John Alexander Tami Buitrago

Prueba de Programación
Preguntas teóricas

Pregunta 1: ¿Qué es Python?

Python es un lenguaje de programación interpretado orientado a objetos.

Pregunta 2: ¿Consideras que Python es mejor a Java? ¿Por qué?

No, cada uno de los lenguajes de programación tiene sus pros y sus contras. En particular me gusta por legibilidad y facilidad de uso, Python.

Pregunta 3: ¿Cuántos tipos de datos existen en Python?

Los tipos incorporados de Python son:

- String
- Lista
- Número
- Diccionario
- Tupla

Pregunta 4: ¿Cuál es la diferencia entre tupla y lista?

Las tuplas no pueden modificarse, mientras que las listas sí.

Pregunta 5: ¿Qué es PEP8?

PEP son siglas que representan a Python Enhancement Proposal. Y que especifican cómo formatear el código de Python para que tenga una máxima legibilidad.

Pregunta 6: ¿Qué es "pickling" y "unpickling"?

Ocurre cuando un módulo dentro del lenguaje Python es aceptado y convertido en un módulo string para luego ser volcado en el archivo. Por el contrario, unpickling es cuando retiras el módulo string del archivo.

Pregunta 7: ¿Qué es "lambda"?

Es una función anónima la cual puede tener diferentes números de parámetros, pero una sola sentencia. A diferencia de las funciones definidas con **def** las funciones lambda no poseen un nombre.

Pregunta 8: ¿Cómo se administra la memoria dentro del lenguaje Python?

El conjunto privado de Python es el encargado de administrar la memoria, como hay un espacio finito de memoria lo que realiza es encontrar espacios libres donde ocupar memoria para así asignarlos, los datos que ya no son usados, son eliminados.

Pregunta 9: ¿Qué es "pass"?

Es una declaración nula que se usa como marcador de posición con el fin de saltar errores o bien omitir la ejecución de un bloque de código.

Pregunta 10: ¿Puedes copiar un objeto en el lenguaje Python?

Sí, por medio del comando copy()

Pregunta 11; ?Cómo borrar un archivo dentro de Python?

Para eliminar un archivo es necesario importar el módulo OS, para posteriormente usar la función remove. (os.remove())

Pregunta 12: ¿Qué es un "diccionario"?

Es un tipo de dato en Python con características especiales en el cual se puede almacenar cualquier tipo de valor, identificando cada uno de estos elementos por medio de una clave.

Pregunta 13: ¿Es Python un lenguaje de programación interpretado?

Sí, ya que el código fuente, se convierte en lenguaje de programación para posteriormente ser cambiado a código de computadora y así poder ser ejecutado.

Pregunta 14: ¿Cómo Python se considera un lenguaje orientado a objetos?

Como los datos y las operaciones que se realizan con los datos se agrupan en unidades lógitas denominadas objetos, dichos objetos representan un concepto del programa por medio de características denominadas atributos.

Pregunta 15: ¿Qué es “slicing"?

Es la selección de múltiples objetos de diferentes lugares como por ejemplo de una lista, un string.

Pregunta 16: Según la guía de estilos PEP8 ¿cómo deben escribirse las constantes?

Las constantes son generalmente definidas a nivel módulo, escritas todas con letras mayúsculas y con guiones bajos separando las palabras.

Pregunta 17: ¿Qué es virtualenv y como lo inicias con un interpretador de Python especifico?

Virtualenv es un programa que permite crear entornos virtuales de Python. Se debe instalar desde la consola de Python con el comando (pip install virtualenv), en mi caso particular puedo inicializar el entorno virtual desde la consola de anaconda promp colocando en la ubicación que desee el comando (virtualenv nombre_del_entorno_virtual).

Ejercicio práctico
(Ejercicio_practico.py)

En el correo, se adjunta un archivo JSON con tweets. Debes leer el archivo en Python y obtener las siguientes métricas:

1. Numero de tweets

```
import json

with open('tweets_test.json') as file:
   data = json.load(file)

print("El número de Tweets es", len(data))
```

2. Lista de los usuarios autores de los tweets (tip: el nombre de usuario está en user > screen_name).

```
Usuarios=[]
for n in data.keys():
   Usuarios.append(data[n]["user"]["screen_name"])
   #print(data[n]["user"]["screen_name"])
Usuarios1=set(Usuarios)
print("Lista de usuarios:\n","\n".join(Usuarios1))
```

3. Totalizador del numero de seguidores de todos los usuarios (tip: los seguidores están en user > followers_count)

```
Total_seguidores=0
for n in data.keys():
   Total_seguidores=Total_seguidores + data[n]["user"]["followers_count"]
print("El número de seguidores de todos los usuarios es ",Total_seguidores)
```

4. Obtén una lista de todos los usuarios mencionados en los tweets

```
Usuarios_menc=[]
for n in data.keys():
   for i in data[n]["real_entities"]["user_mentions"]:
      Usuarios_menc.append(i["screen_name"])
      #print(data[n]["user"]["screen_name"])
Usuarios_menc1=set(Usuarios_menc)
print("Lista de usuarios mencionados en todos los tweets:\n","\n".join(Usuarios1))
```

5. Obtén el total de tweets con tendencia 0, 1 y 2 respectivamente

```
Tend_0=0
Tend_1=0
Tend_2=0
```

```
for n in data.keys():
    print(n)
    if data[n]["tendencia"] == "0":
        Tend_0=Tend_0+1
    elif data[n]["tendencia"] == "1":
        Tend_1=Tend_1+1
    elif data[n]["tendencia"] == "2":
        Tend_2=Tend_2+1

print("Total de Tweets con tendencia 0 es: ",Tend_0)
print("Total de Tweets con tendencia 1 es: ",Tend_1)
print("Total de Tweets con tendencia 2 es: ",Tend_2)
```

Análisis de sentimientos
(Analisis_de_sentimientos.py)

El objetivo es detectar el “discurso de odio” en los tweets.

