Reparación de dispositivos celulares

Los datos del modelo corresponden a:

* Teléfono
* Marca
* Estado de teléfono

¿Qué tipo de modelo aplicarías: regresión lineal, regresión logística o árbol de decisión? Y ¿por qué consideras que ese modelo es adecuado?

* Modelo de regresión logística, el cual genera una respuesta positiva o negativa.

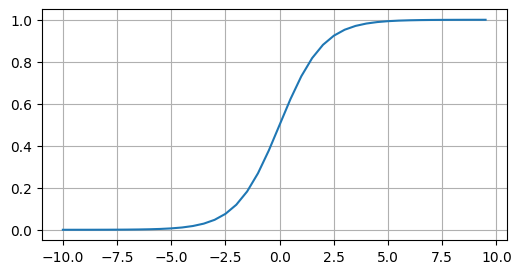
¿Por qué es diferente a la programación regular?

* El modelo regular, se basa en variables o instrucciones de forma manual que siguen unos pasos para realizar un proceso. Mientras que, machine learning, trabaja con variables que generan una predicción mediante el análisis de datos basado en instrucciones.

¿Por qué crees que los modelos como los vistos en la lección no logran llegar a un 100% de respuestas correctas?

* Debido al margen de error, este está atado a la generación de predicciones, por tal motivo, el 100% de las respuestas no son correctas, además que tiene ciertas limitantes de procesamiento y análisis de datos. Igualmente, varía según la máquina utilizada.

2. El código Python que se muestra a continuación dibuja una función. ¿Qué tipo de función es?

Es una función de tipo logística que se genera en un rango que varía entre 0 y 1 lo que resulta en

kikourios@gmail.com

El código Python que se muestra a continuación es un ejemplo de árbol de decisión. ¿El ejemplo realiza Clasificación?, ¿Cuál es su predicción? ¿Puedes realizar su gráfico de Árbol de decisión?

Implementa la clasificacion, toma unas pruebas con respecto a la prediccion.

from sklearn.datasets import load\_iris

from sklearn.model\_selection import train\_test\_split

from sklearn.tree import DecisionTreeClassifier

x,y=load\_iris(return\_X\_y=True)

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.5, random\_state=0)

clf=DecisionTreeClassifier()

clf.fit(x\_train,y\_train)

y\_pred=clf.predict(x\_test)

N= y\_test.shape[0]

C=(y\_test==y\_pred).sum()

print("Total points: %d Correctly labeled points: %d" %(N,C))

from sklearn.tree import plot\_tree #Importa las funciones del arbol de decisicion

import matplotlib.pyplot as plt #Importa la libreria para graficar

from sklearn.datasets import load\_iris #Utiliza conjunto de datos de muestra y load iris toma muestra de iris

#como ejemplo para imprimir el grafico como ejemplo

iris = load\_iris() # Carga el conjunto de datos del iris dentro del arbol de decision

plt.figure(figsize=(15,8)) #Crea la figura con un tamaño de 15x8

plot\_tree(clf, filled=True, # Rellena los nodos para una mejor visualizacion

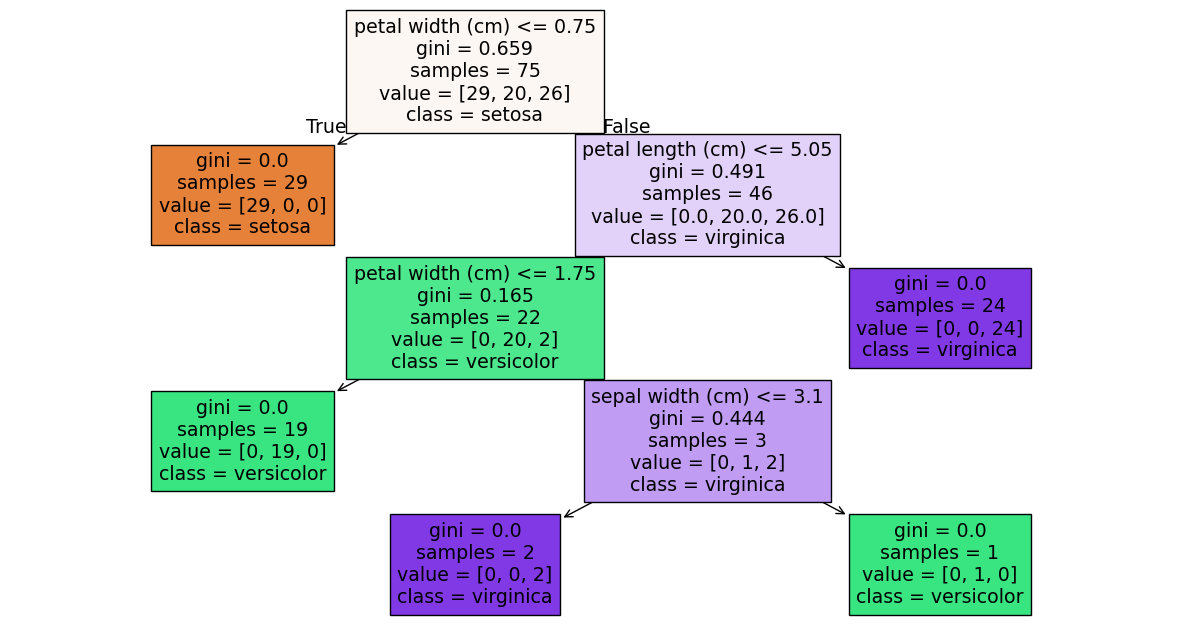
feature\_names=iris.feature\_names, # se usa el atributo nombre feature\_names

#Toma un nombre o caractaristica

class\_names=iris.target\_names) # se usa el atributo target\_names con el cual nombramos el objeto

#Imprimimos el arbol de decision

plt.show()



El arbol de decision describe 3 tipos de flores segun tamaño de petalo y tallo segun estos datos determina el tipo de lirio setosa, virginica o versicolor.