

14

나이브 베이즈 분류기 실습

[나이브베이즈 분류 실습]

우선, 어떤 상황에서 베이즈 분류를 쓰는지 알아야합니다.

- 빈도주의적 추론이 어려울 때, == 데이터 사이즈가 작을 때

00:00

14-2 나이브베이즈 분류

베이즈 정리

- 나이브 베이즈 알고리즘의 기본이 되는 개념
- 우측을 변수의 사전 확률과 사후 확률 사이의 관계를 나타내는 정리
- 사전 A, B가 있을 때, 사건 B가 일어난 것을 전제로 한 사건 A의 조건부 확률을 구하고자 한다.
- 하지만 현재 가지고 있는 정보는 사건 A가 일어난 것을 전제로 한 사건 B의 조건부 확률과 A의 확률, B의 확률 뿐이다.
- 이때, 앞에 구하고자 했던 것을 다음과 같이 구할 수 있다는 것이 베이즈 정리이다.

$$P(A_i|B) = \frac{P(A_i \cap B)}{P(B)}$$

$$= \frac{P(A_i \cap B)}{P(A_1 \cap B) + P(A_2 \cap B) + \dots + P(A_n \cap B)}$$

$$= \frac{P(B|A_i)P(A_i)}{P(B|A_1)P(A_1) + P(B|A_2)P(A_2) + \dots + P(B|A_n)P(A_n)}$$

(단 k가 1, 2, 3, ..., n)

나이브 베이즈 분류

종류

1. GaussianNB: 정규분포 나이브베이즈
- 독립변수가 연속형일 때
2. BernoulliNB: 베르누이분포 나이브베이즈
- 독립변수가 0, 1의 이진형일 때

[가우시안 나이브 베이즈]

02:11

1. GaussianNB

```

In [4]: import pandas as pd
        from sklearn.model_selection import train_test_split

        credit = pd.read_csv("../data/credit_final.csv")
        X = credit[credit.columns.difference(['credit.rating'])]
        y = credit['credit.rating']

        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, stratify=y, random_state=2021)
        X_train.head()

Out[4]:
   account.balance  age  apartment.type  bank.credits  credit.amount  credit.duration.months  credit.purpose  current.assets  dependents  employment.duration
915              1   22                2            1          2462                    18              2              3              1
704              1   67                2            2          1169                    6              3              1              1
992              1   51                3            1          7511                    18              1              2              2
633              1   30                2            1          3059                    36              2              2              1
952              2   39                2            1          1188                    21              4              2              2

In [5]: from sklearn.naive_bayes import GaussianNB
        import pandas as pd

In [6]: gnb=GaussianNB()
        gnb.fit(X_train,y_train)
        y_pred=gnb.predict(X_test)
        print("accuracy score : ", gnb.score(X_test, y_test))
  
```

[베르누이 나이브베이즈 실습]

06:00

Jupyter 14-2_나이브베이지스_분류(실습) (unsaved changes)

```
gnb2.fit(X_train,y_train)
y_pred2=gnb2.predict(X_test)
print("accuracy_score : ", gnb2.score(X_test, y_test))

accuracy_score : 0.6733333333333333
```

2. BernoulliNB naive bayes

```
In [12]: df_t = pd.read_csv('../data/titanic.csv')
X = pd.get_dummies(data=df_t[['class','sex','embark_town']],columns=['class','sex','embark_town'])
y = df_t['survived']

In [121]: from sklearn.naive_bayes import BernoulliNB
bernoulli= BernoulliNB()
bernoulli.fit(X, y)

Out[121]: MultinomialNB()

In [122]: bernoulli.class_log_prior_
Out[122]: array([-0.48424599, -0.9575399])

In [123]: np.exp(bernoulli.class_log_prior_)
Out[123]: array([0.61616162, 0.38383838])

In [124]: X
Out[124]:
```

