

ЛАБОРАТОРНА РОБОТА № 6

Наївний Байєс в Python

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

Git: <https://github.com/flekXD/SAI>

Завдання 1.

Використовую данні з пункту 2 визначити відбудеться матч при наступних погодних умовах чи ні: Розрахунки провести з використанням Python.

2, 7, 12	Outlook = Overcast Humidity = High Wind = Strong
----------	--

```
import numpy as np

# Функція для розрахунку ймовірності для кожної таблиці (Outlook, Humidity, Wind)
def table_operations(smt, table, row_index):
    total_sum = np.sum(table)

    # Сума чисел в рядку
    row_sum = np.sum(table[row_index])

    # Сума чисел в стовпці для Yes та No
    column_Y = np.sum(table[:, 0]) # Сума стовпця "Yes"
    column_N = np.sum(table[:, 1]) # Сума стовпця "No"

    # Обчислення ймовірності для Yes та No
    res_Y = (smt / column_Y) * (column_Y / total_sum) / (row_sum / total_sum)
    res_N = (smt / column_N) * (column_N / total_sum) / (row_sum / total_sum)

    return res_Y, res_N

# Функція для розрахунку ймовірностей для матчу
def calculate_probability_for_match(outlook, humidity, wind):
    # Таблиці ймовірностей для Outlook, Humidity, Wind
    outlook_table = np.array([
        [3, 2], # Sunny (Yes, No)
        [4, 0], # Overcast (Yes, No)
        [3, 2]  # Rainy (Yes, No)
    ])

    humidity_table = np.array([
        [3, 4], # High (Yes, No)
        [6, 1]  # Normal (Yes, No)
    ])

    wind_table = np.array([
```

		Кириченко О С			ДУ «Житомирська політехніка».21.121.00.000 – Лр1	Арк.
		Голенко М. Ю.				1
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        [6, 2], # Strong (Yes, No)
        [3, 3] # Weak (Yes, No)
    ])

# Задано умови для Outlook, Humidity, Wind
if outlook == 'Sunny':
    outlook_row_index = 0
elif outlook == 'Overcast':
    outlook_row_index = 1
elif outlook == 'Rainy':
    outlook_row_index = 2
else:
    raise ValueError("Invalid outlook condition")

if humidity == 'High':
    humidity_row_index = 0
elif humidity == 'Normal':
    humidity_row_index = 1
else:
    raise ValueError("Invalid humidity condition")

if wind == 'Strong':
    wind_row_index = 0
elif wind == 'Weak':
    wind_row_index = 1
else:
    raise ValueError("Invalid wind condition")

# Розрахунок ймовірностей для Outlook, Humidity, Wind
outlook_Y, outlook_N = table_operations(3, outlook_table, outlook_row_index) #
Потрібно передати правильне значення smt
humidity_Y, humidity_N = table_operations(3, humidity_table,
humidity_row_index) # Аналогічно
wind_Y, wind_N = table_operations(6, wind_table, wind_row_index) # І для
вітру, передаємо правильне число smt

# Розрахунок P(Yes) та P(No) для всіх умов
P_yes_result = outlook_Y * humidity_Y * wind_Y * 9/14 # P(Yes) (наприклад, для
конкретного загального числа)
P_no_result = outlook_N * humidity_N * wind_N * 5/14 # P(No)

P_yes = P_yes_result / (P_yes_result + P_no_result)
P_no = P_no_result / (P_yes_result + P_no_result)
return P_yes, P_no

# Задано умови
outlook = 'Overcast'
humidity = 'High'
wind = 'Strong'

# Розрахунок ймовірностей для Yes і No
P_yes, P_no = calculate_probability_for_match(outlook, humidity, wind)

```

		Кириченко О С			ДУ «Житомирська політехніка».21.121.00.000 – Лр1	Арк.
		Голенко М. Ю.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```
# Виведення результатів
print(f"P(Yes|Outlook={outlook}, Humidity={humidity}, Wind={wind}) = {P_yes:.6f}")
print(f"P(No|Outlook={outlook}, Humidity={humidity}, Wind={wind}) = {P_no:.6f}")
```

```
PS D:\123> python lab6_task_1.py
P(Yes|Outlook=Overcast, Humidity=High, Wind=Strong) = 0.642857
P(No|Outlook=Overcast, Humidity=High, Wind=Strong) = 0.357143
```

З результатів прогнозування бачимо що ймовірність складає 64% на те чи буде гра при таких погодних умовах

Завдання 2.

