COMPLEJIDAD DE CODIFICACIÓN:

PRÁCTICA VOLUNTARIA

Autor: Adrián Pérez Peinador

Introducción/motivación

La principal motivación de esta práctica es afianzar los conceptos de índice de Gini y diversidad, para terminar comprender en profundidad los términos que se usan al hablar de complejidad de una codificación. En esta práctica se nos pide programar un algoritmo que calcule ambos valores para una muestra de una variable aleatoria dada. Además se nos pide usarlo para dos casos. Primero se propone un ejemplo sencillo para comprobar la corrección del algoritmo implementado y después se nos pide aplicarlo a la variable aleatoria $S_{English}$ de la segunda práctica.

Material usado

En el código se hace uso de varias librerías de Python. En concreto se usaron librerías genéricas como os (para abrir las muestras de cada alfabeto y poder tratarlas), numpy (para los arrays de numpy) o matplotlib (a la hora de graficar las distribuciones y funciones de probabilidad). Por otro lado se usó la librería collections para contar los elementos de la muestra. Asimismo, se usaron las muestras de texto proporcionadas por el profesor.

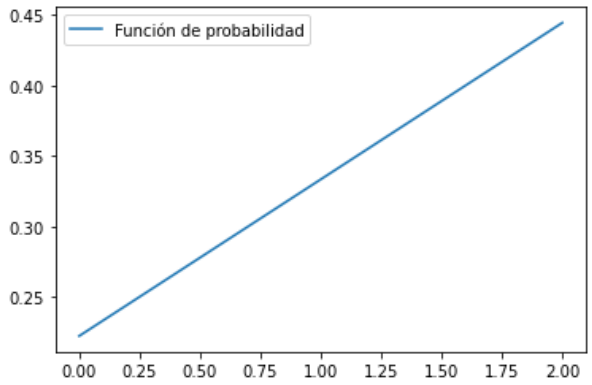
Resultados

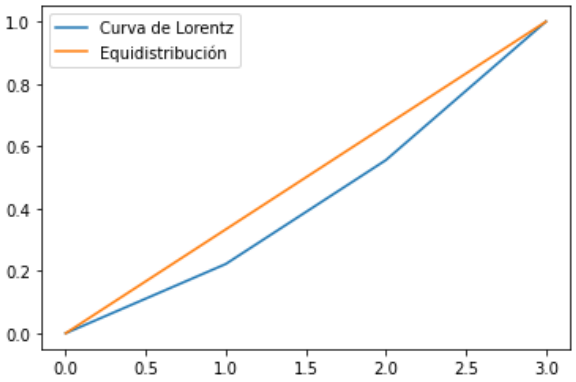
A continuación se exponen los resultados obtenidos para cada apartado pedido.

Apartado 1:

En este apartado se implementa la función *complejidadCod*, que es la parte central de la práctica. Esta función recibe como parámetro la muestra de la variable aleatoria a tratar y se encarga de lo siguiente. Primero procesa los datos para mostrar la función de probabilidad de la variable aleatoria, así como su curva de Lorentz comparándola con la equidistribución. Finalmente calcula la diversidad (^2D), el área bajo la curva de Lorentz y el índice de Gini de la muestra, escribiéndolos por consola.

También se pide en este apartado calcular éstos valores para la muestra dada. En este caso se obtienen los siguientes datos y gráficas.

