T7_NB ЗАВДАННЯ

ВИБІР СІМЕЙСТВА РОЗПОДІЛІВ В НАЇВНОМУ БАЙЄСІ

В цьому завданні Вам пропонується **з'ясувати, який розподіл краще використовувати в наївному байєсівської класифікаторі залежно від виду ознак.**

Завантажте датасета digits і breast_cancer з sklearn.datasets. Виведіть кілька рядків з навчальних вибірок і подивіться на ознаки. За допомогою sklearn.cross_validation.cross_val_score з типовими налаштуваннями і викликом методу mean () у об'єкта типу numpy.ndarray, порівняйте якість роботи наївних байесовских класифікаторів на цих двох датасета. Для порівняння пропонується використовувати BernoulliNB, MultinomialNB і GaussianNB. Наскільки отримані результати узгоджуються з рекомендаціями з лекцій?

Два датасета, звичайно, ще не привід робити далекосяжні висновки, але при бажанні ви можете продовжити дослідження на інших вибірках.

Для здачі завдання, дайте відповідь на наведені нижче питання.

Встановлення найновіших біблотек, щоб не було проблем з числовими відповідями

In [ ]:
```
import sklearn
import numpy as np
import pandas as pd
```

In [ ]:
```
# Перевірочні дані для лабораторної роботи отримані з використанням таких версій бібліотек
!pip install "scikit-learn == 0.24.2"
!pip install "numpy == 1.22.4"
```

```
Requirement already satisfied: scikit-learn==0.24.2 in c:\users\admin\anaconda3\lib\site-packages (0.24.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\admin\anaconda3\lib\site-packages (from scikit-learn==0.24.2) (2.2.0)
Requirement already satisfied: numpy>=1.13.3 in c:\users\admin\anaconda3\lib\site-packages (from scikit-learn==0.24.2) (1.22.4)
Requirement already satisfied: joblib>=0.11 in c:\users\admin\anaconda3\lib\site-packages (from scikit-learn==0.24.2) (1.1.0)
Requirement already satisfied: scipy>=0.19.1 in c:\users\admin\anaconda3\lib\site-packages (from scikit-learn==0.24.2) (1.9.1)
Requirement already satisfied: numpy==1.22.4 in c:\users\admin\anaconda3\lib\site-packages (1.22.4)
```

In [ ]:
```
# Перевірити версії бібліотек можна таким чином (перед цим їх потрібно імпортувати)
print('sklearn',sklearn.__version__)
print('numpy',np.__version__)
```

```
sklearn 0.24.2
numpy 1.22.4
```

Завантажуємо необхідні бібліотеки.

In [ ]:
```
from sklearn import datasets
```

In [ ]:
```
from sklearn.naive_bayes import BernoulliNB, MultinomialNB, GaussianNB
```

In [ ]:
```
from sklearn.model_selection import cross_val_score
```

Завантажуємо датасет **digits**. Та позначаємо незалежні та залежні змінні.

In [ ]:
```
digits = datasets.load_digits()
X_digits = digits.data
y_digits = digits.target
```

In [ ]:
```
type(digits) # sklearn.utils.Bunch
```

Out[ ]:
```
sklearn.utils.Bunch
```

Щоб вивести кілька рядків з навчальної вибірки і подивитися на ознаки потрібно привести тип до *dataframe*.

In [ ]:
```
digits_DF = pd.DataFrame(data = np.c_[digits['data'], digits['target']],
                         columns = digits['feature_names'] + ['target'])
```

In [ ]:
```
digits_DF.head()
```

Out[ ]:

	pixel_0_0	pixel_0_1	pixel_0_2	pixel_0_3	pixel_0_4	pixel_0_5	pixel_0_6	pixel_0_7	pixel_1_0	pixel_1_1	...	pixel_6_7	pixel_7_0	pixel_7_1	pixel_7_2	pixel_7_3	pixel_7_4	pixel_7_5	pixel_7_6	pixel_7_7	target
0	0.0	0.0	5.0	13.0	9.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	6.0	13.0	10.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	12.0	13.0	5.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	11.0	16.0	10.0	0.0	0.0	1.0
2	0.0	0.0	0.0	4.0	15.0	12.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	3.0	11.0	16.0	9.0	0.0	2.0
3	0.0	0.0	7.0	15.0	13.0	1.0	0.0	0.0	0.0	8.0	...	0.0	0.0	0.0	7.0	13.0	13.0	9.0	0.0	0.0	3.0
4	0.0	0.0	0.0	1.0	11.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	2.0	16.0	4.0	0.0	0.0	4.0

5 rows × 65 columns

Завантажуємо датасет **breast_cancer**. Та позначаємо незалежні та залежні змінні.

In [ ]:
```
breast_cancer = datasets.load_breast_cancer()
X_breast_cancer = breast_cancer.data
y_breast_cancer = breast_cancer.target
```

In [ ]:
```
type(breast_cancer) # sklearn.utils.Bunch
```

Out[ ]:
```
sklearn.utils.Bunch
```

Щоб вивести кілька рядків з навчальної вибірки і подивитися на ознаки потрібно привести тип до *dataframe*.

