CENTRO DE CIENCIAS BÁSICAS

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

METAHEURÍSTICAS I

7° "A"

ACTIVIDAD 5_01. MACHINE LEARNING

Profesor: Francisco Javier Luna Rosas

Alumnos:

Almeida Ortega Andrea Melissa

Espinoza Sánchez Joel Alejandro

Flores Fernández Óscar Alonso

Gómez Garza Dariana

González Arenas Fernando Francisco

Orocio García Hiram Efraín

Fecha de Entrega: Aguascalientes, Ags.,**19** de noviembre de 2021

Actividad 5_01. Machine Learning

Antecedentes

Como ya sabemos inteligencia artificial es la combinación de algoritmos planteados con el propósito de crear máquinas que presenten las mismas capacidades que el ser humano. El machine learning es una rama de la computación de la IA que se basa en el entrenamiento de algoritmos de aprendizaje automático a partir de datos anteriores.

Esos tipos de aprendizajes se dividen en supervisados, no supervisados, semi-supervisado y por refuerzo; pero en esta ocasión nos centraremos en el aprendizaje supervisado.

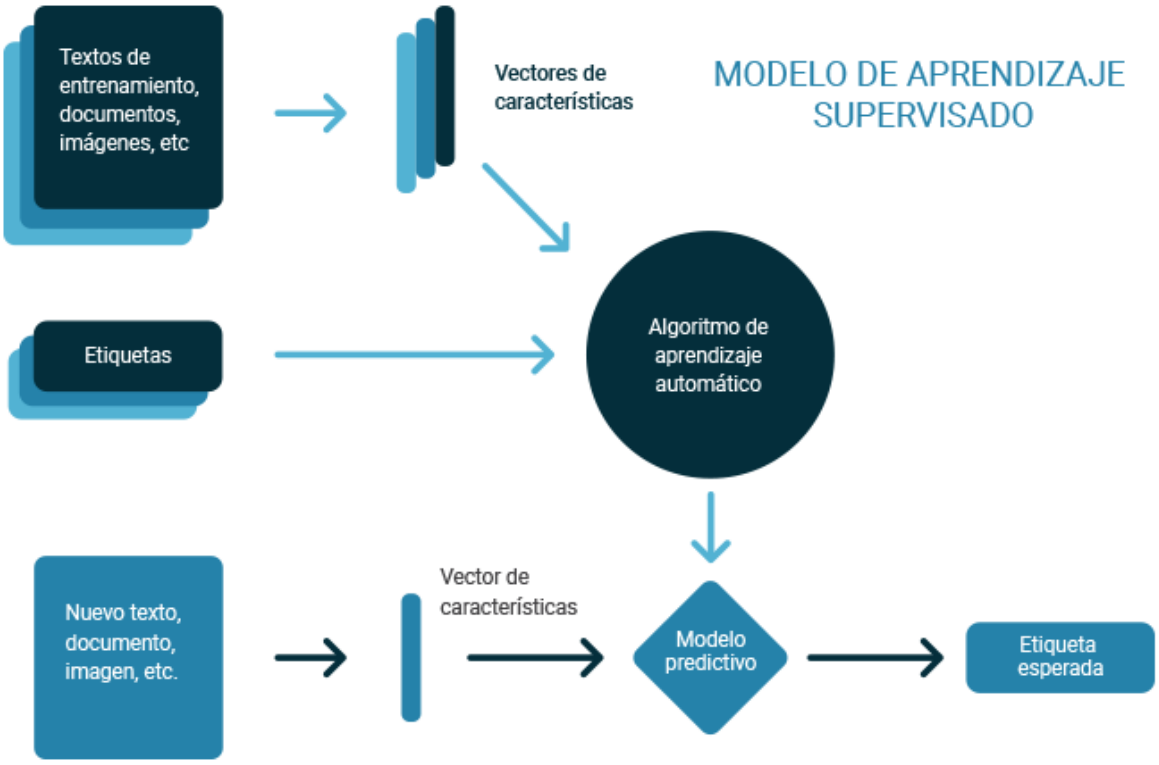
El aprendizaje supervisado necesita conjuntos de datos etiquetados, es decir, le decimos al modelo qué es lo que queremos que aprenda. Dependiendo del tipo de etiqueta, dentro del aprendizaje supervisado existen dos tipos de modelos:

- **Los modelos de clasificación**, que producen como salida una etiqueta discreta, es decir, una etiqueta dentro de un conjunto finito de etiquetas posibles. A su vez, los modelos de clasificación pueden ser binarios si tenemos que predecir entre dos clases o etiquetas (enfermedad o no enfermedad, clasificación de correos electrónicos como “spam” o no “spam”) o multiclase, cuando se tiene que clasificar más de dos clases (clasificación de imágenes de animales, análisis de sentimientos, etc.).

