**UNIVERSIDAD COMPLUTENSE DE MADRID**

**FACULTAD DE ESTUDIOS ESTADÍSTICOS**

**MASTER EN MINERÍA DE DATOS E INTELIGENCIA DE NEGOCIOS**



**Práctica 3**

**Por:**

Roberto Bonilla Ibarra

Mario Esteban Merchán Illescas

Memoria Técnica

Madrid, 11 de Mayo de 2023

1. Objetivo

El objetivo de este trabajo es extraer los valores de cierre del precio de las acciones de una compañía específica utilizando una API. Para predecir el valor del día siguiente, utilizaremos un algoritmo de aprendizaje profundo LSTM[[1]](#footnote-1). Además, aprovecharemos la técnica de web scraping para analizar los sentimientos en los titulares de noticias más recientes relacionadas con la empresa. Al combinar estos enfoques, buscamos obtener resultados más sólidos y completos.

1. Librerías y módulos

Antes de comenzar a especificar las funciones que hemos usado para crear cada una de las secciones del trabajo, se mencionarán las librerías y módulos de Python, al igual que su funcionalidad:

Librerías para tratamiento de datos por medio de API (funciones\_generales):

* **Pandas**: Librería para el manejo de datos con Python
* **Math**: Uso de funciones matemáticas
* **Datetime**: Trabajar con fechas y tiempo
* **Matplotlib**: Graficar resultados
* **alpha\_vantage:** Extracción de datos por medio de API de Alpha vantage.
* **sklearn:** Para escalado de datos

Librerías para modelo de aprendizaje profundo (modelo\_lstm):

* **tensorflow:** Librería de python para Deep Learning.
* **keras:** Librería de código abierto para experimentación de redes neuronales en poco tiempo.
* **numpy:** Crear vectores y matrices, reduciendo tiempos.

Librerías para web scraping (web\_scraping):

* **selenium:** Librería utilizada para automatizar el uso de un buscador.
* **bs4:** Librería para lectura y extracción de datos de archivos con formato HTML.

Librerías para análisis de sentimientos (sentimientos\_viz):

* **wordcloud:** Librería de python para realizar nubes en palabras.
* **seaborn:** Librería para un análisis rápido de resultados.

Librerías para módulo principal:

* **json:** Librería para lectura de archivo configuración en json.
* **pymongo:** Librería para trabajar con Mongo DB.
* **tensorflow:** Librería para interacción con modelo de aprendizaje profundo.

1. Módulo de extracción de datos por API (funciones\_generales)

En este módulo nos enfocaremos en extraer los datos de la API de vantage por medio de Python, adicional usaremos matplotlib para graficar los datos.

Las funciones utilizadas son:

1. **descargar\_datos(config, plot=False)**:
   * Utilidad: Descarga los datos de una empresa utilizando la API de Alpha Vantage.
   * Parámetros:
     + **config**: Un diccionario que contiene la configuración necesaria para acceder a la API y extraer los datos.
     + **plot** (opcional): Un booleano que indica si se debe realizar una visualización de los datos descargados.
   * Salida: Devuelve una serie de datos que incluyen las fechas, los precios de cierre, el número de puntos de datos y el rango de fechas.
2. **obtener\_data\_pandas(fechas, valores, corte)**:
   * Utilidad: Crea un objeto DataFrame de Pandas a partir de las fechas y valores proporcionados.
   * Parámetros:
     + **fechas**: Una lista de fechas.
     + **valores**: Una lista de valores correspondientes a las fechas.
     + **corte**: Una fecha utilizada para filtrar el DataFrame.
   * Salida: Devuelve un DataFrame de Pandas que contiene las fechas y los valores de cierre, así como un arreglo numpy del objetivo (columna de cierre).
3. **split\_train\_val\_test(array, test\_size, val\_size, window\_size)**:
   * Utilidad: Divide los datos en conjuntos de entrenamiento, validación y prueba, y realiza la normalización de los datos.
   * Parámetros:
     + **array**: Un arreglo numpy de los datos de entrada.
     + **test\_size**: El tamaño del conjunto de prueba.
     + **val\_size**: El tamaño del conjunto de validación.
     + **window\_size**: El tamaño de la ventana para la creación de secuencias.
   * Salida: Devuelve los conjuntos de datos normalizados de entrenamiento, validación y prueba, así como los objetivos correspondientes. También devuelve el escalado que se usó para la normalización de datos.
4. **plot\_predictions(config, predicted\_stock\_price, X\_train, fechas, data\_target, split\_name)**:
   * Utilidad: Realiza la visualización de las predicciones del modelo.
   * Parámetros:
     + **config**: Un diccionario que contiene la configuración del modelo y los datos.
     + **predicted\_stock\_price**: Una serie de tiempo con las predicciones del modelo.
     + **X\_train**: Los datos de entrenamiento utilizados para realizar las predicciones.
     + **fechas**: Una lista de fechas.
     + **data\_target**: Un DataFrame de Pandas que contiene los datos de destino.
     + **split\_name**: El nombre de la división de datos utilizada para la visualización.
   * Salida: Muestra una gráfica con los datos de entrenamiento, las predicciones y, en caso de ser necesario, los datos de prueba. También devuelve el valor de error RMSE y la lista de fechas utilizadas para las predicciones.
5. Módulo de aprendizaje profundo LSTM (modelo\_lstm)

El módulo de aprendizaje profundo permite crear una clase de un modelo LSTM especializado en predecir el valor de la acción, junto con una técnica de learning rate [[2]](#footnote-2)cícliclo.

