Procesamiento del Lenguaje Natural

Rodrigo S. Cortez Madrigal

Actividad Sumativa 2

Agregue al corpus más ejemplos de tweets positivos y negativos, por lo menos el doble. Observe si hay algún cambio en el score y explique, con base en los ejemplos agregados, por qué existe ese cambio?

In [1]:
```python
import numpy as np
import pandas as pd
import re
import spacy
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

import plotly
from plotly import graph_objs as go
from plotly import express as px
from plotly.subplots import make_subplots

import spacy

plotly.offline.init_notebook_mode(connected=False)
```

In [2]:
```python
# Cargar tweets

tweets = pd.read_csv('tweets.csv', encoding='utf-8', index_col=0)
```

In [3]:
```python
tweets
```

Out[3]:

	Datetime	Tweet Id	Text	Username	sentiment	sentiment_score	emotion	emotion_score
0	2022-09-30 23:29:15+00:00	1575991191170342912	@Logitech @apple @Google @Microsoft @Dell @Len...	ManjuSreedaran	neutral	0.853283	anticipation	0.587121
1	2022-09-30 21:46:35+00:00	1575965354425131008	@MK_habit_addict @official_stier @MortalKombat...	MiKeMcDnet	neutral	0.519470	joy	0.886913
2	2022-09-30 21:18:02+00:00	1575958171423752203	As @CRN celebrates its 40th anniversary, Bob F...	jfollett	positive	0.763791	joy	0.960347
3	2022-09-30 20:05:24+00:00	1575939891485032450	@dell your customer service is horrible especi...	daveccarr	negative	0.954023	anger	0.983203
4	2022-09-30 20:03:17+00:00	1575939359160750080	@zacokalo @Dell @DellCares @Dell give the man ...	heycamella	neutral	0.529170	anger	0.776124
...	...	...	...	...	...	...	...	...
24965	2022-01-01 02:02:04+00:00	1477097760931336198	@ElDarkAngel2 @GamersNexus @Dell I wouldn't ev...	Eodart	negative	0.682981	anger	0.906309
24966	2022-01-01 01:57:34+00:00	1477096631300415496	@kite_real @GamersNexus @Dell I didn't really ...	Eodart	positive	0.743940	joy	0.951701
24967	2022-01-01 01:36:36+00:00	1477091355629432833	Hey @JoshTheFixer here it is....27 4K UHD USB-...	Corleone250	neutral	0.654463	anticipation	0.471185
24968	2022-01-01 01:31:30+00:00	1477090070830141442	@bravadogaming @thewolfpena @Alienware @intel ...	MrTwistyyy	neutral	0.794049	anticipation	0.747014
24969	2022-01-01 00:59:37+00:00	1477082048900726784	@rabia_ejaz @Dell Stopped buying windows lapto...	IDevourNehari	positive	0.733861	joy	0.958346

24970 rows × 8 columns

In [4]:
```python
tweets = tweets[['Text', 'sentiment']]
tweets = tweets.sample(n=2000, random_state=1)
```

In [5]:
```python
tweets
```

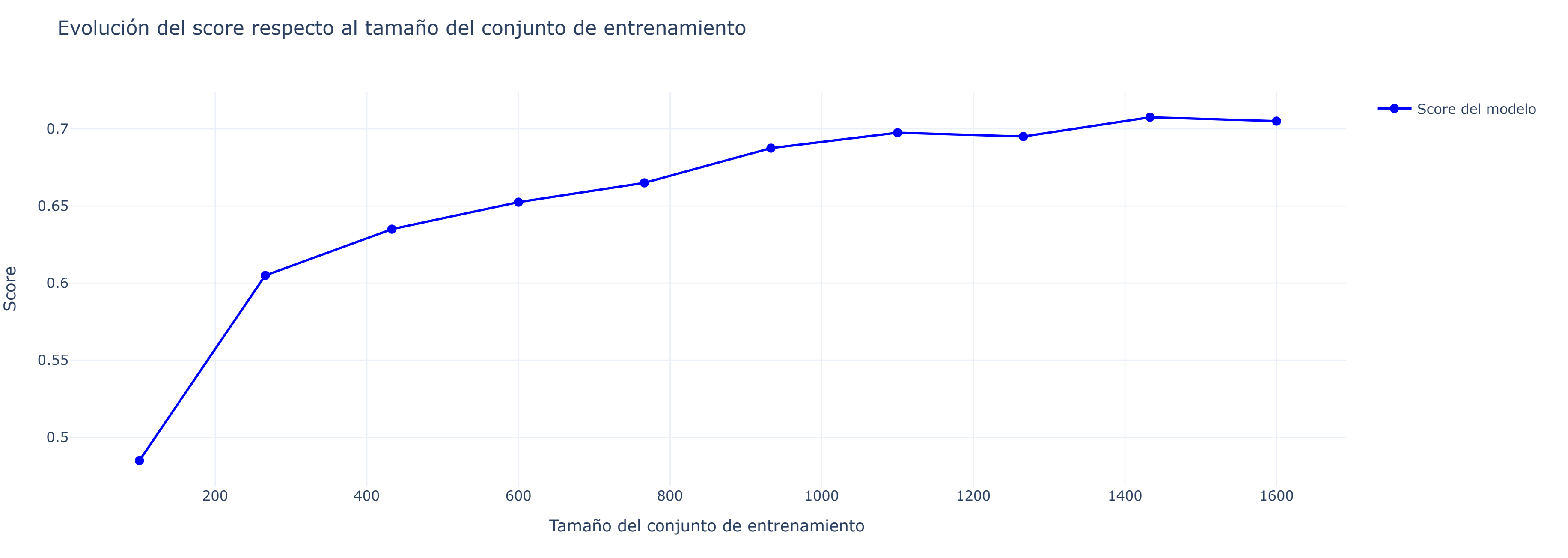
Out[5]:

	Text	sentiment
17359	well....i got approved for @Dell financing tod...	positive
5518	@Dell like a seasoned dell dude 20 years into ...	positive
11523	@shannonrwatts @HeatherThomasAF @HP @Oracle @A...	negative
22196	@Dell you should consider making heaters. Your...	negative
3510	#facebookdown @TheRock @steveaustinBSR @WWEUn...	neutral
...	...	...
20898	@sunnykaushal54 @MichaelDell @Dell Same proble...	negative
15702	@GrowlyBiteBite @AnsgarTOdinson @Alienware @Al...	positive
6398	Couple months later @Dell was running a contes...	neutral
18466	@Dell - my new computer blows. It's slower tha...	negative
3085	@PakmanFPS @intelcanada @Dell Big moves Pak bi...	positive

2000 rows × 2 columns

In [6]:
```python
nlp = spacy.load("es_core_news_sm")

tuits_train, tuits_test = train_test_split(tweets, test_size=0.2, random_state=42)

X_train = tuits_train['Text'].to_numpy()
y_train = tuits_train['sentiment'].to_numpy()
X_test = tuits_test['Text'].to_numpy()
y_test = tuits_test['sentiment'].to_numpy()

def preprocess_docs(docs, nlp):

    processed_docs = []
    for doc in docs:
        spacy_doc = nlp(doc)
        tokens = [token.text for token in spacy_doc if not token.is_stop and not token.is_punct]
        processed_docs.append(' '.join(tokens))
    return processed_docs
```

In [7]:
```python
def eval(X_train, y_train, X_test, y_test, nlp, vectorizer, modelo, t=10):
    scores = []
    tamaños = np.linspace(100, len(X_train), num=t, dtype=int)

    for tamaño in tamaños:
        # Ajustar el tamaño del conjunto de entrenamiento
        subset_X_train = X_train[:tamaño]
        subset_y_train = y_train[:tamaño]

        # Preprocesar y vectorizar
        X_train_transformed = vectorizer.fit_transform(preprocess_docs(subset_X_train, nlp))
        X_test_transformed = vectorizer.transform(preprocess_docs(X_test, nlp))

        # Entrenar el modelo
        modelo.fit(X_train_transformed, subset_y_train)

        # Evaluar el modelo
        score = modelo.score(X_test_transformed, y_test)
        scores.append(score)

    # Crear el gráfico con Plotly
    fig = go.Figure()
    fig.add_trace(go.Scatter(
        x=tamaños,
        y=scores,
        mode='lines+markers',
        name='Score del modelo',
        line=dict(color='blue'),
        marker=dict(size=8)
    ))

    fig.update_layout(
        title='Evolución del score respecto al tamaño del conjunto de entrenamiento',
        xaxis_title='Tamaño del conjunto de entrenamiento',
        yaxis_title='Score',
        template='plotly_white',
        showlegend=True
    )

    fig.show()



# Llamar a la función con los datos actuales
modelo = LogisticRegression()
vectorizer = TfidfVectorizer()
eval(X_train, y_train, X_test, y_test, nlp, vectorizer, modelo)
```



Ademas de que los modelos puede mejorar su evaluación con mayor cantidad de ejemplos en el conjunto de entrenamiento, también pueden depender en gran medida de cómo el vectorizador TF-IDF (Term Frequency-Inverse Document Frequency) representa los nuevos datos agregados al corpus.

Recordemos que lo que hace TF-IDF es asignar pesos a las palabras en función de su frecuencia en un documento y su "rareza" en el corpus completo. Lo que significa que:

- Si los nuevos ejemplos contienen palabras únicas o poco frecuentes en el corpus original, estas palabras tendrán un peso mayor en la representación TF-IDF. Esto puede ayudar al modelo a identificar mejor los patrones en los datos basados en estas palabras clave.

- Si los nuevos ejemplos contienen palabras comunes, estas palabras tendrán un peso bajo, y el modelo no mejoraría significativamente con los nuevos datos, o incluso podría confundirse empeorar si los ejemplos no son representativos.