- **Los modelos de regresión** producen como salida un valor real.

En los algoritmos de aprendizaje supervisado se genera un modelo predictivo, basado en datos de entrada y salida. La palabra clave “supervisado” viene de la idea de tener un conjunto de datos previamente etiquetado y clasificado, es decir, tener un conjunto de muestra, el cual ya se sabe a qué grupo, valor o categoría pertenecen los ejemplos. Con este grupo de datos que llamamos datos de entrenamiento, se realiza el ajuste al modelo inicial planteado. Es de esta forma

como el algoritmo va “aprendiendo” a clasificar las muestras de entrada comparando el resultado del modelo, y la etiqueta real de la muestra, realizando las compensaciones respectivas al modelo de acuerdo con cada error en la estimación del resultado.



Existen muchos algoritmos de aprendizaje supervisado que nos ayudan a clasificar, por ejemplo:

- Árboles de decisión
- Clasificación de Naïve Bayes
- Regresión por mínimos cuadrados
- Regresión Logística
- Support Vector Machines (SVM)
- Métodos “Ensemble” (Conjuntos de clasificadores).

Hablando un poco del modelo Support Vector Machine, es o, mejor dicho, son un conjunto de algoritmos de aprendizaje supervisado desarrollados por Vladimir Vapnik y su equipo en los laboratorios AT&T. Que funciona a grandes rasgos de la

siguiente manera; Dado un conjunto de puntos, subconjunto de un conjunto mayor (espacio), en el que cada uno de ellos pertenece a una de dos posibles categorías, un algoritmo basado en SVM construye un modelo capaz de predecir si un punto nuevo (cuya categoría desconocemos) pertenece a una categoría o a la otra.

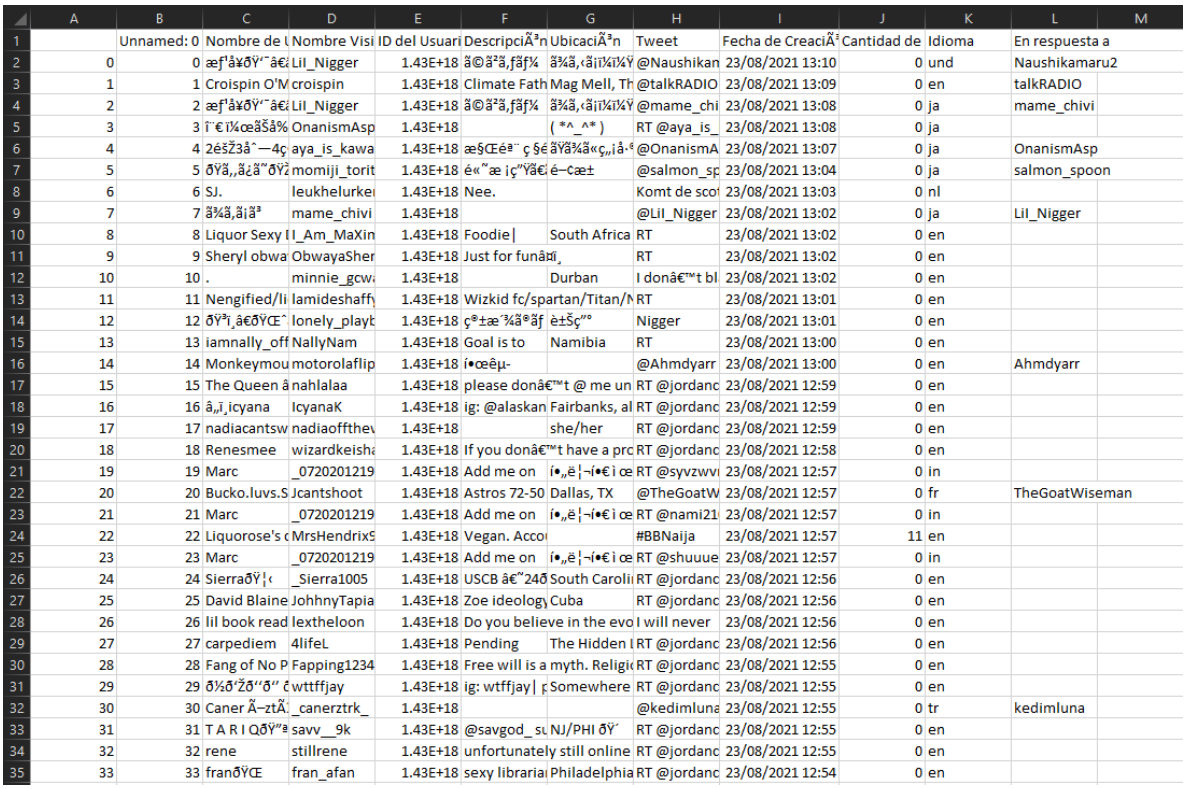
Como en la mayoría de los métodos de clasificación supervisada, los datos de entrada (los puntos) son vistos como un vector p-dimensional (una lista ordenada de p números).

