Bayes Naif(Play tennis)

October 30, 2023

0.0.1 MST IASD 2023-2024 (Département Génie Informatique)

Module "Apprentissage automatique" (M. AIT KBIR)

Bayes Naif appliqué aux données avec attributs continus et Discrèts

[47]:
```
import pandas as pd
```

Lecture des données

[48]:
```
dataSet = pd.read_csv("playtennis.csv",delimiter=';')
#dataSet.values # objet array
```

[49]:
```
dataSet
```

[49]:

	Outlook	Temperature	Humidity	Windy	Play
0	sunny	85	85	weak	no
1	sunny	80	90	strong	no
2	overcast	83	86	weak	yes
3	rainy	70	96	weak	yes
4	rainy	68	80	weak	yes
5	rainy	65	70	strong	no
6	overcast	64	65	strong	yes
7	sunny	72	95	weak	no
8	sunny	69	70	weak	yes
9	rainy	75	80	weak	yes
10	sunny	75	70	strong	yes
11	overcast	72	90	strong	yes
12	overcast	81	75	weak	yes
13	rainy	71	91	strong	no

[50]:
```
# Probabilité à priori de Yes
pCYes = sum(dataSet['Play']=='yes')/len(dataSet)
print(pCYes)
```

```
0.6428571428571429
```

Structure des données pour mémoriser les p(xi|wj)

[51]:
```
# Outlook: Discret, Température: Continu, Humidity: Continu, Windy: Discret
typeData = ['D','C','C','D']
```

```
p_xiwj={}
for i in range(0,len(dataSet.columns)-1):
    p_xiwj[dataSet.columns[i]] = {'typeD':typeData[i]}

    if typeData[i] == 'C':
        p_xiwj[dataSet.columns[i]]['no'] = {'mean':'','var':''}
        p_xiwj[dataSet.columns[i]]['yes'] = {'mean':'','var':''}
    else :
        names = set(dataSet[dataSet.columns[i]])
        for name in names:
             p_xiwj[dataSet.columns[i]][name]={'no':'','yes':''}

# Structure de données à remplir
p_xiwj
```

[51]:
```
{'Outlook': {'typeD': 'D',
  'rainy': {'no': '', 'yes': ''},
  'overcast': {'no': '', 'yes': ''},
  'sunny': {'no': '', 'yes': ''}},
 'Temperature': {'typeD': 'C',
  'no': {'mean': '', 'var': ''},
  'yes': {'mean': '', 'var': ''}},
 'Humidity': {'typeD': 'C',
  'no': {'mean': '', 'var': ''},
  'yes': {'mean': '', 'var': ''}},
 'Windy': {'typeD': 'D',
  'weak': {'no': '', 'yes': ''},
  'strong': {'no': '', 'yes': ''}}}
```

Calcul des probabilités

[52]:
```
dataSetByClasse=dataSet.groupby('Play')
dataSetYes = dataSetByClasse.get_group('yes')
dataSetNo = dataSetByClasse.get_group('no')
dataSetYes # Exemples qui correspondent à 'Play'=='yes'
```

[52]:

	Outlook	Temperature	Humidity	Windy	Play
2	overcast	83	86	weak	yes
3	rainy	70	96	weak	yes
4	rainy	68	80	weak	yes
6	overcast	64	65	strong	yes
8	sunny	69	70	weak	yes
9	rainy	75	80	weak	yes
10	sunny	75	70	strong	yes
11	overcast	72	90	strong	yes
12	overcast	81	75	weak	yes

[53]:
```
dataSetYes.values
```

[53]:
```
array([['overcast', 83, 86, 'weak', 'yes'],
       ['rainy', 70, 96, 'weak', 'yes'],
       ['rainy', 68, 80, 'weak', 'yes'],
       ['overcast', 64, 65, 'strong', 'yes'],
       ['sunny', 69, 70, 'weak', 'yes'],
       ['rainy', 75, 80, 'weak', 'yes'],
       ['sunny', 75, 70, 'strong', 'yes'],
       ['overcast', 72, 90, 'strong', 'yes'],
       ['overcast', 81, 75, 'weak', 'yes']], dtype=object)
```

