**ЛАБОРАТОРНА РОБОТА № 6**

**ДОСЛІДЖЕННЯ МЕТОДІВ АНСАМБЛЕВОГО НАВЧАННЯ.**

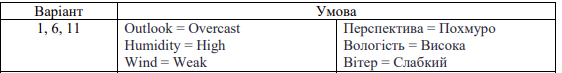
***Мета:*** використовуючи спеціалізовані бібліотеки та мову програмування Python дослідити методи ансамблів у машинному навчанні.

**Хід роботи**

Пункт2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Day | Outlook | Humidity | Wind | Play |
| D1 | Sunny | High | Weak | No |
| D2 | Sunny | High | Strong | No |
| D3 | Overcast | High | Weak | Yes |
| D4 | Rain | High | Weak | Yes |
| D5 | Rain | Normal | Weak | Yes |
| D6 | Rain | Normal | Strong | No |
| D7 | Overcast | Normal | Strong | Yes |
| D8 | Sunny | High | Weak | No |
| D9 | Sunny | Normal | Weak | Yes |
| D10 | Rain | Normal | Weak | Yes |
| D11 | Sunny | Normal | Strong | Yes |
| D12 | Overcast | High | Strong | Yes |
| D13 | Overcast | Normal | Weak | Yes |
| D14 | Rain | High | Strong | No |

Завдання 3. Використовуя данні з пункту 2 визначити відбудеться матч при наступних погодних умовах чи ні: Розрахунки провести з використанням Python.

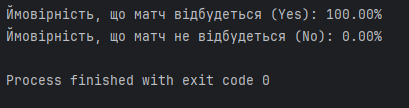


Лістинг програми:

# Частотні таблиці (попередньо обчислені з набору даних)  
P\_Yes = 10 / 14  
P\_No = 4 / 14  
  
# Ймовірності для кожного атрибута для "Yes" та "No"  
likelihoods = {  
 'Outlook': {  
 'Overcast': {'Yes': 4 / 10, 'No': 0 / 4},  
 'Sunny': {'Yes': 3 / 10, 'No': 2 / 4},  
 'Rain': {'Yes': 3 / 10, 'No': 2 / 4}  
 },

'Humidity': {  
 'High': {'Yes': 3 / 9, 'No': 4 / 5},  
 'Normal': {'Yes': 6 / 9, 'No': 1 / 5}  
 },  
 'Wind': {  
 'Strong': {'Yes': 6 / 9, 'No': 2 / 5},  
 'Weak': {'Yes': 3 / 9, 'No': 3 / 5}  
 }  
}  
  
def predict\_match(outlook, humidity, wind):  
 # Обчислення ймовірностей для "Yes"  
 P\_Yes\_given\_conditions = (  
 likelihoods['Outlook'][outlook]['Yes'] \*  
 likelihoods['Humidity'][humidity]['Yes'] \*  
 likelihoods['Wind'][wind]['Yes'] \*  
 P\_Yes  
 )  
  
 # Обчислення ймовірностей для "No"  
 P\_No\_given\_conditions = (  
 likelihoods['Outlook'][outlook]['No'] \*  
 likelihoods['Humidity'][humidity]['No'] \*  
 likelihoods['Wind'][wind]['No'] \*  
 P\_No  
 )  
  
 # Нормалізація значень  
 P\_Total = P\_Yes\_given\_conditions + P\_No\_given\_conditions  
 P\_Yes\_normalized = (P\_Yes\_given\_conditions / P\_Total \* 100) if P\_Total != 0 else 0  
 P\_No\_normalized = (P\_No\_given\_conditions / P\_Total \* 100) if P\_Total != 0 else 0  
  
 return P\_Yes\_normalized, P\_No\_normalized  
  
# Вхідні дані (можна змінювати для різних прогнозів)  
outlook = "Overcast" # Перспектива  
humidity = "High" # Вологість  
wind = "Weak" # Вітер  
  
# Результат прогнозування  
P\_Yes\_result, P\_No\_result = predict\_match(outlook, humidity, wind)  
print(f"Ймовірність, що матч відбудеться (Yes): {P\_Yes\_result:.2f}%")  
print(f"Ймовірність, що матч не відбудеться (No): {P\_No\_result:.2f}%")

