**ЛАБОРАТОРНА РОБОТА № 6**

**Наївний Байєс в Python**

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

**Завдання 3**. Використовуя 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', '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']

}

# Перетворення даних у DataFrame

df = pd.DataFrame(data)

# Словники для маппінгу 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']]

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}")

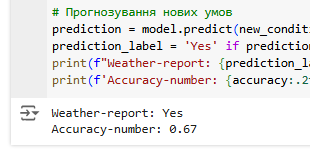


Рис. 1. Результат розрахунку.

***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]

# 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

new\_data\_prediction = predict\_price('AVE', 'Madrid', 'Barcelona', 'Preferente', 55.3, model, X.columns)

print(f"\nПрогнозована ціна для нових даних: {new\_data\_prediction:.2f} EUR")

Результат

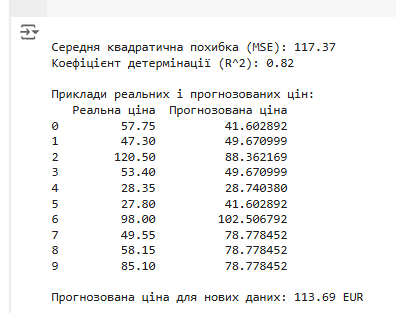


Рис. 2. Результат написання коду байєсівського аналізу до набору даних про ціни на квитки на іспанські високошвидкісні залізниці.

***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 невазначеність 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