[54]:
```
import numpy as np
# Calcul des probabilités conditionnelles
for name in p_xiwj.keys():
    if p_xiwj[name]['typeD']=='C':
        # mean and std de type DataFrame

        p_xiwj[name]['yes']['mean'] = dataSetYes[name].mean()
        p_xiwj[name]['yes']['var']  = dataSetYes[name].std()
        p_xiwj[name]['no']['mean']  = dataSetNo[name].mean()
        p_xiwj[name]['no']['var']   = dataSetNo[name].std()
    else :
        valeurs = set(dataSet[name])
        for val in valeurs:
            p_xiwj[name][val]['yes'] = np.round(sum(dataSetYes[name]==val)/
 ↪len(dataSetYes),2)
            p_xiwj[name][val]['no'] = np.round(sum(dataSetNo[name]==val)/
 ↪len(dataSetNo),2)
# Structure avec Calcul des probabilités conditionnelles
# p_xiwj
```

[55]:
```
p_xiwj
```

[55]:
```
{'Outlook': {'typeD': 'D',
  'rainy': {'no': 0.4, 'yes': 0.33},
  'overcast': {'no': 0.0, 'yes': 0.44},
  'sunny': {'no': 0.6, 'yes': 0.22}},
 'Temperature': {'typeD': 'C',
  'no': {'mean': 74.6, 'var': 7.893034904268446},
  'yes': {'mean': 73.0, 'var': 6.164414002968976}},
 'Humidity': {'typeD': 'C',
  'no': {'mean': 86.2, 'var': 9.731392500562292},
  'yes': {'mean': 79.11111111111111, 'var': 10.215728613814635}},
 'Windy': {'typeD': 'D',
  'weak': {'no': 0.4, 'yes': 0.67},
  'strong': {'no': 0.6, 'yes': 0.33}}}
```

[56]:
```
from scipy.stats import norm

def classifyNB(vec2Classify, pCond, pC1):
    p1 = np.log(pC1)
    p0 = np.log(1-pC1)
    for attrib,val in zip(dataSet.columns[0:-1],vec2Classify):
        if pCond[attrib]['typeD']== 'D':
            p1 = p1+np.log(pCond[attrib][val]['yes'])
            p0 = p0+np.log(pCond[attrib][val]['no'])
        else:
            p1 = p1 + np.log(norm.pdf(val, pCond[attrib]['yes']['mean'],␣
 ↪pCond[attrib]['yes']['var']) )
            p0 = p0 + np.log(norm.pdf(val, pCond[attrib]['no']['mean'],␣
 ↪pCond[attrib]['no']['var']) )

    #print('yes: ',np.round(p1,4),'no: ',np.round(p0,4))

    if p1 > p0:
        return 1
    else:
        return 0
```

Vérification par rapport aux exemples d'apprentissage

[57]:
```
i=0
rst = []
for vect in dataSet.values:
    rst.append(classifyNB(vect[:-1],p_xiwj,pCYes))
    # print(i,' : ',rst[i],' - ',vect[-1])
    i = i+1

print(rst) # Résultats de la classifications
print(dataSet['Play'].values) # Classes d'appartenance des exemples
```

```
[0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0]
['no' 'no' 'yes' 'yes' 'yes' 'no' 'yes' 'no' 'yes' 'yes' 'yes' 'yes' 'yes'
 'no']
```

```
C:\Users\MAK\AppData\Local\Temp\ipykernel_21404\1514126046.py:9: RuntimeWarning:
divide by zero encountered in log
  p0 = p0+np.log(pCond[attrib][val]['no'])
```

Taux de la classification correcte

[58]:
```
from sklearn import metrics
metrics.accuracy_score(rst, dataSet['Play']=='yes')
```

[58]:
```
0.9285714285714286
```

[59]:
```
metrics.confusion_matrix(rst, dataSet['Play']=='yes')
```

[59]:
```
array([[4, 0],
       [1, 9]], dtype=int64)
```

Test : deux nouveaux exemples

[60]:
```
ex = ['sunny',62,90,'strong']
rst= classifyNB(ex,p_xiwj,pCYes)
print(ex,' --> ',rst)

ex = ['rainy',80,82,'weak']
rst= classifyNB(ex,p_xiwj,pCYes)
print(ex,' --> ',rst)
```

```
['sunny', 62, 90, 'strong']  -->  0
['rainy', 80, 82, 'weak']  -->  1
```