Лабораторна робота №6

Наївний Байєс в Python

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

Література

Supervised learning - https://scikit-learn.org/stable/user_guide.html
Naive Bayes Tutorial: Naive Bayes Classifier in Python
https://dzone.com/articles/naive-bayes-tutorial-naive-bayes-classifier-in-pyt
Наивный баейсовский классификатор - http://datascientist.one/naive-bayes/

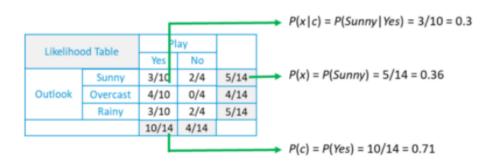
Хід роботи:

Завдання 1. Ретельно опрацювати теоретичні відомості:

- теорему Байєса;
- які типи наївного байєсівського класифікатора є;
- де використовується Наївний Байєс.

Завдання 2. Ретельно розібрати приклад: прогнозування з використанням теореми Байєса.

В ході розбору прикладу в ньому було виявлено декілька неточностей в обчисленнях та значеннях.



в таблиці тотал єс та тотал ні було вказано 10 та 4, в прикладі ж ці значення були 9 та 5, я виправив власну таблицю і вона має ось такий вигляд:

```
data = {
    "Outlook": {
        "rainy": {"Yes": 2, "No": 3},
        "sunny": {"Yes": 3, "No": 2},
        "overcast": {"Yes": 4, "No": 0}
    },
    "Humidity": {
        "high": {"Yes": 3, "No": 4},
        "normal": {"Yes": 6, "No": 1}
    },
    "Wind": {
        "weak": {"Yes": 6, "No": 2},
        "strong": {"Yes": 3, "No": 3}
    }
}
```

Тепер значення порівнялись а таблиця стала навіть логічнішою (при дощі ні має більше значення).

Крім того, в прикладі вказаному в ЛР:

Ймовірність «Yes» в цей день = P(Outlook = Rain|Yes)*P(Humidity= High|Yes)* P(Wind= Weak|Yes)*P(Yes)= <math>2/9*3/9*6/9*9/14=0,0199

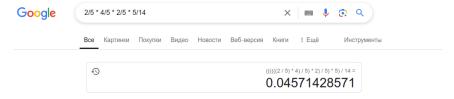
Ймовірність Так в цей день порахована не вірно:

```
(((((((2 / 9) * 3) / 9) * 6) / 9) * 9) / 14 =
```

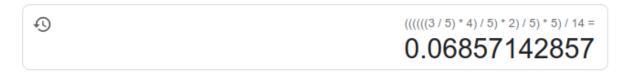
0.03174603174

Як і ось тут:

Імовірність негативної відповіді «No» в цей день = P(Outlook = Rain|No)*P(Humidity= High|No)* P(Wind= Weak|No)*P(No)= 2/5 * 4/5 * 2/5 * 5/14 = 0,0166



Плюс до того, мої значення будуть відрізнятись, бо за таблицею в мене при дощі коефіцієнт на $\mathrm{Hi}\ \%$ а не % тому:



Тому ймовірність на Так в цей день:

Та на Ні:

Перейдемо до 3 завдання:

Завдання 3. Використовуя данні з пункту 2 визначити відбудеться матч при наступних погодних умовах чи ні: Розрахунки провести з використанням Python.

Ще в ході виконання завдання два було реалізовано програмний код на мові Python, який обчислює це все самостійно:

