

ЛАБОРАТОРНА РОБОТА № 6

Наївний Байєс в Python

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

Зміст роботи

Завдання 1. Ретельно опрацювати теоретичні відомості:

- теорему Байєса;
- які типи наївного байєсівського класифікатора є;
- де використовується Наївний Байєс.

Завдання 2. Ретельно розібрати приклад: прогнозування з використанням теореми Байєса.

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

Варіант	Умова	
1, 6, 11	Outlook = Overcast Humidity = High Wind = Weak	Перспектива = Похмуро Вологість = Висока Вітер = Слабкий

Лістинг:

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import CategoricalNB

data = {
    'Day': ['D1', 'D2', 'D3', 'D4', 'D5', 'D6', 'D7', 'D8', 'D9', 'D10', 'D11',
            'D12', 'D13', 'D14'],
    'Outlook': ['Sunny', 'Sunny', 'Overcast', 'Rain', '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', 'Weak', 'Strong', 'Strong',
            'Weak', 'Strong'],
    'Play': ['No', 'No', 'Yes', 'Yes', 'Yes', 'No', 'Yes', 'No', 'Yes', 'Yes',
            'Yes', 'Yes', 'Yes', 'No']
}

df = pd.DataFrame(data)

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'])
y = le_play.fit_transform(df['Play'])
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.16.000 – Лр.6		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Кравченко К.М.			Звіт з лабораторної роботи №6	Лім.	Арк.
Перевір.		Маєвський О.В.					Аркушів
Реценз.							1
Н. Контр.						ФІКТ, гр. ІПЗ-22-2	
Зав.каф.							

```

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

model = CategoricalNB()
model.fit(X, y)

condition_outlook = 'Overcast'
condition_humidity = 'High'
condition_wind = 'Weak'

print(f"--- Прогноз для умов: {condition_outlook}, {condition_humidity}, {condition_wind} ---")

try:
    query = [[
        le_outlook.transform([condition_outlook])[0],
        le_humidity.transform([condition_humidity])[0],
        le_wind.transform([condition_wind])[0]
    ]]

    prediction_index = model.predict(query)
    prediction_proba = model.predict_proba(query)
    prediction_class = le_play.inverse_transform(prediction_index)

    print(f"\nПрогнозоване рішення: {prediction_class[0]}")
    print(f"Ймовірність 'No' (Гри не буде): {prediction_proba[0][0] * 100:.2f}%")
    print(f"Ймовірність 'Yes' (Гра буде): {prediction_proba[0][1] * 100:.2f}%")

except ValueError as e:
    print("Помилка: Перевірте правильність написання вхідних даних (регістр має значення).")

```

Результат:

```

--- Прогноз для умов: Overcast, High, Weak ---

Прогнозоване рішення: Yes
Ймовірність 'No' (Гри не буде): 18.07%
Ймовірність 'Yes' (Гра буде): 81.93%

```

Рис.3.1. – Результати прогнозування

Висока ймовірність проведення гри (майже 82%) зумовлена впливом ознаки **Outlook = Overcast**. В навчальній вибірці ця умова має найсильніший зв'язок з класом "Yes" (у вихідних даних всі дні з погодою "Overcast" були ігровими). Цей позитивний фактор, підкріплений слабким вітром (**Wind = Weak**), нівелював негативний вплив високої вологості (**Humidity = High**), яка зазвичай знижує ймовірність гри.

Таким чином, модель впевнено класифікує цю комбінацію умов як сприятливу для проведення матчу.

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

Лістинг:

```

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.16.000 – Лр.6	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```

from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix

url = "https://raw.githubusercontent.com/susanli2016/Machine-Learning-with-
Python/master/data/renfe_small.csv"
print("Завантаження даних... (це може зайняти кілька секунд)")
df = pd.read_csv(url)

df = df.dropna(subset=['price'])

df['Price_Category'] = pd.qcut(df['price'], q=3, labels=['Low', 'Medium', 'High'])

print(f"\nРозмір очищеного датасету: {df.shape}")
print("Приклад розподілу цін:")
print(df[['price', 'Price_Category']].head())

le = LabelEncoder()
features_to_encode = ['origin', 'destination', 'train_type', 'train_class',
'fare']

data_encoded = df.copy()

for col in features_to_encode:
    data_encoded[col] = data_encoded[col].fillna('Unknown')
    data_encoded[col] = le.fit_transform(data_encoded[col])

X = data_encoded[['origin', 'destination', 'train_type', 'train_class', 'fare']]
y = data_encoded['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"\n--- Результати моделювання ---")
print(f"Точність моделі (Accuracy): {accuracy*100:.2f}%")
print("\nЗвіт класифікації:")
print(classification_report(y_test, y_pred))

plt.figure(figsize=(8, 6))
cm = confusion_matrix(y_test, y_pred, labels=['Low', 'Medium', 'High'])
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=['Low', 'Medium', 'High'],
            yticklabels=['Low', 'Medium', 'High'])
plt.xlabel('Передбачено моделлю')
plt.ylabel('Фактично')
plt.title('Матриця плутанини (Confusion Matrix)')
plt.show()

