**Лабораторна робота № 6**

**Наївний Байєс в Python**

**Мета роботи:** набути навичок працювати з даними і опонувати роботу у Python з використанням теореми Байэса.

**Хід роботи**

**Завдання 6.1.** Ретельно опрацювати теоретичні відомості:

**Завдання 6.2.** Ретельно розібрати приклад: прогнозування з використанням теореми Байєса.

**Завдання 6.3.** Використовуя данні з пункту 2 визначити відбудеться матч при наступних погодних умовах чи ні: Розрахунки провести з використанням Python.

Таблиця. 6.1. Вхідні умови

|  |  |  |
| --- | --- | --- |
| Варіант | Умова | |
| 4,9,14 | Outlook = Sunny  Humidity = Normal  Wind = Strong | Перспектива = Сонячно  Вологість = Нормальна  Вітер = Сильний |

* Загальні ймовірності (Priors):
  + P(Yes) = 9/14
  + P(No) = 5/14
* Ймовірності для «Yes» (з відповідних таблиць):
  + P(Outlook = Sunny∣Yes) = 2/9
  + P(Humidity = Normal∣Yes) = 6/9
  + P(Wind = Strong∣Yes) = 3/9
* Ймовірності для «No» (з відповідних таблиць):
* P(Outlook = Sunny∣No) = 3/5
* P(Humidity = Normal∣No) = 1/5
* P(Wind = Strong∣No) = 3/5

P(Yes∣Sunny, Normal, Strong) = P(Sunny∣Yes) \* P(Normal∣Yes) \* P(Strong∣Yes) \* P(Yes) = 2/9 \* 6/9 \* 3/9 \* 9/14 = 3/70 ≈ 0.03174

P(No∣Sunny, Normal, Strong) = P(Sunny∣ No) \* P(Normal∣No) \* P(Strong∣ No) \* P(No) = 3/5 \* 1/5 \* 3/5 \* 5/14 = 3/140 ≈ 0.02571

P(Yes) = 0.03174 / (0.03174 + 0.02571) = 0.5525

P(No) = 0.02571 / (0.021429 + 0.03174 ) = 0.4475

Ймовірність того, що матч відбудеться (55.25%), вища за ймовірність того, що він не відбудеться (44.75%).

**Лістинг LR\_6\_task\_3.py:**

p\_sunny\_yes = 2/9

p\_normal\_yes = 6/9

p\_strong\_yes = 3/9

p\_sunny\_no = 3/5

p\_normal\_no = 1/5

p\_strong\_no = 3/5

p\_yes = 9/14

p\_no = 5/14

score\_yes = p\_sunny\_yes \* p\_normal\_yes \* p\_strong\_yes \* p\_yes

score\_no = p\_sunny\_no \* p\_normal\_no \* p\_strong\_no \* p\_no

print(f"Показник для 'Yes': {score\_yes:.5f}")

print(f"Показник для 'No': {score\_no:.5f}")

total\_score = score\_yes + score\_no

final\_p\_yes = score\_yes / total\_score

final\_p\_no = score\_no / total\_score

print("\n--- Результати прогнозу ---")

print(f"Ймовірність, що матч відбудеться ('Yes'): {final\_p\_yes:.1%}")

print(f"Ймовірність, що матч не відбудеться ('No'): {final\_p\_no:.1%}")

if final\_p\_yes > final\_p\_no:

    print("\nВисновок: Модель прогнозує, що матч відбудеться. ✅")

else:

    print("\nВисновок: Модель прогнозує, що матч НЕ відбудеться. ❌")

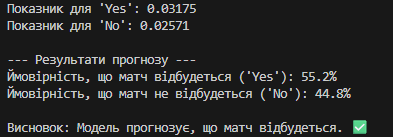


Рис.6.1. Результат програми

**Завдання 6.4.** Застосуєте методи байєсівського аналізу до набору даних про ціни на квитки на іспанські високошвидкісні залізниці.