Лістинг:

```
data = {
    "Outlook": {
          "rainy": {"Yes": 2, "No": 3},
          "sunny": {"Yes": 3, "No": 2},
          "overcast": {"Yes": 4, "No": 0}
},

"Humidity": {
          "high": {"Yes": 3, "No": 4},
          "normal": {"Yes": 6, "No": 1}
},

"Wind": {
          "weak": {"Yes": 6, "No": 2},
          "strong": {"Yes": 3, "No": 3}
}

total_yes = 9
```

```
total no = 5
total = total yes + total no
p yes = total yes / total
p no = total no / total
conditions = {
   "Outlook": input("Enter outlook (sunny, overcast, rainy):
").lower(),
  "Humidity": input("Enter humidity level (high, normal): ").lower(),
   "Wind": input("Enter wind strength (weak, strong): ").lower()
p rain yes = data["Outlook"][conditions["Outlook"]]["Yes"] / total yes
p rain no = data["Outlook"][conditions["Outlook"]]["No"] / total no
p humidity yes = data["Humidity"][conditions["Humidity"]]["Yes"] /
total yes
p humidity no = data["Humidity"][conditions["Humidity"]]["No"] /
total no
p wind yes = data["Wind"][conditions["Wind"]]["Yes"] / total yes
p wind no = data["Wind"][conditions["Wind"]]["No"] / total no
p yes given conditions = p rain yes * p humidity yes * p wind yes *
p yes
p no given conditions = p rain no * p humidity no * p wind no * p no
total_probability = p_yes_given_conditions + p_no_given_conditions
p yes final = p yes given conditions / total probability
p_no_final = p_no_given_conditions / total_probability
print(f"Ймовірність, що матч відбудеться (Yes): {p_yes_final:.2f}")
print(f"Ймовірність, що матч не відбудеться (No): \{p \text{ no final:.2f}\}")
```

Також звіримо його обчилення з тим що в нас вийшло в минулому завданні:

Тому ймовірність на Так в цей день:

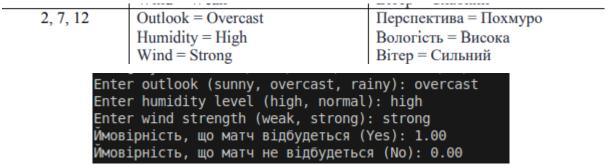
```
 \begin{array}{c} \textcircled{(((((((2/9)*3)/9)*6)/9)*9)/14)/(((((((2/9)*3)/9)*6)/9)*9)/14)+(((((((3/5)*4)/5)*2).\\ \textbf{0.3164556962} \end{array} )
```

Та на Ні:

```
● root@dmytro-X550CL:/home/alhim/Завантаження/ШI/lab6# /bin/python3 /home/alhim/Завантаження/ШI/lab6/lab6.py
Enter outlook (sunny, overcast, rainy): Rainy
Enter humidity level (high, normal): High
Enter wind strength (weak, strong): weak
Ймовірність, що матч відбудеться (Yes): 0.32
Ймовірність, що матч не відбудеться (No): 0.68
○ root@dmytro-X550CL:/home/alhim/Завантаження/ШI/lab6#
```

Загалом программа дозволяє дослідити всі варіанти, але перейдемо саме до нашого та дослідимо інші:

Мій варіант 27, їх всього 15 тому виконуємо 12:



Також виконаємо решту:

Варіант	Умова	
1, 6, 11	Outlook = Overcast	Перспектива = Похмуро
	Humidity = High	Вологість = Висока
	Wind = Weak	Вітер = Слабкий
		
