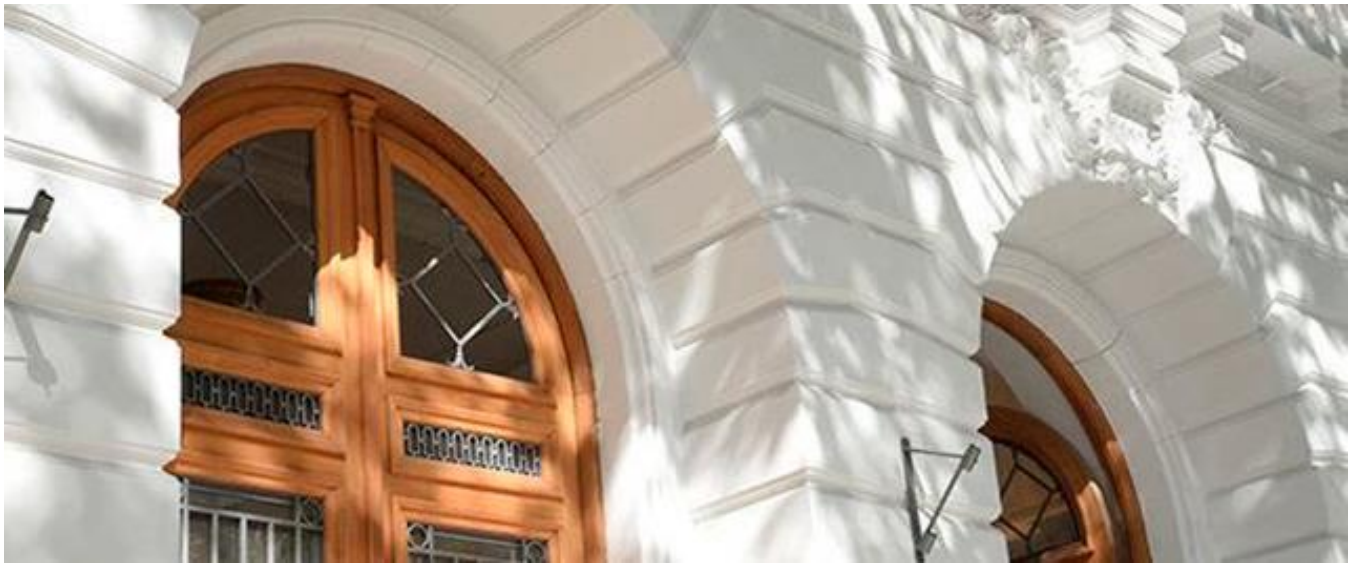
UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

DEPARTAMENTO
INGENIERIA EN SISTEMAS DE INFORMACION



Procesamiento del lenguaje natural

2023

Entrega TP sentimientos

CURSO: K3551

Nombre del alumno: Ilan Trupkin

Resolución de la consigna:

La moneda seleccionada es BTC (Bitcoin) y las páginas seleccionadas para obtener noticas son Bing, Google y Cointelegraph.

Para el análisis de sentimiento se usa OpenAI. Para El análisis de frecuencias y NER se usa Spacy. Como base de datos se usan hojas de cálculo (Excel) trabajadas mediante pandas.

Para obtener el valor en dólares del precio del bitcoin en determinada fecha se usa la API de CriptoCompare.

El código se divide en dos partes:

1) Parte funcional
2) Parte de resultados

Comenzando por la primera parte, esta se divide en 3 celdas.

- La primera contiene la instalación de librerías/bibliotecas a utilizar en el resto del código.
- La segunda celda contiene la importación de las librerías a utilizar
- La tercera celda contiene la definiciones y declaración de las funciones que se utilizan como a su vez de las clases creadas.

Explico brevemente que bibliotecas se utilizan:

- PyTorch para algunas funciones de Machine Learning
- Requests para hacer las consultas a las paginas web
- Beautiful Soup para hacer el scraping
- Textwrap para tratar los strings
- Random para aleatoriedad
- Pandas para manejo de datos
- OpenAI para análisis de sentimiento
- GoogleSearch como api para obtener noticias de Google
- Spacy para análisis NER y frequency
- Matplotlib para hacer graficos

La tercera celda de esta primera parte es la más importante ya que contiene la funcionalidad del código como tal.

