

Support vector machine

Cancer de mama



En esta entrada vamos a explicar como poner en práctica el algoritmo de Máquinas de Vectores de Soporte Clasificación. 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.

Entonces comencemos con el desarrollo del modelo.

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.svm` vamos a importar `SVC`, con esto ya podemos implementar este algoritmo dentro de nuestro programa.

```
#Defino el algoritmo a utilizar
from sklearn.svm import SVC
algoritmo = SVC(kernel = 'linear')
```

Por lo tanto, definimos el algoritmo, acá vamos a configurar un solo parámetro y será el correspondiente al kernel, el cual lo definiremos como "linear". Recuerda que el kernel tiene varias funciones que puedes configurar según lo que te parezca más adecuado.

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

Ahora si ha llegado el momento de entrenar el modelo utilizando los datos de entrenamiento separados previamente.

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

Support vector machine

Cancer de mama



Con el modelo entrenado se realiza una predicción junto con los datos de prueba.

En este momento puedes comparar los datos obtenidos luego de realizar la predicción con los datos reales para ver que tal ha sido el modelo que hemos construido, pero esta vez vamos a aplicar las métricas con las que cuenta los algoritmos de Machine Learning.

```
#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:
[[45  4]
 [ 2 63]]
```

Lo primero que vamos a verificar es la matriz de confusión, para ello lo importamos desde el modulo de sklearn.metrics. El resultado obtenido es el que observamos.

Viendo este resultado ya podemos intuir que el modelo ha sido correcto, pero veamos la precisión del modelo.

```
#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:
0.9402985074626866
```

Para esto importamos esta función y la aplicamos junto con los datos de “y” de prueba y los obtenidos en la predicción. El resultado obtenido es xx. Por lo que consideramos que el modelo cumple con su función.

Recuerda que cada una de estas métricas fueron explicadas anteriormente, por lo que te recomiendo que revises esta información, en caso que no entiendas lo que aplicamos acá.

Support vector machine

Cancer de mama



Este modelo puedes modificarlo por tu cuenta para que veas si puede ser mejorado o en todo caso ver como puede cambiar los resultados si se cambia los parámetros del algoritmo, ya queda de tu parte realizar estas modificaciones.

A continuación encuentras el programa completo:

```
"""
Máquinas Vectores de Soporte Clasificación
"""

##### LIBRERÍAS A UTILIZAR #####
#Se importan la librerías 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 MAQUINAS VECTORES DE SOPORTE #####
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
from sklearn.svm import SVC
algoritmo = SVC(kernel = 'linear')
#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
```

Support vector machine

Cancer de mama



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