La SVM busca un hiperplano que separe de forma óptima a los puntos de una clase de la de otra, que eventualmente han podido ser previamente proyectados a un espacio de dimensionalidad superior.

En ese concepto de "separación óptima" es donde reside la característica fundamental de las SVM: este tipo de algoritmos buscan el hiperplano que tenga la máxima distancia (margen) con los puntos que estén más cerca de él mismo. Por eso también a veces se les conoce a las SVM como clasificadores de margen máximo. De esta forma, los puntos del vector que son etiquetados con una categoría estarán a un lado del hiperplano y los casos que se encuentren en la otra categoría estarán al otro lado.

Evidencias

El archivo CSV de tweets extraídos tiene una apariencia como la siguiente:



	A	B	C	D	E	F	G	H	I	J	K	L	M
1		Unnamed: 0	Nombre de U	Nombre Visi	ID del Usuari	DescripciÃ³n	UbicaciÃ³n	Tweet	Fecha de CreaciÃ³	Cantidad de	Idioma	En respuesta a	
2	0	0	æƒ¹å¥ðŸ‘ˆâ€¿	Lil_Nigger	1.43E+18	ã©ãªãƒãƒ¼	ã¾ã‚‹ã‚ï¼¼ï¼¼	@Naushikan	23/08/2021 13:10	0	und	Naushikamaru2	
3	1	1	Croispin O'M	croispin	1.43E+18	Climate Fath	Mag Mell, Th	@talkRADIO	23/08/2021 13:09	0	en	talkRADIO	
4	2	2	æƒ¹å¥ðŸ‘ˆâ€¿	Lil_Nigger	1.43E+18	ã©ãªãƒãƒ¼	ã¾ã‚‹ã‚ï¼¼ï¼¼	@mame_chi	23/08/2021 13:08	0	ja	mame_chivi	
5	3	3	ï¯€ï¼œæšŠå%	OnanismAsp	1.43E+18		(*^_^*)	RT @aya_is_	23/08/2021 13:08	0	ja		
6	4	4	2éšŽ3åˆ—4ç	aya_is_kawa	1.43E+18	æ§Œéª¨ ç §é	ãŸã¾ã«ç„¡å·¥	@OnanismA	23/08/2021 13:07	0	ja	OnanismAsp	
7	5	5	ðŸŸå„¿ã¯ðŸŽ	momiji_torit	1.43E+18	é«˜æ ¡ç”Ÿé	é–¢æ±	@salmon_sp	23/08/2021 13:04	0	ja	salmon_spoon	
8	6	6	SJ.	leukhelurke	1.43E+18	Nee.		Komt de sco	23/08/2021 13:03	0	nl		
9	7	7	ã¾ã‚ã¡ã³	mame_chivi	1.43E+18			@Lil_Nigger	23/08/2021 13:02	0	ja	Lil_Nigger	
10	8	8	Liquor Sexy I	I_Am_MaXin	1.43E+18	Foodie|	South Africa	RT	23/08/2021 13:02	0	en		
11	9	9	Sheryl obwa	ObwayaSher	1.43E+18	Just for funâ		RT	23/08/2021 13:02	0	en		
12	10	10	.	minnie_gcw	1.43E+18		Durban	I donâ€™t bl	23/08/2021 13:02	0	en		
13	11	11	Nengified/li	lamideshaffy	1.43E+18	Wizkid fc/spartan/Titan/M		RT	23/08/2021 13:01	0	en		