Aquí explico el contenido de esta celda:

- Clase noticia con Titulo, link, descripción, fecha, sentimiento, diccionario de frecuencias, diccionario con NER

```
class Noticia:
  def __init__(self):
      self.Title = ""
      self.Link = ""
      self.Snippet = ""
      self.Date = ""
      self.Sent = None
      self.Frequency = {}
      self.Ner = {}
```

- La función cointelegraph_scrapping() se encarga de obtener una lista con las noticias de la página cointelegraph que hablen sobre el bitcoin.
- La función google_scrapping() se encarga de realizar el mismo procedimiento que la función anterior, pero en este caso sobre las noticias de Google. Obtiene 10 noticias.
- La función bing_scrapping(search) hace lo mismo que las dos anteriores pero sobre Bing Noticias.
- page_scrapping(url) es la función utilizada para que dada una noticia en formato url, filtre solamente el contenido de los párrafos.
- La función sentiment_analysis(text) se encarga de recibir por parámetro un texto y determina si el sentimiento es “positivo”, “negativo” o “neutral” en comparación con el precio del Bitcoin. Esto es llevado a cabo con la API de OpenAI usando GPT3.5. El prompt enviado es el siguiente: “Just answer "positive", "negative" or "neutral" depending on what the following text between slashes means about a posible rise (positive), fall (negative) in the price of bitcoin. Answer "neutral" if you can't identify anything.”. Tome la decisión de usar esta api porque tiene la capacidad de analizar si el texto se relaciona o no con la criptomoneda, también tiene la capacidad de determinar si representa una suba o no del precio. He probado con otras librerías como spacy o HuggingFace y no logre un buen nivel de respuesta como el que obtuve con OpenAI.
- La función search_bitcoin_notices(scrapping_function) se encarga de unificar los resultados del scraping de las paginas web con el scraping de los links de las noticias y devuelve una lista con las noticias y sus párrafos detectados.
- La funcion detect_sentiment_in_notices(notices) recibe varias noticias y devuelve la lista de noticias con sus sentimientos como así también el sentimiento asociado a todas esas noticias.
- detect_sentiment_in_full_notice(parrafos) recibe todos los párrafos de una noticia, los unifica y detecta el sentimiento de la misma.
- sentiment_from_bing2() devuelve la lista de noticias y el sentimiento asociado a la búsqueda por medio de Bing.
- sentiment_from_google() devuelve la lista de noticias y el sentimiento asociado a la búsqueda por medio de Google.
- sentiment_from_cointelegraph() devuelve la lista de noticias y el sentimiento asociado a la búsqueda por medio de Cointelegraph.
- agregar_a_excel(nuevo_dataframe) en caso de que no exista la hoja de cálculos (que es la base de datos) se crea. Si ya existe se agregan las nuevas noticias a la base de datos.
- obtener_frecuency(parrafos, cantidad) esta función devuelve un diccionario con las “cantidad” principales palabras.
- obtener_ner(parrafos) devuelve un array con todos los NER que se encuentren en el texto.

- obtener_sentimiento_por_fecha(fecha) recibe una fecha de tipo string y devuelve el sentimiento analizado para esa fecha según la base de datos.
- graficar_sentimientos_segun_bd() busca las fechas que se usan en la base de datos y grafica cada fecha con su correspondiente sentimiento promedio del día.
- obtener_precio_bitcoin_por_dia(fecha) obtiene mediante la api de CriptoCompare el precio en dólares al que cerró el bitcoin en una fecha determinada.
- comparar_precio_por_fecha_con_dia_anterior_y_siguiente(fecha) devuelve el signo de la diferencia en dólares del precio del bitcoin el día anterior y posterior a una fcha determinada.

Acá finaliza la primera parte.

La segunda parte consta de la puesta en ejecución de las funciones anteriores y la muestra de los resultados:

```
noticiasGoogle, sentGoogle = sentiment_from_google()

hoy = date.today()
dataGoogle = [{'link': noticia.Link, 'fecha': hoy, 'sentimiento': noticia.Sent, 'frequency': str(noticia.Frequency), 'ner': str(noticia.Ner)} for noticia in noticiasGoogle]
dataNoticiasGoogle = pd.DataFrame(dataGoogle)
```

Esta celda ejecuta la función para detectar el sentimiento de las noticias de Google de hoy.

Esto mismo se hace con Bing y con Cointelegraph.

Luego de ejecutar el código anterior para las 3 fuentes de información, se agregan todos los resultados a la base de datos.

```
dataNoticias = pd.concat([dataNoticiasBing, dataNoticiasGoogle, dataNoticiasCointelegraph], axis=0)
dataNoticias.reset_index(drop=True, inplace=True)
agregar_a_excel(dataNoticias)
```

Finalmente se obtiene el sentimiento del día. Es mediante un promedio entre todas las fuentes que se calcula el sentimiento del día. En caso de ser promedio mayor a 0.2, se lo considera en alta. Si es menor a -0.2 se lo considera en baja. Y si se encuentra entre -0.2 y 0.2 se lo denomina neutral.

```
# obtener el sentimiento de hoy
fecha_actual = date.today()
fecha_actual_str = fecha_actual.strftime("%Y-%m-%d")
prom = obtener_sentimiento_por_fecha(fecha_actual_str)
if prom > 0.2:
  print("El mercado esta en alta")
elif prom < -0.2:
  print("El mercado esta en baja")
else: print("El mercado esta en estado neutral")
```

