PLN: Examen 1: Minería de textos para Turismo

Y. Jerahi García Zúñiga

Enlaces

```
!pwd
```

```
/content
```

```
!mkdir -p ~/nltk_data/corpora/stopwords
!mkdir -p ~/nltk_data/tokenizers/punkt
!mkdir -p ~/nltk_data/taggers/averaged_perceptron_tagger
!mkdir -p ~/nltk_data/chunkers/maxent_ne_chunker
!mkdir -p ~/nltk_data/corpora/wordnet
!mkdir -p ~/nltk_data/corpora/omw-1.4
!ls ~/nltk_data/
```

```
chunkers  corpora  tokenizers  taggers
```

```
!pip install -q nltk spacy textblob
!python -m spacy download en_core_web_sm
!python -m textblob.download_corpora
!pip install -q matplotlib seaborn wordcloud
from nltk.corpus import stopwords, wordnet
from nltk.tokenize import word_tokenize, sent_tokenize
from nltk.stem import WordNetLemmatizer
import nltk, spacy, textblob, os, sys, pandas as pd, numpy as np, matplotlib.pyplot as plt, seaborn as sns, wordcloud
print("✅ Todo instalado y listo para usar.")
```

```
✅ Todo instalado y listo para usar.
```

```
!wget -q -O ~/nltk_data/corpora/stopwords.zip https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/packages/corpora/stopwords.zip
!unzip -q -o ~/nltk_data/corpora/stopwords.zip -d ~/nltk_data/corpora/
!ls ~/nltk_data/corpora/stopwords | head -5
```

```
arabic
azerbaijani
danish
dutch
English
```

```
# Descargar los stopwords de NLTK (si aún no están)
import nltk, os, sys, urllib.request, zipfile
nltk.download('stopwords', quiet=True)
# Verificar que se descargaron
from nltk.corpus import stopwords
print("✅ Stopwords en español:", len(stopwords.words('spanish')))
print("✅ Stopwords en inglés:", len(stopwords.words('english')))
```

```
✅ Stopwords en español: 313
✅ Stopwords en inglés: 179
```

```
# Descargar los stopwords de NLTK (si aún no están)
import nltk, os, sys, urllib.request, zipfile
nltk.download('stopwords', quiet=True)
# Verificar que se descargaron
from nltk.corpus import stopwords
print("✅ Stopwords en español:", len(stopwords.words('spanish')))
print("✅ Stopwords en inglés:", len(stopwords.words('english')))
```

```
✅ Stopwords en español: 313
✅ Stopwords en inglés: 179
```

```
# Crear carpeta de datos
!mkdir -p data
# Descargar el archivo CSV de reseñas de Airbnb
!wget -q -O data/airbnb_reviews.csv https://raw.githubusercontent.com/datasets/airbnb-listings/master/data/airbnb-listings.csv
# Verificar que se descargó
!ls -lh data/
```

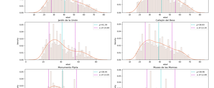
```
total 2.4M
-rw-r--r-- 1 root root 2.4M Oct 30 00:00 airbnb_reviews.csv
```

```
import pandas as pd
df = pd.read_csv('data/airbnb_reviews.csv', sep=';')
print("✅ Dataset cargado:", df.shape)
df.head(2)
```

```
✅ Dataset cargado: (20000, 106)
```

