

Лабораторна робота №6

Наївний Байєс в Python

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

Завдання 1. Використовуючи данні з пункту 2 визначити відбудеться матч при наступних погодних умовах чи ні: Розрахунки провести з використанням Python.

```
from collections import Counter

# Вхідні дані
data = [
    {"Outlook": "Sunny", "Humidity": "High", "Wind": "Weak", "Play": "No"},
    {"Outlook": "Sunny", "Humidity": "High", "Wind": "Strong", "Play": "No"},
    {"Outlook": "Overcast", "Humidity": "High", "Wind": "Weak", "Play": "Yes"},
    {"Outlook": "Rain", "Humidity": "High", "Wind": "Weak", "Play": "Yes"},
    {"Outlook": "Rain", "Humidity": "Normal", "Wind": "Weak", "Play": "Yes"},
    {"Outlook": "Rain", "Humidity": "Normal", "Wind": "Strong", "Play": "No"},
    {"Outlook": "Overcast", "Humidity": "Normal", "Wind": "Strong", "Play": "Yes"},
    {"Outlook": "Sunny", "Humidity": "High", "Wind": "Weak", "Play": "No"},
    {"Outlook": "Sunny", "Humidity": "Normal", "Wind": "Weak", "Play": "Yes"},
    {"Outlook": "Rain", "Humidity": "High", "Wind": "Weak", "Play": "Yes"},
    {"Outlook": "Sunny", "Humidity": "Normal", "Wind": "Strong", "Play": "Yes"},
    {"Outlook": "Overcast", "Humidity": "High", "Wind": "Strong", "Play": "Yes"},
    {"Outlook": "Overcast", "Humidity": "Normal", "Wind": "Weak", "Play": "Yes"},
    {"Outlook": "Rain", "Humidity": "High", "Wind": "Strong", "Play": "No"},
]

# Функція для підрахунку умовних ймовірностей
def calculate_conditional_probability(attribute, value, target, target_value, data):
    relevant = [row for row in data if row[target] == target_value]
    count_matches = sum(1 for row in relevant if row[attribute] == value)
    return count_matches / len(relevant) if relevant else 0

# Функція для підрахунку загальної ймовірності
def calculate_total_probability(target_value, conditions, data):
    target_count = sum(1 for row in data if row["Play"] == target_value)
    total_count = len(data)
    target_probability = target_count / total_count

    conditional_probabilities = [
        calculate_conditional_probability(attr, val, target="Play", target_value, data)
        for attr, val in conditions.items()
    ]

    combined_probability = target_probability
    for prob in conditional_probabilities:
        combined_probability *= prob

    return combined_probability
```

Рис 1.1 – лістинг програми

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.121.15.000 – Лр.6		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Прокопчук О.С			Звіт з лабораторної роботи №6	Лім.	Арк.
Перевір.		Голенко М.Ю.					Аркушів
Реценз.							1
Н. Контр.							4
Зав.каф.						ФІКТ, гр. ІПЗ-21-1(2)	

```

# Умови для розгляду за варіантом
conditions = {"Outlook": "Rain", "Humidity": "High", "Wind": "Strong"}

# Обчислення ймовірностей
prob_yes = calculate_total_probability(target_value: "Yes", conditions, data)
prob_no = calculate_total_probability(target_value: "No", conditions, data)

# Нормалізація
total = prob_yes + prob_no

if total != 0:
    normalized_yes = prob_yes / total
    normalized_no = prob_no / total
else:
    normalized_yes = 0
    normalized_no = 0

# Вивід результатів
print("Чи відбудеться матч?")
print(f"Умови: {conditions}")
print(f"Ймовірність, що матч відбудеться 'Yes' --> {normalized_yes:.2f}")
print(f"Ймовірність, що матч НЕ відбудеться 'No' --> {normalized_no:.2f}")

```

Рис 1.2 – лістинг програми

```

C:\Users\Администратор\PycharmProjects\.venv\Scripts\python.exe C:\U
Чи відбудеться матч?
Умови: {'Outlook': 'Rain', 'Humidity': 'High', 'Wind': 'Strong'}
Ймовірність, що матч відбудеться 'Yes' --> 0.32
Ймовірність, що матч НЕ відбудеться 'No' --> 0.68

```

Рис 1.3 – результат виконання

Завдання 2. Застосуєте методи байєсівського аналізу до набору даних про ціни на квитки на іспанські високошвидкісні залізниці.

Вхідні дані: https://raw.githubusercontent.com/susanli2016/MachineLearning-with-Python/master/data/renfe_small.csv

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.121.15.000 – Лр.6	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

```

import pandas as pd
from collections import defaultdict

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

# Зберігається відповідні стовпці та опускайте пропущені значення
data = data[["price", "train_type", "origin", "destination", "train_class"]].dropna()

# Розділ цін на 'low', 'medium', 'high'
price_bins = [data["price"].min(), data["price"].quantile(0.33), data["price"].quantile(0.66), data["price"].max()]
data["price_category"] = pd.cut(data["price"], bins=price_bins, labels=["low", "medium", "high"])

# Обчислення частотних розподілів
frequency_distributions = defaultdict(lambda: defaultdict(int))
total_per_category = defaultdict(int)

for _, row in data.iterrows():
    category = row["price_category"]
    total_per_category[category] += 1
    for feature in ["train_type", "origin", "destination", "train_class"]:
        frequency_distributions[feature][(row[feature], category)] += 1

# Обчислення умовних ймовірностей
def conditional_probability(feature, value, category): 1 usage new *
    match_count = frequency_distributions[feature].get((value, category), 0)
    category_total = total_per_category[category]
    return match_count / category_total if category_total else 0

# Визначення калькулятора ймовірностей за теоремою Байєса
def calculate_category_probability(category, conditions): 1 usage new *
    prior = total_per_category[category] / len(data)
    likelihood = 1
    for feature, value in conditions.items():
        likelihood *= conditional_probability(feature, value, category)
    return prior * likelihood

# Визначте характеристики квитка для прогнозування
ticket_conditions = {
    "train_type": "AVE",
    "origin": "MADRID",
    "destination": "SEVILLA",
    "train_class": "Turista"
}

```

Рис 1.4 – лістинг програми

```

# Розрахувати апостеріорні ймовірності для кожної цінової категорії
posterior_probabilities = {}
for category in total_per_category.keys():
    posterior_probabilities[category] = calculate_category_probability(category, ticket_conditions)

# Нормалізація ймовірностей
total_probability = sum(posterior_probabilities.values())
normalized_probabilities = {cat: prob / total_probability for cat, prob in posterior_probabilities.items()}

print("Ймовірності для кожної категорії вартості квитка -->")
for category, probability in normalized_probabilities.items():
    print(f"{str(category).capitalize()}: {probability:.2f}")

```

Рис 1.5 – лістинг програми

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.121.15.000 – Лр.6	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

C:\Users\Администратор\PycharmProjects\.venv\Scripts\pyt
Ймовірності для кожної категорії вартості квитка -->
Low: 0.13
High: 0.37
Medium: 0.50
Nan: 0.00

```

Рис 1.6 – результат виконання

Висновок: На лабораторній роботі мала змогу набути навичок праці з даними і опанувати роботу у Python з використанням теореми Байєса.

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.121.15.000 – Лр.6	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		