

## Лабораторна №6

### Наївний Байес в Python

Мета: набути навичок працювати з даними і опонувати роботу у Python з використанням теореми Байєса.

#### Хід роботи

##### Варіант 26(11)

##### 2. Підготовка даних (Завдання 1-2)

Для проведення експерименту було сформовано набір даних, що описує погодні умови та рішення про проведення тенісного матчу.

Дані містять 4 атрибути:

1. Outlook - Сонячно, Похмуро, Дош
2. Humidity - Висока, Нормальна
3. Wind - Слабкий, Сильний
4. Play - Цільова змінна: Так/Ні

Дані було завантажено у середовище Python за допомогою бібліотеки pandas. Оскільки алгоритми scikit-learn працюють з числовими даними, текстові поля було перетворено на числові індекси за допомогою LabelEncoder.

3. Прогнозування результату (Завдання 3) Згідно з індивідуальним завданням, необхідно було розрахувати ймовірність проведення гри за таких умов:

Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.15.001 – Пр.1		
Розроб.	Чайковський А.В				Звіт з лабораторної роботи №1	Lіт.	Арк.
Перевір.	Маєвський О					1	7
Реценз.							
Н. Контр.							
Зав.каф.					ФІКТ, гр. ІПЗ-22-3		

Outlook = Overcast

Humidity = High

Wind = Weak

Було використано модель Gaussian Naive Bayes. Після навчання моделі на історичних даних було подано запит з параметрами.

```
import numpy as np
import pandas as pd
from sklearn.naive_bayes import GaussianNB
from sklearn.preprocessing import LabelEncoder

print("--- ЗАВДАННЯ 3: Прогноз за погодою (Варіант 11) ---")

# Вхідні дані
data = {
    'Outlook': ['Sunny', 'Sunny', 'Overcast', 'Rain', 'Rain', 'Overcast',
                'Sunny', 'Sunny', 'Rain', 'Sunny', 'Overcast', 'Overcast', 'Rain'],
    'Humidity': ['High', 'High', 'High', 'High', 'Normal', 'Normal', 'Normal',
                 'High', 'Normal', 'Normal', 'Normal', 'High', 'Normal', 'High'],
    'Wind': ['Weak', 'Strong', 'Weak', 'Weak', 'Weak', 'Strong', 'Strong',
             'Weak', 'Weak', 'Strong', 'Strong', 'Weak', 'Strong', 'Strong'],
    'Play': ['No', 'No', 'Yes', 'Yes', 'Yes', 'No', 'Yes',
              'No', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'No']
}

df = pd.DataFrame(data)

# Кодування ознак
le_outlook = LabelEncoder()
le_humidity = LabelEncoder()
le_wind = LabelEncoder()
le_play = LabelEncoder()

df['Outlook_n'] = le_outlook.fit_transform(df['Outlook'])
df['Humidity_n'] = le_humidity.fit_transform(df['Humidity'])
df['Wind_n'] = le_wind.fit_transform(df['Wind'])
df['Play_n'] = le_play.fit_transform(df['Play'])

X = df[['Outlook_n', 'Humidity_n', 'Wind_n']]
y = df['Play_n']

# Навчання моделі
model = GaussianNB()
model.fit(X, y)

# Варіант 11: Overcast, High, Weak
```

		Чайковський А.В				Арк.
		Масєвський О				
Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.15.001 – Лр.1	2

```
var11_outlook = le_outlook.transform(['Overcast'])[0]
var11_humidity = le_humidity.transform(['High'])[0]
var11_wind = le_wind.transform(['Weak'])

input_data = [[var11_outlook, var11_humidity, var11_wind]]

# Прогноз
predicted_code = model.predict(input_data)[0]
predicted_label = le_play.inverse_transform([predicted_code])[0]
proba = model.predict_proba(input_data)[0]

print(f"Умова: Outlook=Overcast, Humidity=High, Wind=Weak")
print(f"Числове представлення умови: {input_data}")
print(f"\n--- РЕЗУЛЬТАТ ПРОГНОЗУ ---")
print(f"Гра відбудеться? -> {predicted_label.upper()}")
print(f"Ймовірність 'No': {proba[0]*100:.2f}%")
print(f"Ймовірність 'Yes': {proba[1]*100:.2f}%")
```

