**Лабораторна робота №6**

Наївний Байєс в Python

**Виконав:** ІПЗ-21-3 Осипчук Антон Олексійович

**Github:** <https://github.com/AntonOsypchuk1/ai_lab/tree/main/lab6>

**Завдання.** Використовуючи дані з пункту 2 визначити відбудеться матч при наступних погодних умовах чи ні:

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

Лістинг програми:

# Вхідні дані

data = {

"Outlook": {"Sunny": {"Yes": 3/10, "No": 2/4}},

"Humidity": {"Normal": {"Yes": 6/9, "No": 1/5}},

"Wind": {"Strong": {"Yes": 6/9, "No": 2/5}}

}

# Ймовірності для гри "Yes"

p\_yes = (

data["Outlook"]["Sunny"]["Yes"] \*

data["Humidity"]["Normal"]["Yes"] \*

data["Wind"]["Strong"]["Yes"]

)

# Ймовірності для гри "No"

p\_no = (

data["Outlook"]["Sunny"]["No"] \*

data["Humidity"]["Normal"]["No"] \*

data["Wind"]["Strong"]["No"]

)

# Нормалізація

p\_yes\_normalized = p\_yes / (p\_yes + p\_no)

p\_no\_normalized = p\_no / (p\_yes + p\_no)

print(f"Ймовірність гри 'Yes': {p\_yes\_normalized:.2f}")

print(f"Ймовірність гри 'No': {p\_no\_normalized:.2f}")

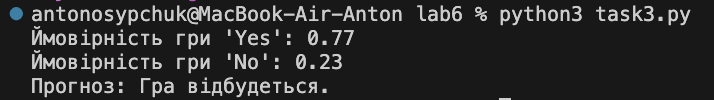
if p\_yes\_normalized > p\_no\_normalized:

print("Прогноз: Гра відбудеться.")

else:

print("Прогноз: Гра не відбудеться.")

Результат:



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

Лістинг програми:

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.naive\_bayes import GaussianNB

from sklearn.preprocessing import LabelEncoder, KBinsDiscretizer

from sklearn.metrics import accuracy\_score

url = "https://raw.githubusercontent.com/susanli2016/Machine-Learning-with-Python/master/data/renfe\_small.csv"

data = pd.read\_csv(url)

data = data.dropna()

le = LabelEncoder()

categorical\_columns = ['origin', 'destination', 'train\_type', 'fare']

for col in categorical\_columns:

data[col] = le.fit\_transform(data[col])

discretizer = KBinsDiscretizer(n\_bins=3, encode='ordinal', strategy='uniform')

data['price\_category'] = discretizer.fit\_transform(data[['price']])

X = data[['origin', 'destination', 'train\_type', 'fare']]

y = data['price\_category']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, 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"Точність моделі: {accuracy:.2f}")

Результат:

