

ЛАБОРАТОРНА РОБОТА № 6

Наївний Байєс в Python

Мета: використовуючи спеціалізовані бібліотеки та мову програмування Python дослідити методи ансамблів у машинному навчанні.

Варіант 1

Хід роботи:

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.

```
from collections import Counter

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"},
]
```

					ДУ «Житомирська політехніка».24.121.01.000 – Лр6			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Барабаш В.В.			Звіт з лабораторної роботи		Літ.	Арк.
Перевір.		Черняк І.О.						1
Керівник							ФІКТ Гр. ІПЗ-21-3	
Н. контр.								
Зав. каф.								
							4	

```

play_count = Counter([row["Play"] for row in data])
outlook_count = Counter([(row["Outlook"], row["Play"]) for row in data])
humidity_count = Counter([(row["Humidity"], row["Play"]) for row in data])
wind_count = Counter([(row["Wind"], row["Play"]) for row in data])

total_play_yes = play_count["Yes"] / len(data)
total_play_no = play_count["No"] / len(data)

p_outlook_yes = outlook_count[("Overcast", "Yes")] / play_count["Yes"]
p_outlook_no = outlook_count[("Overcast", "No")] / play_count["No"]

p_humidity_yes = humidity_count[("High", "Yes")] / play_count["Yes"]
p_humidity_no = humidity_count[("High", "No")] / play_count["No"]

p_wind_yes = wind_count[("Weak", "Yes")] / play_count["Yes"]
p_wind_no = wind_count[("Weak", "No")] / play_count["No"]

p_yes = p_outlook_yes * p_humidity_yes * p_wind_yes * total_play_yes
p_no = p_outlook_no * p_humidity_no * p_wind_no * total_play_no

p_yes_normalized = p_yes / (p_yes + p_no)
p_no_normalized = p_no / (p_yes + p_no)

print(f"Ймовірність, що матч відбудеться (Yes): {p_yes_normalized:.2f}")
print(f"Ймовірність, що матч не відбудеться (No): {p_no_normalized:.2f}")

```

```

➡ Ймовірність, що матч відбудеться (Yes): 1.00
   Ймовірність, що матч не відбудеться (No): 0.00

```

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

```

import pandas as pd
from itertools import product

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

columns_of_interest = ["price", "train_type", "origin", "destination",
                        "train_class"]
data_cleaned = data[columns_of_interest].dropna()

price_labels = ["low", "medium", "high"]

```

		Барабаш В.В.			ДУ «Житомирська політехніка».24.121.01.000 – Лр6	Арк.
		Черняк І.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

```

data_cleaned["price_category"] = pd.cut(data_cleaned["price"], bins=3,
labels=price_labels)

def count_occurrences(dataframe, group_by_cols):
    grouped = dataframe.groupby(group_by_cols).size()
    return grouped.to_dict()

train_type_frequencies = count_occurrences(data_cleaned, ["train_type",
"price_category"])
origin_frequencies = count_occurrences(data_cleaned, ["origin",
"price_category"])
destination_frequencies = count_occurrences(data_cleaned, ["destination",
"price_category"])
class_frequencies = count_occurrences(data_cleaned, ["train_class",
"price_category"])
price_category_counts =
data_cleaned["price_category"].value_counts().to_dict()

def conditional_probability(category, feature_value, frequency_table,
total_counts):
    if total_counts.get(category, 0) == 0:
        return 0
    return frequency_table.get((feature_value, category), 0) /
total_counts[category]

selected_train_type = "AVE"
selected_origin = "SEVILLA"
selected_destination = "MADRID"
selected_class = "Preferente"

def calculate_probabilities(dataset, train_type, origin, destination,
train_class):
    total_records = len(dataset)
    probabilities = {}

    for category in price_category_counts.keys():
        p_train_type = conditional_probability(
            category, train_type, train_type_frequencies,
price_category_counts
        )
        p_origin = conditional_probability(
            category, origin, origin_frequencies, price_category_counts
        )
        p_destination = conditional_probability(
            category, destination, destination_frequencies,
price_category_counts
        )
        p_class = conditional_probability(
            category, train_class, class_frequencies, price_category_counts
        )

```

		Барабаш В.В.			ДУ «Житомирська політехніка».24.121.01.000 – Лр6	Арк.
		Черняк І.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		3

```

    prior_prob = price_category_counts[category] / total_records

    probabilities[category] = p_train_type * p_origin * p_destination *
p_class * prior_prob

    total_prob = sum(probabilities.values())
    if total_prob > 0:
        probabilities = {k: v / total_prob for k, v in probabilities.items()}

    return probabilities

resulting_probabilities = calculate_probabilities(
    data_cleaned, selected_train_type, selected_origin, selected_destination,
selected_class
)

formatted_result = {cat: f"{prob:.2f}" for cat, prob in
resulting_probabilities.items()}

print(
    f"Ймовірності для цінкових категорій за параметрами "
    f"({selected_train_type}, {selected_class}, {selected_origin},
{selected_destination}):"
)
print(formatted_result)

```

```

Ймовірності для цінкових категорій за параметрами (AVE, Preferente, SEVILLA, MADRID):
{'low': '0.92', 'medium': '0.08', 'high': '0.00'}

```

Посилання на Github:

https://github.com/Vladislav2533/SHI_Barabash_Vlad_IPZ_21_3

Висновки: використав спеціалізовані бібліотеки та мову програмування Python, та дослідив методи ансамблів у машинному навчанні.

		Барабаш В.В.			ДУ «Житомирська політехніка».24.121.01.000 – Лр6	Арк.
		Черняк І.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		4