NaiveBayes

January 31, 2018

1 Naïve Bayes Classifier

http://scikit-learn.org/stable/modules/naive_bayes.html
  https://www.countbayesie.com/blog/2015/2/18/bayes-theorem-with-lego

```
In [2]: import pandas as pd
        import numpy as np
```

1.1 Importar los datos en Pandas

```
In [3]: dataframe = pd.read_excel('https://benlarsonsite.files.wordpress.com/2016/05/logi21.xls
```

```
In [4]: dataframe.head()
```

```
Out[4]:    Score  ExtraCir  Accepted
        0    982         0         0
        1   1304         1         1
        2   1256         0         1
        3   1562         1         1
        4    703         0         0
```

Columnas:

- Score – Puntuación del examen
- ExtraCir – El estudiante estaba involucrado en actividades extra-curriculares
- Accepted – El estudiante fue aceptado

1.2 Split the data

```
In [5]: y = dataframe.pop('Accepted')
        X = dataframe

        y.head()
```

```
Out[5]: 0    0
        1    1
        2    1
        3    1
        4    0
        Name: Accepted, dtype: int64
```

```
In [6]: X.head()

Out[6]:     Score  ExtraCir
        0    982         0
        1   1304         1
        2   1256         0
        3   1562         1
        4    703         0
```

Scikit-learn 1 - Importar algoritmo
    2 - Instanciar clasificador
    3 - Entrenar
    4 - Predecir

```
In [7]: # 1 - Importar algoritmo
        from sklearn.naive_bayes import MultinomialNB

In [8]: # 2 - Instanciar clasificador
        classifier = MultinomialNB()

In [9]: # 3 - Entrenar
        classifier.fit(X,y)

Out[9]: MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

1.3   Predict and profit

```
In [14]: #--score: 1200, ExtraCir = 1
         sample_1 =  np.array([1200,1]).reshape((1, -1))

         print(classifier.predict(sample_1))
         print(classifier.predict_proba(sample_1))

[1]
[[ 0.31343672  0.68656328]]


In [20]: #--score: 1000, ExtraCir =  0
         sample_2 =  np.array([2,1]).reshape((1, -1))

         print(classifier.predict(sample_2))
         print(classifier.predict_proba(sample_2))

[1]
[[ 0.19327899  0.80672101]]
```