

CENTRO DE ENSEÑANZA TÉCNICA INDUSTRIAL



**Análisis de Sentimientos por Consola con Dataset de Hugging Face usando
Python**

Proyecto

Victor Alejandro Dominguez Cruz

21310228

6°G

Visión Artificial

15 de junio de 2025

1. Objetivos del Proyecto

- Diseñar un sistema capaz de clasificar frases en inglés según su sentimiento: positivo o negativo.
- Utilizar un dataset de Hugging Face (sst2)
- Entrenar un modelo de aprendizaje automático.
- Crear una aplicación por consola que permita ingresar frases y recibir una clasificación de sentimiento.

2. Explicación del Proyecto

Este proyecto implementa un sistema de análisis de sentimientos mediante aprendizaje automático. Se basa en el dataset "Stanford Sentiment Treebank" (sst2) disponible en Hugging Face. Se empleó un vectorizador TfidfVectorizer para convertir texto en vectores numéricos y un clasificador LogisticRegression para predecir sentimientos.

La salida se limita a dos clases:

- Positivo (1)
- Negativo (0)

El sistema final consiste en:

1. Un código py para entrenamiento (entrenar_modelo_sentimientos.py).
2. Un código py para prueba por consola (analisis_sentimientos_consola.py).

3. Códigos Implementados

Código 1

Entrenar_modelo_sentimientos.py

```
from datasets import load_dataset

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import classification_report

import joblib


dataset = load_dataset("sst2")

train_texts = dataset["train"]["sentence"]

train_labels = dataset["train"]["label"]

test_texts = dataset["validation"]["sentence"]

test_labels = dataset["validation"]["label"]


vectorizer = TfidfVectorizer(max_features=5000)

X_train = vectorizer.fit_transform(train_texts)

X_test = vectorizer.transform(test_texts)


model = LogisticRegression(max_iter=1000)

model.fit(X_train, train_labels)
```

```
y_pred = model.predict(X_test)

print(classification_report(test_labels, y_pred))


joblib.dump(model, "modelo_sentimientos.pkl")

joblib.dump(vectorizer, "vectorizador.pkl")
```

Codigo 2

analisis_sentimientos_consola.py

```
import joblib


model = joblib.load("modelo_sentimientos.pkl")

vectorizer = joblib.load("vectorizador.pkl")


print("Análisis de Sentimientos (escribe 'salir' para terminar):")

while True:

    frase = input("> ")

    if frase.lower() == "salir":

        break

    X = vectorizer.transform([frase])

    pred = model.predict(X)[0]

    sentimiento = "Positivo" if pred == 1 else "Negativo"

    print(f"{frase}' => {sentimiento}\n")
```

4. Pruebas Realizadas

Se probaron frases como:

- > I love this
- 'I love this' => Positivo
- > I hate you
- 'I hate you' => Negativo
- > Your dog is beautiful
- 'Your dog is beautiful' => Positivo
- > My aunt is very ugly
- 'My aunt is very ugly' => Negativo
- > My dad is the best
- 'My dad is the best' => Positivo
- > You are dumb
- 'You are dumb' => Negativo
- > You are very intelligent
- 'You are very intelligent' => Positivo
- Precisión del modelo (según clasificación): alrededor del 90%

5. Errores y Soluciones

- Error: Todo se clasificaba como negativo con pocos datos.
 - Solución: Se entrenó el modelo usando el dataset completo sst2. Esto se agregó al documento entrenador
- Error: Palabras no reconocidas al usar frases nuevas.

- Solución: Se aumentó el vocabulario usando `max_features=5000` en el vectorizador.
- Error: "Archivo no encontrado" al ejecutar el script.
 - Solución: Se aseguró que los archivos `.pkl` estuvieran en la misma carpeta.

6. Conclusiones

El proyecto fue exitoso al lograr un sistema que analiza sentimientos de frases en tiempo real desde consola. El uso de un dataset real mejoró significativamente la precisión. Esta base puede ampliarse a una aplicación web, un bot o integrarse en procesos de atención al cliente automatizada.

7. Archivos Generados

- `entrenar_modelo_sentimientos.py`
- `analisis_sentimientos_consola.py`
- `modelo_sentimientos.pkl`
- `vectorizador.pkl`

The screenshot shows a Visual Studio Code editor window with a project named "66_21310228_Proyecto". The Explorer sidebar on the left lists the following files: `analisis_sentimientos_consola.py`, `entrenar_modelo_sentimientos.py`, `matriz_confusion.png`, `modelo_sentimientos.pkl`, and `vectorizador.pkl`. The main editor area displays the code for `analisis_sentimientos_consola.py`, which includes imports for `joblib` and `os`, and logic to load the `modelo_sentimientos.pkl` and `vectorizador.pkl` files. The TERMINAL panel at the bottom shows the execution of the script, displaying sentiment analysis results for various input phrases. The status bar at the bottom indicates the current line is 7, column 8, with 4 spaces, UTF-8 encoding, LF line endings, and the Python interpreter is set to 3.13.2.

```

1  import joblib
2
3  # Cargar modelo y vectorizador
4  model = joblib.load("modelo_sentimientos.pkl")
5  vectorizer = joblib.load("vectorizador.pkl")

```

```

> Your dog is beautiful
'Your dog is beautiful' => Positivo

> My aunt is very ugly
'My aunt is very ugly' => Negativo

> My dad is the best
'My dad is the best' => Positivo

> You are an asshole
'You are an asshole' => Positivo

> You are dumb
'You are dumb' => Negativo

> You are very intelligent
'You are very intelligent' => Positivo

> & C:/Users/vicdc/AppData/Local/Programs/Python/Python313/python.exe c:/Users/vicdc/Desktop/66_21310228_Proyecto/analisis_sentim
ientos_consola.py
'& C:/Users/vicdc/AppData/Local/Programs/Python/Python313/python.exe c:/Users/vicdc/Desktop/66_21310228_Proyecto/analisis_sentimi
entos_consola.py' => Positivo

> This is awful
'This is awful' => Negativo

> This is terrific
'This is terrific' => Positivo

```