14	12	12	ðŸ³ï¸â€ðŸŒˆ	lonely_playb	1.43E+18	ç°±æ´¾ã°ãƒ	è±Šç”°	Nigger	23/08/2021 13:01	0	en		
15	13	13	iamnally_off	NallyNam	1.43E+18	Goal is to	Namibia	RT	23/08/2021 13:00	0	en		
16	14	14	Monkeymou	motorolaflip	1.43E+18	í•œêµ-		@Ahmdyarr	23/08/2021 13:00	0	en	Ahmdyarr	
17	15	15	The Queen â	nahlalaa	1.43E+18	please donâ€™t @ me un		RT @jordanc	23/08/2021 12:59	0	en		
18	16	16	â„ï¸icyana	IcyanaK	1.43E+18	ig: @alaskan	Fairbanks, al	RT @jordanc	23/08/2021 12:59	0	en		
19	17	17	nadiacantsw	nadiaoffthev	1.43E+18		she/her	RT @jordanc	23/08/2021 12:59	0	en		
20	18	18	Renesmee	wizardkeisha	1.43E+18	If you donâ€™t have a prc		RT @jordanc	23/08/2021 12:58	0	en		
21	19	19	Marc	_0720201219	1.43E+18	Add me on	í•„ë¦¬í•€ì œ	RT @syvzwv	23/08/2021 12:57	0	in		
22	20	20	Bucko.luvs.S	Jcantshoot	1.43E+18	Astros 72-50	Dallas, TX	@TheGoatW	23/08/2021 12:57	0	fr	TheGoatWiseman	
23	21	21	Marc	_0720201219	1.43E+18	Add me on	í•„ë¦¬í•€ì œ	RT @nami21	23/08/2021 12:57	0	in		
24	22	22	Liquorose's c	MrsHendrix9	1.43E+18	Vegan. Acco		#BBNaija	23/08/2021 12:57	11	en		
25	23	23	Marc	_0720201219	1.43E+18	Add me on	í•„ë¦¬í•€ì œ	RT @shuuue	23/08/2021 12:57	0	in		
26	24	24	SierraðŸ¦‹	_Sierra1005	1.43E+18	USCB â€™24ð	South Caroli	RT @jordanc	23/08/2021 12:56	0	en		
27	25	25	David Blaine	JohhnyTapia	1.43E+18	Zoe ideology	Cuba	RT @jordanc	23/08/2021 12:56	0	en		
28	26	26	lil book read	lextheloon	1.43E+18	Do you believe in the evo		I will never	23/08/2021 12:56	0	en		
29	27	27	carpediem	4lifeL	1.43E+18	Pending	The Hidden I	RT @jordanc	23/08/2021 12:56	0	en		
30	28	28	Fang of No P	Fapping1234	1.43E+18	Free will is a myth. Religi		RT @jordanc	23/08/2021 12:55	0	en		
31	29	29	ð½ð‘Žð‘“ð‘	wttffjay	1.43E+18	ig: wtffjay| ç	Somewhere	RT @jordanc	23/08/2021 12:55	0	en		
32	30	30	Caner Ã–ztÃ	_canerztrk_	1.43E+18			@kedimluna	23/08/2021 12:55	0	tr	kedimluna	
33	31	31	T A R I QðŸ”¥	savv__9k	1.43E+18	@savgod_ su	NJ/PHI ðŸ´	RT @jordanc	23/08/2021 12:55	0	en		
34	32	32	rene	stillrene	1.43E+18	unfortunately still online		RT @jordanc	23/08/2021 12:55	0	en		
35	33	33	franðŸŒ	fran_afan	1.43E+18	sexy librarian	Philadelphia	RT @jordanc	23/08/2021 12:54	0	en		