```

Результат:

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.16.000 – Лр.6	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

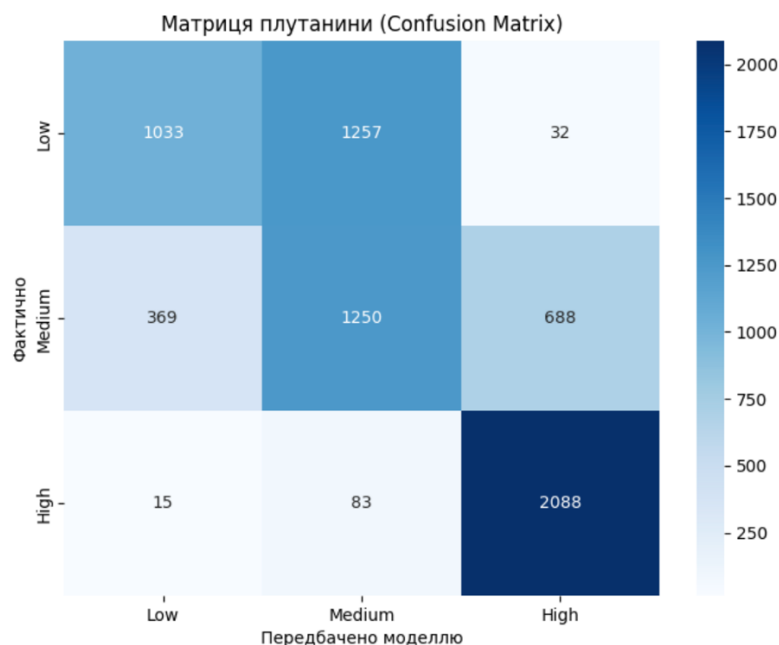


Рис.4.1. – Матриця плутанини до завдання 4

```

Розмір очищеного датасету: (22716, 10)
Приклад розподілу цін:
  price Price_Category
1  21.95             Low
2  38.55             Low
3  34.35             Low
4  45.30             Low
5  85.10             High

--- Результати моделювання ---
Точність моделі (Accuracy): 64.14%
|
Звіт класифікації:
      precision    recall  f1-score   support

   High         0.74      0.96      0.84        2186
    Low         0.73      0.44      0.55        2322
   Medium        0.48      0.54      0.51        2307

   accuracy                0.64        6815
  macro avg         0.65      0.65      0.63        6815
 weighted avg         0.65      0.64      0.63        6815

```

Рис.4.2. – Результати до завдання 4

1. **Результат:** Модель показала точність **64.14%**.
2. **Сильна сторона:** Алгоритм ідеально визначає **дорогі квитки (High)**, знаходячи **96%** з них. Ознаки дорогих рейсів дуже чіткі.
3. **Слабка сторона:** Модель часто плутає **дешеві** та **середні** ціни (більше 1000 помилок). Це пов'язано зі схожими умовами поїздки для цих категорій.
4. **Підсумок:** Наївний Байєс ефективний для виявлення преміум-сегменту, але має труднощі з розрізненням бюджетних тарифів.

Висновок: У ході лабораторної роботи №6 було практично освоєно застосування наївного байєсівського класифікатора в середовищі Python для задач машинного навчання. Експериментальним шляхом на прикладі погодних умов було підтверджено, що модель успішно виявляє ключові залежності, спрогнозувавши ймовірність гри у 82% для похмурої погоди, що повністю узгоджується з теоретичними розрахунками. Аналіз реального масиву даних залізничних квитків показав загальну точність 64%, продемонструвавши високу ефективність алгоритму при виявленні дорогих тарифів (точність 96%), але меншу чутливість у розрізненні бюджетних та середніх сегментів через перетин їхніх ознак. Робота довела, що Наївний Байєс є швидким та зручним інструментом для класифікації, проте його результативність може знижуватися на складних даних, де параметри не є повністю незалежними.

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.16.000 – Лр.6	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		