**Лістинг LR\_6\_task\_4.py:**

import pandas as pd

import pymc as pm

import arviz as az

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

df = pd.read\_csv('renfe\_small.csv')

df\_clean = df.dropna(subset=['price', 'train\_type', 'fare']).copy()

df\_clean['train\_type\_code'] = pd.Categorical(df\_clean['train\_type']).codes

df\_clean['fare\_code'] = pd.Categorical(df\_clean['fare']).codes

train\_type\_labels = pd.Categorical(df\_clean['train\_type']).categories

fare\_labels = pd.Categorical(df\_clean['fare']).categories

print("Дані успішно підготовлені.")

print(f"Категорії типів поїздів: {list(train\_type\_labels)}")

print(f"Категорії тарифів: {list(fare\_labels)}")

plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)

sns.histplot(df\_clean['price'], kde=True)

plt.title('Розподіл цін на квитки')

plt.xlabel('Ціна (€)')

plt.ylabel('Частота')

plt.subplot(1, 2, 2)

sns.boxplot(x='train\_type', y='price', data=df\_clean)

plt.title('Ціна залежно від типу поїзда')

plt.xlabel('Тип поїзда')

plt.ylabel('Ціна (€)')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()

with pm.Model() as renfe\_model:

    beta0 = pm.Normal('beta0', mu=0, sigma=10)

    beta\_train\_type = pm.Normal('beta\_train\_type', mu=0, sigma=5, shape=len(train\_type\_labels))

    beta\_fare = pm.Normal('beta\_fare', mu=0, sigma=5, shape=len(fare\_labels))

    alpha = pm.Exponential('alpha', 1.0)

    mu = pm.math.exp(beta0 + beta\_train\_type[df\_clean['train\_type\_code']] + beta\_fare[df\_clean['fare\_code']])

    beta = alpha / mu

    price = pm.Gamma('price', alpha=alpha, beta=beta, observed=df\_clean['price'])

with renfe\_model:

    idata = pm.sample(2000, tune=1000, cores=1)

    idata.extend(pm.sample\_posterior\_predictive(idata))

print("Модель успішно навчена.")

summary = az.summary(idata, var\_names=['beta0', 'beta\_train\_type', 'beta\_fare', 'alpha'], hdi\_prob=0.94)

print("\nЗведення параметрів моделі:")

print(summary)

summary\_train\_type = az.summary(idata, var\_names=['beta\_train\_type'], hdi\_prob=0.94)

summary\_train\_type.index = [f"beta\_train\_type[{label}]" for label in train\_type\_labels]

print("\nЗведення коефіцієнтів типу поїзда:")

print(summary\_train\_type)

summary\_fare = az.summary(idata, var\_names=['beta\_fare'], hdi\_prob=0.94)

summary\_fare.index = [f"beta\_fare[{label}]" for label in fare\_labels]

print("\nЗведення коефіцієнтів тарифу:")

print(summary\_fare)

mean\_coeffs\_train = idata.posterior['beta\_train\_type'].mean(dim=('chain', 'draw')).values

print(f"\nБазовий тип поїзда: {train\_type\_labels[0]}")

for i, label in enumerate(train\_type\_labels[1:], 1):

    effect = np.exp(mean\_coeffs\_train[i] - mean\_coeffs\_train[0])

    print(f"Поїзд типу '{label}' у {effect:.2f} разів дорожчий/дешевший, ніж '{train\_type\_labels[0]}'")

az.plot\_ppc(idata, num\_pp\_samples=100)

plt.title('Апостеріорна предиктивна перевірка')

plt.xlabel('Ціна (€)')

plt.show()

az.plot\_forest(idata, var\_names=['beta\_train\_type', 'beta\_fare'], combined=True, hdi\_prob=0.94)

plt.title('Апостеріорні розподіли для коефіцієнтів')

plt.show()

Для тесту було взято 2500 строк, щоб робота програми не тривала більше 3 годин.