| | Scrape ID | Last scraped | Name | Summary | Space | Description | Experiences offered | Neighborhood overview | Notes | Transit | Access | House rules | Thumbnail url | Medium url | Picture url | XL picture url | Host ID | Host URL | Host name | Host since | Host location | Host about | Host response time | Host response rate | Host acceptance rate | Host is superhost | Host thumbnail url | Host picture url | Host neighbourhood | Host listings count | Host total listings count | Host verifications | Host identity verified | City | City portal | Neighborhood | City district | Zipcode | State | Market | Smart location | Country code | Country | Property type | Room type | Accommodates | Bathrooms | Bedrooms | Beds | Bed type | Amenities | Square feet | Price | Weekly price | Monthly price | Security deposit | Cleaning fee | Guests included | Extra people | Minimum nights | Maximum nights | Calendar updated | Has availability | Availability 30 | Availability 60 | Availability 90 | Availability 365 | Calendar last scraped | First review | Last review | Requires license | License | Jurisdiction names | Instant bookable | Cancellation policy | Require guest profile picture | Require guest phone verification | Calculated host listings count | Reviews per month | Geolocation |
|---|---|---

```
[0]: 2,
[1]: 2,
[2]: 2,
[3]: 2,
[4]: 2,
[5]: 2,
[6]: 2,
[7]: 2,
[8]: 2,
[9]: 2,
[10]: 2,
[11]: 2,
[12]: 2,
[13]: 2,
[14]: 2,
[15]: 2,
[16]: 2,
[17]: 2,
[18]: 2,
[19]: 2,
```



