



TECNOLOGIA SUPERIOR

BIG DATA E INTELIGENCIA DE NEGOCIO

Prácticas de Análisis de Sentimientos – Ciencia de Datos

WILLIAM ESTUARDO JIMÉNEZ MIGUEZ

william.jimenez@cenestur.edu.ec

Profesor(a): YADIRA FRANCO

yadira.franco@cenestur.edu.ec

Quito, Ecuador

2025

a. Introducción

En el presente proyecto se aplican técnicas de análisis de sentimientos y emociones a partir de la transcripción de contenido audiovisual proveniente de plataformas como YouTube, TikTok y entrevistas reales. El objetivo es identificar, clasificar y visualizar las emociones expresadas en testimonios, opiniones o relatos personales.

Para ello, se seleccionaron 10 videos de mínimo 2 minutos, cuyos audios fueron transcritos usando herramientas como Whisper, YouTube Transcript o Speech-to-Text de Google. Los textos fueron limpiados, divididos en frases y procesados mediante modelos de inteligencia artificial basados en Python (TextBlob, VADER, Transformers) para detectar emociones como alegría, tristeza, enojo, sorpresa y otras.

Finalmente, los resultados se visualizaron en gráficos (circular, línea temporal, heatmap y burbujas) y se documentaron en un informe técnico con hallazgos, patrones y conclusiones. Este trabajo demuestra cómo el procesamiento de lenguaje natural (NLP) puede aplicarse para analizar contenido emocional desde videos reales.

b. Objetivos

Objetivo Principal

Aplicar técnicas de análisis de sentimientos y emociones a partir de transcripciones de videos reales, mediante herramientas de procesamiento de lenguaje natural, con el fin de identificar patrones emocionales, visualizar resultados y generar un informe interpretativo con hallazgos relevantes.

Objetivos Específicos

- Seleccionar y transcribir el contenido hablado de al menos 10 videos de YouTube, TikTok, entrevistas o testimonios con duración mínima de 2 minutos.
- Limpiar y preparar los textos transcritos: corregir errores, dividir en frases coherentes y conservar expresiones emocionales.
- Aplicar modelos de análisis de sentimientos en Python utilizando librerías como TextBlob, VADER y Transformers, o mediante plataformas no-code.
- Clasificar las emociones expresadas en cada frase dentro de las categorías: positiva, negativa, neutra y opcionalmente: alegría, tristeza, enojo, sorpresa y miedo.
- Visualizar los resultados mediante gráficos estadísticos como barras, pastel, líneas, mapas de calor y burbujas.
- Redactar un informe técnico que sintetice los hallazgos, patrones emocionales y conclusiones del análisis realizado.

c. Desarrollo

- 1. Seleccionar 10 videos de YouTube, TikTok, entrevistas o testimonios (mínimo 2 minutos cada uno).**

Link de los videos:

- <https://www.youtube.com/watch?v=v4OlxfPug6E>
- https://www.youtube.com/watch?v=eUbtI_tskEg
- https://www.youtube.com/watch?v=BT1FDbJQs_0
- <https://www.youtube.com/watch?v=bIWv8T0iE98>
- <https://www.youtube.com/watch?v=DWxoCjI7di8>
- <https://www.youtube.com/watch?v=DJ1Qf74mfgw>
- <https://www.youtube.com/watch?v=JRSviGOwSUE>
- <https://www.youtube.com/watch?v=Ly9oMOaGmXI>
- <https://www.youtube.com/watch?v=1VpFJVmwCd8>

- 2. Extraer el texto hablado usando herramientas como YouTube Transcript, Whisper o Google Speech-to-Text.**

Se utiliza la herramienta Whisper y se adjunta los archivos.

- 3. Limpiar y preparar los textos: corregir errores, dividir en frases, conservar emociones.**

- **Corrección de errores**

Condición aplicada:

Se evitó aplicar corrección ortográfica automática con herramientas como TextBlob.correct() o autocorrecciones de modelos, porque solicitaste conservar la emoción y estructura del texto original.

Argumento:

- La corrección ortográfica puede suavizar o alterar expresiones emocionales (como "¡noooo!" → "¡no!").
- Para análisis de sentimientos y comprensión de tono emocional, es mejor preservar el lenguaje tal como fue escrito (incluso con errores).

- **División en frases**

Condición aplicada:

Se dividió el texto usando expresiones regulares, con el siguiente patrón:

python

CopiarEditar

`re.split(r'(?<=[.!?¡¿])\s+', texto)`

Argumento:

- Esta expresión reconoce delimitadores naturales del español: punto (.), signos de exclamación (¡!) e interrogación (¿?).
- No requiere librerías externas como nltk, lo que evita errores de descarga o compatibilidad.
- Se mantiene el contexto emocional por frase y no por párrafo completo.

Conservación de emociones

Condición aplicada:

Se conservó la redacción original del texto, sin filtrar, corregir o truncar expresiones.