Con esta próxima celda se puede analizar el mercado de Bitcoin en determinada fecha según la base de datos.

```
# consulta por fecha a la base de datos:
prom = obtener_sentimiento_por_fecha('2023-10-23')
if prom > 0.2:
  print("El mercado esta en alta")
elif prom < -0.2:
  print("El mercado esta en baja")
else: print("El mercado esta en estado neutral")
```

```
El mercado esta en alta
```

Según los resultados de la celda anterior el detector de emociones predijo que el mercado estuvo en alta.

En la próxima celda se puede ver cuál fue el estado del mercado de Bitcoin en la fecha determinada:

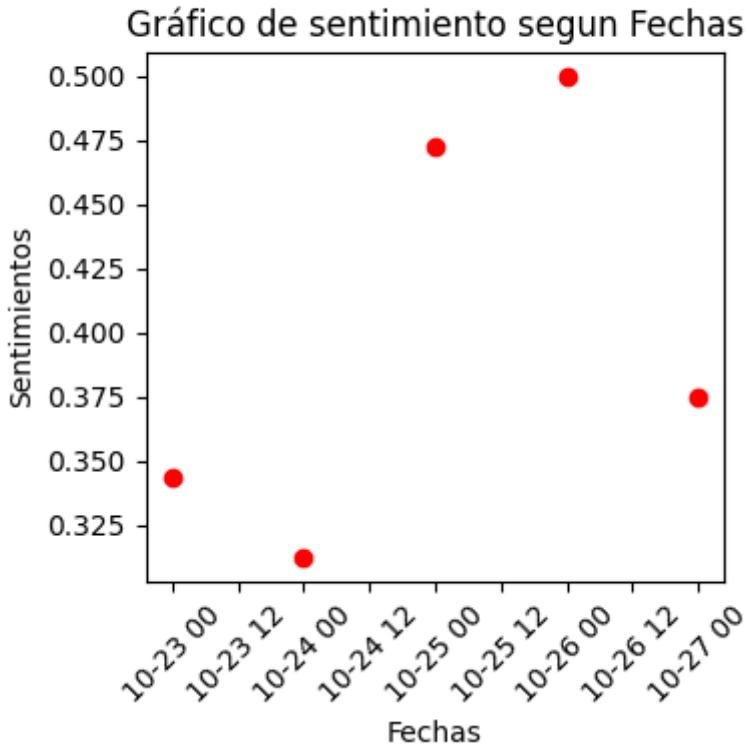
```
#comparo el resultado predicho en la celda anterior con el real segun la misma fecha
resul = comparar_precio_por_fecha_con_dia_anterior_y_siguiente('2023-10-23')
if resul > 0:
  print("El mercado realmente estuvo en alta")
else:
  print("El mercado realmente estuvo en baja")
```

```
El mercado realmente estuvo en alta
```

De esta forma se concluye que la predicción fue correcta.

En esta celda se realiza un gráfico con los sentimientos encontrados día a día:

```
#Grafico con los sentimientos segun las fechas
graficar_sentimientos_segun_bd()
```



Referencias:

TheCodex. (2020, 21 julio). [Python Project] Sentiment Analysis and Visualization of Stock news [Vídeo]. YouTube. https://www.youtube.com/watch?v=o-zM8onpQZY

Análisis del sentimiento en el trading de criptomonedas: guía para principiantes | KuCoin Learn. (2023, 28 julio). KuCoin Learn. https://www.kucoin.com/es/learn/trading/sentiment-analysis-in-crypto-trading-a-beginners-guide

Pratikpv. (s. f.). GitHub - Pratikpv/Google_News_scraper_and_Sentiment_Analyzer: downloads news articles from Google News and uses pre-trained NLP models to perform sentiment analysis. GitHub. https://github.com/pratikpv/google_news_scraper_and_sentiment_analyzer

Sc, P. O. M. (2022, 7 enero). Using sentiment analysis to predict the stock market. Medium. https://medium.com/analytics-vidhya/using-sentiment-analysis-to-predict-the-stock-market-77100295d753

OpenAI Platform. (s. f.). https://platform.openai.com/overview

Cointelegraph. (2019, 15 julio). Bitcoin News by Cointelegraph. Cointelegraph. https://cointelegraph.com/tags/bitcoin

SPACY · Industrial-strength Natural language processing in Python. (s. f.). https://spacy.io/

Crypto Compare. (s. f.). Cryptocurrency Prices, Portfolio, Forum, Rankings. CryptoCompare. https://www.cryptocompare.com/