Tarea #: 3
Tema: Clasificación de datos utilizando texto
Fecha entrega: 11:59 pm Junio 05 de 2023
Objetivo: Utilizar modelos de regresión logística y árboles para crear un modelo de clasificación utilizando datos reales .
Entrega: Crear una rama utilizando el mismo repositorio de la tarea 1 y 2, crear otra carpeta llamada tarea 3, solucionar el problema y crear un pull request sobre la master donde me debe poner como reviewer (entregas diferentes tienen una reducción de 0.5 puntos)..

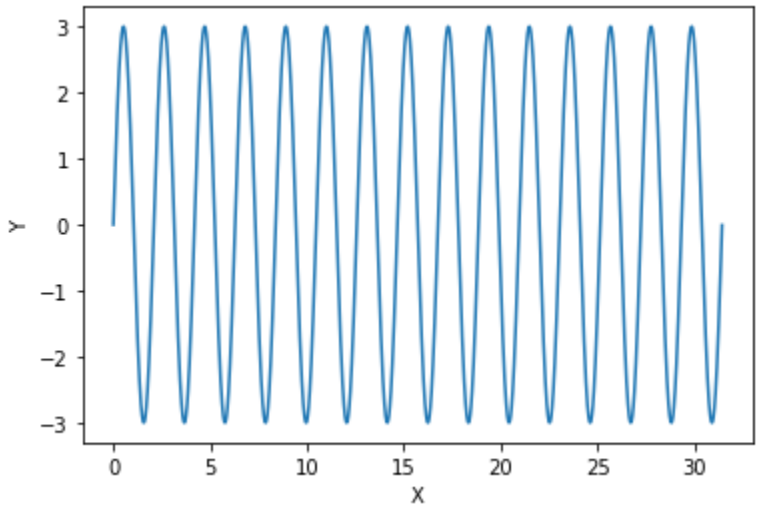
1. Clasificación y la entropía como función objetivo de la clasificación (15%).
    a. Utilizar la siguiente tabla para crear un árbol de clasificación de la variable Y a mano con 2 niveles.

ingresos	estrato	credito
1	2	0
5	2	0
1	4	1
6	0	0
8	5	1
4	0	0
3	5	1
6	2	0
3	5	1
3	2	0
9	2	1
1	2	0

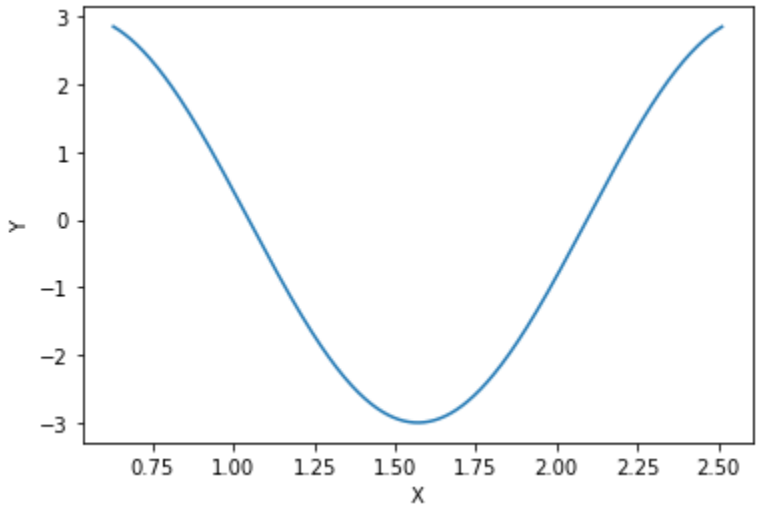
3	0	0
1	3	0
7	2	0
9	1	1
8	3	1
7	2	0
6	3	0
9	2	1
0	3	0
2	4	1
3	4	1
1	3	0
0	5	1
5	4	1
9	5	1
0	5	1
0	4	1
3	3	0

b. Entrene un árbol utilizando sklearn.tree.DecisionTreeClassifier y compare los resultados. ¿Qué puede concluir son iguales?

2. Gradiente descendiente (15%), estaba en un parque de la ciudad y escuche un ruido constante. Entonces tomé mi celular y realice la grabación del sonido, observando la siguiente gráfica.



Cualquier sonido senoidales tiene la forma y_hat =a * sen(bx + c) + d, nuestro error debería ser LSE (Least Square Error) entonces nuestra función de error se puede escribir como e(x) = (y - y_hat)^2. Ahora con la función de error tenemos que derivar a,b,c y d=0 (Calcula el gradiente que son las derivadas parciales con respecto a a,b,c) y aplicar el algoritmo de gradiente descendiente (Considerando que el gradiente funciona solo con funciones convexas vamos a tomar la parte de la señal convexa, para encontrar los parámetros).



Importar el archivo senal.csv en la carpeta tarea3. Y aplicar el algoritmo de gradiente descendente que consiste en:

- Calcular los gradientes que son las derivadas parciales e iniciar los valores en random
- Iterar por muchos epochs (Muchas iteraciones a los datos)
- Por cada epoch tomar un numero de muestras aleatorias, calcular el gradiente y ajustar los valores anteriores utilizando un parámetro q es el learning rate.
- Termina cuando el numero de epochs es alcanzado,