El archivo puede encontrarse adjunto en este entregable con el nombre *“Team 10 Scrapping (Primera Limpieza).csv”*.

Al hacer el proceso de lemantización se generó un nuevo archivo llamado *“Lemantizacion2.csv”* con una visualización como la de a continuación:

A	B	C	D	E	F	G	H	I	J	K	L
RT @ jordancoduu : @ poochivicious Did he just call me a nigger ?											
@ inteledits sir aap to nigger ho											
@ poochivicious Did he just call me a nigger ?											
@ KnockBrands Negrito Bimbo with a `` white afro kid '' But now its call `` nito '' cuss `` Negrito '' be basically `` nigger '' and `` Nito '' Is the same but More frien											
@ Charlesrat74 If it be up to you negros wrestle would have all nigger champion											
@ 0X1HONG His as just look like he problematic .. maybe would call me a nigger or two											
Django : “ .. Every single word that come out of Calvin Candie 's mouth be nothing but horseshit but he be right about one thing . I be that one nigger in ten th											
Precisava disso ?											
@ dking_go @ Tegs114 @ niccjennings @ BigBadCrabCake @ ballinhog @ itspreciouzzzz @ Big_Eti @ ChrisExcel102 @ EvrybodyHatesAP Stfu niggér											
@ DerrickNAACP Do you plan on go on after USTA.3 Black men be sue Black NYPD LT be call nigger by Black parking attend 2017 US Open Tennis a disgrace.NA											
nigger											
RT @ Phil_Lewis_ : Tarrant Alabama city council member Tommy Bryant have come under fire after he stand up and loudly ask “ Do we have a ho…											
isa just call me a nigger SERIOUS NOT CLICKBAIT											
My quest4 # Justice4Vera & amp ; Title VII 1964 CRA victim be # Divine i.e . Unwed dead mom # EulaMaeRichardson firing by Mr Carter4 `` mention that nigger											
`` house nigger fuck allow to fuck leave ''											
RT @ ChiefKeef : Fuck u nigger http : //t.co/pBhUZY1LIB											
FOOLISH NIGGER ? ? €\^\ * \+\+ http : //t.co/ul47C6dUzV											
A fly roach just land on my front door . I ’ m pretty sure bitch call me a nigger .											
Unprofessional nigger b.s exactly as I say it Automatically blame money and power											
NIGGER											
twitter term people for say |k¥ $ but im get call a nigger and its cricket ? ?											
RT @ StanCollymore : Sharna Walker . Worcester . Racist . In custody as we speak . Sacked from a job . `` Black cunt '' nigger '' . http : //t.co/JGY…											
If you think this nigger will be the governor of California You 're on crack											
“ Cut your hair . You should cut your hair . You know white people and corporate America cut your hair . Cut your hair . You should trim your beard . Cut your h											
Did you just call him a nigger ? http : //t.co/6iSti1IQbN											
RT @ Patrick42594797 : @ MatthewTPage @ USinNigeria @ AsstSecStateAF @ ECAatState US be now act on frustration @ StateDept Afgahnistan be a g…											
@ Puer_Hispano Creo que he interactuado con domicanos equivocados se parecen a los puerto riqueños de Internet domados por los Yankis y el movimient											
@ N0IRXMINGS @ diiorhoseokk @ MOONSOFHOPE @ shotasocean @ GLOHOBI @ ELILICIOUSSS @ ksjkult @ horsoek @ makhylisgf @ s0ulscared @ billboardc											
@ MookieSZN50 NIGGER											
I love it dude It take the edge off of all the nigger bullshit we have to deal with ! ! ! I wan na see a war											
Take a hint nigger											
i could take my gun right now and shoot my jaw off and live but some fucking nigger in tarkov will grace my chin once with 9mm and i drop dead instantly											
@ reef1x_ @ Mongraal That ’ s my nigger *											
Wendy stans come for crush like she didn ’ t just call me a nigger last week y ’ all sit down											
@ GuiltyG3kk Nigger activity http : //t.co/QRxZdFT18P											
I feel like Pulp Fiction have `` only '' the one `` dead nigger storage '' dialouge/scene but Reservoir Dogs have like 4 or 5 of them moment . Sheesh .											

Finalmente, se genera una evaluación de análisis de sentimientos en un archivo llamado *"Sentimiento.csv"* con la siguiente visualización:

	A	B	C	D	E
1	{'neg': 0.243	'neu': 0.567	'pos': 0.19	'compound': -0.3818}	
2	{'neg': 0.243	'neu': 0.567	'pos': 0.19	'compound': -0.3818}	
3	{'neg': 0.243	'neu': 0.567	'pos': 0.19	'compound': -0.3818}	
4	{'neg': 0.683	'neu': 0.317	'pos': 0.0	'compound': -0.6486}	
5	{'neg': 0.243	'neu': 0.567	'pos': 0.19	'compound': -0.3818}	
6	{'neg': 0.382	'neu': 0.412	'pos': 0.206	'compound': -0.6124}	
7	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	
8	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	
9	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	
10	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	
11	{'neg': 0.264	'neu': 0.736	'pos': 0.0	'compound': -0.6486}	
12	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	
13	{'neg': 0.28	'neu': 0.72	'pos': 0.0	'compound': -0.6868}	
14	{'neg': 0.234	'neu': 0.602	'pos': 0.163	'compound': -0.5386}	
15	{'neg': 0.199	'neu': 0.694	'pos': 0.106	'compound': -0.4588}	
16	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	
17	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	
18	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	
19	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	
20	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	
21	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	
22	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	
23	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	
24	{'neg': 0.207	'neu': 0.728	'pos': 0.065	'compound': -0.5707}	
25	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	
26	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	
27	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	
28	{'neg': 0.417	'neu': 0.583	'pos': 0.0	'compound': -0.6486}	
29	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	
30	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	
31	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	
32	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	
33	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	
34	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	
35	{'neg': 0.0	'neu': 1.0	'pos': 0.0	'compound': 0.0}	

Con base en este archivo, se limpió el contenido no deseado, dejando únicamente la columna de compound que pondera la componente negativa, la neutra y la positiva y da un resultado único para el tweet. Esta información fue la que se procesó en el código siguiente (véase el anexo 1 para revisar el código completo).

