

Лабораторна робота № 6

Наївний Байес в Python

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

Хід роботи

Завдання 6.2-3. Ретельно розібрати приклад: прогнозування з використанням теореми Байєса. Використовуя 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	Перспектива = Похмуро Вологість = Висока Вітер = Сильний
----------	--	--

Лістинг LR_6_task3

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

data = {
    'Outlook': ['Sunny', 'Sunny', 'Overcast', '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', 'Strong', 'Strong', 'Weak', 'Strong', 'Strong'],
    'Play': ['No', 'No', 'Yes', 'Yes', 'Yes', 'No', 'Yes',
              'No', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'No']
}
df = pd.DataFrame(data)

encoders = {}
for col in ['Outlook', 'Humidity', 'Wind', 'Play']:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    encoders[col] = le

X = df[['Outlook', 'Humidity', 'Wind']]
y = df['Play']

model = GaussianNB()
model.fit(X, y)
```

Змн.	Арк.	№ докум.	Підпис	Дата
Розроб.	Свистанюк Н.О.			
Перевір.	Масевський О.В.			
Керівник				
Н. контр.				
Зас. каф.				

ДУ «Житомирська політехніка». 25.121.22.000–Пр6

Звіт з
лабораторної роботи

ФІКТ Гр. ІПЗ-22-3

Літ.	Арк.	Аркушів
	1	5

```

var_outlook = 'Overcast'
var_humidity = 'High'
var_wind = 'Strong'

input_data = [
    encoders['Outlook'].transform([var_outlook])[0],
    encoders['Humidity'].transform([var_humidity])[0],
    encoders['Wind'].transform([var_wind])[0]
]

prediction_idx = model.predict([input_data])[0]
prediction_label = encoders['Play'].inverse_transform([prediction_idx])[0]
probabilities = model.predict_proba([input_data])[0]

print(f"Вхідні умови (Варіант 22): {var_outlook}, {var_humidity}, {var_wind}")
print("-" * 40)
print(f"Прогнозоване рішення: {prediction_label}")
print(f"Ймовірність 'No': {probabilities[0]:.4f}")
print(f"Ймовірність 'Yes': {probabilities[1]:.4f}")

if probabilities[1] > probabilities[0]:
    print("\nВисновок: Модель прогнозує, що гра відбудеться.")
else:
    print("\nВисновок: Модель прогнозує, що гра НЕ відбудеться.")

```

Вхідні умови (Варіант 22): Overcast, High, Strong

Прогнозоване рішення: Yes

Ймовірність 'No': 0.0359

Ймовірність 'Yes': 0.9641

Висновок: Модель прогнозує, що гра відбудеться.

Рис.6.3.Результат виконання завдання

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

Застосувати методи байєсівського аналізу до набору даних про ціни на квитки. Оскільки ціна є неперервною величиною, а інші параметри (тип поїзда, клас) є категоріальними, доцільно використати Гауссівський найвний байєсівський класифікатор. Мета моделі спрогнозувати клас квитка на основі ціни та типу поїзда. Це дозволить оцінити, як ціна впливає на ймовірність належності квитка до певного класу.

Лістинг LR_6_task4

```

import pandas as pd
import seaborn as sns

```

		Свистанюк Н.О.		
		Маєвський О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

```
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
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"
try:
    df = pd.read_csv(url)
    print("Дані завантажено з інтернету.")
except:
    df = pd.read_csv('renfe_small.csv')
    print("Дані завантажено з локального файлу.")

df = df.dropna(subset=['price', 'train_class', 'train_type', 'fare'])

df = df.sample(5000, random_state=42)

print(f"Кількість записів для аналізу: {len(df)}")

plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
sns.histplot(df['price'], kde=True, color='skyblue')
plt.title('Розподіл цін на квитки')
plt.xlabel('Ціна (€)')
plt.ylabel('Частота')

plt.subplot(1, 2, 2)
sns.boxplot(x='train_type', y='price', data=df)
plt.title('Ціна залежно від типу поїзда')
plt.xlabel('Тип поїзда')
plt.ylabel('Ціна (€)')
plt.xticks(rotation=45)

plt.tight_layout()
plt.show()

le_type = LabelEncoder()
le_class = LabelEncoder()

df['train_type_code'] = le_type.fit_transform(df['train_type'])
y = le_class.fit_transform(df['train_class'])