Argumento:

- La emoción en el texto muchas veces se transmite por:
 - Repetición de letras (e.g. "¡Nooo!")
 - Uso informal de palabras
 - Signos de exclamación/interrogación repetidos

Cualquier normalización o corrección automática podría neutralizar estos elementos emocionales, afectando negativamente un posterior análisis de sentimientos o tono.

Etap	Herramienta usada	Condición aplicada	Justificación
Corrección errores	Ninguna (se omitió)	No se corrige ortografía	Preservar intenciones emocionales y estilo original
División en frases	Expresiones regulares	Separación por signos naturales en español	Precisión, compatibilidad, y claridad en análisis posterior
Conservación emoción	Lectura sin modificación	No se modifican ni normalizan expresiones afectivas	Permitir análisis semántico y afectivo más fiel

4. **Aplicar análisis de sentimientos usando Python (TextBlob, VADER, Transformers) o plataformas no-code.**
5. **Clasificar las emociones: positiva, negativa, neutral, y opcionalmente otras como enojo, tristeza, alegría.**
6. **Visualizar los resultados: gráficas de barras, pastel, nube de palabras, etc.**
7. **Redactar un informe con hallazgos, patrones y conclusiones.**

Análisis de emociones:

Emociones posibles con este modelo

- joy (alegría)
- anger (enojo)
- sadness (tristeza)
- fear (miedo)
- love (afecto)
- surprise (sorpresa)

Se adjunta el código desarrollado en Python:

```

# Librerías
import os
import re
import ssl
import pandas as pd
from docx import Document
from transformers import pipeline

# Ignorar verificación de certificados
ssl._create_default_https_context = ssl._create_unverified_context

# Cargar modelo de emociones
emotion_pipeline = pipeline("text-classification", model="j-hartmann/emotion-english-distilroberta-base",
top_k=1)

# Rutas
ruta_entrada = r"C:\DATA\PERSONAL\UNIVERSIDAD\5 NIVEL\HERRAMIENTAS PARA CIENCIA DE DATOS\PROYECTO\TEXTO"
ruta_salida = r"C:\DATA\PERSONAL\UNIVERSIDAD\5 NIVEL\HERRAMIENTAS PARA CIENCIA DE DATOS\PROYECTO\RESULTADOS"
archivo_docx = os.path.join(ruta_salida, "Limpieza_emociones.docx")
archivo_excel = os.path.join(ruta_salida, "Limpieza_emociones.xlsx")

# Leer archivo Word
def leer_docx(ruta_archivo):
    doc = Document(ruta_archivo)
    texto = ""
    for p in doc.paragraphs:
        texto += p.text.strip() + " "
    return texto.strip()

# Dividir en frases
def dividir_en_frases(texto):
    frases = re.split(r'(?=[.!?;:])', texto)
    return [f.strip() for f in frases if f.strip()]

# Documento Word
doc_salida = Document()
doc_salida.add_heading("Limpieza y Clasificación de Emociones", level=1)

# Lista de resultados
datos_excel = []

# Procesar archivos
for archivo in os.listdir(ruta_entrada):
    if archivo.endswith(".docx"):
        ruta_completa = os.path.join(ruta_entrada, archivo)
        texto = leer_docx(ruta_completa)
        frases = dividir_en_frases(texto)

        doc_salida.add_heading(f"Archivo: {archivo}", level=2)

        for i, frase in enumerate(frases, 1):
            resultado = emotion_pipeline(frase[:512])[0][0]
            emotion = resultado['label']
            score = round(resultado['score'], 3)

            doc_salida.add_paragraph(f"{i}. {frase}\n    > Emoción: {emotion} (confianza: {score})")
            datos_excel.append({
                "Archivo": archivo,
                "Frase": frase,
                "Emoción": emotion,
                "Confianza": score,
                "Indice": i
            })

# Guardar Word
doc_salida.save(archivo_docx)

# Guardar Excel con gráficos
df = pd.DataFrame(datos_excel)
with pd.ExcelWriter(archivo_excel, engine='xlsxwriter') as writer:
    df.to_excel(writer, sheet_name='Emociones', index=False)

    # Gráfico 2: Pastel
    resumen = df['Emoción'].value_counts().reset_index()
    resumen.columns = ['Emoción', 'Frecuencia']
    resumen.to_excel(writer, sheet_name='Resumen', index=False)
    wb = writer.book
    ws = writer.sheets['Resumen']

    pie = wb.add_chart({'type': 'pie'})
    pie.add_series({
        'name': 'Distribución de emociones',
        'categories': ['Resumen', 1, 0, len(resumen), 0],
        'values': ['Resumen', 1, 1, len(resumen), 1]
    })
    pie.set_title({'name': 'Gráfico circular de emociones'})
    ws.insert_chart('E2', pie)

    # Gráfico 3: Línea temporal
    df_linea = df[['Indice', 'Emoción']].copy()
    df_linea['EmociónIndex'] = df_linea['Emoción'].astype('category').cat.codes
    df_linea.to_excel(writer, sheet_name='Evolucion', index=False)
    ws3 = writer.sheets['Evolucion']
    linea = wb.add_chart({'type': 'line'})
    linea.add_series({
        'name': 'Evolución emocional',
        'categories': ['Evolucion', 1, 0, len(df_linea), 0],
        'values': ['Evolucion', 1, 2, len(df_linea), 2],
        'marker': {'type': 'circle'}
    })
    linea.set_title({'name': 'Emociones a lo largo del texto'})
    linea.set_x_axis({'name': 'Frase'})
    linea.set_y_axis({'name': 'Emoción (codificada)'})
    ws3.insert_chart('E2', linea)

    # Gráfico 4: Heatmap (tabla)
    heatmap_data = df.groupby(['Archivo', 'Emoción']).size().unstack(fill_value=0)
    heatmap_data.to_excel(writer, sheet_name='Heatmap')

    # Gráfico 5: Burbuja como scatter con tamaños
    burbujas = df.groupby('Emoción').agg({'Confianza': 'mean', 'Frase': 'count'}).reset_index()
    burbujas.columns = ['Emoción', 'ConfianzaMedia', 'Frecuencia']
    burbujas.to_excel(writer, sheet_name='Burbujas', index=False)
    ws5 = writer.sheets['Burbujas']

    bubble_chart = wb.add_chart({'type': 'scatter', 'subtype': 'bubble'})
    bubble_chart.add_series({
        'name': 'Emociones',
        'categories': ['Burbujas', 1, 0, len(burbujas), 0],
        'values': ['Burbujas', 1, 1, len(burbujas), 1],
        'bubble_sizes': ['Burbujas', 1, 2, len(burbujas), 2],
        'data_labels': {'value': True}
    })
    bubble_chart.set_title({'name': 'Gráfico de burbujas: Emoción'})
    bubble_chart.set_x_axis({'name': 'Emoción'})
    bubble_chart.set_y_axis({'name': 'Confianza promedio'})
    bubble_chart.set_legend({'position': 'bottom'})
    ws5.insert_chart('E2', bubble_chart)

print(f"Word generado: {archivo_docx}")
print(f"Excel con gráficos (2-5) generado: {archivo_excel}")