Por lo tanto, el impacto de los nuevos ejemplos en el score dependerá de:

- La calidad y representatividad de los nuevos ejemplos.
- Cómo TF-IDF ajusta los pesos de las palabras en el corpus expandido.

Para observar como cambia la matriz TF-IDF hagamos el siguiente experimento

In [8]:
```python
tuits = pd.DataFrame({
    'Text': [
        "Me encanta estar en Pátzcuarp, es increíble!",  # Positivo
        "El servicio fue excelente, muy recomendado.",  # Positivo
        "Estoy muy feliz con los resultados.",  # Positivo
        "No me gustó para nada, fue una pérdida de tiempo.",  # Negativo
        "Estoy muy triste por haber perdido el autobus.",  # Negativo
        "El servicio al cliente fue terrible, no lo recomiendo."  # Negativo
    ],
    'sentiment': [
        'positivo',  # Etiqueta positiva
        'positivo',
        'positivo',
        'negativo',  # Etiqueta negativa
        'negativo',
        'negativo'
    ]
})

# Veamos como cambia la matriz TF-IDF al añadir más datos con los tuits inventados

vectorizer = TfidfVectorizer()

for i in range(1, 6):
    tuitsaux = tuits.loc[0:i]
    xaux = vectorizer.fit_transform(preprocess_docs(tuitsaux['Text'], nlp))
    # Convertir la matriz dispersa a un DataFrame
    dfaux = pd.DataFrame(xaux.toarray(), columns=vectorizer.get_feature_names_out())
    # Visualizar la matriz TF-IDF
    fig = go.Figure(data=[go.Table(
        header=dict(values=list(dfaux.columns),
                    fill_color='paleturquoise',
                    align='left'),
        cells=dict(values=[dfaux[col] for col in dfaux.columns],
                   fill_color='lavender',
                   align='left'))
    ])
    fig.update_layout(title_text=f"Matriz TF-IDF con {i+1} tuits")
    fig.show()
```

Matriz TF-IDF con 2 tuits

encanta	excelente	increíble	pátzcuarp	recomendado	servicio
0.5773502691896257	0	0.5773502691896257	0.5773502691896257	0	0
0	0.5773502691896257	0	0	0.5773502691896257	0.5773502691896257

Matriz TF-IDF con 3 tuits

encanta	excelente	feliz	increíble	pátzcuarp	recomendado	resultados	servicio
0.5773502691896257	0	0	0.5773502691896257	0.5773502691896257	0	0	0
0	0.5773502691896257	0	0	0	0.5773502691896257	0	0.5773502691896257
0	0	0.7071067811865476	0	0	0	0.7071067811865476	0

Matriz TF-IDF con 4 tuits

encanta	excelente	feliz	gustó	increíble	pátzcuarp	pérdida	recomendado	resultados	servicio	tiempo
0.577350269189	0	0	0	0.577350269189	0.577350269189	0	0	0	0	0
0	0.577350269189	0	0	0	0	0	0.577350269189	0	0.577350269189	0
0	0	0.707106781186	0	0	0	0	0	0.707106781186	0	0
0	0	0	0.577350269189	0	0	0.577350269189	0	0	0	0.577350269189

Matriz TF-IDF con 5 tuits

autobus	encanta	excelente	feliz	gustó	increíble	perdido	pátzcuarp	pérdida	recomendadc	resultados	servicio	tiempo	triste
0	0.577350269	0	0	0	0.577350269	0	0.577350269	0	0	0	0	0	0
0	0	0.577350269	0	0	0	0	0	0	0.577350269	0	0.577350269	0	0
0	0	0	0.707106781	0	0	0	0	0	0	0.707106781	0	0	0
0	0	0	0	0.577350269	0	0	0	0.577350269	0	0	0	0.577350269	0
0.577350269	0	0	0	0	0	0.577350269	0	0	0	0	0	0	0.577350269

Matriz TF-IDF con 6 tuits

autobus	cliente	encanta	excelente	feliz	gustó	increíble	perdido	pátzcuarp	pérdida	recomenda	recomiend	resultados	servicio	terrible	tiempo	triste
0	0	0.5773502	0	0	0	0.5773502	0	0.5773502	0	0	0	0	0	0	0	0
0	0	0	0.6117125	0	0	0	0	0	0	0.6117125	0	0	0.5016130	0	0	0
0	0	0	0	0.7071067	0	0	0	0	0	0	0	0.7071067	0	0	0	0
0	0	0	0	0	0.5773502	0	0	0	0.5773502	0	0	0	0	0	0.5773502	0
0.5773502	0	0	0	0	0	0	0.5773502	0	0	0	0	0	0	0	0	0.5773502
0	0.5218234	0	0	0	0	0	0	0	0	0	0.5218234	0	0.4279027	0.5218234	0	0