Atividade NB

December 16, 2020

1 Parte 0

The probability of correctly diagnosing a rare disease is 0.70. When diagnosed correctly, the probability of cure is 0.90. If not diagnosed correctly, the probability of cure is 0.40. If the patient with the disease is cured, what is the probability that he has been diagnosed correctly?

[5]:
```
# Seu codigo aqui
A = 0.7   # Diagnóstico OK
B_A = 0.9 # Curado, dado que diagnóstico OK
A_B = 0   # Diagnóstico OK, dado que curado
C_D = 0.4 # Curado, dado que diagnóstico NOK
C = 0.3   # Diagnóstico NOK
D = B = 0 # Curado

D = (C * B_A) / C_D
B = D
A_B = (A * B_A)/ B
A_B
```

[5]:
```
0.9333333333333332
```

2 Parte 1

Crie um classificador Naive Bayes para o MNIST dataset

[7]:
```
from sklearn.datasets import fetch_openml
mnist = fetch_openml('mnist_784', version=1)
mnist.keys()
```

[7]:
```
dict_keys(['data', 'target', 'frame', 'categories', 'feature_names',
'target_names', 'DESCR', 'details', 'url'])
```

[8]:
```
# Seu codigo aqui
X, y = mnist["data"], mnist["target"]
```

[9]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33)
```

[10]:
```
from sklearn.naive_bayes import GaussianNB
```

[14]:
```
nb = GaussianNB()
nb.fit(X_train, y_train).predict(X_test)
```

[14]:
```
array(['9', '9', '0', …, '7', '8', '0'], dtype='<U1')
```

[16]:
```
nb.score(X_test, y_test)
```

[16]:
```
0.5564935064935065
```

[18]:
```
from sklearn.naive_bayes import MultinomialNB
```

[19]:
```
mnb = MultinomialNB()
mnb.fit(X_train, y_train)
```

[19]:
```
MultinomialNB()
```

[20]:
```
mnb.score(X_test, y_test)
```

[20]:
```
0.8293506493506494
```

[21]:
```
from sklearn import metrics
```

[22]:
```
y_pred = mnb.predict(X_test)
```

[23]:
```
acc = metrics.accuracy_score(y_test, y_pred)
acc
```

[23]:
```
0.8293506493506494
```

3 Parte 2 - Desafio

Implemente o classificador para o MNIST utilizando apenas a biblioteca numpy e auxiliares. Não utilize o `scikit-learn`.

[1]:
```
# Seu codigo aqui
```