



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE COMPUTO



Nombre de los alumnos: Sandoval Muñoz Emanuel, Valerio López José

Eduardo, Trejo Sierra Héctor

Matriculas: 2022670104, 2022670002, 2021630065

Maestr@: CAMACHO VAZQUEZ VANESSA ALEJANDRA

Materia: MACHINE LEARNING

Grupo: 6CV1

Trabajo: Clasificadores usando tweets etiquetados con 3 emociones diferentes.

Código:

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split, cross_val_score, cross_val_predict
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix, precision_score, recall_score, f1_score
import pandas as pd

# Ruta al archivo con los tweets
ruta_datos = r'C:\Users\emanu\Documents\Escuela\PROGRAMACION\python\Machine\twets_etiquetados\TWEETS_EMOCIONES_NEGATIVAS.txt'

# Leer el archivo y extraer los tweets
tweets = []
with open(ruta_datos, 'r', encoding='utf-8') as file:
    lines = file.readlines()

current_tweet = []
for line in lines:
    if line.startswith('==== Frase'):
        if current_tweet:
            tweets.append(' '.join(current_tweet))
            current_tweet = []
        elif 'Palabra:' in line:
            word = line.split('Palabra:')[1].split()[0]
            current_tweet.append(word)
```

```
# Añadir el último tweet si lo hay
if current_tweet:
    tweets.append(' '.join(current_tweet))

# Asignar etiquetas según los hashtags
y = []
for i, tweet in enumerate(tweets):
    if i < 50:
        y.append('Ira')
    elif 50 ≤ i < 100:
        y.append('Tristeza')
    elif i ≥ 100:
        y.append('Miedo')

print(y)
```

```

# Crear el vectorizador TF-IDF
tfidf_vectorizer = TfidfVectorizer(ngram_range=(1, 1))

# Ajustar el vectorizador y transformar los datos de texto
tfidf_matrix = tfidf_vectorizer.fit_transform(tweets)

# Obtener el vocabulario (características)
feature_names = tfidf_vectorizer.get_feature_names_out()

# Convertir la matriz TF-IDF a DataFrame para mejor manejo
df_tfidf = pd.DataFrame(tfidf_matrix.toarray(), columns=feature_names)

# Guardar los vectores TF-IDF en un archivo
df_tfidf.to_csv( path_or_buf: "vectores_tweets_Uni_emonegativas.txt", index=False)

print("Vectores TF-IDF guardados exitosamente.")

# Imprimir vocabulario
print("Vocabulario (Características): ")
print(feature_names)

# Imprimir matriz TF-IDF
print("\nMatriz TF-IDF")
print(df_tfidf)

```

```

# Construcción del clasificador Naive Bayes
algoritmo_nb = MultinomialNB()
y_pred_nb = cross_val_predict(algoritmo_nb, tfidf_matrix, y, cv=10)

# Evaluación del clasificador Naive Bayes
precision_nb = precision_score(y, y_pred_nb, average='macro', zero_division=0)
recall_nb = recall_score(y, y_pred_nb, average='macro', zero_division=0)
f1_nb = f1_score(y, y_pred_nb, average='macro', zero_division=0)

print(f"Presicion del modelo Naive Bayes: {precision_nb}")
print(f"Recall del modelo Naive Bayes: {recall_nb}")
print(f"F1-score del modelo Naive Bayes: {f1_nb}")

# Construcción del clasificador SVM
algoritmo_svm = SVC(kernel='linear')
y_pred_svm = cross_val_predict(algoritmo_svm, tfidf_matrix, y, cv=10)

# Evaluación del clasificador SVM
precision_svm = precision_score(y, y_pred_svm, average='macro', zero_division=0)
recall_svm = recall_score(y, y_pred_svm, average='macro', zero_division=0)
f1_svm = f1_score(y, y_pred_svm, average='macro', zero_division=0)

```

Ejecución:

Matriz TF-IDF											
	00	01	02	03	04	...	ángel	él	época	últimos	única
0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.000000	0.0	0.0	0.0
1	0.0	0.0	0.258841	0.0	0.0	...	0.0	0.000000	0.0	0.0	0.0
2	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.000000	0.0	0.0	0.0
3	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.000000	0.0	0.0	0.0
4	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.000000	0.0	0.0	0.0
..
65	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.000000	0.0	0.0	0.0
66	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.163604	0.0	0.0	0.0
67	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.000000	0.0	0.0	0.0
68	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.000000	0.0	0.0	0.0
69	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.000000	0.0	0.0	0.0
[70 rows x 1160 columns]											

```
Presicion del modelo Naive Bayes: 0.35714285714285715  
Recall del modelo Naive Bayes: 0.5  
F1-score del modelo Naive Bayes: 0.4166666666666667  
Presicion del modelo SVM: 0.35714285714285715  
Recall del modelo SVM: 0.5  
F1-score del modelo SVM: 0.4166666666666667
```