Segundo avanze del PIA

Equipo 2

Grupo 012, martes y jueves de 18:30 a 20:00 hrs
Integrantes:
1395501 Jose Pedro Treviño Hernandez
1647656 Omar Alejandro Delgado Lozano

BASE DE DATOS: Trip Advisor Hotel Reviews

La base de datos de Trip Advisor Hotel Reviews contiene mas de 20,000 reviews o reseñas de diferentes hoteles ademas de calificaciones dadas por los huspedes.

In [46]:
```
#librerias necesarias
import pandas as pd
import seaborn as sns
import re
import nltk
from nltk import word_tokenize,sent_tokenize
import nltk as nlp
import matplotlib as plt
import numpy as np
from nltk.stem.porter import PorterStemmer
from nltk.corpus import stopwords
import missingno as msno
import warnings
warnings.filterwarnings("ignore")
import plotly.graph_objects as go
import spacy
import tensorflow as tf
from wordcloud import WordCloud, STOPWORDS
import ktrain
from ktrain import text

from collections import Counter
%matplotlib inline
```

Se planea usar estas librerrias ya que son las que satisfacen las necesidades de busqueda de palabras y almacenamiento de ellas,ademas de ayudarnos para la visualizacion de datos en forma de graficos. por lo tanto dichas librerias son las que se estaran manejando.

Base de datos

Aqui se despliega la base de datos que seleccionamos https://www.kaggle.com/andrewmvd/trip-

In [47]:
```
df = pd.read_csv('tripadvisor_hotel_reviews.csv');
```

Out[47]:

	Review	Rating
0	nice hotel expensive parking got good deal sta...	4
1	ok nothing special charge diamond member hilto...	2
2	nice rooms not 4* experience hotel monaco seat...	3
3	unique, great stay, wonderful time hotel monac...	5
4	great stay great stay, went seahawk game aweso...	5
...	...	...
20486	best kept secret 3rd time staying charm, not 5...	5
20487	great location price view hotel great quick pl...	4
20488	ok just looks nice modern outside, desk staff ...	2
20489	hotel theft ruined vacation hotel opened sept ...	1
20490	people talking, ca n't believe excellent ratin...	2

In [48]:
```
df.shape
```

Out[48]:
```
(20491, 2)
```

In [49]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20491 entries, 0 to 20490
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Review  20491 non-null  object
 1   Rating  20491 non-null  int64 
dtypes: int64(1), object(1)
memory usage: 320.3+ KB
```

La base de datos cuenta con 2 columnas la primera de tipo object, y la segunda de tipo int ademas de contar con 20491 filas x 2 columnas ya mencionadas.

In [50]:
```
df.columns
```

Out[50]:
```
Index(['Review', 'Rating'], dtype='object')
```

In [51]:
```
df.isnull()
```

Out[51]:

	Review	Rating
0	False	False
1	False	False

	Review	Rating
2	False	False
3	False	False
4	False	False
...	...	...
20486	False	False
20487	False	False
20488	False	False
20489	False	False
20490	False	False

In [52]:
```
df.isnull().sum()
```

Out[52]:
```
Review    0
Rating    0
dtype: int64
```

Primera parte del PIA

Como la base de datos solo cuenta con dos columnas, estas dos solo son necesarias, no es necesario el eliminar ni modificar las columnas ya que seran utiles para los siguientes pasos.

In [53]:
```
baseD = df
baseD.to_csv('baseD_editada.csv', index=False)
```

In [54]:
```
df['Rating'].median() #nuestra la media
```

Out[54]:
```
4.0
```

In [55]:
```
df['Rating'].mean() #nuestra la mediana
```

Out[55]:
```
1.5203624326830831
```

In [56]:
```
df['Rating'].std() #nuestra la desviacion estandar
```

Out[56]:
```
1.2330297776952035
```

In [57]:
```
review_list=[]

for review in df.Review:
    review=re.sub("[^a-zA-z]"," ",review)
    review=review.lower()
    review=nltk.word_tokenize(review)
    lemma=nlp.WordNetLemmatizer()
    review=[lemma.lemmatize(word) for word in review]
    review=" ".join(review)
    review_list.append(review)
```

In [58]:
```
from sklearn.feature_extraction.text import CountVectorizer
```

