

SENTIMENT ANALYSIS WITH PROBABILITY SCORES FROM COMMENTS USING PYTHON WITH PYSENTIMENTO

MARÍA FERNANDA RUBÍ EGUEZ
22/08/2025

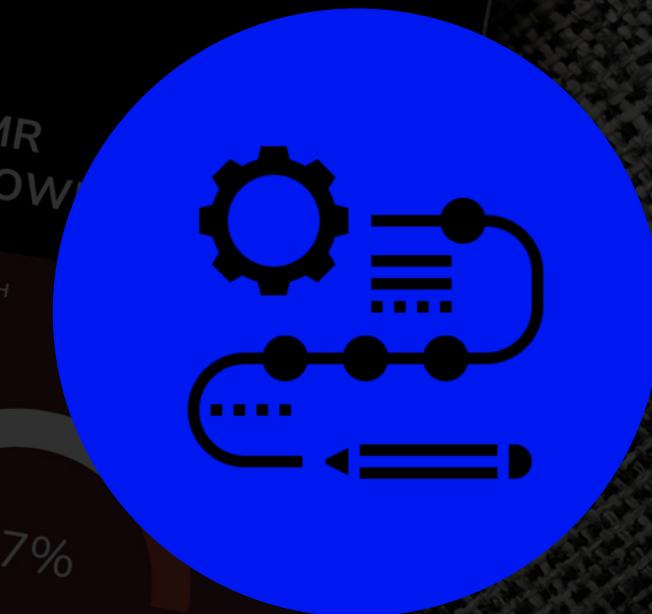
TOOLS AND LIBRARIES:
PYTHON, PANDAS, MATPLOTLIB,
SEABORN, PYSENTIMENTO

INTRODUCTION TO THE PROBLEM



GOAL

Sentiment Analysis with Probability Scores from Comments using Python with Pysentimiento



METHOD

- Analyze Data
- Get Charts

TOOLS USED

- Python (Jupyter Notebook): For data analysis, predictive modeling, and visualization in an interactive environment.
- Pandas: For structuring, cleaning, and manipulating tabular data efficiently using DataFrames.
- Pysentimiento: For performing sentiment analysis and text classification (NLP) to interpret customer opinions and reviews.
- Matplotlib: For building customizable charts such as line, bar, histogram, and scatter plots.
- Seaborn: For creating professional and statistical visualizations (heatmaps, boxplots, regression plots) with an elegant style.

Sentiment analysis (positive, negative, neutral)

```
#Import the Library
from pysentimiento import create_analyzer
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

#Create a sentiment analyzer in Spanish
analyzer = create_analyzer(task="sentiment", lang="es")

#Comments list
comments = [
    "El servicio fue excelente y rápido.",
    "Estoy satisfecho con la entrega.",
    "Me encantó la atención que recibí.",
    "No volvería a comprar aquí, fue una mala experiencia.",
    "Todo fue perfecto, superó mis expectativas.",
    "El envío llegó en buen estado, excelente.",
    "Buena calidad y buen precio.",
    "El soporte técnico no resolvió mi problema.",
    "Estoy muy feliz con mi compra.",
    "No me gustó para nada el servicio al cliente."
]
```

DATA EXPLORATION AND STRUCTURE

Data source:
Simulated comment data

Variables:

- Different comments, including positive, negative, and neutral

MODEL DEVELOPMENT

To analyze the comments in the code, we use the analyzer.predict method.

This allows us to evaluate the context of each comment and determine whether it is positive or negative, storing the result in a sentiment column.

In addition, we calculate a probability score ranging from 0 to 1. The Probability column reflects the model's confidence in the predicted class. Values close to 1 indicate high confidence in the prediction, while values near 0 indicate low confidence.

```
#Sentiment analysis
results = []
for comment in comments:
    result = analyzer.predict(comment)
    results.append({
        "Comment": comment,
        "Feelings": result.output,
        "Prob_POS": round(result.probas["POS"], 2),
        "Prob_NEG": round(result.probas["NEG"], 2),
        "Probability": round(result.probas[result.output], 2)
    })
#Create the DataFrame
df_results = pd.DataFrame(results)

#Show result
print("🔍 Sentiment Analysis (BERT model in Spanish):")
display(df_results)
```