Primero se importaron las librerías:

```
#%% Importamos las librerías
import pandas as pd
import os
import numpy as np
import cv2
import matplotlib.pyplot as plt
import pickle
import random
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC

from sklearn.preprocessing import StandardScaler
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
```

Se cargó el archivo con el análisis de sentimientos, se definieron las categorías y se declaró la variable de datos:

```
#%% Cargamos los datos de análisis de sentimientos
datosRacismo = pd.read_csv("Sentimiento.csv")

#%% Declaramos las categorías
categories = ['Negativo', 'Neutro', 'Positivo']

#%% Declaramos la variable que mantendrá los datos a lo largo del procedimiento Random Forest
data = []
```

Se limpió la información para dejar únicamente la columna de compound del análisis de sentimientos:

```
#%% Se revisa por categoría cada imagen del directorio
for category in categories:
    label = categories.index(category)

    for i in datosRacismo.iloc[:,3]:
        numberData = i.replace(" \'compound\': ","")
        numberData = numberData.replace("} ","")
        data.append([float(numberData),round(float(numberData)) + 1])
```

Posteriormente se realiza la primera parte de análisis supervisado:

```
#%% Se crea el documento data1 en el mismo directorio de archivos con el análisis Random Forest
pick_in = open('data1.pickle', 'wb')
pickle.dump(data, pick_in)
pick_in.close()

#%% Se carga la información del archivo data1 en la variable data1
pick_in = open('data1.pickle', 'rb')
data1 = pickle.load(pick_in)
pick_in.close()

#%% Se elige un individuo al azar
random.shuffle(data1)
features1 = []
label1 = []
```

Finalmente se realiza el procedimiento del algoritmo Random Forest y se muestran los resultados:

```
#%% Se llena el arreglo de características de cada individuo
for feature, labelint in data1:
    features1.append([feature,0])
    label1.append(labelint)
```

```
#%% Se realiza la clasificación a partir del entrenamiento usando 75% de la población
xtrain, xtest, ytrain, ytest = train_test_split(features1, label1, test_size = 0.20, random_state = 0)

sc = StandardScaler()

xtrain = sc.fit_transform(xtrain)
xtest = sc.fit_transform(xtest)

classifier = RandomForestClassifier(n_estimators = 12, random_state = 0)
classifier.fit(xtrain, ytrain)

ypred = classifier.predict(xtest)

print("Matriz de confusión:")
print(confusion_matrix(ytest, ypred))

print("Reporte de clasificación:")
print(classification_report(ytest, ypred))

print("Precisión:")
print(accuracy_score(ytest, ypred))
```

Los resultados que arroja el algoritmo en una ejecución son los siguientes (véase el anexo 2 para ver más ejecuciones):

```
Matriz de confusión:
[[421884      0      0]
 [    20 182436      0]
 [     0     33  24739]]
Reporte de clasificación:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    421884
           1       1.00      1.00      1.00    182456
           2       1.00      1.00      1.00     24772

    accuracy                           1.00    629112
   macro avg       1.00      1.00      1.00    629112
weighted avg       1.00      1.00      1.00    629112

Precisión:
0.9999157542695101
```

La matriz de confusión está ordenada de izquierda a derecha (en columnas y de arriba hacia abajo en filas) por sentimiento de racismo negativo, neutro y positivo.

Como se puede observar, el algoritmo tiene una precisión muy grande, pues en la matriz de confusión, se observa que la diagonal, es decir, las predicciones acertadas componen a la mayoría, mientras que son pocos los tweets que llegan a ser mal validados.

Conclusiones

Andrea Melissa Almeida Ortega: Como se puede ver, es de suma importancia la implementación de una inteligencia artificial para clasificar cualquier cosa, en este caso, tweets. Cuando hablamos de proyectos grandes, ayuda a que empresas también sepan la opinión de personas a través de ellos.

Joel Alejandro Espinoza Sánchez: Gracias a la actividad, nos permitimos explorar el planteamiento de un problema de aprendizaje desde la selección del algoritmo, ya que no todos los algoritmos están diseñados para la solución de una misma tarea. Su diversidad permite la solución de problemas orientados a distintos tipos de datos, predicciones y objetivos y Random Forest era el idóneo.

Pude orientar personalmente un enfoque de investigación para proponer el mejor algoritmo con base en lo que deseábamos predecir y el conjunto de datos con lo que realizaríamos el procedimiento.

