

ЛАБОРАТОРНА РОБОТА № 6

Наївний Байєс в Python

Мета роботи: набути навичок працювати з даними і опонувати роботу у Python 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 – 12.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import CategoricalNB
from sklearn.metrics import accuracy_score

# Дані
data = {
    'Outlook': ['Sunny', 'Sunny', 'Overcast', 'Rain', 'Rain', 'Rain', 'Over-
cast', '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']
}

# Перетворення даних у DataFrame
df = pd.DataFrame(data)

# Словники для мапінгу категоріальних значень
mappings = {
    'Outlook': {'Sunny': 0, 'Overcast': 1, 'Rain': 2},
    'Humidity': {'High': 0, 'Normal': 1},
    'Wind': {'Weak': 0, 'Strong': 1},
    'Play': {'No': 0, 'Yes': 1}
}

# Перетворення категорій у числові значення
for column, mapping in mappings.items():
    df[column] = df[column].map(mapping)

# Розділення на ознаки та цільову змінну
X = df[['Outlook', 'Humidity', 'Wind']]
```

					ДУ «Житомирська політехніка».24.121.12.000 – Лр6			
Змн.	Арк.	№ докум.	Підпис	Дата				
Виконав		Левкович О.О.			Звіт з лабораторної роботи		Лім.	Арк.
Перевір.		Іванов Д.А.						1
Керівник								5
Н. контр.							ФІКТ Гр. ІПЗк-23-1	
Зав. каф.								

```

y = df['Play']

# Розділення на тренувальні та тестові дані
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Модель Наївного Байєса
model = CategoricalNB()
model.fit(X_train, y_train)

# Прогнозування
y_pred = model.predict(X_test)

# Оцінка точності
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')

# Прогноз для нових умов
# Умови: Outlook=Overcast, Humidity=High, Wind=Strong
new_conditions = pd.DataFrame({
    'Outlook': [mappings['Outlook']['Overcast']],
    'Humidity': [mappings['Humidity']['High']],
    'Wind': [mappings['Wind']['Strong']]
})

# Прогнозування нових умов
prediction = model.predict(new_conditions)
prediction_label = 'Yes' if prediction[0] == mappings['Play']['Yes'] else 'No'
print(f"Прогноз: {prediction_label}")

```

```

# Прогнозування нових умов
prediction = model.predict(new_conditions)
prediction_label = 'Yes' if prediction[0] == mappings['Play']['Yes'] else 'No'
print(f"Weather-report: {prediction_label}")
print(f'Accuracy-number: {accuracy:.2f}')

```

⇒ Weather-report: Yes
Accuracy-number: 0.67

Рис. 1. Результат розрахунку.

		Левкович О.О.			ДУ «Житомирська політехніка». 24.121.12.000 – Лр6	Арк.
		Іванов Д.А.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновок: Модель Наївного Байєса успішно передбачає цільову змінну для категоріальних даних. Код демонструє перетворення даних, навчання моделі та прогнозування, забезпечуючи простоту та зрозумілість.

Результат прогнозу для умов мого варіанту - 12 - Outlook=Overcast, Humidity=High, Wind=Strong показує, що модель передбачає значення "Yes". Це означає, що за заданих умов модель прогнозує позитивний результат, тобто буде гра.

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

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Функція для прогнозування ціни на основі нових даних
def predict_price(train_type, origin, destination, train_class, fare, model,
X_columns):
    # Створення DataFrame з новими даними
    input_data = pd.DataFrame({
        'train_type': [train_type],
        'origin': [origin],
        'destination': [destination],
        'train_class': [train_class],
        'fare': [fare]
    })

    # Застосування one-hot encoding до нових даних
    input_data_encoded = pd.get_dummies(input_data)

    # Вирівнювання нових даних з тренувальним набором
    input_data_encoded = input_data_encoded.reindex(columns=X_columns,
fill_value=0)

    # Прогнозування ціни
    predicted_price = model.predict(input_data_encoded)
    return predicted_price[0]
```

		Левкович О.О.			ДУ «Житомирська політехніка». 24.121.12.000 – Лр6	Арк.
		Іванов Д.А.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