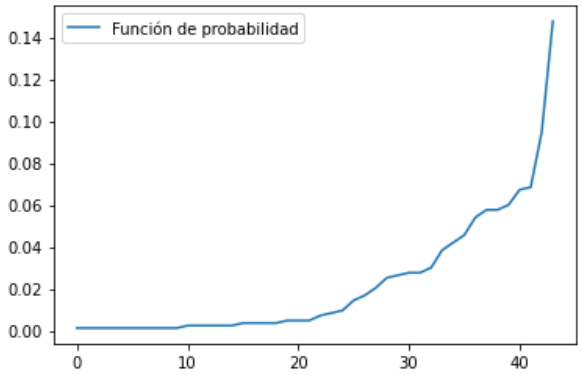


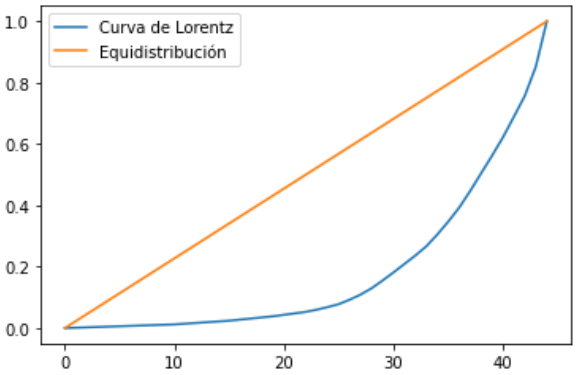


- Muestra: {0, 1, 0, 0, 0, 2, 1, 1, 0, 2, 0, 1, 2, 0, 2, 0, 1, 1}
- Área bajo la curva: 0.4259
- Índice de Gini: 0.1481
- Diversidad: 2.7931
- Número de elementos: 3

Apartado 2:

En este apartado solo se pide calcular el índice de Gini y la diversidad para la variable aleatoria de la segunda práctica. Para esta variable se obtiene lo siguiente.





- Área bajo la curva: 0.1819
- Índice de Gini: 0.6363
- Diversidad: 15.7412
- Número de elementos: 44

Conclusión

A la vista de los resultados podemos concluir que en el segundo ejemplo se tiene una variable aleatoria con una distribución mucho más desigual. Fijándonos en el índice de Gini nos damos cuenta de que el segundo caso más que cuadruplica al primero, y gráficamente la diferencia de distribución es clara. Por otro lado, si observamos la diversidad (^2D) nos damos cuenta de que en el primer caso está mucho más cerca del número de elementos, lo que concuerda con que la curva de Lorentz sea más parecida a la equidistribución. En el segundo, sin embargo, no llega a la mitad pues hay muchos elementos que apenas aparecen.

Podemos concluir que estos dos indicadores, índice de Gini y diversidad, van de la mano y basta interpretarlos adecuadamente para entender cómo de desigual es la distribución de una variable aleatoria.

Código

A continuación se anexa el código escrito para esta práctica, comentado para aclarar al lector lo que se hace en cada momento.

```python
import os
import numpy as np
import matplotlib.pyplot as plt
from collections import Counter


def complejidadCod(muestra):
    #### Contamos cuantos caracteres hay
    tab = Counter(muestra)
    weights = np.array(list(tab.values()))
    prob = weights/float(np.sum(weights))
    prob.sort()

    # Se crean los vectores con la equidistribución y la curva de Lorentz
    n = len(prob)
    prob_xy = np.linspace(0.,1.,n+1)
    prob_acum = []
    prob_acum.append(0)
    for i in range(n):
        prob_acum.append(prob_acum[i] + prob[i])

    # Se representan gráficamente
    plt.plot(prob)
    plt.title("Función de probabilidad")
    plt.show()
    plt.plot(prob_acum)
    plt.plot(prob_xy)
    plt.legend(("Curva de Lorentz", "Equidistribución"))
    plt.show()

    # Calculo correspondiente a la diversidad e indice de Gini
    A = 0.5 # Parte correspondiente a los triangulos superiores de los trapecios
    D = 0
    for i in range(n):
        D += prob[i]**2
        A += prob_acum[i]
    A /= n
    D = 1./D
    GI = 1 - 2*A

    print("Area bajo la curva: ", A)
    print("Indice de Gini: ", GI)
    print("Diversidad: ", D)
    print("Numero de elementos: ", n)
```

En el código superior se implementa *complejidadCod*, la función troncal de la práctica. En la imagen debajo situada se hace uso de ella para los dos casos que se piden en el enunciado.

```
muestra1 = [0, 1, 0, 0, 0, 2, 1, 1, 0, 2, 0, 1, 2, 0, 2, 0, 1, 1]

print("Muestra de ejemplo\n")
complejidadCod(muestra1)


#### Vamos al directorio de trabajo
os.getcwd()

with open('GCOM2023_pract1_auxiliar_eng.txt', 'r',encoding="utf8") as file:
      en = file.read()

print("-----------------\nS_English\n")
complejidadCod(en)
```