```

d. Funciones integradas

1. Lee archivos .docx desde la carpeta.
2. Limpia y divide el texto en frases.
3. Clasifica emociones con el modelo j-hartmann/emotion-english-distilroberta-base.
4. Genera un archivo Word con resultados detallados.
5. Genera un archivo Excel con:
 - Todas las frases con su emoción y confianza.
 - Una segunda hoja con un gráfico de barras que resume las emociones.

e. Hallazgos relevantes

Distribución General de Emociones (Gráfico Circular)

Las emociones más frecuentes fueron: joy, sadness y surprise. Se evidenció una tendencia positiva o neutra en la mayoría de los documentos. Emociones como anger y fear fueron minoritarias.

Evolución Emocional por Frase (Gráfico de Líneas)

Algunos documentos muestran picos de tristeza o sorpresa en secciones específicas. Otros presentan una progresión positiva, con un aumento de frases clasificadas como joy.

Heatmap: Emociones por Documento

Permite visualizar qué documentos presentan mayor carga emocional negativa o positiva. Un archivo en particular mostró una alta concentración de frases con sadness, lo cual podría relacionarse con un tema sensible o narrativo personal.

Gráfico de Burbujas: Confianza vs Frecuencia

Joy no solo fue frecuente, sino también tuvo una alta confianza media, indicando claridad emocional en las frases. Algunas emociones como surprise tuvieron alta frecuencia pero baja confianza, lo que sugiere ambigüedad emocional en la redacción.

d. Patrones detectados

- ✓ Frases cortas y exclamativas tienden a clasificarse como joy o surprise.
- ✓ Oraciones largas, con negaciones o reflexiones suelen asociarse a sadness.
- ✓ La emoción love apareció en contextos con lenguaje afectivo o enfático ("me encanta", "es maravilloso").

f. Conclusiones

- ✓ Existe una predominancia emocional positiva en los textos analizados, lo cual podría estar relacionado con su propósito (comunicativo, educativo, reflexivo).
- ✓ Los documentos presentan variaciones emocionales internas, detectables a través del análisis secuencial de frases.
- ✓ El modelo de clasificación ha permitido una evaluación automatizada precisa, útil para análisis de tono, percepción emocional y evaluación de estilo discursivo.

g. Recomendaciones

- ✓ Aplicar este tipo de análisis en documentos de opinión, encuestas o retroalimentaciones para entender el tono emocional.
- ✓ En contextos educativos o psicológicos, este enfoque puede servir como herramienta para detectar estados afectivos reflejados en la escritura.
- ✓ En textos con alta carga negativa, considerar estrategias de mejora del contenido o revisión de contexto.

h. Entregables

TIPO	CONTENIDO	ARCHIVO
Word	Frases + emoción y nivel de confianza	Limpieza_emociones.docx
Excel	Datos y gráficos	Limpieza_emociones.xlsx
Gráficos	Circular, línea, heatmap, burbujas	Incluidos en el Excel

GIBHUB: <https://github.com/Willyejm/AnalsisSentimientos.git>