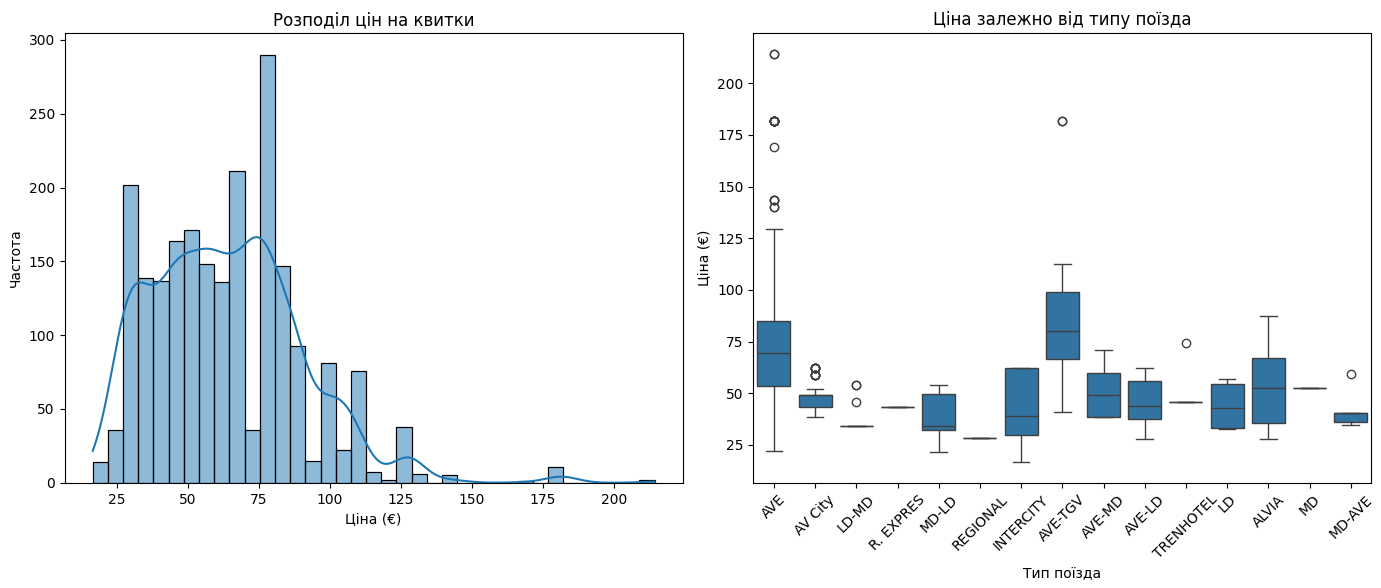


Рис.6.2. Гістограма цін на квитки + Діаграма розмаху цін за типами поїздів

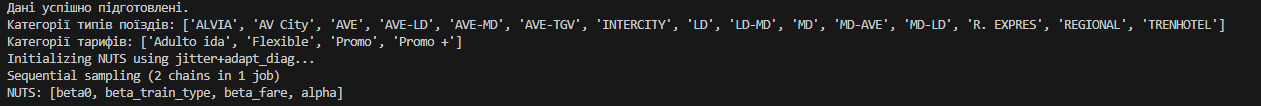


Рис.6.3. Вивід в консоль на цьому етапі

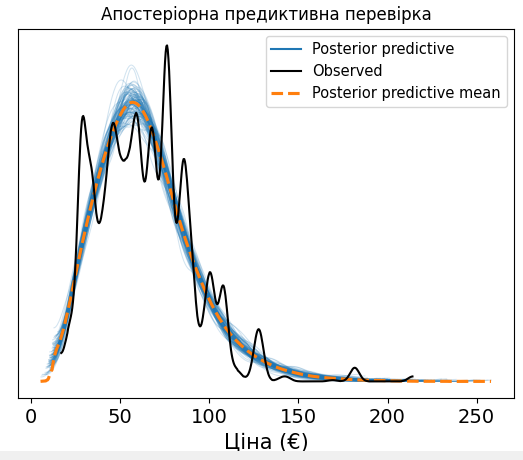


Рис.6.4. Графік апостеріорної предиктивної перевірки (PPC)

