

2000080110_ML Skill6

September 2, 2021

```
[20]: #bayesian classifier to predict if person donates blood or not from scratch
import sklearn
import pandas as pd
import numpy as np
data=pd.read_csv(r'E:\M&L excel\transfusion.csv')
data.dropna(inplace=True)
print(data.isna().sum())
data.info()
```

```
Recency (months)          0
Frequency (times)         0
Monetary (c.c. blood)     0
Time (months)             0
whether he/she donated blood in March 2007  0
dtype: int64
<class 'pandas.core.frame.DataFrame'>
Int64Index: 748 entries, 0 to 747
Data columns (total 5 columns):
 #   Column                                Non-Null Count  Dtype
---  -
 0   Recency (months)                    748 non-null   int64
 1   Frequency (times)                   748 non-null   int64
 2   Monetary (c.c. blood)               748 non-null   int64
 3   Time (months)                       748 non-null   int64
 4   whether he/she donated blood in March 2007  748 non-null   int64
dtypes: int64(5)
memory usage: 35.1 KB
```

```
[21]: print(data)
```

```
   Recency (months)  Frequency (times)  Monetary (c.c. blood) \
0                  2                  50                12500
1                  0                  13                3250
2                  1                  16                4000
3                  2                  20                5000
4                  1                  24                6000
..                ...                ...                ...
743                23                  2                 500
744                21                  2                 500
```

745	23	3	750
746	39	1	250
747	72	1	250

	Time (months)	whether he/she donated blood in March 2007
0	98	1
1	28	1
2	35	1
3	45	1
4	77	0
..
743	38	0
744	52	0
745	62	0
746	39	0
747	72	0

[748 rows x 5 columns]

```
[22]: from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
X=data.drop(['whether he/she donated blood in March 2007'],axis=1)
Y=data['whether he/she donated blood in March 2007']
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.2)
```

```
[23]: prior = data.groupby('whether he/she donated blood in March 2007').size().
    ↪div(len(data))
print(prior)
```

```
whether he/she donated blood in March 2007
0    0.762032
1    0.237968
dtype: float64
```

```
[24]: likelihood = {}
likelihood['Recency (months)'] = data.groupby(['whether he/she donated blood in_
    ↪March 2007', 'Recency (months)']).size().div(len(data)).div(prior)
likelihood['Frequency (times)'] = data.groupby(['whether he/she donated blood_
    ↪in March 2007', 'Frequency (times)']).size().div(len(data)).div(prior)
likelihood['Monetary (c.c. blood)'] = data.groupby(['whether he/she donated_
    ↪blood in March 2007', 'Monetary (c.c. blood)']).size().div(len(data)).
    ↪div(prior)
likelihood['Time (months)'] = data.groupby(['whether he/she donated blood in_
    ↪March 2007', 'Time (months)']).size().div(len(data)).div(prior)
print (likelihood)
```

```
{'Recency (months)': whether he/she donated blood in March 2007  Recency
(months)}
```

0	0	0.005263
	1	0.012281
	2	0.187719
	3	0.014035
	4	0.171930
	5	0.007018
	6	0.005263
	7	0.012281
	8	0.008772
	9	0.028070
	10	0.005263
	11	0.124561
	12	0.010526
	13	0.007018
	14	0.117544
	15	0.003509
	16	0.103509
	18	0.003509
	21	0.071930
	23	0.085965
	25	0.001754
	35	0.001754
	38	0.003509
	39	0.001754
	40	0.001754
	72	0.001754
	74	0.001754
1	0	0.011236
	1	0.011236
	2	0.370787
	3	0.028090
	4	0.325843
	5	0.011236
	6	0.011236
	7	0.005618
	8	0.005618
	9	0.022472
	11	0.073034
	13	0.005618
	14	0.050562
	16	0.022472
	17	0.005618
	20	0.005618
	21	0.016854
	22	0.005618
	23	0.005618
	26	0.005618

dtype: float64, 'Frequency (times)': whether he/she donated blood in March 2007

Frequency (times)

0

1	0.242105
2	0.163158
3	0.128070
4	0.085965
5	0.073684
6	0.061404
7	0.054386
8	0.031579
9	0.031579
10	0.014035
11	0.028070
12	0.019298
13	0.007018
14	0.015789
15	0.008772
16	0.015789
17	0.003509
18	0.001754
19	0.001754
22	0.001754
23	0.001754
24	0.005263
38	0.001754
44	0.001754

