SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

YUCATAN

INSTITUTO TECNOLÓGICO SUPERIOR DE VALLADOLID

por el saber universal y la grandeza del espíritu

Organismo Público Descentralizado del Gobierno del Estado de Yucatán
Clave del C.T. 31EIT0002S

Nombre del alumno:

Adrián Guadalupe Balam Jiménez

Miguel Angel Tun Tuz

Rogelio Canche Canche

Nombre del profesor o tutor:

M.M. JORGE MANUEL POOL CEN

Grado y Grupo:

8vo semestre grupo A

Materia:

INTELIGENCIA ARTIFICIAL

Actividad:

Software de Machine Learning

Fecha de última modificación:

05/06/2020

Instrucción:

Desarrollar con python y scikit learning un código que genera una toma de desición con un algoritmo de clasificación, por ejemplo, el problema del Titanic que determina si sobrevive o no, puede apoyarse en los siguientes enlaces: https://www.aprendemachinelearning.com/arbol-de-decision-en-python-clasificacion-y-prediccion/ El entregable es un programa documentado en un informe con un glosario de las funciones.

Conceptos:

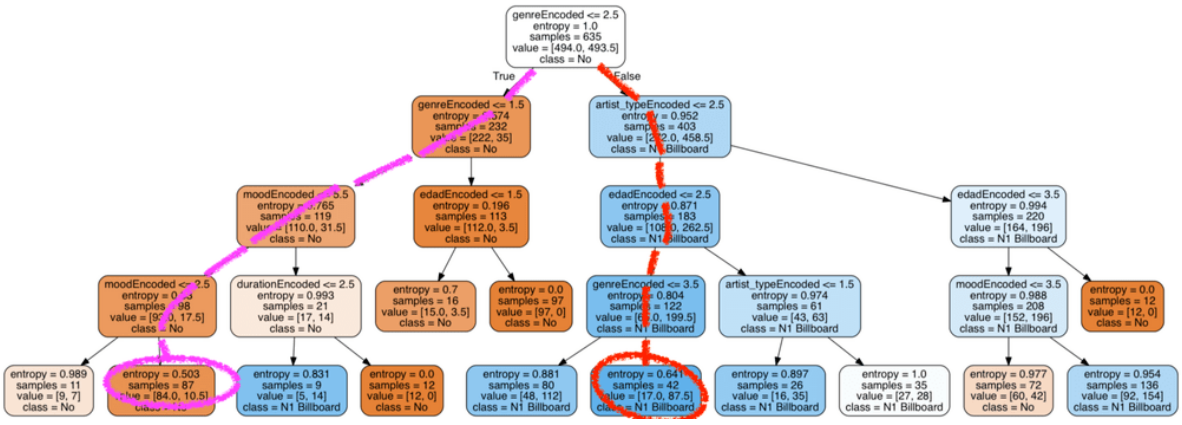
Analizaremos a los cantantes y bandas que lograron un puesto número uno en las listas de Billboard Hot 100 e intentaremos predecir quién será el próximo Éxito a fuerza de Inteligencia Artificial.

Primero hay que saber que los árboles de decisión son representaciones gráficas de posibles soluciones a una decisión basadas en ciertas condiciones, es uno de los algoritmos de aprendizaje supervisado más utilizados en machine learning y pueden realizar tareas de clasificación o regresión.

Los árboles de decisión tienen un primer nodo llamado raíz (root) y luego se descomponen el resto de atributos de entrada en dos ramas o mas planteando una condición que puede ser cierta o falsa.

Para realizar este ejercicio, utilizaremos una Jupyter notebook con código python y la librería Scikit learn.

Árbol de decisiones



Glosario de funciones:

x, y, hue : nombres de variables endata

Entradas para trazar datos de formato largo. Ver ejemplos de interpretación.

datos : DataFrame

Conjunto de datos de forma larga (ordenada) para el trazado. Cada columna debe corresponder a una variable, y cada fila debe corresponder a una observación.

fila, col : nombres de variables en data, opcional

Variables categóricas que determinarán las facetas de la cuadrícula.

col_wrap : int, opcional

"Ajuste" la variable de columna en este ancho, de modo que las facetas de la columna abarquen varias filas. Incompatible con una rowfaceta.

estimador: invocable que asigna vectores -> escalar, opcional

Función estadística para estimar dentro de cada bin categórico.

ci : float o None, opcional

Tamaño de los intervalos de confianza para dibujar alrededor de los valores estimados. Si Noneno se realizará el arranque, no se dibujarán barras de error**.**

pima = pd.read_csv("artistas.csv", header=None, names=col_names):

Esta función permite cargar el dataset o base de datos del cual hará las predicciones.

pima = pima.drop(0, axis=0)

Esta función realiza la eliminación de la primera fila con el fin de evitar la compatibilidad de tipo de datos.

print(pima.head())

Función que permite imprimir la cabeza del dataset o base de datos.

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,

random_state=1) #

Esta función divide los datos de entrenamiento y prueba dando como prioridad la cantidad del 30% en el conjunto de pruebas.

clf = clf.fit(X_train,y_train)

Función para entrenar el árbol de decisiones.

print("Porcentaje de exactitud:",metrics.accuracy_score(y_test, y_pred))

Función que mide el porcentaje de exactitud del árbol de decisión.

decision_tree.fit(X_train, y_train)

Función para indicar el dataset o base de datos de entrenamiento al árbol de decisiones.

Y_pred = logreg.predict(X_test)

Función para predecir con regresión logística con base a datos de prueba.

Y_pred = svc.predict(X_test)

Función para predecir con soporte de vectores con base a datos de prueba.

Y_pred = knn.predict(X_test)

Función para predecir con vecinos más cercanos con base a datos de prueba.

pima.drop([artistas], axis = 1, inplace=True)

Función para eliminar la columna de artistas

out_logreg = pd.DataFrame({ 'Edad' : ids, 'Salida': prediccion_logreg })

Funcion de regresion logistica.

out_svc = pd.DataFrame({ 'Edad' : ids, 'Salida': prediccion_svc })

Función de soporte de vectores.

out_knn = pd.DataFrame({ datos : ids, 'Salida': prediccion_knn })

Función de vecinos más cercanos.

export_graphviz(decision_tree, out_file=dot_data, filled=True,

rounded=True, special_characters=True)

Función para darle características al árbol de decisiones.

Image(graph.create_png())

Función para generar el árbol de decisiones.