Pre-presentación Proyecto Webscraping

Integrantes:

- Rodrigo Bruna Durán (GitHub: rbrunaduran)
- José Mora González (GitHub: josemoragonzalez)
- Miguel Abarzúa Ludueña (GitHub: MiguelAbarzua)

Contexto del Proyecto

UAI
UNIVERSIDAD ADOLFO IBÁÑEZ

Actualmente Chile es un país sumamente dependiente de otras naciones respecto de sus necesidades alimenticias. Esto se traduce en una debilidad para la economía pues frente a situaciones como la guerra en Ucrania o desastres naturales, se generan enormes variaciones de precio debido a la escasez. Frente a esta situación nos hemos enfocado en analizar la situación productiva chilena en alimentos lácteos, con la finalidad de conocer si esta actividad productiva tiene una tendencia positiva o negativa, y establecer un panel de control que permita monitorear estos comportamientos.

Con lo anterior queremos visibilizar esta situación dejando el análisis en un plataforma abierta para que cualquier ciudadano pueda conocer esta información en visualizaciones de rápido entendimiento.

Objetivos

UAI
UNIVERSIDAD ADOLFO IBÁÑEZ

- Obtener datos históricos sobre la producción de productos de origen lácteo mediante procedimiento scraping con lenguaje python en sitio web de ODEPA.
- Almacenar los datos encontrados en un formato tabular mediante lenguaje python y guardarlos en un archivo de formato .csv
- Visualizar la información mediante plots y/o visualizador BI para comprender desde la perspectiva gráfica los datos.

Pasos a seguir

UAI
UNIVERSIDAD ADOLFO IBÁÑEZ

- Analizar la web https://www.odepa.gob.cl/avance-mensual-y-series-de-tiempo-de-productos-por-planta-o-region-de-la-industria-lactea

- Crear un scraper que extraiga la información necesaria.
- Almacenar los datos en archivos csv.
- Generar visualizaciones públicas de actualización mensual usando streamlit.
- Generar un panel mediante Power BI para analizar interactivamente la información.

Pseudocódigo (1/4)

UAI
UNIVERSIDAD ADOLFO IBÁÑEZ

Importar librerías pandas, requests y BeautifulSoup

Creación de variables:

- URL ((tipo string) = https://aplicativos.odepa.gob.cl/avancemensual.do
- headers (tipo diccionario) = { }
- page (tipo string) = solicitud de almacenamiento del sitio web
- campos (tipo diccionario) = {'mesini': 'cboMesIni', 'anioini' : 'cboAgnoIni', 'mesfin' : 'cboMesFin', 'aniofin' : 'cboAgnoFin','producto' : 'cboProducto', 'region' : 'cboRegion'}
- parametros (tipo diccionario) = { }
- soup (tipo objeto html) = parsear(page)

comentario: definición variables para acotar años de revisión, definir productos y regiones a almacenar:

- Anios (tipo lista) = [min(parametros[anioini]), max(parametros[aniofin])]
- productos (lista) = parametros[producto]
- regiones (lista) = parametros[region]

Pseudocódigo (2/4)

UAI
UNIVERSIDAD ADOLFO IBÁÑEZ

Creación de funciones:

comentario: Función para obtener los parámetros de cada elemento del formulario

Función: get_element (Recibe: lista)
repositorio (tipo lista) = [ ]
Ciclo: desde x = 0 hasta largo (lista)
añadir en repositorio [valor de x, texto x]

comentario: obtención de parámetros del formulario y se guardan en el diccionario 'parametros'

Ciclo: desde key = 0, value = 0 hasta largo(campos.ítems)
temporal = igualdad entre id y value
valor = listado valores option de temporal
parámetros[key] = valor

Pseudocódigo (3/4)

UAI
UNIVERSIDAD ADOLFO IBÁÑEZ

comentario: ciclos para recorrer todos los parámetros y hacer los request

dffinal (tipo data frame) = pd.DataFrame( )

Ciclo: desde producto = 0 hasta largo(productos)
    Ciclo desde región = 0 hasta largo(regiones)
        Payload (tipo diccionario) = {'dataExport': '', 'compressed': 'false', 'fileNameExcel':'', 'decimales':'', 'cboMesIni': '01', 'cboAgnoIni': valor mínimo de año almacenado en la variable anios, 'cboMesFin': '06', 'cboAgnoFin': valor máximo de año almacenado en la variable anios, 'cboProducto': producto i en productos, 'rdoTipo': 'region', 'cboRegion': region i en regiones, 'rdoFormatoTabla': 'mesagno'}

        page = request ("POST", url, headers=headers, data=payload)

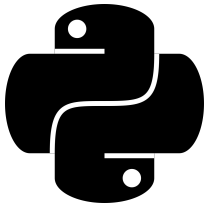
Pseudocódigo (4/4)

UAI
UNIVERSIDAD ADOLFO IBÁÑEZ

```
ciclo: si page != error entonces
                soup (tipo objeto) = parsear(page)
                columns (tipo lista) = [ ]
                data (tipo lista) = [ ]
                table = soup.encontrar('table')
                table_body = table.encontrar('tbody')
                table_head = table.encontrar('thead')
                fields = table_head.find_all('th')
                rows = table_body.find_all('tr')

                ciclo: desde field = 0 hasta largo(fields)
                        columns = agregar field.text

                ciclo: desde row = 0 hasta largo (rows)
                        cols = agregar filas en cada columna

                tabla = guardar como data frame (atributos data y columns)

guardar tabla en csv.
```

Entregables

- Archivo con columnas:
  - Meses: Meses del año en palabras (string).
  - Producto: Productos lacteos, ejemplo: leche condensada en litros (string).
  - Region: nombre de la regiones (string).
  - Año: Año de la producción (integer).
  - Valor: Cantidad producida de cada producto (integer).



- Archivo con código de scraping (.py).
- Archivo con código de plots (.py)
- Archivo con código para levantar una aplicación en Streamlit (.py).

- Archivo con dashboard (.pbix).