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

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score

# Завантаження даних
url = "https://raw.githubusercontent.com/susanli2016/Machine-Learning-with-Python/refs/heads/master/data/renfe_small.csv"
df = pd.read_csv(url)

# Видаляємо рядки, де ціна відсутня або не є числом
df = df.dropna(subset=['price'])
df['price'] = pd.to_numeric(df['price'], errors='coerce')
df = df.dropna(subset=['price'])

# Переводимо ціни в категорії (наприклад, дешево і дорого)
df['price_category'] = df['price'].apply(lambda x: 1 if x > 100 else 0) #
Створення стовпця price_category

# Вибір ознак для моделювання
features = ['origin', 'destination', 'train_type', 'train_class'] # Використовуємо
ці стовпці для моделювання
df_dummies = pd.get_dummies(df[features], drop_first=True) # Перетворюємо
категоріальні змінні в числові

# Додаємо стовпець price_category назад в df_dummies
df_dummies['price_category'] = df['price_category'].astype(int)

# Розділяємо дані на тренувальні та тестові
X = df_dummies.drop('price_category', axis=1)
y = df_dummies['price_category']
```

		Кириченко О С			ДУ «Житомирська політехніка».21.121.00.000 – Лр1	Арк.
		Голенко М. Ю.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

# Створення і тренування моделі наївного баєсівського класифікатора
model = GaussianNB()
model.fit(X_train, y_train)

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

# Фільтрація даних для цін менше 25 євро
low_price_count = df[df['price'] < 25].shape[0]

# Загальна кількість записів
total_count = df.shape[0]

# Ймовірність отримати ціну менше 25 євро
probability = low_price_count / total_count

print(f"Ймовірність отримати ціну менше 25 євро: {probability:.2%}")

print(f"Точність моделі: {accuracy:.2f}")

```

```

PS D:\123> python lab6_task_2.py
Ймовірність отримати ціну менше 25 євро: 2.37%
Точність моделі: 0.52

```

У цій роботі проаналізовано ціни на залізничні квитки та виконано їх класифікацію на "дорогі" й "дешеві" за допомогою наївного баєсівського класифікатора. Точність моделі склала 52%, що свідчить про її обмежену ефективність. Ймовірність отримати квиток за ціною менше 25 євро склала лише 2.37%, що вказує на рідкість бюджетних квитків. Для покращення результатів рекомендується використовувати більш складні моделі та додаткові ознаки, як-от сезонність або дати бронювання.

Висновок

У ході виконання лабораторної роботи №6 було досягнуто мету, поставлену на початку. Я здобув практичні навички роботи з даними та опанував використання теореми Байєса в Python. Було розглянуто основні принципи роботи Наївного Байєса, його переваги та обмеження.

Також реалізовано алгоритм Наївного Байєса для класифікації даних, що дозволило закріпити теоретичні знання на практиці. Аналіз отриманих

		Кириченко О С			ДУ «Житомирська політехніка».21.121.00.000 – Лр1	Арк.
		Голенко М. Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

результатів показав, що Наївний Байєс є ефективним інструментом для розв'язання задач класифікації, особливо для великих наборів даних.

		Кириченко О С			ДУ «Житомирська політехніка».21.121.00.000 – Лр1	Арк.
		Голенко М. Ю.				5
Змн.	Арк.	№ докум.	Підпис	Дата		