1

1	0.112360
2	0.106742
3	0.078652
4	0.073034
5	0.112360
6	0.095506
7	0.067416
8	0.073034
9	0.033708
10	0.033708
11	0.033708
12	0.016854
13	0.028090
14	0.022472
15	0.005618
16	0.022472
17	0.011236
19	0.005618
20	0.011236
21	0.011236
22	0.005618
26	0.005618
33	0.005618

	34	0.005618
	41	0.005618
	43	0.005618
	46	0.005618
	50	0.005618
dtype: float64, 'Monetary (c.c. blood)': whether he/she donated blood in March 2007 Monetary (c.c. blood)		
0	250	0.242105
	500	0.163158
	750	0.128070
	1000	0.085965
	1250	0.073684
	1500	0.061404
	1750	0.054386
	2000	0.031579
	2250	0.031579
	2500	0.014035
	2750	0.028070
	3000	0.019298
	3250	0.007018
	3500	0.015789
	3750	0.008772
	4000	0.015789
	4250	0.003509
	4500	0.001754
	4750	0.001754
	5500	0.001754
	5750	0.001754
	6000	0.005263
	9500	0.001754
	11000	0.001754
1	250	0.112360
	500	0.106742
	750	0.078652
	1000	0.073034
	1250	0.112360
	1500	0.095506
	1750	0.067416
	2000	0.073034
	2250	0.033708
	2500	0.033708
	2750	0.033708
	3000	0.016854
	3250	0.028090
	3500	0.022472
	3750	0.005618
	4000	0.022472
	4250	0.011236

	4750	0.005618
	5000	0.011236
	5250	0.011236
	5500	0.005618
	6500	0.005618
	8250	0.005618
	8500	0.005618
	10250	0.005618
	10750	0.005618
	11500	0.005618
	12500	0.005618

dtype: float64, 'Time (months)': whether he/she donated blood in March 2007

Time (months)

0	2	0.052632
	4	0.070175
	9	0.007018
	10	0.001754
	11	0.040351
		...
1	79	0.011236
	86	0.005618
	87	0.005618
	89	0.005618
	98	0.028090

Length: 130, dtype: float64}

```
[45]: #let new record be (1,10,1250,98)
# Probability that the person will donate
p_yes = likelihood['Recency (months)'][1][2] * likelihood['Frequency_
→(times)'][1][10] * likelihood['Monetary (c.c. blood)'][1][1250] *
→likelihood['Time (months)'][1][98] * prior[1]
# Probability that the person will NOT donate
p_no = likelihood['Recency (months)'][0][2] * likelihood['Frequency_
→(times)'][0][10] * likelihood['Monetary (c.c. blood)'][0][1250] *
→likelihood['Time (months)'][0][98] * prior[0]
print('person donates blood')
print ('Yes : ',p_yes*100000)
print ('No : ',p_no*100000)
```

```
person donates blood
Yes : 0.938714618490108
No : 0.1297677858315234
```

```
[46]: #bayesian classifier to predict class of wine from features by inbuilt sklearn.
→naive_bayes library
from sklearn import datasets
wine = datasets.load_wine()
```

```
[47]: print("Features: ", wine.feature_names)
      print("Labels: ", wine.target_names)
      print(wine.target)
```

```
Features:  ['alcohol', 'malic_acid', 'ash', 'alcalinity_of_ash', 'magnesium',
'total_phenols', 'flavanoids', 'nonflavanoid_phenols', 'proanthocyanins',
'color_intensity', 'hue', 'od280/od315_of_diluted_wines', 'proline']
Labels:  ['class_0' 'class_1' 'class_2']
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2]
```

```
[48]: from sklearn.model_selection import train_test_split
      X_train, X_test, y_train, y_test = train_test_split(wine.data, wine.target,
      ↪test_size=0.3, random_state=109)
```

```
[49]: from sklearn.naive_bayes import GaussianNB
      gnb = GaussianNB()
      gnb.fit(X_train, y_train)
      y_pred = gnb.predict(X_test)
```

```
[50]: from sklearn import metrics
      print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
```

Accuracy: 0.9074074074074074