	class_First	class_Second	class_Third	sex_female	sex_male	embark_town_Cherrybourg	embark_town_Quenstown	embark_town_Southampton
0	0	0	1	0	1	0	0	1
1	0	0	1	0	1	0	0	1
2	0	0	1	0	1	0	0	1
3	0	0	1	0	1	0	0	1
4	0	0	1	0	1	0	0	1
5	0	0	1	0	1	0	0	1
6	0	0	1	0	1	0	0	1
7	0	0	1	0	1	0	0	1
8	0	0	1	0	1	0	0	1
9	0	0	1	0	1	0	0	1
10	0	0	1	0	1	0	0	1
11	0	0	1	0	1	0	0	1
12	0	0	1	0	1	0	0	1
13	0	0	1	0	1	0	0	1
14	0	0	1	0	1	0	0	1
15	0	0	1	0	1	0	0	1
16	0	0	1	0	1	0	0	1
17	0	0	1	0	1	0	0	1
18	0	0	1	0	1	0	0	1
19	0	0	1	0	1	0	0	1
20	0	0	1	0	1	0	0	1
21	0	0	1	0	1	0	0	1
22	0	0	1	0	1	0	0	1
23	0	0	1	0	1	0	0	1
24	0	0	1	0	1	0	0	1
25	0	0	1	0	1	0	0	1
26	0	0	1	0	1	0	0	1
27	0	0	1	0	1	0	0	1
28	0	0	1	0	1	0	0	1
29	0	0	1	0	1	0	0	1
30	0	0	1	0	1	0	0	1
31	0	0	1	0	1	0	0	1
32	0	0	1	0	1	0	0	1
33	0	0	1	0	1	0	0	1
34	0	0	1	0	1	0	0	1
35	0	0	1	0	1	0	0	1
36	0	0	1	0	1	0	0	1
37	0	0	1	0	1	0	0	1
38	0	0	1	0	1	0	0	1
39	0	0	1	0	1	0	0	1
40	0	0	1	0	1	0	0	1
41	0	0	1	0	1	0	0	1
42	0	0	1	0	1	0	0	1
43	0	0	1	0	1	0	0	1
44	0	0	1	0	1	0	0	1
45	0	0	1	0	1	0	0	1
46	0	0	1	0	1	0	0	1
47	0	0	1	0	1	0	0	1
48	0	0	1	0	1	0	0	1
49	0	0	1	0	1	0	0	1
50	0	0	1	0	1	0	0	1
51	0	0	1	0	1	0	0	1
52	0	0	1	0	1	0	0	1
53	0	0	1	0	1	0	0	1
54	0	0	1	0	1	0	0	1
55	0	0	1	0	1	0	0	1
56	0	0	1	0	1	0	0	1
57	0	0	1	0	1	0	0	1
58	0	0	1	0	1	0	0	1
59	0	0	1	0	1	0	0	1
60	0	0	1	0	1	0	0	1
61	0	0	1	0	1	0	0	1
62	0	0	1	0	1	0	0	1
63	0	0	1	0	1	0	0	1
64	0	0	1	0	1	0	0	1
65	0	0	1	0	1	0	0	1
66	0	0	1	0	1	0	0	1
67	0	0	1	0	1	0	0	1
68	0	0	1	0	1	0	0	1
69	0	0	1	0	1	0	0	1
70	0	0	1	0	1	0	0	1
71	0	0	1	0	1	0	0	1
72	0	0	1	0	1	0	0	1
73	0	0	1	0	1	0	0	1
74	0	0	1	0	1	0	0	1
75	0	0	1	0	1	0	0	1
76	0	0	1	0	1	0	0	1
77	0	0	1	0	1	0	0	1
78	0	0	1	0	1	0	0	1
79	0	0	1	0	1	0	0	1
80	0	0	1	0	1	0	0	1
81	0	0	1	0	1	0	0	1
82	0	0	1	0	1	0	0	1
83	0	0	1	0	1	0	0	1
84	0	0	1	0	1	0	0	1
85	0	0	1	0	1	0	0	1
86	0	0	1	0	1	0	0	1
87	0	0	1	0	1	0	0	1
88	0	0	1	0	1	0	0	1
89	0	0	1	0	1	0	0	1
90	0	0	1	0	1	0	0	1
91	0	0	1	0	1	0	0	1
92	0	0	1	0	1	0	0	1
93	0	0	1	0	1	0	0	1
94	0	0	1	0	1	0	0	1
95	0	0	1	0	1	0	0	1
96	0	0	1	0	1	0	0	1
97	0	0	1	0	1	0	0	1
98	0	0	1	0	1	0	0	1
99	0	0	1	0	1	0	0	1

[멀티노미얼 나이브베이지스 실습]

08:12

Jupyter 14-2_나이브베이지스_분류(실습) (autosaved)

```
In [20]: # 1등석, 여성, 채르부르인 사람의 생존율 예측
X_test= [[1, 0, 0, 0, 1, 1, 0, 0]]
bernoulli.predict_proba(X_test)

Out[20]: array([[0.41528384, 0.58471616]])
```

3. Multinomial naive bayes

- 독립변수가 이산형이 아닌 범주형 변수인 경우 사용

```
In [146]: from sklearn.naive_bayes import MultinomialNB
import numpy as np

In [147]: X = np.random.randint(5, size=(6, 100))
y = np.array([1, 2, 3, 4, 5, 6])
```