# Завантаження даних
url = "https://raw.githubusercontent.com/susanli2016/Machine-Learning-with-Python/master/data/renfe_small.csv"
data = pd.read_csv(url)

# Видалення пропущених значень
data = data.dropna()

# Вибір змінних для аналізу
selected_columns = ['train_type', 'origin', 'destination', 'train_class', 'fare', 'price']
data = data[selected_columns]

# Розділення ознак і цільової змінної
X = data[['train_type', 'origin', 'destination', 'train_class', 'fare']]
y = data['price']

# Перетворення категоріальних змінних на числові (one-hot encoding)
X = pd.get_dummies(X)

# Розділення на тренувальний і тестовий набори
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Створення та навчання моделі
model = LinearRegression()
model.fit(X_train, y_train)

# Прогнозування на тестових даних
y_pred = model.predict(X_test)

# Оцінка моделі
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"\nСередня квадратична похибка (MSE): {mse:.2f}")
print(f"Коефіцієнт детермінації (R^2): {r2:.2f}")

# Приклади реальних і прогнозованих цін
print("\nПриклади реальних і прогнозованих цін:")
comparison = pd.DataFrame({'Реальна ціна': y_test[:10].values, 'Прогнозована ціна': y_pred[:10]})
print(comparison)

# Прогноз для нових даних

```

		Левкович О.О.			ДУ «Житомирська політехніка». 24.121.12.000 – Лр6	Арк.
		Іванов Д.А.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

```
new_data_prediction = predict_price('AVE', 'Madrid', 'Barcelona', 'Prefer-
ente', 55.3, model, X.columns)
print(f"\nПрогнозована ціна для нових даних: {new_data_prediction:.2f} EUR")
```

Результат

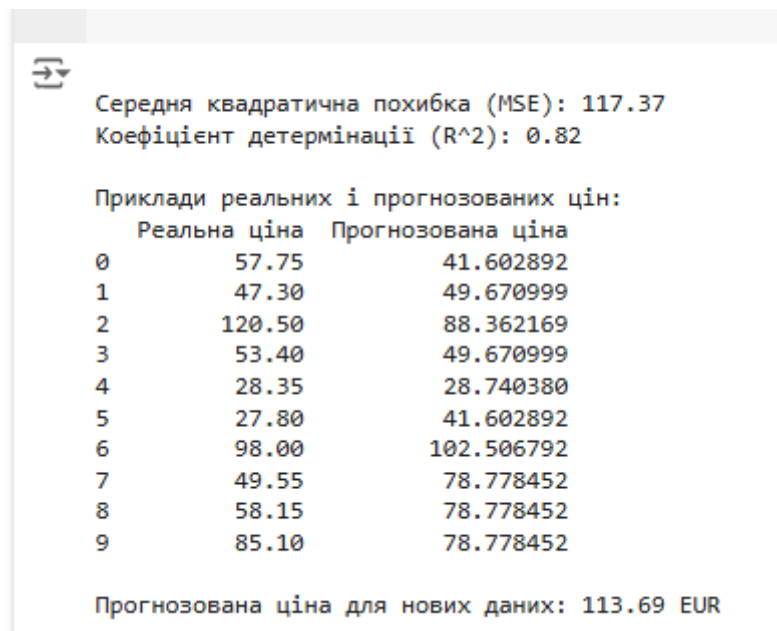


Рис. 2. Результат написання коду байєсівського аналізу до набору даних про ціни на квитки на іспанські високошвидкісні залізниці.

Висновок: У цьому завданні для прогнозування цін на квитки на іспанські високошвидкісні залізниці було використано методи байєсівського аналізу, які дозволяють оцінювати не лише прогноз, але й неважаченість у моделях.

Байєсівська лінійна регресія дає змогу враховувати розподіли ймовірностей для параметрів, що дозволяє створювати довірчі інтервали для прогнозованих значень і оцінювати варіативність результатів.

Github - <https://github.com/TAMOTO24/-Intelligen-Systems>

		Левкович О.О.			ДУ «Житомирська політехніка». 24.121.12.000 – Лр6	Арк.
		Іванов Д.А.				5
Змн.	Арк.	№ докум.	Підпис	Дата		