X = df[['price', 'train_type_code']]

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

		<i>Свистанюк Н.О.</i>				
		<i>Масєвський О.В.</i>				
Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка».25.121.22.000 – Прб	Арк. 3

```

model = GaussianNB()
model.fit(X_train, y_train)

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

print(f"\nТочність моделі: {accuracy:.2%}")
print("\nДетальний звіт класифікації:")
print(classification_report(y_test, y_pred, target_names=le_class.classes_))

plt.figure(figsize=(8, 6))
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=le_class.classes_,
            yticklabels=le_class.classes_)
plt.title('Матриця плутанини (Confusion Matrix)')
plt.ylabel('Справжній клас')
plt.xlabel('Передбачений моделлю клас')
plt.show()

```

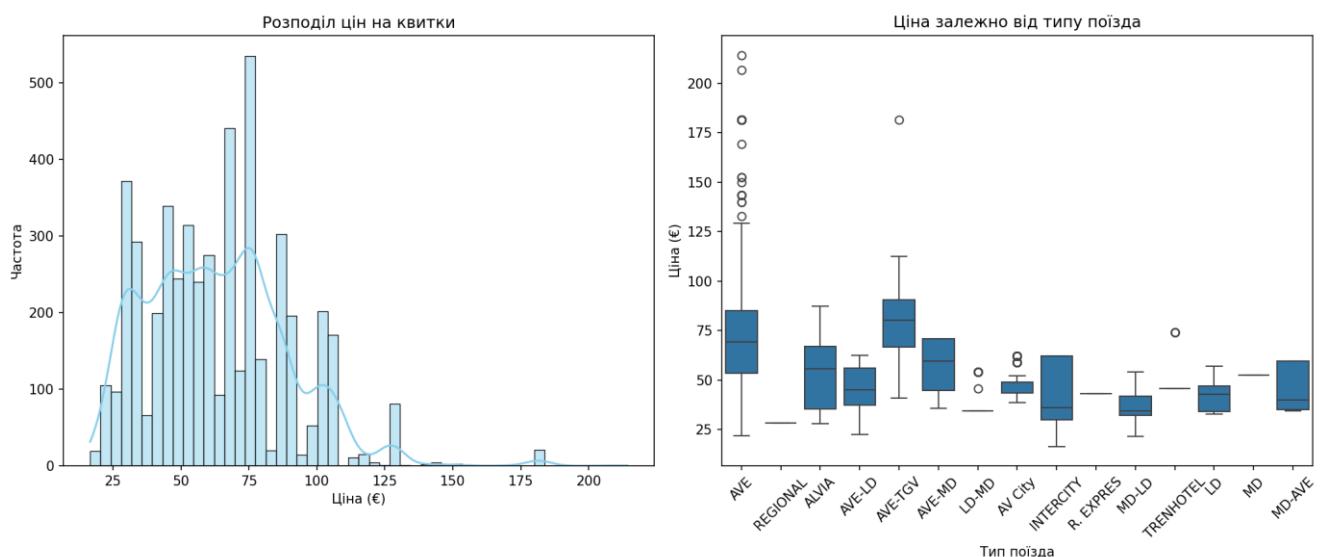


Рис.6.4.Результат виконання завдання

		Свистанюк Н.О.			ДУ «Житомирська політехніка». 25.121.22.000 – Пр6	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