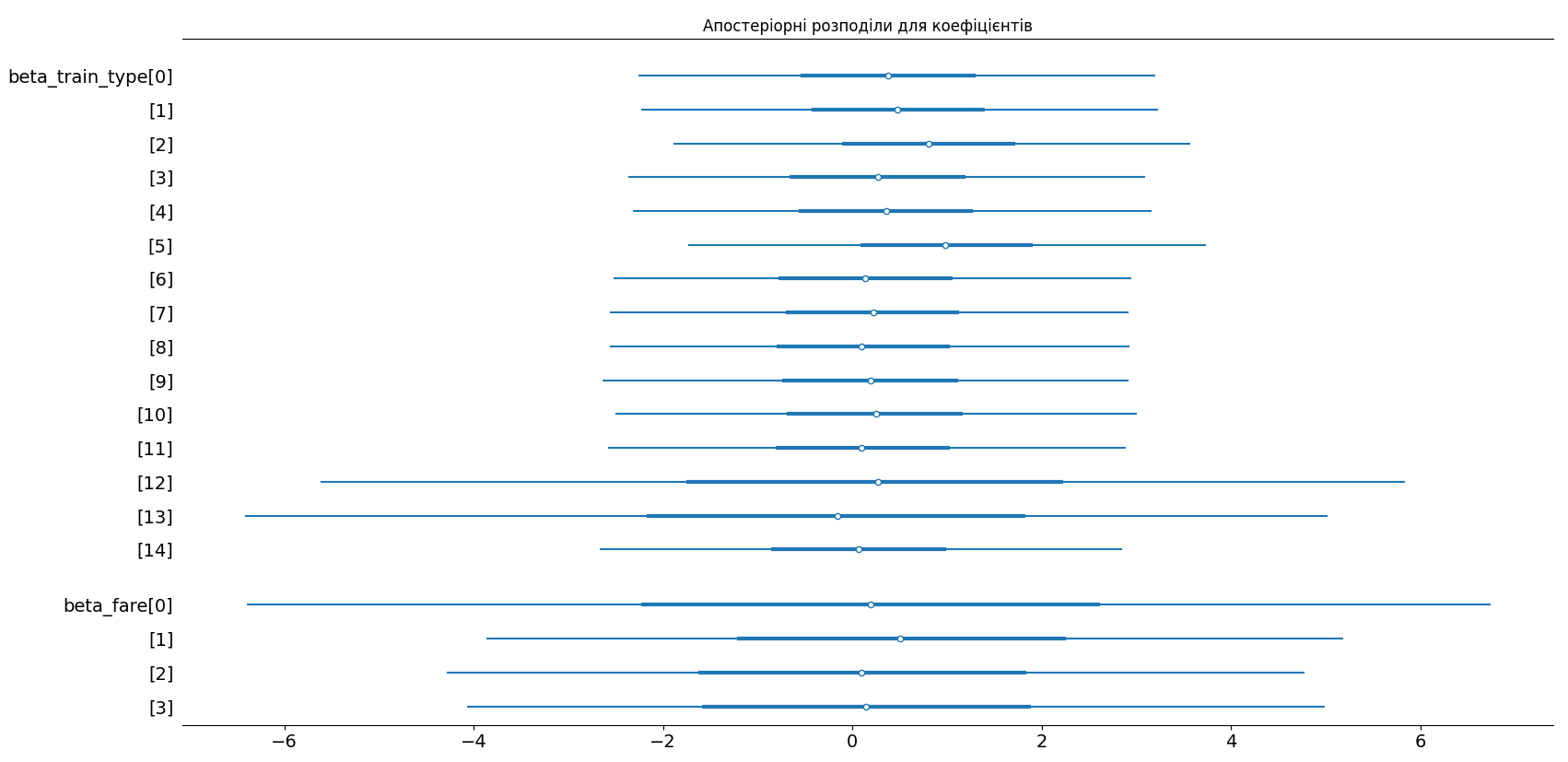


Рис.6.5. «Лісовий» графік коефіцієнтів моделі

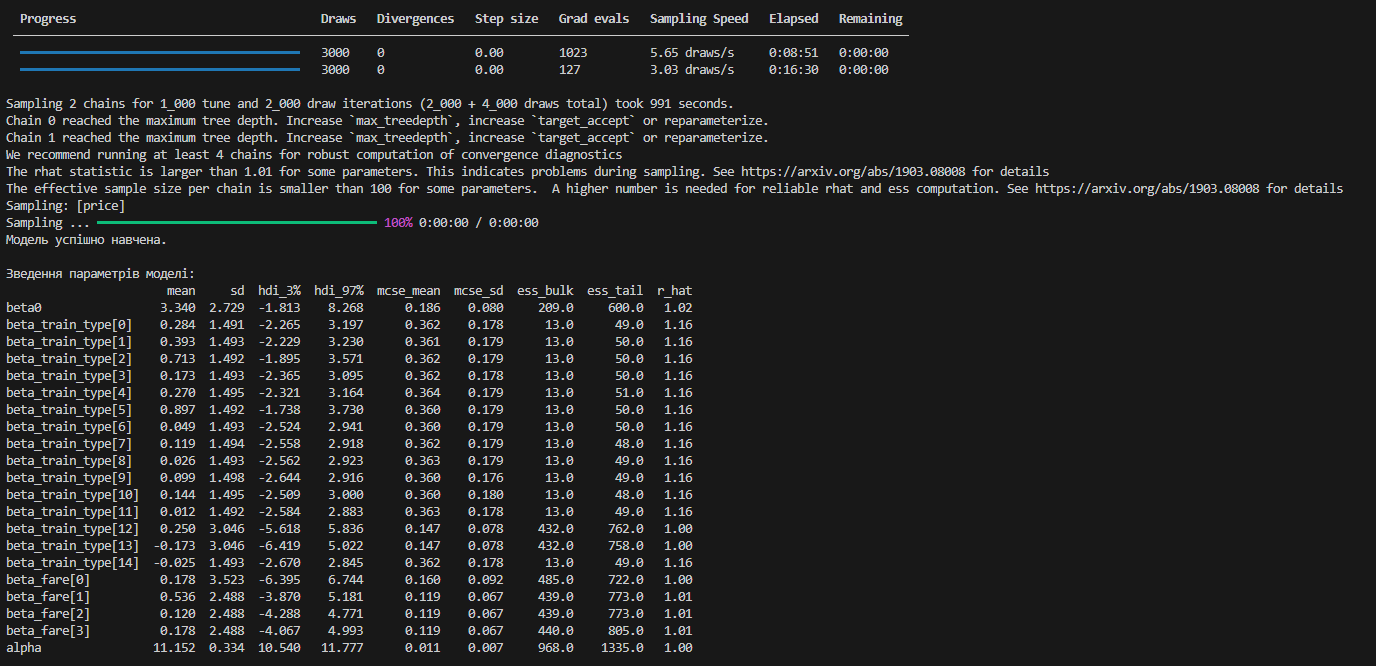


Рис.6.6. Вивід в консоль

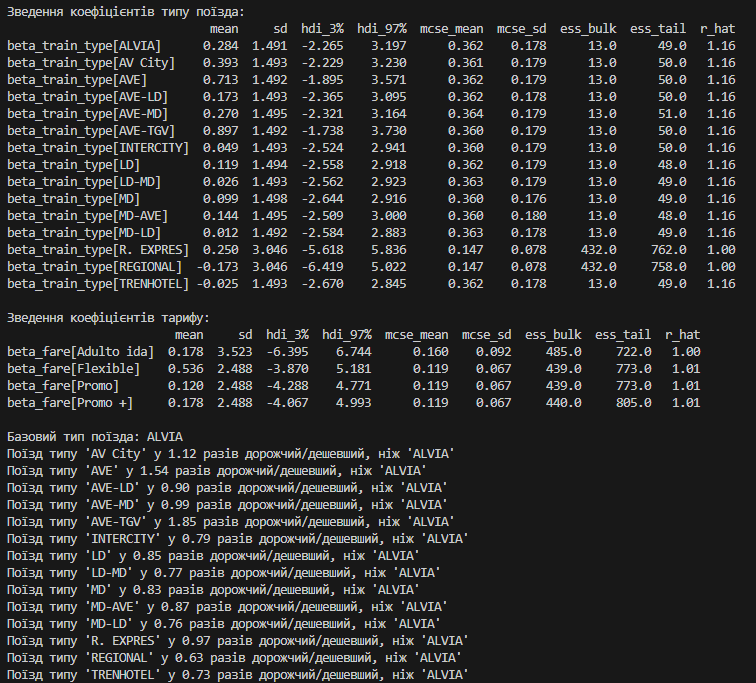
****

Рис.6.7. Вивід в консоль

**Посилання на гіт: https://github.com/VadymLeus/Y4S1-AIS**

**Висновок:** в ході виконання лабораторної роботи були набуті навички працювати з даними і опонувано роботу у Python з використанням теореми Байэса.