Naive Bayes
Detección cáncer de mama

En esta entrada vamos a explicar cómo poner en práctica el algoritmo de Naive Bayes. Con anterioridad ya explicamos la parte teórica e inclusive lo necesario para implementar este algoritmo utilizando la librería de Python, scikit learn, ahora ha llegado el momento de ver la parte práctica.

Para esta entrada continuaremos desarrollando el proyecto que hemos venido trabajando a lo largo de los algoritmos de clasificación que es el de determinar si un paciente tiene o no cáncer de seno.

Este conjunto de datos se encuentra disponible dentro de la librería scikit learn y como es el mismo problema que hemos venido desarrollando con anterioridad acá nos enfocaremos en construir el modelo, mas no en entender los datos o en realizar el preprocesamiento de los mismos, esto ya lo hicimos con anterioridad. Si quieres ver explicado esta parte te recomiendo que busques la información práctica del algoritmo de Regresión Logística en donde fue explicado esta parte.

En este momento ya tenemos separados los datos de “X” y “y”. Ahora es el momento de separar los datos de entrenamiento y prueba, para ello utilizamos la instrucción de train_test_split, la cual nos facilita bastante este procedimiento.

```
from sklearn.model_selection import train_test_split

#Separo los datos de "train" en entrenamiento y prueba para probar los algoritmos
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

Seguidamente definimos el algoritmo, en este caso le indicamos a nuestro programa, de sklearn.naive_bayes vamos a importar GaussianNB, con esto ya podemos implementar este algoritmo dentro de nuestro programa.

```
#Naive Bayes
from sklearn.naive_bayes import GaussianNB

algoritmo = GaussianNB()
```

Por tal motivo entrenamos el modelo junto a los datos de entrenamiento.

```
#Entreno el modelo
algoritmo.fit(X_train, y_train)
```

Naive Bayes
Detección cáncer de mama

Seguidamente realizamos una predicción junto a los datos de pruebas que separamos previamente.

```
#Realizo una predicción
y_pred = algoritmo.predict(X_test)
```

Verifiquemos como es el modelo utilizando las métricas de los problemas de clasificación, para ello vamos a comenzar obteniendo la matriz de confusión. Para esto importamos del modulo skk¡learn.metrics, confusion_matrix, y aplicamos esta instrucción junto a los datos de prueba y los obtenidos en la predicción realizada previamente.

```
#Verifico la matriz de Confusión
from sklearn.metrics import confusion_matrix
matriz = confusion_matrix(y_test, y_pred)
print('Matriz de Confusión:')
print(matriz)
```

```
Matriz de Confusión:
[[38  1]
 [ 0 75]]
```

El resultado acá, es que tenemos 113 datos predichos correctamente y 1 dato erróneo obtenido luego de realizar la predicción.

Viendo este resultado podemos concluir que el modelo predijo la gran mayoría de los datos por lo que es un buen modelo que podemos utilizar.

Ahora veamos la precisión del mismo, para esto importamos precision_score del modulo sklearn.metrics y lo implementamos de igual forma junto a los datos de entrenamiento y los predichos.

```
#Calculo la precisión del modelo
from sklearn.metrics import precision_score
precision = precision_score(y_test, y_pred)
print('Precisión del modelo:')
print(precision)
```

```
Precisión del modelo:
0.9868421052631579
```

Naive Bayes
Detección cáncer de mama

El resultado obtenido acá es de 0,986. Por lo que consideramos que el modelo cumple con su función.

A continuación se encuentra el código completo:

```
"""
Naive Bayes
"""
########## LIBRERÍAS A UTILIZAR ##########
#Se importan la librerias a utilizar
from sklearn import datasets
########## PREPARAR LA DATA ##########
#Importamos los datos de la misma librería de scikit-learn
dataset = datasets.load_breast_cancer()
print(dataset)
########## ENTENDIMIENTO DE LA DATA ##########
#Verifico la información contenida en el dataset
print('Información en el dataset:')
print(dataset.keys())
print()
#Verifico las características del dataset
print('Características del dataset:')
print(dataset.DESCR)
#Seleccionamos todas las columnas
X = dataset.data
#Defino los datos correspondientes a las etiquetas
y = dataset.target
########## IMPLEMENTACIÓN DE NAIVE BAYES ##########
from sklearn.model_selection import train_test_split
#Separo los datos de "train" en entrenamiento y prueba para probar los algoritmos
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
#Defino el algoritmo a utilizar
#Naive Bayes
from sklearn.naive_bayes import GaussianNB
algoritmo = GaussianNB()
#Entreno el modelo
algoritmo.fit(X_train, y_train)
#Realizo una predicción
y_pred = algoritmo.predict(X_test)
#Verifico la matriz de Confusión
from sklearn.metrics import confusion_matrix
matriz = confusion_matrix(y_test, y_pred)
print('Matriz de Confusión:')
print(matriz)
#Calculo la precisión del modelo
```

Naive Bayes
Detección cáncer de mama


```
from sklearn.metrics import precision_score
precision = precision_score(y_test, y_pred)
print('Precisión del modelo:')
print(precision)
```