

## ЛАБОРАТОРНА РОБОТА № 6

### Наївний Байєс в Python

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

**Github** - <https://github.com/andreyzaets/lab6>

#### Task 3

```
from collections import Counter

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

play_count = Counter([row["Play"] for row in data])
outlook_count = Counter([(row["Outlook"], row["Play"]) for row in data])
humidity_count = Counter([(row["Humidity"], row["Play"]) for row in data])
wind_count = Counter([(row["Wind"], row["Play"]) for row in data])

total_play_yes = play_count["Yes"] / len(data)
total_play_no = play_count["No"] / len(data)

# Для 8 варіанту події наступні
# Outlook = Sunny
# Humidity = High
# Wind = Weak
p_outlook_yes = outlook_count[("Sunny", "Yes")] / play_count["Yes"]
p_outlook_no = outlook_count[("Sunny", "No")] / play_count["No"]
p_humidity_yes = humidity_count[("High", "Yes")] / play_count["Yes"]
p_humidity_no = humidity_count[("High", "No")] / play_count["No"]
```

```

p_wind_yes = wind_count[("Weak", "Yes")] / play_count["Yes"]
p_wind_no = wind_count[("Weak", "No")] / play_count["No"]

p_yes = p_outlook_yes * p_humidity_yes * p_wind_yes * total_play_yes
p_no = p_outlook_no * p_humidity_no * p_wind_no * total_play_no

p_yes_normalized = p_yes / (p_yes + p_no)
p_no_normalized = p_no / (p_yes + p_no)
print(f"Ймовірність, що матч відбудеться (Yes): {p_yes_normalized:.2f}")
print(f"Ймовірність, що матч не відбудеться (No): {p_no_normalized:.2f}")

```

```

"D:\work\шляк 4 курса\СШІ\laba6\laba6\venv\Scripts\python.exe" "D:\work\шляк 4 курса\СШІ\laba6\laba6\LR_6_task_3.py"
Ймовірність, що матч відбудеться (Yes): 0.38
Ймовірність, що матч не відбудеться (No): 0.62

Process finished with exit code 0

```

## Task 4

```

from collections import Counter
import pandas as pd

url = "https://raw.githubusercontent.com/susanli2016/Machine-Learning-with-Python/master/data/renfe_small.csv"
df = pd.read_csv(url)
df = df[["price", "train_type", "origin", "destination", "train_class"]].dropna()
df["price_category"] = pd.cut(df["price"], bins=3, labels=["low", "medium", "high"])

train_type_count = Counter(zip(df["train_type"], df["price_category"]))
origin_count = Counter(zip(df["origin"], df["price_category"]))
destination_count = Counter(zip(df["destination"], df["price_category"]))
class_count = Counter(zip(df["train_class"], df["price_category"]))
price_category_count = Counter(df["price_category"])

def calculate_probability(category, attribute, attribute_count, price_category_count):
    return (
        attribute_count.get((attribute, category), 0) /
        price_category_count[category]
        if price_category_count[category]
        else 0
    )

```

```

train_type = "AVE"
origin = "MADRID"
destination = "BARCELONA"
train_class = "Preferente"
probabilities = {}
for category in price_category_count.keys():
    p_train_type = calculate_probability(
        category, train_type, train_type_count, price_category_count
    )
p_origin = calculate_probability(
    category, origin, origin_count, price_category_count
)
p_destination = calculate_probability(
    category, destination, destination_count, price_category_count
)
p_class = calculate_probability(
    category, train_class, class_count, price_category_count
)
p_category = price_category_count[category] / len(df)

probabilities[category] = (
    p_train_type * p_origin * p_destination * p_class * p_category
)

total_prob = sum(probabilities.values())
for category in probabilities:
    probabilities[category] /= total_prob if total_prob else 1
probabilities = list(probabilities.values())
formatted_probabilities = list(map("{:.2f}".format, probabilities))
print(
    f"Ймовірності для кожної категорії вартості квитка з параметрами({train_type}, {train_class}, {origin}, {destination}): ",
    formatted_probabilities)

```

```

Ймовірності для кожної категорії вартості квитка з параметрами (AVE, Preferente, MADRID, BARCELONA): ['0.21', '0.67', '0.11']

```

Де застосовується наївний Байєс?

- Фільтрації спаму
- Аналізу емоційного забарвлення текстів
- Категоризації новин
- Прогнозу погоди
- Медичної діагностики

Поясніть теорему Байєса.

Теорема Байєса описує спосіб визначення умовної ймовірності. Вона дозволяє обчислювати ймовірність події, базуючись на наявній інформації про

інші події, що можуть бути пов'язані з нею. Згідно з теоремою Байєса:

Які типи наївного байєсівського класифікатора існують? Основні типи наївного байєсівського класифікатора:

- Поліноміальний наївний Байєс використовується для даних, де ознаки є частотністю подій, наприклад, при класифікації текстів.
- Бернуллі наївний Байєс обробляє двійкові дані (0 або 1) та підходить для задач, де важливо лише наявність або відсутність ознаки.
- Гаусовий наївний Байєс передбачає нормальний розподіл безперервних значень і застосовується для обробки числових даних з безперервними значеннями.

**Висновок:** Під час виконання лабораторної роботи ми дослідили наївний байєсівський класифікатор, його основні типи та принцип роботи за теоремою Байєса. На прикладі було продемонстровано побудову частотних та правдоподібнісних таблиць, що дозволяють розраховувати ймовірності для прогнозу подій. Виконали практичне завдання на прогнозування можливості гри залежно від погодних умов, а також застосували метод наївного Байєса до набору даних про ціни на квитки.