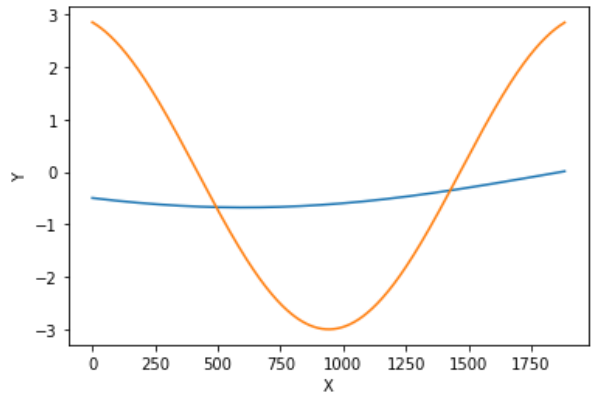
```
def y_predict(a,b,c,d,x):
    return a * math.sin(b*x + c) + d

lr = 0.01
n = len(x)
batch = 500
epochs =
rsl = []
a = random.random()
b = random.random()
c = random.random()
d = 0
for i in range(epochs):
    a_gradiente = 0
    b_gradiente = 0
    c_gradiente = 0
    d_gradiente = 0
    e = 0
    for m in range(batch):
        ix = int(random.uniform(0,n))
        e +=  (y[ix] - y_predict(a,b,c,d,x[ix]))* (y[ix] - y_predict(a,b,c,d,x[ix]))
        a_gradiente +=
        b_gradiente +=
        c_gradiente +=

    a = a - lr * a_gradiente/batch
    b = b - lr * b_gradiente/batch
    c = c - lr * c_gradiente/batch

    e = e/batch
    rsl.append([a,b,c,d,e])
    print(f"error:{e} period:{b} amplitude:{a} constant: {c} ")
```

Utiliza los parámetros del vector rst y crea la siguiente tabla con los datos del dataset y la función predict aprendida hasta ese epoch..

Epoch=10 (Ejemplo blue es lo aprendido)	Epoch=400
	

Epoch=1000	Epoch=2000

Por último utiliza los parámetros aprendidos, y crea la señal por un periodo más largo de tiempo long_s = [y_predict(a,b,c,d,xi) for xi in np.arange(x_min , 20*math.pi, 0.001).astype(np.float32)] , dibuja la señal, y escuchala utilizando la libraria (import sounddevice as sd y el método sd.play(long_s))

3. Utilizando el servicio del dataset de datos abiertos https://www.datos.gov.co/Ciencia-Tecnolog-a-e-Innovaci-n/Contenidos-Youtube/v98q-23dp , vamos a hacer un clasificador del tipo de contenido basados en el titulo:
    a. Crear un código python para invocar el servicio https://www.datos.gov.co/resource/v98q-23dp.json?$query=SELECT%0A%20%20%60titulo%60%2C%0A%20%20%60descripcion%60%2C%0A%20%20%60url_lista_de_reproduccion%60%2C%0A%20%20%60plataforma%60 y leer los datos en un dataframe (Puede utilizar la libreria request), la variable objetivo es predecir la columna categoria (Se debe transformar en Entretenimiento, Deportes, Película y Animación, Educación y Otros) utilizando el texto del titulo. Para eso es necesario hacer las siguientes tareas :
        i. Utilizar todos los datos, y explorar el numero total de palabras únicas en todos los títulos de train y el numero total de repeticiones (Crear un diccionario para saber si la palabra ya fue observada antes e intentar remover tildes y poner el texto en minúsculas), ¿Cuántas palabras hay en el data set?, generar la siguiente matriz.

	Palabra 1	Palabra 2	Palabra 3	…	Palabra n
Titulo 1	0 (#veces palabra 1 en titulo1)				
Titulo 2		2 (#veces palabra 2 en titulo2)			
Titulo m					

        ii. Crear un gráfico de barras con las 10 palabras más comunes, ¿son útiles?, normalmente las palabras más comunes son llamadas stop words. Y corresponden a los artículos o preposiciones. Generar nuevamente los gráficos. Y una matriz de correlación utilizando dummy

variables para las variables objetivo "categoria". También generar la matrix nueva eliminando los stopwords y dividimos por el numero total de palabras en cada título. Esta matriz se llama TF (Term Frequency)[1]

	Palabra 1	Palabra 2	Palabra 3	…	Palabra n
Titulo 1	0 (#veces palabra 1 en titulo1)/ Total de palabras en el Titulo 1				
Titulo 2		2 (#veces palabra 2 en titulo2)/# Total de palabras en el Titulo 2			
Titulo m					

iii. Ahora vamos a crear un vector contando el numero de titulos que tiene cada palabra. Este vector tiene "n" elementos y el valor es el número de títulos que tiene la palabra. Por último vamos a transformar el numero calculando el log(Total de documentos/ (Numero de documentos con la Palabra i + 1)). El +1 es para q no quede el numero indeterminado cuando algún valor es "0", el vector es llamado IDF(Inverse Document Frequency) [2]

	Palabra 1	Palabra 2	Palabra 3	…	Palabra n
	log((m+1) / (# titulos con la Palabra 1 + 1))				

iv. Ahora vamos a multiplicar el vector TF (La matriz) * IDF (Vector transpuesto), El resultado es una matriz de m títulos por n

[1] https://www.learndatasci.com/glossary/tf-idf-term-frequency-inverse-document-frequency/
[2] https://www.learndatasci.com/glossary/tf-idf-term-frequency-inverse-document-frequency/

palabras. Y dividir el dataset en test y train. ***En producción se debería dividir los datos antes de calcular el TF-IDF , y para calcular la matriz TFIDF en testing es necesario calcular la frecuencia de las palabras en testing utilizando el orden y las palabras en training y utilizar el IDF de training, pero no es necesario para esta tarea***.

v. Utilizando la matriz vamos a entrenar 3 modelos, una regresión logística, un random forest y un GBM. Vamos a crear una matriz de confusión y vamos a comparar los 3 modelos. ¿Cuál es el mejor modelo?, incluir métricas como accuracy, precision y recall para cada modelo.