3, 8, 13	Outlook = Sunny	Перспектива = Сонячно
	Humidity = High	Вологість = Висока
	Wind = Weak	Вітер = Слабкий
4, 9, 14	Outlook = Sunny	Перспектива = Сонячно
	Humidity = Normal	Вологість = Нормальна
	Wind = Strong	Вітер = Сильний
5, 10, 15	Outlook = Rain	Outlook = Дощ
	Humidity = High	Вологість = Висока
	Wind = Strong	Вітер = Сильний

```
    root@dmytro-X550CL:/home/alhim/Завантаження/ШI/lab6# /bin/python3 /home/alhim/Завантаження/ШI/lab6/lab6.py Enter outlook (sunny, overcast, rainy): overcast Enter humidity level (high, normal): high Enter wind strength (weak, strong): weak Ймовірність, що матч відбудеться (Yes): 1.00 Ймовірність, що матч відбудеться (No): 0.00
    root@dmytro-X550CL:/home/alhim/Завантаження/ШI/lab6# /bin/python3 /home/alhim/Завантаження/ШI/lab6/lab6.py Enter outlook (sunny, overcast, rainy): Sunny Enter humidity level (high, normal): high Enter wind strength (weak, strong): weak Ймовірність, що матч відбудеться (Yes): 0.51 Ймовірність, що матч відбудеться (No): 0.49
    root@dmytro-X550CL:/home/alhim/Завантаження/ШI/lab6# /bin/python3 /home/alhim/Завантаження/ШI/lab6/lab6.py Enter outlook (sunny, overcast, rainy): sunny Enter humidity level (high, normal): normal Enter wind strength (weak, strong): strong Ймовірність, що матч відбудеться (Yes): 0.74 Ймовірність, що матч не відбудеться (Yes): 0.74
    Ймовірність, що матч не відбудеться (No): 0.26
```

```
● root@dmytro-X550CL:/home/alhim/Завантаження/ШI/lab6# /bin/python3 /home/alhim/Завантаження/ШI/lab6/lab6.py
Enter outlook (sunny, overcast, rainy): rainy
Enter humidity level (high, normal): high
Enter wind strength (weak, strong): strong
Ймовірність, що матч відбудеться (Yes): 0.13
Ймовірність, що матч не відбудеться (No): 0.87
```

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

– Bхідні дані: https://raw.githubusercontent.com/susanli2016/Machine-Learning-with-Python/master/data/renfe small.csv

Лістинг:

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import BayesianRidge
from sklearn.metrics import mean_squared_error

data = pd.read_csv('data.txt')
```

```
data = data.dropna(subset=['price'])
data['origin enc'] = <u>LabelEncoder</u>().fit_transform(data['origin'])
data['destination enc'] =
<u>LabelEncoder().fit transform(data['destination'])</u>
data['train type enc'] =
<u>LabelEncoder().fit transform(data['train type'])</u>
data['train class enc'] =
<u>LabelEncoder().fit transform(data['train class'])</u>
data['fare enc'] = <u>LabelEncoder</u>().fit transform(data['fare'])
X = data[['origin enc', 'destination enc', 'train_type_enc',
'train class enc', 'fare enc']]
y = data['price']
X_train, X_test, y_train, y_test = train_test_split(X, y,
model = <u>BayesianRidge</u>()
model.fit(X train, y train)
y pred = model.predict(X test)
mse = mean squared error(y test, y pred)
print(f"Середньоквадратична похибка на тестовому наборі: {mse}")
def predict price(origin, destination, train type, train class, fare):
    origin enc =
<u>LabelEncoder().fit(data['origin']).transform([origin])[0]</u>
    destination enc =
<u>LabelEncoder().fit(data['destination']).transform([destination])[0]</u>
    train_type enc =
<u>LabelEncoder().fit(data['train type']).transform([train type])[0]</u>
    train class enc =
<u>LabelEncoder().fit(data['train class']).transform([train class])[0]</u>
    fare enc = LabelEncoder().fit(data['fare']).transform([fare])[0]
    input data = pd.DataFrame([[origin enc, destination enc,
train type enc, train class enc, fare enc]],
                               columns=['origin enc', 'destination enc',
'train type enc', 'train class enc', 'fare enc'])
    predicted price = model.predict(input data)[0]
    return predicted price
```

```
origin = input("Enter the origin station: ")
destination = input("Enter the destination station: ")
train_type = input("Enter the train type (e.g., AVE): ")
train_class = input("Enter the train class (e.g., Turista): ")
fare = input("Enter the fare type (e.g., Promo): ")

predicted_price = predict_price(origin, destination, train_type,
train_class, fare)
print(f"The predicted price is: {predicted_price}")
```

```
Середньоквадратична похибка на тестовому наборі: 218.33846291905388
Enter the origin station: MADRID
Enter the destination station: VALENCIA
Enter the train type (e.g., AVE): AVE
Enter the train class (e.g., Turista): Turista
Enter the fare type (e.g., Promo): Promo
The predicted price is: 44.10358351408544
```

GitHub: https://github.com/Alhim616/AI Labs Yanushevych