Reporte de Servicio Social.

Ángel Edmundo Hernández Martínez

1. Ecuación logistica y la complejidad

Para familiarizarnos con el estudio de la complejidad y el caos tratamos el caso de la Ecuación Logística. La ecuación logística es un modelo de crecimiento poblacional publicado por Pierre Verhulst. Que se expresa con la siguiente ecuación iterativa;

$$x_{t+1} = r \cdot x_t(1 - x_t), \text{ donde } 0 \leq x_0 \leq 1$$

Donde

2. Creación de una base de datos

2.1. Extracción de datos.

Para extraer los distintos datos de los distintos índices económicos se empleó la API de YahooFinance, conocida como Yfinance. Se tomó esta decisión porque la mayoría de las API's con las mismas funciones eran de paga, y, por otro lado, era más fácil obtener los datos de esta manera que extrayendo los datos de forma directa aplicando técnicas de Web Scrapping.

El uso de Yfinance es bastante simple, aún así creo esclarecedor una breve explicación de su funcionamiento. Yfinance se basa en un sistema de Tickets que son la forma de comunicarnos con el sitio Web de Yahoo Finance. Para emplear los tickets necesitamos de la clave con la que están designados los índices económicos, la cual puede ser consultada desde la página de YahooFinance. Por ejemplo, la clave del IPC MEXICO es "ˆMXX".

Con el ticket podemos consultar información del índice económico como sus acciones, sus dividendos, sus ganancias, etcétera. A continuación presentamos un ejemplo de código para consultar la información relacionada a un índice económico.

```
import yfinance as yf
IPC = yf.Ticker("^MXX")
# Muestra acciones (dividendos, splits)
IPC.actions
# Muestra dividendos
IPC.dividens
# Muestra splits
IPC.splits
# Existen mas metodos, que pueden ser consultados en la documentacion
```

Ahora con la posibilidad de obtener la información, necesitamos guardarla de algún modo. Y queríamos que fuera lo antes posible porque los datos con granularidad menor a un día en las series de tiempo se eliminaban después de 7 días.

La solución fue que mientras se diseñaba la base de datos, los datos se almacenaran en formato ".csv". Para ello hice un script que almacenaba los datos en la computadora y que se activaba de forma automática todos los días. Ahora comentaré las partes de dicho código, y explicaré la forma en que lo automaticé.

La primera parte del código es la importación de las funciones y librerias que ocuparemos, a saber;

```
import yfinance as yf
from pathlib import Path
from datetime import date, timedelta, datetime
import time
```

La librería `pathlib` sirve

La siguiente parte del código consiste en la enumeración de los tickets de los distintos índices económicos de los que queremos extraer la información.

```
SP500 = yf.Ticker("^GSPC") #S&P 500
Nasdaq =yf.Ticker("^NDX") #Nasdaq 100
Nasdaqcomp = yf.Ticker("^IXIC") #Nasdaq Composite
DWJ = yf.Ticker("^DJI") #Dow JOnes Industrial Average
IPC = yf.Ticker('^MXX') #IPC (BMV)
STOXX50E = yf.Ticker('^STOXX50E') #STOXX50
Nikkei = yf.Ticker('^N225') #Nikkei
CSI300 = yf.Ticker('000300.SS') #CS1300
```

```
tickets = [SP500, Nasdaq, Nasdaqcomp, DWJ, IPC, STOXX50E, Nikkei, CSI300]
ticketsname = ["SP500", "Nasdaq", "Nasdaqcomp", "DWJ", "IPC", "STOXX50", "Nikkei", "CSI300"]

#corregir

for i in range(7):
    start = datetime.today().replace(hour = 0, minute = 0, second = 0 ) - timedelta(days= i +1)
    end = datetime.today().replace(hour = 23) - timedelta(days = i +1)

        #agregar avg de open, high, low, close


    for i in range(len(tickets)):
        historial = tickets[i].history(start=start, end=end , interval="1m") #Obtener los tickets
        #Elegimos el path en el que se guardaran los csv, queremos que vaya a una carpeta adecuada a cada indicador
        filepath = Path('Info/{almacenamiento}/{name}_{date}.csv'.format(almacenamiento = ticketsname[i] ,name = tic
        filepath.parent.mkdir(parents=True, exist_ok=True)
        #guardar grafica
        historial.to_csv(filepath, header=True, index=True)
```

Por otro lado para la automatización del script ocupé el administrador de tareas de windows.

2.2. Diseño de base de datos

El diseño de una base de datos

2.3. Implementación y automatización de la base de datos

El plan original era

3. Análisis de distintos índices

3.1. Análisis general de los índices económicos

3.2. Análisis sobre complejidad de los índices económicos

3.2.1. Entropía aproximada

4. Bibliografía

https://algotrading101.com/learn/yfinance-guide/