Proyecto Final

Algoritmos de aprendizaje automático para tareas de Clasificación y Análisis Inteligente de Datos.	Fecha: 30/11/2022
Alumno: Victor Alberto Lopez Cardona	Exp: 747175

Rúbrica.

Entregar en un archivo zip o rar en la actividad en CANVAS:

1. código fuente

2. reporte de proyecto

Rúbrica:

40 puntos menos - Si no anexan el código fuente

40 puntos menos - Si no anexan reporte de proyecto

20 puntos menos - Si esta incompleto su reporte, recuerden poner referencias y conclusiones.

40 puntos menos - Si la simulación es incorrecta

Objetivo general.

Programar alguno de los siguientes algoritmos.

- COLOREADO DE GRAFOS EMPLEANDO EL METODO DE ABSORCIÓN
- COLOREADO DE GRAFOS EMPLEANDO ALGORITMOS GENETICOS
- ALGORITMO DE HUFFMAN (Aplicando Árboles binarios alfabéticos óptimos)
- **CLASIFICADOR DE TEXTOS USANDO REDES BAYESIANAS INGENUAS**
- ARBOLES DE DECISIÓN MEDIANTE ID3, REPTREE Ó J48

NAIVE BAYES CLASIFICADOR DE TEXTOS :

Entrada : Texto a clasificar. (Spam_data)

	type	message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...
5	spam	FreeMsg Hey there darling it's been 3 week's n...
6	ham	Even my brother is not like to speak with me. ...
7	ham	As per your request 'Melle Melle (Oru Minnamin...
8	spam	WINNER!! As a valued network customer you have...
9	spam	Had your mobile 11 months or more? U R entitle...

Salida : Modelo Clasificador.

- Clasificador de clase, aquella con la probabilidad posterior más alta como resultado de la predicción.

```
El mensaje Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat... ha sido detectado como seguro
El mensaje Ok lar... Joking wif u oni... ha sido detectado como seguro
El mensaje Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's ha sido detectado como SPAM
El mensaje U dun say so early hor... U c already then say... ha sido detectado como seguro
El mensaje Nah I don't think he goes to usf, he lives around here though ha sido detectado como seguro
El mensaje FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send, £1.50 to rcv ha sido detectado como seguro
El mensaje Even my brother is not like to speak with me. They treat me like aids patent. ha sido detectado como seguro

Precision: 97.21823402727925
```

Algoritmo:

- Tabla de frecuencias del conjunto de Datos
- Tabla de probabilidad calculando las correspondientes a que ocurran los diversos eventos.
- Aplicación de la ecuación Naive Bayes para calcular la probabilidad posterior de cada clase.

Codigo.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import string
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score
import nltk
try:
    nltk.corpus.stopwords.words('english')
except:
    nltk.download('stopwords')
```

```
#remover letras que apocan poca info
def clean_data(mnsj):
    msg = [x for x in mnsj if x not in string.punctuation]
    msg = ''.join(msg)
    clean_msg = [x for x in msg.split() if x.lower not in
nltk.corpus.stopwords.words('english')]
    return clean_msg

#leer informacion
all_data = pd.read_csv('Spam_data',sep='\t',names=["type","message"])
all_data.head(10)
all_data.groupby('type').describe()
#determinar atributos que nos puedan indicar si es SPAM o no
all_data['length']=all_data['message'].apply(len)
all_data.length.describe()
#transformar la informacion a vectores
vector = CountVectorizer(analyzer=clean_data).fit(all_data['message'])
df_vector = vector.transform(all_data['message'])
frecuencias=TfidfTransformer().fit(df_vector)
#obtener una tabla de frecuencias
frecuencias_data_frame=frecuencias.transform(df_vector)
print(frecuencias_data_frame.shape)
#entrenar el modelo
modelo = MultinomialNB().fit(frecuencias_data_frame,all_data['type'])
#predecir todos los datos con el modelo
resultados = modelo.predict(frecuencias_data_frame)
print(resultados)
#imprimir los primeros mensajes y obtener si es seguro o SPAM
for i in range(0,50):
    if resultados[i] == "ham":
        print(f"El mensaje {all_data['message'][i]} ha sido detectado como
seguro")
    elif resultados[i] == "spam":
        print(f"El mensaje {all_data['message'][i]} ha sido detectado como
SPAM")
#calcular la precision
puntaje = accuracy_score(all_data['type'], resultados)
print(f'Precision: {puntaje*100}')
```

Conclusiones.

El objetivo del proyecto es hacer uso de redes bayesianas ingenuas para clasificar un mensaje como SPAM o como mensaje seguro, para hacer esto podemos hacer uso de 3 distintos algoritmos, que son el Multinomial, Gaussiano, y de Bernoulli, para este proyecto se opto por utilizar el algoritmo multinomial ya que para nuestro set de información que es un texto, este algoritmo funciona mejor.

El proyecto logra clasificar los mensajes como SPAM o mensaje seguro con un 97% de precisión aproximadamente.
Como áreas de mejora o trabajo futuro, podemos normalizar la información que tenemos, hay algunos mensajes que contienen una palabra mal escrita y podemos detectar esta palabra y corregirla para tener una tabla de frecuencias correcta en nuestro proyecto, esto nos llevara a un modelo de predicción mas preciso.

En el código podemos observar que tenemos una función clean_data(), en esta función estamos eliminando palabras que no nos aportan información valiosa al mensaje, por lo que las eliminamos por cada uno de los mensajes que tenemos, esto mejoro la precisión de nuestro modelo en un 30%, ya que la mayoría de las palabras que se repetían en los mensajes eran “the”, “a”, is“, y el modelo identificaba erróneamente los mensajes que contenían SPAM.

Bibliografía.

https://ccc.inaoep.mx/~esucar/Clases-mgp/Proyectos/MGP_RepProy_Abr_29.pdf
https://www.inf.ed.ac.uk/teaching/courses/inf2b/learnnotes/inf2b-learn-note07-2up.pdf
https://www.geeksforgeeks.org/naive-bayes-classifiers/