

# Python

## Procesamiento de Lenguaje Natural

### Análisis de Sentimientos



Rogelio Ferreira Escutia

Profesor / Investigador  
Tecnológico Nacional de México  
Campus Morelia



# Análisis de Sentimientos

# Análisis de Sentimientos

- El "Análisis de Sentimientos" es el procedimiento para identificar, extraer y cuantificar los estados afectivos de las personas ó grupos de personas, utilizando el "Procesamiento del Lenguaje Natural", el "Análisis del Texto" y la "Lingüística Computacional".
- Su rango de aplicación es muy grande ya que se puede aplicar para determinar la situación emocional de los clientes en cualquier proceso de servicios y ventas así como el de clasificar la intención de cualquier texto.

# Aplicaciones

- **Algunas de las aplicaciones actuales mas usadas son:**
  - **Análisis de las experiencias de usuarios con un producto o servicio.**
  - **Análisis de discursos políticos.**
  - **Análisis de artículos y libros.**

# Polaridad

- **La forma más fácil de hacer un "Análisis de Sentimientos" es detectando la "polaridad" de una palabra, frase ó texto, y se pueden clasificar en 3:**
  - **Positivo.**
  - **Negativo.**
  - **Neutral.**

# Técnicas

- **Existen varias técnicas para hacer estos análisis y cada día surgen mas debido a la necesidad y cantidad de casos donde se puede aplicar.**
- **Básicamente podríamos tener la siguiente clasificación de técnicas:**
  - **Manual: Generando listas de palabras y/o frases positivas para identificarlas.**
  - **Automática: Usando bibliotecas que ya incluyen sus algoritmos propios para hacer el análisis.**

# Ejemplo:

- **Analizar un texto para determinar su polaridad (positivo, negativo ó neutral).**
- **Lo primero es definir el texto a analizar, para este ejemplo se utilizará el libro completo de “Don Quijote de la Mancha” de Miguel de Cervantes Saavedra (de dominio público).**

# Análisis de Sentimientos (1):

- **Bibliotecas a utilizar:**

```
# Bibliotecas a utilizar
import re                # Manejo de expresiones regulares
import nltk              # Para procesar lenguaje natural
from nltk.tokenize import word_tokenize # Para Tokenizar un texto
from nltk.corpus import stopwords    # Cargar las "Stopwords" del español
```



# Análisis de Sentimientos (2):

- Cargar a memoria el archivo de texto del libro de “Don Quijote de la Mancha” que descargamos previamente y lo convertimos todo a minúsculas (por las comparaciones que se harán más adelante):

```
# Cargar archivo de texto a memoria
with open('texto_001_don_quijote.txt', 'r') as archivo_en_memoria:
    texto = archivo_en_memoria.read()

# Convertir todo el texto a minúsculas
texto_minusculas = texto.lower()
```

# Análisis de Sentimientos (3):

- Eliminamos caracteres y símbolos no deseados y “tokenizamos”:

```
# Eliminar símbolos y caracteres especiales usando expresiones regulares
texto_sin_simbolos = re.sub(r'[^w\s]', '', texto_minusculas)

# Convertimos a tokens todo el texto y lo imprimimos en pantalla
tokens_de_mi_texto = word_tokenize(texto_sin_simbolos)
print('\nTokens Totales: ', len(tokens_de_mi_texto))
```

# Análisis de Sentimientos (4):

- Cargamos las “stopwords” y filtramos el texto:

```
# Cargamos las "stopwords" del español (las palabras que no nos aportan información)
palabras_vacias = set(stopwords.words('spanish'))

# Filtramos los tokens eliminando las "stopwords"
lista_final = []
for palabra in tokens_de_mi_texto:
    if palabra not in palabras_vacias:
        lista_final.append(palabra)
```

# Análisis de Sentimientos (5):

- Imprimimos la cantidad de “tokens” ya sin “stopwords”:

```
# Impresión total de Tokens sin StopWords  
print("Total de Tokens sin Stopwords: ", len(lista_final))
```

- Hasta este momento se observará en pantalla lo siguiente:

```
Tokens Totales: 187009  
Total de Tokens sin Stopwords: 91055
```

# Análisis de Sentimientos (6):

- Construimos 2 listas, una de palabras conocidas comúnmente como positivas y otra para las negativas. En este ejemplo se hicieron las 2 listas con 5 elementos cada una:

```
# Lista de palabras positivas y negativas en español
palabras_positivas = ['bueno', 'encantado', 'bonito', 'increíble', 'superior']
palabras_negativas = ['malo', 'decepcionado', 'feo', 'terrible', 'inferior']
```

- Nuestro contador de palabras positivas y negativas lo inicializamos a cero:

```
# Inicializar a cero las cantidades positivas y negativas
numero_de_positivas = numero_de_negativas = 0
```

# Análisis de Sentimientos (7):

- Ahora buscamos si dentro de nuestro libro hay alguna coincidencia de alguna palabra que está en nuestra lista de palabras positivas y negativas, y si es así, incrementamos su respectivo contador:

```
# Buscamos las palabras positivas y negativas dentro del texto
for elemento in lista_final:
    if elemento in palabras_positivas:
        numero_de_positivas = numero_de_positivas + 1
    if elemento in palabras_negativas:
        numero_de_negativas = numero_de_negativas + 1
```

# Análisis de Sentimientos (8):

- Ya solo falta imprimir el total de palabras positivas y negativas y la polaridad del texto:

```
# Impresión de resultados
print("\nNúmero de palabras positivas encontradas: ", numero_de_positivas)
print("Número de palabras negativas encontradas: ", numero_de_negativas)
print("\nPolaridad del texto:")
polaridad = numero_de_positivas - numero_de_negativas
if polaridad>0: print("Positiva\n")
elif polaridad<0: print("Negativa\n")
else: print("Neutral\n")
```

- Y la salida es:

```
Número de palabras positivas encontradas: 100
Número de palabras negativas encontradas: 25

Polaridad del texto:
Positiva
```



## Rogelio Ferreira Escutia

Profesor / Investigador  
Tecnológico Nacional de México  
Campus Morelia



[rogelio.fe@morelia.tecnm.mx](mailto:rogelio.fe@morelia.tecnm.mx)



[rogeplus@gmail.com](mailto:rogeplus@gmail.com)



[xumarhu.net](http://xumarhu.net)



[@rogeplus](https://twitter.com/rogeplus)



[https://www.youtube.com/  
channel/UC0on88n3LwTKxJb8T09sGjg](https://www.youtube.com/channel/UC0on88n3LwTKxJb8T09sGjg)



[rogelioferreiraescutia](https://www.linkedin.com/in/rogelioferreiraescutia)