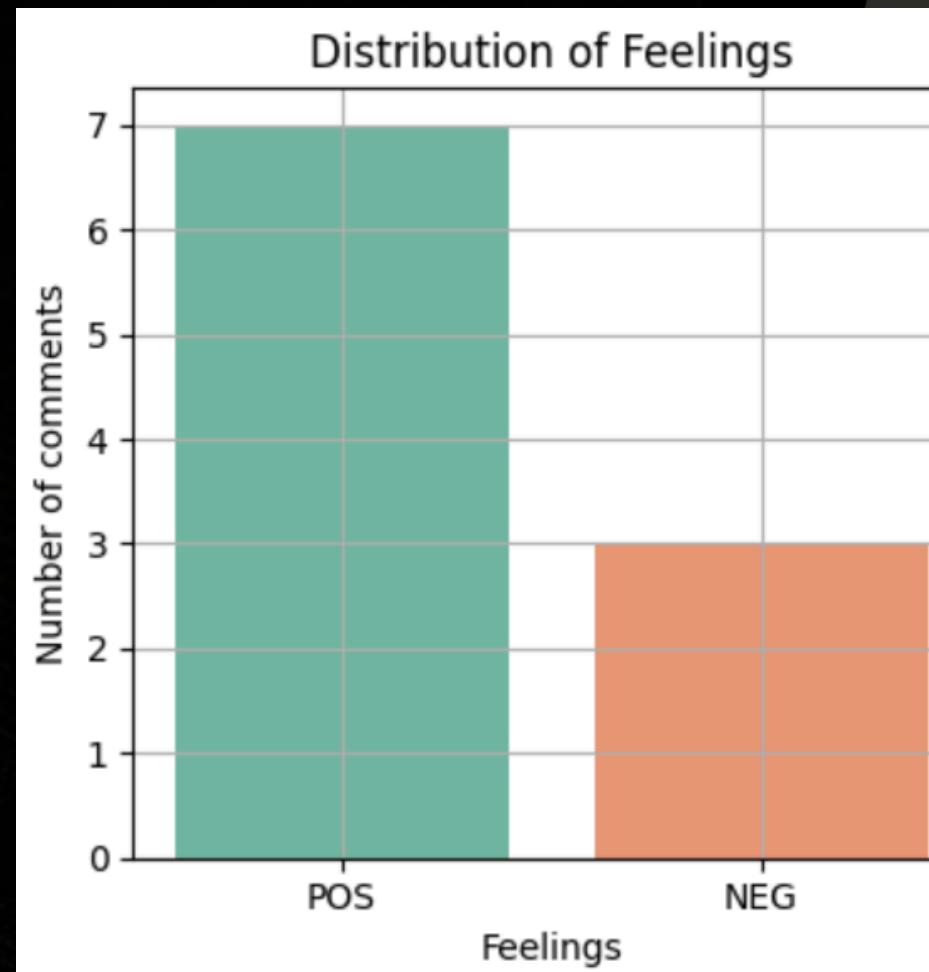
✓ 1.0s

	Comment	Feelings	Prob_POS	Prob_NEG	Probability
0	El servicio fue excelente y rápido.	POS	0.96	0.00	0.96
1	Estoy satisfecho con la entrega.	POS	0.94	0.00	0.94
2	Me encantó la atención que recibí.	POS	0.97	0.00	0.97
3	No volvería a comprar aquí, fue una mala exper...	NEG	0.02	0.87	0.87
4	Todo fue perfecto, superó mis expectativas.	POS	0.94	0.01	0.94
5	El envío llegó en buen estado, excelente.	POS	0.89	0.01	0.89
6	Buena calidad y buen precio.	POS	0.88	0.01	0.88
7	El soporte técnico no resolvió mi problema.	NEG	0.02	0.76	0.76
8	Estoy muy feliz con mi compra.	POS	0.95	0.00	0.95
9	No me gustó para nada el servicio al cliente.	NEG	0.01	0.96	0.96

MODEL EVALUATION

A bar chart visualization was created to compare the number of positive comments versus the number of negative comments.

The results show that 70% of the comments are positive and 30% are negative, meaning that there are more than twice as many positive comments (2.33 times) compared to negative ones.



INDUSTRY BACKGROUND

```
#Percentage of positive and negative comments
pcg = df_results["Feelings"].value_counts(normalize=True) * 100
print(pcg)

#Store in variables
pos_percent = pcg.get("POS", 0)
neg_percent = pcg.get("NEG", 0)

print(f"Percentage of positive comments: {pos_percent:.2f}%")
print(f"Percentage of negative comments: {neg_percent:.2f}%")

#Ratio of positive to negative comments
if neg_percent > 0:
    ratio = pos_percent / neg_percent
    print(f"POS/NEG ratio: {ratio:.2f}")
else:
    print("There are no negative comments, so the ratio cannot be calculated.")

Feelings
POS    70.0
NEG    30.0
Name: proportion, dtype: float64
Percentage of positive comments: 70.00%
Percentage of negative comments: 30.00%
POS/NEG ratio: 2.33
```

GROUND

THE INDUSTRY'S HISTORY

I WANT TO SAY

THANK YOU

FOR YOUR ATTENTION