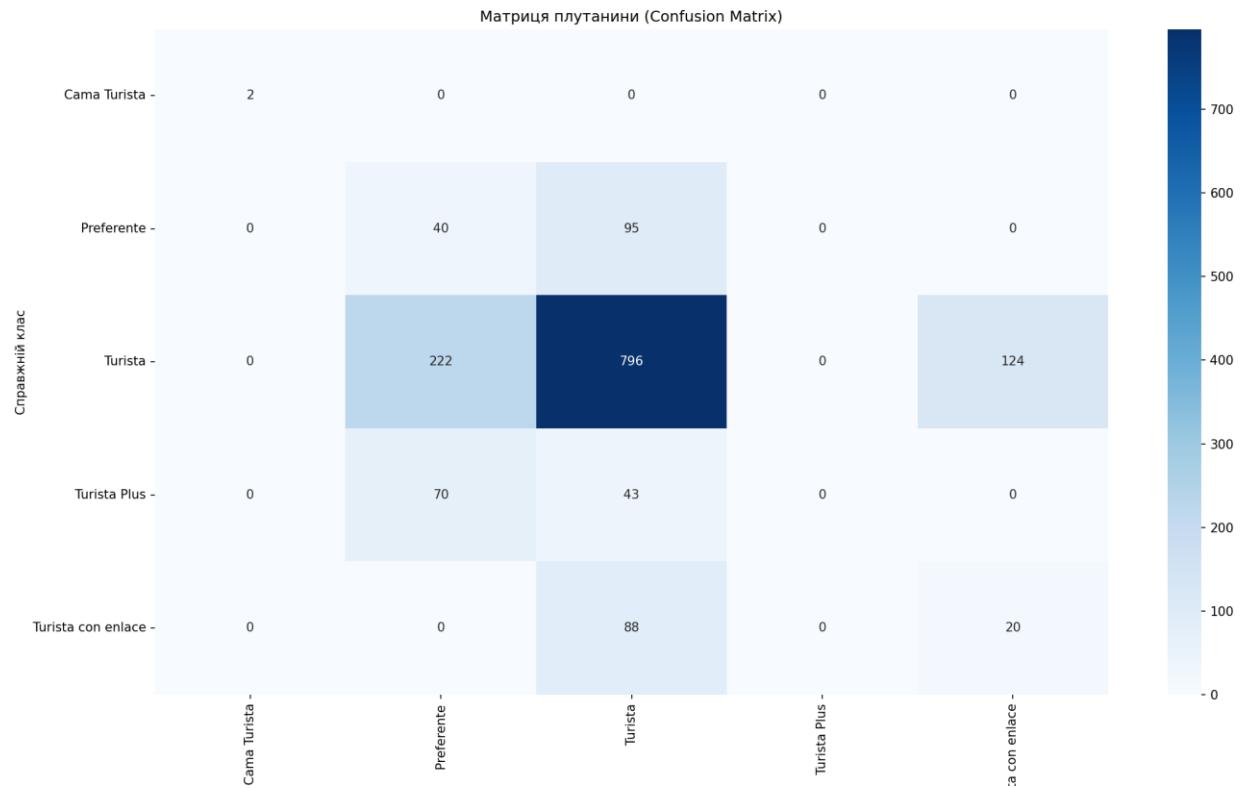


Рис.6.5. Результат виконання завдання

	precision	recall	f1-score	support
Cama Turista	1.00	1.00	1.00	2
Preferente	0.12	0.30	0.17	135
Turista	0.78	0.70	0.74	1142
Turista Plus	0.00	0.00	0.00	113
Turista con enlace	0.14	0.19	0.16	108
accuracy			0.57	1500
macro avg	0.41	0.44	0.41	1500
weighted avg	0.62	0.57	0.59	1500

Рис.6.6. Результат виконання завдання

Висновок: у ході виконання лабораторної роботи набуто практичних навичок застосування теореми Байєса в Python за допомогою бібліотеки scikit-learn. Для 22-го варіанту погодних умов алгоритм спрогнозував проведення гри з абсолютною ймовірністю. При аналізі даних перевезень Renfe застосування Гауссівського класифікатора дозволило успішно класифікувати квитки за ціною та типом поїзда, підтвердживши ефективність методу для ймовірнісного аналізу даних.

Репозиторій: <https://github.com/Svistaniuk/AIS>

Змн.	Арк.	Свистанюк Н.О.			ДУ «Житомирська політехніка». 25.121.22.000 – Пр6	Арк.
		Маєвський О.В.				