Óscar Alonso Flores Fernández: Poder trabajar de nuevo con Random Forest nos permite comprender el cómo hacer uso de diversas fuentes de recursos. En este caso el programa es sencillo y a una escala muy pequeña, pero escalar poco a poco con el fin de realmente mejorar la eficiencia en futuros proyectos de mayor tamaño será imperativo para ser capaces de cubrir las demandas de cada tarea llevada a cabo.

Dariana Gómez Garza: En este proyecto final pudimos agrandar el conocimiento que teníamos en aprendizaje supervisado, ya que nunca habíamos hecho una inteligencia artificial a base de tweets en algún curso.

Es interesante la amplia gama de opciones que hay para realizar este tipo de programas, por ejemplo, encontramos mucha información sobre Tensorflow y cómo era más sencillo clasificarlo así en lugar de realizarlo con el aprendizaje supervisado; pero el punto de este último parcial era tratar de realizarlo con este tema (Random Forest) en lugar de redes neuronales.

Me gustó mucho como quedó nuestro proyecto final y también era muy interesante como el algoritmo detectaba las emociones de los tweets y las clasificaba.

Fernando Francisco González Arenas: El reconocimiento de emociones en los tweets por medio de aprendizaje supervisado es una técnica muy útil que tiene múltiples usos en la vida diaria de las personas alrededor del mundo en la actualidad, se puede usar para fines de seguridad, clasificación de grupos de población, detectar emociones y sentimientos de las personas, etc.

Con la realización de esta práctica investigamos formas de detectar emociones en los tweets y clasificarlos para saber si contenía palabras racistas o no, lo cual puede tener muchas aplicaciones prácticas en el futuro, incluso para futuros proyectos de los integrantes de este mismo equipo.

Hiram Efraín Orocio García: La clasificación de emociones es útil al momento de hacer uso de recursos, datos, servicios, entre otras cosas así que aprender a implementar aun así sea en una aplicación básica como para saber qué opinan sobre un producto o en este caso saber que emociones transmiten las personas mediante las publicaciones, en un futuro lo podríamos implementar en problemas mayores donde el compartir estos recursos, serán de gran ayuda.

Referencias

- Igual, L. (2017). *Introduction to Data Science*. Barcelona: Springer.
- Rebala, G. (2019). *An Introduction to Machine Learning*. California: Springer.
- Sarkar, D. (2018). *Practical Machine Learning with Python*. Chicago: Apress.
- Unpingco, J. (2019). *Python for Probability, Statistics and Machine Learning*. California: Springer.

Anexos

Anexo 1: Código de Machine Learning con Random Forest en Python:

```
#%% Presentación
'''
   Universidad Autónoma de Aguascalientes

         Centro de Ciencias Básicas
 Departamento de Ciencias de la Computación
              Metaheurísticas I
                   7° "A"

                Actividad 5.01

   Profesor: Francisco Javier Luna Rosas

  Alumnos:
      Almeida Ortega Andrea Melissa
      Espinoza Sánchez Joel Alejandro
      Flores Fernández Óscar Alonso
      Gómez Garza Dariana
      González Arenas Fernando Francisco
      Orocio García Hiram Efraín

  Fecha de Entrega: 19 de noviembre del 2021

Descripción: Machine Learning (Random Forest como Aprendizaje Supervisado)
para análisis de sentimientos
'''

#%% Importamos las librerías
import pandas as pd
import os
import numpy as np
import cv2
import matplotlib.pyplot as plt
import pickle
import random
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC

from sklearn.preprocessing import StandardScaler
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from  sklearn.metrics  import  classification_report,  confusion_matrix,
accuracy_score

#%% Cargamos los datos de análisis de sentimientos
datosRacismo = pd.read_csv("Sentimiento.csv")
```

```
#%% Declaramos las categorías
categories = ['Negativo', 'Neutro', 'Positivo']

#%% Declaramos la variable que mantendrá los datos a lo largo del
procedimiento Random Forest
data = []

#%% Se revisa por categoría cada imagen del directorio
for category in categories:
    label = categories.index(category)

    for i in datosRacismo.iloc[:,3]:
        numberData = i.replace(" \'compound\': ","")
        numberData = numberData.replace("} ","")
        data.append([float(numberData),round(float(numberData)) + 1])

#%% Se crea el documento data1 en el mismo directorio de archivos con el
análisis Random Forest
pick_in = open('data1.pickle', 'wb')
pickle.dump(data, pick_in)
pick_in.close()

#%% Se carga la información del archivo data1 en la variable data1
pick_in = open('data1.pickle', 'rb')
data1 = pickle.load(pick_in)
pick_in.close()

#%% Se elige un individuo al azar
random.shuffle(data1)
features1 = []
label1 = []

#%% Se llena el arreglo de características de cada individuo
for feature, labelint in data1:
    features1.append([feature,0])
    label1.append(labelint)

#%% Se realiza la clasificación a partir del entrenamiento usando 75% de
la población
xtrain, xtest, ytrain, ytest = train_test_split(features1, label1,
test_size = 0.20, random_state = 0)

sc = StandardScaler()

xtrain = sc.fit_transform(xtrain)
xtest = sc.fit_transform(xtest)

classifier = RandomForestClassifier(n_estimators = 12, random_state = 0)
classifier.fit(xtrain, ytrain)
```

```
ypred = classifier.predict(xtest)

print("Matriz de confusión:")
print(confusion_matrix(ytest, ypred))

print("Reporte de clasificación:")
print(classification_report(ytest, ypred))

print("Precisión:")
print(accuracy_score(ytest, ypred))
```