1. **crear\_learning\_rate\_ciclico(config, X\_train):**
   * Utilidad: Crea una tasa de aprendizaje cíclica para utilizarla en el entrenamiento de un modelo de aprendizaje profundo.
   * Parámetros:
     + **config:** Un diccionario que contiene la configuración del modelo de aprendizaje profundo.
     + **X\_train:** Un conjunto de datos de entrenamiento.
   * Salida: Devuelve un objeto Optimizer de TensorFlow que contiene una tasa de aprendizaje cíclica.
2. **class MyModel(keras.Model):**
   * Utilidad: Define la arquitectura de un modelo de aprendizaje profundo.
   * Parámetros:
     + **config**: Un diccionario que contiene la configuración del modelo de aprendizaje profundo.
   * Métodos:
     + **init**(self): Define las capas del modelo.
     + **call(**self, inputs): Define el flujo de datos a través del modelo. Devuelve las predicciones del modelo.
3. Módulo de web scraping (web\_scraping)

En este módulo nos enfocaremos en extraer los datos de la página **Yahoo Finance** por medio de las librerías especializadas en web scraping.

Las funciones utilizadas son:

1. **extraer\_noticias(config)**:
   * Utilidad: Esta función extrae titulares de noticias de una página web utilizando Selenium y BeautifulSoup.
   * Parámetros:
     + **config**: Un diccionario de configuración que contiene la URL de la página de noticias.
   * Salida: Devuelve una lista de titulares de noticias extraídos.
2. **classify\_positive(text, threshold=0)**:
   * Utilidad: Esta función clasifica un texto como positivo o negativo basado en su puntuación de sentimiento.
   * Parámetros:
     + **text**: El texto a clasificar.
     + **threshold** (opcional): Umbral utilizado para determinar la clasificación. El valor predeterminado es 0.
   * Salida: Devuelve 1 si el texto se clasifica como positivo según el umbral dado, de lo contrario devuelve 0.
3. **predicciones\_sentimientos(titulares, threshold\_value)**:
   * Utilidad: Esta función realiza predicciones de sentimientos para una lista de titulares de noticias utilizando la función **classify\_positive()** y genera un DataFrame de Pandas con los resultados.
   * Parámetros:
     + **titulares**: Una lista de titulares de noticias.
     + **threshold\_value**: Umbral utilizado para la clasificación de sentimientos.
   * Salida: Devuelve un DataFrame de Pandas que contiene los titulares, la clasificación de sentimientos, los detalles de polaridad y otros datos relacionados, como el umbral y la marca de tiempo.
4. Módulo de análisis de sentimientos (sentimientos\_viz)

El módulo de análisis de sentimientos se encuentra enfocado a usar el modelo preentrenado VADER el cuál es el estado del arte en la materia de clasificación de texto.

1. **bar\_plot(df):**
   * Utilidad: Crea un gráfico de barras a partir de los datos en un DataFrame.
   * Parámetros:
     + **df:** DataFrame que contiene las columnas 'neu', 'pos' y 'neg'.
   * Salida:
     + Gráfico de barras a partir del análisis de sentimientos.
2. **casos\_plot(df, config):**
   * Utilidad: Crea un gráfico de conteo de casos a partir de un DataFrame y un archivo de configuración.
   * Parámetros:
     + **df:** DataFrame que contiene la columna 'Positivos'.
     + **config:** Configuración que contiene información sobre el símbolo.
   * Salida:
     + Gráfico de conteo de casos de noticias positivas con seaborn
3. **word\_cloud(df):**
   * Utilidad: Crea una nube de palabras a partir de los titulares en noticias.
   * Parámetros:
     + **df:** DataFrame que contiene la columna 'Titulares'.
   * Salida:
     + Gráfico de nube de palabras
4. Módulo principal

Este módulo es el orquestador del trabajo, en primera instancia carga el archivo de configuración (config.json) para descargar los datos usando la API de Alpha Vantage, después realiza las transformaciones de datos necesarias para entrenar un modelo de aprendizaje profundo LSTM y analiza los resultados en las predicciones. El siguiente paso que toma es analizar los titulares más recientes de la compañía en la que se enfoca la predicción de la acción, usando técnicas de web scraping y análisis de sentimientos guarda los resultados finales donde da una perspectiva completa del mercado.

1. Conclusión

El uso de técnicas de extracción y almacenamiento de datos para la creación de inteligencia artificial ha hecho posible el desarrollo de múltiples soluciones por la comunidad de científicos de datos. La medición de resultados y progreso del algoritmo es tan importante como el desarrollo del modelo y es por lo que las técnicas de almacenamiento juegan un rol fundamental en el presente y futuro de la ciencia de datos.

1. LSTM (Long Short Memory) red neuronal recurrente para predecir secuencias en los datos. [↑](#footnote-ref-1)
2. Porcentaje de cambio en pesos de cada iteración conforme al ajuste en la función de error. [↑](#footnote-ref-2)