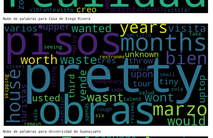

```
# Análisis comparativo
print("\n" + "="*80)
print("ANÁLISIS COMPARATIVO")
print("="*80)
# Para cada dataset, mostrar estadísticas por categoría
for dataset_name, df in [('Dataset 1', df1), ('Dataset 2', df2)]:
    print(f"\n{dataset_name}:")
    print("-"*40)
    for cat in sorted(df['categoria'].unique):
        cat_data = df[df['categoria'] == cat]
        print(f"\{cat}:")
        print(f"  Muestras: {len(cat_data)}")
        print(f"  Sentimiento promedio: {cat_data['sentiment'].mean:.3f}")
        print(f"  Desviación estándar: {cat_data['sentiment'].std:.3f}")
        print(f"  Mediana: {cat_data['sentiment'].median:.3f}")
        print(f"  Mínimo: {cat_data['sentiment'].min:.3f}")
        print(f"  Máximo: {cat_data['sentiment'].max:.3f}")
        print(f"  Positivos: {(cat_data['sentiment'] > 0).sum} ({(cat_data['sentiment'] > 0).mean*100:.1f}%)")
        print(f"  Neutros: {(cat_data['sentiment'] = 0).sum} ({(cat_data['sentiment'] = 0).mean*100:.1f}%)")
        print(f"  Negativos: {(cat_data['sentiment'] < 0).sum} ({(cat_data['sentiment'] < 0).mean*100:.1f}%)")
# Crear visualizaciones comparativas
fig, axes = plt.subplots(2, 2, figsize=(15, 12))
for idx, (dataset_name, df) in enumerate(zip(['Dataset 1', 'Dataset 2'], [df1, df2])):
    ax = axes[idx]
    # Boxplot por categoría
    categorias = df['categoria'].unique
    data_boxplot = [df[df['categoria'] == cat]['sentiment'] for cat in categorias]
    bp = ax.boxplot(data_boxplot, labels=categorias, patch_artist=True)
    ax.set_title(f'{dataset_name}\n Por Categoría')
    ax.set_ylabel('Sentimiento')
    ax.set_xlabel('Categoría')
    ax.grid(True, alpha=0.3)
    # Histograma por categoría
    ax = axes[idx + 2]
    for cat in categorias:
        cat_data = df[df['categoria'] == cat]['sentiment']
        ax.hist(cat_data, bins=20, alpha=0.5, label=cat)
    ax.set_title(f'{dataset_name}\n Distribución')
    ax.set_ylabel('Frecuencia')
    ax.set_xlabel('Sentimiento')
    ax.legend
    ax.grid(True, alpha=0.3)
plt.tight_layout
plt.savefig('comparacion_categorias.png', dpi=300, bbox_inches='tight')
plt.show
```

```
# Análisis temporal (si hay datos)
print("\n" + "="*80)
print("ANÁLISIS TEMPORAL")
print("="*80)
for dataset_name, df in [('Dataset 1', df1), ('Dataset 2', df2)]:
    if 'fecha' in df.columns:
        print(f"\n{dataset_name}:")
        print("-"*40)
        df['fecha'] = pd.to_datetime(df['fecha'], errors='coerce')
        df['mes'] = df['fecha'].dt.month
        df['año'] = df['fecha'].dt.year
        for cat in sorted(df['categoria'].unique):
            cat_data = df[df['categoria'] == cat]
            print(f"\{cat}:")
            print(f"  Sentimiento promedio por mes:")
            monthly = cat_data.groupby('mes')['sentiment'].mean
            for mes, sent in monthly:
                print(f"    Mes {mes}: {sent:.3f}")
# Crear visualización temporal
fig, axes = plt.subplots(2, 2, figsize=(15, 12))
for idx, (dataset_name, df) in enumerate(zip(['Dataset 1', 'Dataset 2'], [df1, df2])):
    if 'fecha' in df.columns:
        df['fecha'] = pd.to_datetime(df['fecha'], errors='coerce')
        df['mes'] = df['fecha'].dt.month
        df['año'] = df['fecha'].dt.year
        ax = axes[idx]
        for cat in df['categoria'].unique:
            cat_data = df[df['categoria'] == cat]
            monthly = cat_data.groupby('mes')['sentiment'].mean
            ax.plot(monthly.index, monthly.values, marker='o', label=cat)
        ax.set_title(f'{dataset_name}\n Por Mes')
        ax.set_ylabel('Sentimiento')
        ax.set_xlabel('Mes')
        ax.legend
        ax.grid(True, alpha=0.3)
plt.tight_layout
plt.savefig('analisis_temporal.png', dpi=300, bbox_inches='tight')
plt.show
```

```
# Análisis de palabras clave
print("\n" + "="*80)
print("ANÁLISIS DE PALABRAS CLAVE")
print("="*80)
from collections import Counter
from wordcloud import WordCloud
for dataset_name, df in [('Dataset 1', df1), ('Dataset 2', df2)]:
    print(f"\n{dataset_name}:")
    print("-"*40)
    for cat in sorted(df['categoria'].unique):
        cat_data = df[df['categoria'] == cat]
        print(f"\{cat}:")
        print(f"  Longitud promedio: {cat_data['text_length'].mean:.1f} caracteres")
        print(f"  Longitud mediana: {cat_data['text_length'].median:.1f} caracteres")
# Crear nuevo dataset con todos los datos
df_combined = pd.concat([df1, df2], ignore_index=True)
df_combined['dataset'] = ['Dataset 1'] * len(df1) + ['Dataset 2'] * len(df2)
# Guardar
df_combined.to_csv('datasets_combinados.csv', index=False)
print("\n✓ Datasets combinados guardados en: datasets_combinados.csv")
```

```
"Ya que todas las cosas que pasaron hasta ahora no tienen ningún sentido, tal vez el tiempo mismo sea un error, tal vez el tiempo mismo sea una mentira, tal vez el tiempo mismo sea el culpable de todo lo que ha pasado y de todo lo que no ha pasado. Tal vez el tiempo mismo sea el responsable de todo lo que he hecho y de todo lo que no he hecho. Tal vez el tiempo mismo sea el responsable de todo lo que he sido y de todo lo que no he sido. Tal vez el tiempo mismo sea el responsable de todo lo que voy a hacer y de todo lo que no voy a hacer. Tal vez el tiempo mismo sea el responsable de todo lo que voy a ser y de todo lo que no voy a ser. Tal vez el tiempo mismo sea el responsable de todo lo que voy a decir y de todo lo que no voy a decir. Tal vez el tiempo mismo sea el responsable de todo lo que voy a pensar y de todo lo que no voy a pensar. Tal vez el tiempo mismo sea el responsable de todo lo que siento y de todo lo que no siento
```

Nube de palabras para Casa de Diego Ribera


piso
obras
pety
months
years
marzo
bien
worth
house
wasnt
would
unknown
wanted
waste
lee
usted
upon
tiny
tour
tomar
creo
varios
upper
vendedor
tarde
year
viva
viaje
viajar
viajando
viajando
viajando