In [ ]:
```
breast_cancer_DF = pd.DataFrame(data = breast_cancer.data, columns = breast_cancer.feature_names)
```

In [ ]:
```
breast_cancer_DF.head()
```

Out[ ]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst radius	worst texture	worst perimeter	worst area	worst smoothness	worst compactness	worst concavity	worst concave points	worst symmetry	worst fractal dimension
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	...	25.38	17.33	184.60	2019.0	0.1622	0.6656	0.7119	0.2654	0.4601	0.11890
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	...	24.99	23.41	158.80	1956.0	0.1238	0.1866	0.2416	0.1860	0.2750	0.08902
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	...	23.57	25.53	152.50	1709.0	0.1444	0.4245	0.4504	0.2430	0.3613	0.08758
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	...	14.91	26.50	98.87	567.7	0.2098	0.8663	0.6869	0.2575	0.6638	0.17300
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	...	22.54	16.67	152.20	1575.0	0.1374	0.2050	0.4000	0.1625	0.2364	0.07678

5 rows × 30 columns

За допомогою sklearn.cross_validation.cross_val_score з типовими налаштуваннями і викликом методу mean () у об'єкта типу numpy.ndarray, порівняємо якість роботи наївних байесовских класифікаторів на цих двох датасета. Для порівняння пропонується використовувати BernoulliNB, MultinomialNB і GaussianNB.

In [ ]:
```
Bernoulli = BernoulliNB()
Multinomial = MultinomialNB()
Gaussian = GaussianNB()
```

Розглянемо датасет **digits**.

In [ ]:
```
score_Bernoulli_digits   = cross_val_score(Bernoulli, X_digits, y_digits).mean()
score_Multinomial_digits = cross_val_score(Multinomial, X_digits, y_digits).mean()
score_Gaussian_digits    = cross_val_score(Gaussian, X_digits, y_digits).mean()
```

In [ ]:
```
print('score_Bernoulli_digits   ', score_Bernoulli_digits)
print('score_Multinomial_digits ', score_Multinomial_digits)
print('score_Gaussian_digits    ', score_Gaussian_digits)
```

```
score_Bernoulli_digits    0.8241736304549674
score_Multinomial_digits  0.8703497369235531
score_Gaussian_digits     0.8069281956050759
```

Розглянемо датасет **breast_cancer**.

In [ ]:
```
score_Bernoulli_breast_cancer   = cross_val_score(Bernoulli, X_breast_cancer, y_breast_cancer).mean()
score_Multinomial_breast_cancer = cross_val_score(Multinomial, X_breast_cancer, y_breast_cancer).mean()
score_Gaussian_breast_cancer    = cross_val_score(Gaussian, X_breast_cancer, y_breast_cancer).mean()
```

In [ ]:
```
print('score_Bernoulli_breast_cancer   ', score_Bernoulli_breast_cancer)
print('score_Multinomial_breast_cancer ', score_Multinomial_breast_cancer)
print('score_Gaussian_breast_cancer    ', score_Gaussian_breast_cancer)
```

```
score_Bernoulli_breast_cancer    0.6274181027790716
score_Multinomial_breast_cancer  0.8963204471355379
score_Gaussian_breast_cancer     0.9385188635305075
```

Завдання 1

Яка максимальна якість класифікації на датасеті **breast_cancer**?

Результат округліть до чотирьох чисел після коми та завантажіть у відповідну комірку Google Forms. Використовувати правила округлення.

In [ ]:
```
mas_breast_cancer = [score_Bernoulli_breast_cancer,
                     score_Multinomial_breast_cancer,
                     score_Gaussian_breast_cancer]

ans1 = round(np.max(mas_breast_cancer), 4)
print(ans1)
```

```
0.9385
```

Завдання 2

Яка максимальну якість класифікації на датасеті **digits**?

Результат округліть до чотирьох чисел після коми та завантажіть у відповідну комірку Google Forms. Використовувати правила округлення.

In [ ]:
```
mas_digits = [score_Bernoulli_digits,
              score_Multinomial_digits,
              score_Gaussian_digits]
ans2 = round(np.max(mas_digits), 4)
print(ans2)
```

```
0.8703
```

Завдання 3

Виберіть вірні твердження і запишіть їх номери через пробіл (в порядку зростання номера):

1. На дійсних ознаках найкраще спрацював наївний байесовский класифікатор з розподілом Бернуллі?
2. На дійсних ознаках найкраще спрацював наївний байесовский класифікатор з поліноміальним розподілом?
3. поліноміальний розподіл краще показав себе на вибірці з цілими невід'ємними значеннями ознак?
4. На дійсних ознаках найкраще спрацював нормальний розподіл?

Потрібно підготувати Jnotebook, в якому ви знаходили відповіді на завдання, та завантажити у Classroom.

In [ ]:
```
# 3 - score_Multinomial_digits 0.8703 найкращий у вибірці digits (з цілими невід'ємними значеннями ознак)
# 4 - score_Gaussian_breast_cancer 0.9385 найкращий у вибірці breast_cancer (з дійсними ознаками)

ans3 = ['3', '4']
print(ans3[0], ans3[1])
```

```
3 4
```