Anexo 2: Ocho ejecuciones del código de Random Forest:

Ejecución 1:

```
Matriz de confusión:
[[421884      0      0]
 [    20 182436      0]
 [     0     33  24739]]
Reporte de clasificación:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    421884
           1       1.00      1.00      1.00    182456
           2       1.00      1.00      1.00     24772

    accuracy                           1.00    629112
   macro avg       1.00      1.00      1.00    629112
weighted avg       1.00      1.00      1.00    629112

Precisión:
0.9999157542695101
```

Ejecución 2:

```
Matriz de confusión:
[[421621      0      0]
 [     0 182854      0]
 [     0      0  24637]]
Reporte de clasificación:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    421621
           1       1.00      1.00      1.00    182854
           2       1.00      1.00      1.00     24637

    accuracy                           1.00    629112
   macro avg       1.00      1.00      1.00    629112
weighted avg       1.00      1.00      1.00    629112

Precisión:
1.0
```

Ejecución 3:

```
Matriz de confusión:
[[422121      0      0]
 [    12 182175      0]
 [     0     34  24770]]
Reporte de clasificación:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    422121
           1       1.00      1.00      1.00    182187
           2       1.00      1.00      1.00     24804

    accuracy                           1.00    629112
   macro avg       1.00      1.00      1.00    629112
weighted avg       1.00      1.00      1.00    629112

Precisión:
0.9999268810641031
```

Ejecución 4:

```
Matriz de confusión:
[[421871     38      0]
 [     0 182654     31]
 [     0      0  24518]]
Reporte de clasificación:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    421909
           1       1.00      1.00      1.00    182685
           2       1.00      1.00      1.00     24518

    accuracy                           1.00    629112
   macro avg       1.00      1.00      1.00    629112
weighted avg       1.00      1.00      1.00    629112

Precisión:
0.9998903215961545
```

Ejecución 5:

```
Matriz de confusión:
[[421815      0      0]
 [    21 182513      0]
 [     0     30  24733]]
Reporte de clasificación:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    421815
           1       1.00      1.00      1.00    182534
           2       1.00      1.00      1.00     24763

    accuracy                           1.00    629112
   macro avg       1.00      1.00      1.00    629112
weighted avg       1.00      1.00      1.00    629112

Precisión:
0.9999189333536794
```

Ejecución 6:

```
Matriz de confusión:
[[422502     43      0]
 [     0 181813     34]
 [     0      0  24720]]
Reporte de clasificación:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    422545
           1       1.00      1.00      1.00    181847
           2       1.00      1.00      1.00     24720

    accuracy                           1.00    629112
   macro avg       1.00      1.00      1.00    629112
weighted avg       1.00      1.00      1.00    629112

Precisión:
0.9998776052594769
```

Ejecución 7:

```
Matriz de confusión:
[[422274     37      0]
 [     0 181887      0]
 [     0     36  24878]]
Reporte de clasificación:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    422311
           1       1.00      1.00      1.00    181887
           2       1.00      1.00      1.00     24914

    accuracy                           1.00    629112
   macro avg       1.00      1.00      1.00    629112
weighted avg       1.00      1.00      1.00    629112

Precisión:
0.9998839634278157
```

Ejecución 8:

```
Matriz de confusión:
[[423196     51      0]
 [     0 181283    110]
 [     0      0  24472]]
Reporte de clasificación:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    423247
           1       1.00      1.00      1.00    181393
           2       1.00      1.00      1.00     24472

    accuracy                           1.00    629112
   macro avg       1.00      1.00      1.00    629112
weighted avg       1.00      1.00      1.00    629112

Precisión:
0.9997440837243606
```