```
max_features=500

count_vectorizer=CountVectorizer(max_features=max_features,stop_words="en
```

In [59]:
```
sparce_matrix
```

Out[59]:
```
CountVectorizer(max_features=500, stop_words='english')
```

In [60]:

```
Top 500 Palabras mas usadas: ['able', 'absolutely', 'access', 'activit
y', 'actually', 'afternoon', 'air', 'airport', 'amazing', 'amenity', '
american', 'amsterdam', 'area', 'arrival', 'arrived', 'ask', 'asked',
'ate', 'attraction', 'available', 'average', 'away', 'awesome', 'bad',
'bag', 'balcony', 'bar', 'barcelona', 'basic', 'bath', 'bathroom', 'be
ach', 'beautiful', 'bed', 'bedroom', 'beer', 'believe', 'best', 'bette
r', 'big', 'bit', 'block', 'book', 'booked', 'booking', 'bottle', 'bre
akfast', 'bring', 'brought', 'buffet', 'building', 'bus', 'business',
'busy', 'buy', 'ca', 'cab', 'cafe', 'called', 'came', 'cana', 'car', '
card', 'care', 'casino', 'center', 'central', 'centre', 'certainly', '
chair', 'change', 'charge', 'cheap', 'check', 'checked', 'cheese', 'ch
ild', 'choice', 'choose', 'chose', 'city', 'clean', 'cleaned', 'close
', 'club', 'coffee', 'cold', 'come', 'comfortable', 'coming', 'comment
', 'complaint', 'complimentary', 'concierge', 'conditioning', 'conveni
ent', 'cool', 'corner', 'cost', 'country', 'couple', 'course', 'custom
er', 'daily', 'daughter', 'day', 'deal', 'decent', 'decided', 'decor',
'decorated', 'definitely', 'desk', 'did', 'different', 'dining', 'dinn
er', 'dirty', 'disappointed', 'distance', 'doe', 'dollar', 'dominican
', 'dont', 'door', 'double', 'downtown', 'drink', 'early', 'easy', 'ea
```

In [61]:
```
X = df['Review'].copy()
```

In [62]:
```
def data_cleaner(review):

    # remove digits
    review = re.sub(r'\d+',' ', review)

    #removing stop words
    review = review.split()
    review = " ".join([word for word in review if not word in stop_words]

    #Stemming
    #review = " ".join([ps.stem(w) for w in review])

    return review

ps = PorterStemmer()
stop_words = stopwords.words('english')

X_cleaned = X.apply(data_cleaner)
```

Out[62]:

```
0    nice hotel expensive parking got good deal sta...
1    ok nothing special charge diamond member hilto...
```

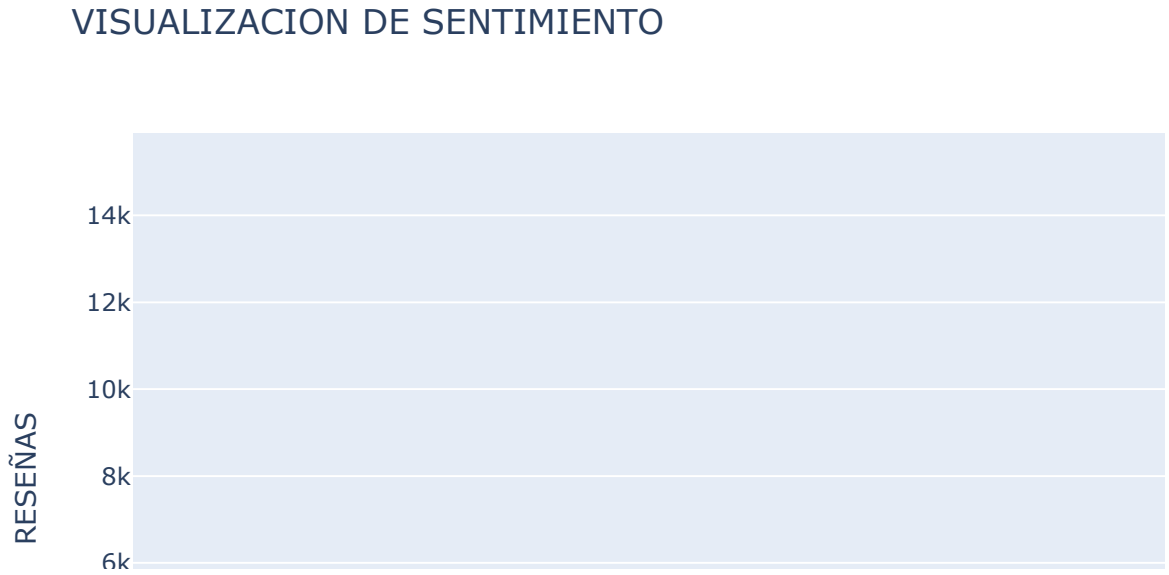
In [63]:
```
pos = [4, 5]
neg = [1, 2]
neu = [3]