- Decimos que un tweet contiene discurso de odio si tiene un sentimiento racista o sexista asociado con él. Por lo tanto, la tarea es clasificar los tweets racistas o sexistas de otros tweets.

- Dada una muestra de entrenamiento de tweets y etiquetas, donde la etiqueta '1' denota que el tweet es racista/sexista y la etiqueta '0' denota que el tweet no es racista/sexista, su objetivo es predecir las etiquetas en el conjunto de datos de prueba.

➢ Base de datos: analisis_sentimientos.zip

```
# -*- coding: utf-8 -*-
"""
Created on Thu Dec 23 11:02:21 2021

@author: John Tami
"""

""" Lectura de los datos de entrenamiento y de test """
import pandas as pd

train  = pd.read_csv('train.csv')
test = pd.read_csv('test.csv')

#%%
""" Limpieza de los Tweets """

import numpy as np
import re

# Función para limpieza de datos
def eliminar_patrones(tex_original, patron):
    r = re.findall(patron, tex_original)
    for i in r:
        tex_original = re.sub(i, '', tex_original)
    return tex_original

# Eliminar palabras que inicien con @ (@usuarios)
train['tweet_limpio'] = np.vectorize(eliminar_patrones)(train['tweet'], "@[\w]*")
test['tweet_limpio'] = np.vectorize(eliminar_patrones)(test['tweet'], "@[\w]*")

# Eliminar números, puntuaciones y caracteres especiales
train['tweet_limpio'] = train['tweet_limpio'].str.replace("[^a-zA-Z#]", " ")
test['tweet_limpio'] = test['tweet_limpio'].str.replace("[^a-zA-Z#]", " ")

# Eliminar palabras con menos de 4 letras de longitud
train['tweet_limpio'] = train['tweet_limpio'].apply(lambda x: ' '.join([w for w in x.split() if len(w)>3]))
test['tweet_limpio'] = test['tweet_limpio'].apply(lambda x: ' '.join([w for w in x.split() if len(w)>3]))


#%%
"""Tokenización"""
```

```
tokenized_tweet_train = train['tweet_limpio'].apply(lambda x: x.split())
tokenized_tweet_test = test['tweet_limpio'].apply(lambda x: x.split())


from nltk.stem.porter import *
stemmer = PorterStemmer()

tokenized_tweet_train = tokenized_tweet_train.apply(lambda x: [stemmer.stem(i) for i in x]) # stemming
tokenized_tweet_test = tokenized_tweet_test.apply(lambda x: [stemmer.stem(i) for i in x])

#%%

"""                Nube de  palabras - Palabras negativas            """

import matplotlib.pyplot as plt
from wordcloud import WordCloud

negative_words = ' '.join([text for text in train['tweet_limpio'][train['label'] == 1]])
wordcloud = WordCloud(width=800, height=500,
random_state=21, max_font_size=110).generate(negative_words)
plt.figure(figsize=(10, 7))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis('off')
plt.show()

#%%
""" Extraer hashtag dependiendo de si es o no racista o sexista"""

# Función para extraer los hashtags
def hashtag_extract(x):
    hashtags = []
    for i in x:
        ht = re.findall(r"#(\w+)", i)
        hashtags.append(ht)
    return hashtags

# Extraer HT de Tweets no racistas, ni sexistas
HT_normal = hashtag_extract(train['tweet_limpio'][train['label'] == 0])

# Extraer HT de Tweets racistas y sexistas
HT_negativo = hashtag_extract(train['tweet_limpio'][train['label'] == 1])


#%%

"""""Entrenamiento para clasificación de tweets"""

from sklearn.feature_extraction.text import CountVectorizer

bow_vectorizer = CountVectorizer(max_df=0.90, min_df=2, max_features=1000, stop_words='english')
# bag-of-words feature matrix
train_bow = bow_vectorizer.fit_transform(train['tweet_limpio'])
test_bow = bow_vectorizer.fit_transform(test['tweet_limpio'])

from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import f1_score
```

```
# splitting data into training and validation set
xtrain_bow, xvalid_bow, ytrain, yvalid = train_test_split(train_bow, train['label'], random_state=42, test_size=0.3)

#%%
"""Regresión Logística"""

lreg = LogisticRegression()
lreg.fit(xtrain_bow, ytrain) # training the model

prediction = lreg.predict_proba(xvalid_bow) # predicting on the validation set
prediction_int = prediction[:,1] >= 0.3 # if prediction is greater than or equal to 0.3 than 1 else 0
prediction_int = prediction_int.astype(np.int)

f1_score(yvalid, prediction_int) # calculating f1 score

#%%
"""Predicción y exportación de los datos"""

test_pred = lreg.predict_proba(test_bow)
test_pred_int = test_pred[:,1] >= 0.3
test_pred_int = test_pred_int.astype(np.int)
test['label'] = test_pred_int
result = test[['id','tweet','label']]
result.to_csv('prediccion.csv', index=False) # writing data to a CSV file
```