Виконання програми:



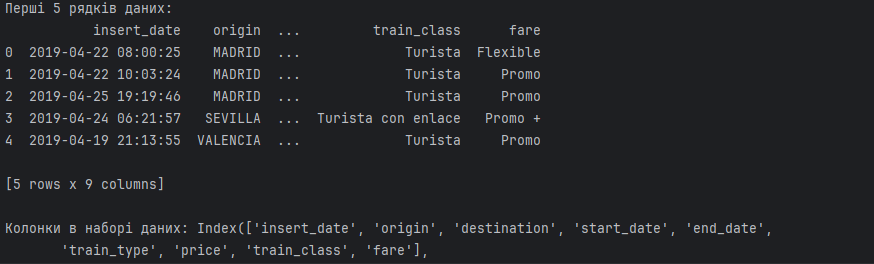
Завдання 4. Застосуєте методи байєсівського аналізу до набору даних про ціни на квитки на іспанські високошвидкісні залізниці.

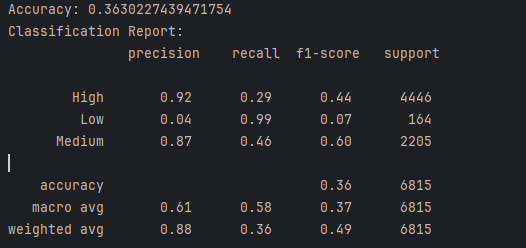
− Вхідні дані: <https://raw.githubusercontent.com/susanli2016/Machine-Learning-with-Python/master/data/renfe_small.csv>

Лістинг програми:

import pandas as pd  
import numpy as np  
from sklearn.model\_selection import train\_test\_split  
from sklearn.naive\_bayes import GaussianNB  
from sklearn.metrics import accuracy\_score, classification\_report  
  
url = "https://raw.githubusercontent.com/susanli2016/Machine-Learning-with-Python/master/data/renfe\_small.csv"  
data = pd.read\_csv(url)  
  
print("Перші 5 рядків даних:")  
print(data.head())  
print("\nКолонки в наборі даних:", data.columns,"\n")  
  
# Видалення рядків з пропущеними значеннями  
data = data.dropna()  
  
# Видаляємо текстові/дата-часові колонки, які не потрібні для моделі  
data = data.drop(columns=['insert\_date', 'origin', 'destination', 'start\_date', 'end\_date', 'train\_class'])  
  
# Список колонок, які хочемо закодувати  
columns\_to\_encode = ['train\_type', 'fare']  
  
# Фільтруємо, залишаючи тільки ті, що наявні в даних  
columns\_to\_encode = [col for col in columns\_to\_encode if col in data.columns]  
  
# Перетворення категоріальних змінних на числові за допомогою One-Hot Encoding  
data = pd.get\_dummies(data, columns=columns\_to\_encode)  
  
# Створимо категорії цін: низька (< 25), середня (25-50), висока (> 50)  
data['price\_category'] = pd.cut(data['price'], bins=[0, 25, 50, np.inf], labels=['Low', 'Medium', 'High'])  
  
# Визначимо вхідні та вихідні змінні  
X = data.drop(['price', 'price\_category'], axis=1)  
y = data['price\_category']  
  
# Розділимо дані на тренувальну та тестову вибірки  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)  
  
# Ініціалізація та навчання наївного байєсівського класифікатора  
model = GaussianNB()  
model.fit(X\_train, y\_train)  
  
# Прогнозуємо на тестовій вибірці  
y\_pred = model.predict(X\_test)  
  
# Оцінюємо точності моделі  
print("Accuracy:", accuracy\_score(y\_test, y\_pred))  
print("Classification Report:\n", classification\_report(y\_test, y\_pred))

Використання програми:





Посилання на ГітХаб: <https://github.com/KaidanovychBohdan/SystemOfAI>

**Висновок**: в ході виконання лабораторної роботи опрацював спеціалізовані бібліотеки та мову програмування Python дослідити методи ансамблів у машинному навчанні.