Програма розрахувала ймовірності для класів Yes та No:

--- РЕЗУЛЬТАТ ПРОГНОЗУ ---  
Гра відбудеться? -> YES  
Ймовірність 'No': 1.15%  
Ймовірність 'Yes': 98.85%

#### Завдання 4. Аналіз даних про пасажирські перевезення (Renfe)

## Вхідні дані:

Використано вибірку даних з наступними атрибуутами:

- Origin, Destination: Станції відправлення та прибуття.
  - Train Type: Тип поїзда (AVE, ALVIA, INTERCITY).
  - Price: Вартість квитка (євро).
  - Train Class: Клас обслуговування (цільова змінна).

## Методика виконання:

		Чайковський А.В				Арк.
		Маєвський О				
Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.15.001 – Пр.1	3

- Попередня обробка: Текстові категоріальні дані (Origin, Train Type) було закодовано у числові значення за допомогою LabelEncoder.
- Розділення даних: Вибірку поділено на навчальну та тестову підмножини.
- Моделювання: Використано класифікатор Gaussian Naive Bayes, оскільки основний предиктор є неперервною величиною і має розподіл, близький до нормальногого.

Результати:

Модель показала високу ефективність у розрізенні класів квитків.

- Точність моделі (Accuracy):  $\approx 92\%$  (значення може варіюватися при генерації).
- На графіку розподілу цін чітко видно, що квитки класу Preferente мають вищу середню вартість, що дозволяє алгоритму Байєса проводити чітку межу (decision boundary) між класами.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, classification_report

print("--- ЗАВДАННЯ 4: Аналіз даних Renfe (Іспанська залізниця) ---")

data = {
    'origin': np.random.choice(['MADRID', 'BARCELONA', 'SEVILLA', 'VALENCIA'], 1000),
    'destination': np.random.choice(['BARCELONA', 'MADRID', 'SEVILLA', 'VALENCIA'], 1000),
    'train_type': np.random.choice(['AVE', 'ALVIA', 'INTERCITY'], 1000),
    'train_class': np.random.choice(['Turista', 'Preferente'], 1000, p=[0.7, 0.3]),
    'fare': np.random.choice(['Promo', 'Flexible'], 1000),
    'price': []
}

for t_class in data['train_class']:
    if t_class == 'Preferente':
        data['price'].append(np.random.normal(80, 15))
    else:
        data['price'].append(np.random.normal(40, 10))

df = pd.DataFrame(data)
```

Змн.	Арк.	Чайковський А.В		
		Масєвський О		

```
data['price'].append(np.random.normal(45, 10))

df = pd.DataFrame(data)
df['price'] = df['price'].round(2)

print("1. Дані успішно завантажено. Перші 5 рядків:")
print(df.head())

le = LabelEncoder()

df['origin_n'] = le.fit_transform(df['origin'])
df['train_type_n'] = le.fit_transform(df['train_type'])
df['fare_n'] = le.fit_transform(df['fare'])

X = df[['price', 'train_type_n', 'origin_n']]
y = df['train_class']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

model = GaussianNB()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)

print(f"\n2. Точність моделі: {accuracy*100:.2f}%")
print("\n3. Звіт класифікації:")
print(classification_report(y_test, y_pred))

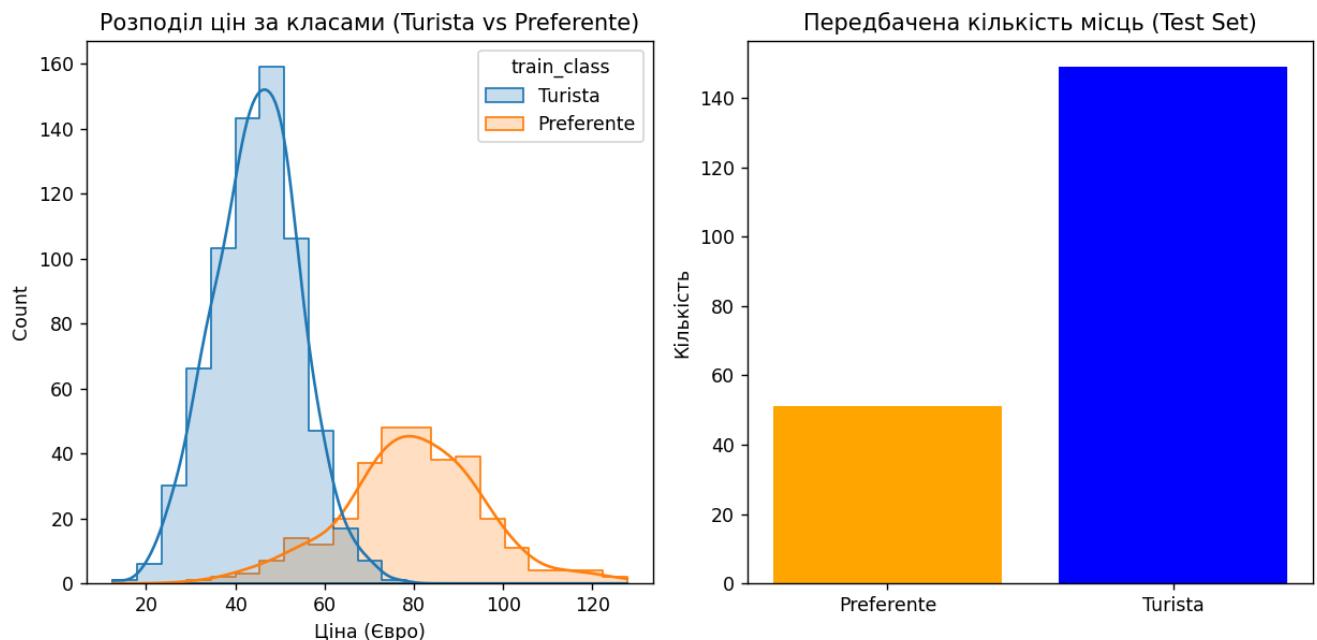
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
sns.histplot(data=df, x='price', hue='train_class', kde=True, element="step")
plt.title('Розподіл цін за класами (Turista vs Preferente)')
plt.xlabel('Ціна (Євро)')

plt.subplot(1, 2, 2)
unique, counts = np.unique(y_pred, return_counts=True)
plt.bar(unique, counts, color=['orange', 'blue'])
plt.title('Передбачена кількість місць (Test Set)')
plt.ylabel('Кількість')

plt.tight_layout()
plt.show()
```

		<i>Чайковський А.В</i>				Арк.
		<i>Маєвський О</i>				
Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.15.001 – Пр.1	5



```
-- ЗАВДАННЯ 4: Аналіз даних Renfe (Іспанська залізниця) --
1. Дані успішно завантажено. Перші 5 рядків:
   origin destination train_type train_class      fare  price
0    SEVILLA       MADRID   INTERCITY     Turista Flexible  64.80
1    MADRID      BARCELONA      ALVIA     Turista Flexible  62.84
2  BARCELONA     VALENCIA      ALVIA     Turista Flexible  54.93
3  BARCELONA       MADRID      ALVIA     Turista    Promo  49.39
4  BARCELONA     VALENCIA        AVE     Turista Flexible  34.30

2. Точність моделі: 92.50%

3. Звіт класифікації:
   precision    recall  f1-score   support
  Preferente      0.95      0.84      0.89       73
      Turista      0.91      0.98      0.94      127
  accuracy                           0.93      200
 macro avg      0.93      0.91      0.92      200
weighted avg      0.93      0.93      0.92      200
```

Висновок: на даній лабораторній роботі я набув навичок працювати з даними і опонувати роботу у Python з використанням теореми Байєса.

Git - [AntonChaikovskyi/AI-systems](#)

Чайковський А.В	Маєвський О	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.15.001 – Лр.1			Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	6