Herwin Alejandro Rosado Molina
Código: 2019188012

Pruebas Unitarias Ejercicio de Datos Bayesianos

#	Líneas de Código
1	import pandas as pd
2	Datos = pd.read_csv('Datos_Balegria.csv')
3	Datos.head(7)
4	features_train = Datos.iloc[0:7, 0:4]
5	target_train = Datos.iloc[0:7, 4]
6	from sklearn import preprocessing
7	le = preprocessing.LabelEncoder()
8	f0 = le.fit_transform(features_train.iloc[0:7, 0])
9	f1 = le.fit_transform(features_train.iloc[0:7, 1])
10	f2 = le.fit_transform(features_train.iloc[0:7, 2])
11	f3 = le.fit_transform(features_train.iloc[0:7, 3])
12	label = le.fit_transform(target_train)
13	features = list(zip(f0,f1,f2,f3))
14	print(features)
15	print(label)
16	from sklearn.naive_bayes import GaussianNB
17	model1 = GaussianNB()
18	model1.fit(features, label)
19	Predicted = model1.predict([[0,0,0,2]])
20	print(Predicted)
21	le.inverse_transform([2])
	Pruebas Unitarias
23	from sklearn.naive_bayes import GaussianNB
24	def bayesiano(num1, num2, num3, num4):
25	return model1.predict([[num1,num2,num3,num4]])
26	import unittest
27	class PruebasFunciones(unittest.TestCase):
28	def test_evaluar(self):
29	self.assertEqual(2,bayesiano(0,0,0,2))
30	unittest.main(argv =['ignored', '-v'], exit = False)

Herwin Alejandro Rosado Molina
Código: 2019188012

In [33]:
```
from sklearn.naive_bayes import GaussianNB
model1 = GaussianNB()
model1.fit(features, label)
#Predicted = model1.predict([[1,1,0,2]])
#Predicted = model1.predict([[0,1,2,1]])
#Predicted = model1.predict([[0,0,2,1]])
Predicted = model1.predict([[0,0,0,2]])
print(Predicted)
#El sistema debe implementar el algoritmo de Naive Bayes para mostrar las
#predicciones de campaña de los clientes con base a un perfil especifico equivalente al resto de los datos de la tabla
```

```
[2]
```

In [35]:
```
le.inverse_transform([2])
```

Out[35]:
```
array(['Viajes'], dtype=object)
```

In [17]:
```
from sklearn.naive_bayes import GaussianNB
def bayesiano(num1, num2, num3, num4):
    #model1 = GaussianNB()
    #model1.fit(features, label)
    return model1.predict([[num1,num2,num3,num4]])

import unittest
class PruebasFunciones(unittest.TestCase):
    def test_evaluar(self):
        self.assertEqual(2,bayesiano(0,0,0,2))
unittest.main(argv=['ignored', '-v'], exit=False)
```

```
test_evaluar (__main__.PruebasFunciones) ... ok

----------------------------------------------------------------------
Ran 1 test in 0.001s

OK
```

Out[17]:
```
<unittest.main.TestProgram at 0x26053162b38>
```