def sentiment(rating):
  if rating in pos:
    return 2
  elif rating in neg:
    return 0
  else:
```

In [64]:
```
df['Sentiment'] = df['Rating'].apply(sentiment)
```

Out[64]:

	Review	Rating	Sentiment
0	nice hotel expensive parking got good deal sta...	4	2
1	ok nothing special charge diamond member hilto...	2	0
2	nice rooms not 4* experience hotel monaco seat...	3	1
3	unique, great stay, wonderful time hotel monac...	5	2
4	great stay great stay, went seahawk game aweso...	5	2

In [65]:
```
fig = go.Figure([go.Bar(x=df.Sentiment.value_counts().index, y=df.Sentime
fig.update_layout(
    title="VISUALIZACION DE SENTIMIENTO",
    xaxis_title="SENTIMIENTO",
    yaxis_title="RESEÑAS")
```



2 - positivo (4, 5) 1 - neutral (3) 0 - Negativo (1, 2)

In [66]:
```
nlp = spacy.load('en_core_web_sm')

def normalize(msg):

    doc = nlp(msg)
    res = []

    for token in doc:
        if(token.is_stop or token.is_punct or token.is_space):
            pass
        else:
            res.append(token.lemma_.lower())
```

In [*]:
```
df['Review'] = df['Review'].apply(normalize)
```

In [*]:
```
words_collection = Counter([item for sublist in df['Review'] for item in
freq_word_df = pd.DataFrame(words_collection.most_common(15))
freq_word_df.columns = ['frequently_used_word','count']
```

In [*]:
```
```

In [*]:
```
valores=freq_word_df
ax=valores.plot.bar(x="palabras",y="cantidad de veses usada",rot=90)
```

palabras mas usadas en general

In [*]:
```
word_list = [item for sublist in df['Review'] for item in sublist]
word_string = " ".join(word_list)

wordcloud = WordCloud(stopwords=STOPWORDS,
                          background_color='white',
                      max_words=60000,
                      width=1000,
                      height=650
```

In [*]:
```
plt.figure(figsize=(20,10))
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```

positivas

In [*]:
```
pos_df = df[df['Sentiment'] == 2]
words_collection = Counter([item for sublist in pos_df['Review'] for item
freq_word_df = pd.DataFrame(words_collection.most_common(15))
freq_word_df.columns = ['frequently_used_word','count']
```

In [*]:
```
word_list_pos = [item for sublist in pos_df['Review'] for item in sublist
word_string_pos = " ".join(word_list)

wordcloud = WordCloud(stopwords=STOPWORDS,
                      background_color='white',
                      max_words=40000,
                      width=1000,
                      height=650
```

In [*]:
```
plt.figure(figsize=(20,10))
plt.imshow(wordcloud)
plt.axis('off')
```

negativas

In [*]:
```
neg_df = df[df['Sentiment'] == 0]
words_collection = Counter([item for sublist in neg_df['Review'] for item
freq_word_df = pd.DataFrame(words_collection.most_common(15))
freq_word_df.columns = ['frequently_used_word','count']
```

In [*]:
```
word_list_neg = [item for sublist in neg_df['Review'] for item in sublist
word_string_neg = " ".join(word_list)

wordcloud = WordCloud(stopwords=STOPWORDS,
                      background_color='white',
                      max_words=10000,
                      width=1000,
                      height=650
```

In [*]:
```
plt.figure(figsize=(20,10))
plt.imshow(wordcloud)
plt.axis('off')
```