovljev složenost. Pokažite kako kraći programi imaju veću apriornu vjerojatnost.

Kauzalno zaključivanje

Implementirajte Pearl-ove kauzalne grafove. Pokažite razliku između korelacije i kauzalnosti. Simulirajte Simpson-ov paradoks.

Maksimalna entropija

Implementirajte princip maksimalne entropije za izbor vjerojatnosnih distribucija. Pokažite kako ograničenja određuju optimalnu distribuciju.

Vjerojatnosno programiranje

Stvorite jednostavan vjerojatnosni programski jezik gdje varijable mogu biti distribucije. Implementirajte inferenciju kroz MCMC ili variational Bayes.

Svaki zadatak postupno gradi razumijevanje vjerojatnosti kao generalizacije logike, omogućavajući studentima da istraže dublje veze između logike, vjerojatnosti i racionalnog zaključivanja.

7.1.10 Zaključak

Kroz ovu implementaciju istražili smo teoriju vjerojatnosti kao prirodno proširenje logike sudova:

1. **Generalizacija istinitosti** - od $\perp, \top$ do $[0, 1]$
2. **Kolmogorovljevi aksiomi** kao temelj matematičke teorije
3. **Bayesovo zaključivanje** za ažuriranje vjerovanja
4. **Filozofske implikacije** za racionalnost i odlučivanje

Pascal je anticipirao modernu teoriju odlučivanja pokazujući kako matematički pristupiti neizvjesnosti:

"Razum nas ne može odlučiti između kršćanstva i ateizma, ali morate birati kako ćete živjeti" - Pascal

Vjerojatnost premošćuje jaz između čiste logike i nesigurnog svijeta iskustva. Ona ne daje apsolutnu istinu, ali omogućava racionalno navigiranje kroz neizvjesnost.

Kroz praktično programiranje otkrivamo da je vjerojatnost više od matematičkog alata - ona je jezik za izražavanje stupnjeva vjerovanja, kvantificiranje neizvjesnosti i donošenje racionalnih odluka u svijetu gdje je potpuno znanje nedostižno.

"Život je umijeće izvlačenja dovoljnih zaključaka iz nedovoljnih premisa" - Samuel Butler

Teorija vjerojatnosti je upravo to umijeće, formalizirano u elegantnom matematičkom okviru.