Tarea final sobre python de Software en Matemáticas

Pedro Antonio Fondevila Franco

January 2020

1 Introducción

Enmarcado en la asignatura de Software en Matemáticas del Máster en Matemáticas de la Universidad de Cádiz vamos a realizar un pequeño proyecto que englobe practicamente todos los puntos aprendidos durante las lecciones sobre Python.

En concreto proponemos realizar una clase que ayude al tratamiento estadístico de un base de datos. Hemos elegido la base de datos tempus3 del Instituto Nacional de Estadística debido a que estos datos ya están pretratados y no tendrán ruido por lo que no tendremos que realizar ningún tipo de filtrado cuando queramos presentarlos.

2 Primeros pasos

Afortunadamente ya existe una api para conectarse con tempus3 pero es necesario conocer de antemano una URL, para encontrar esta dirección tendremos que ir a la página Glosario códigos INE donde tendremos que elegir que tabla queremos hacer el tratamiento estadístico.

Los pasos a seguir son los siguientes, elegimos la tabla que queramos y generamos el link que genera el json necesario, en este caso sería link al json

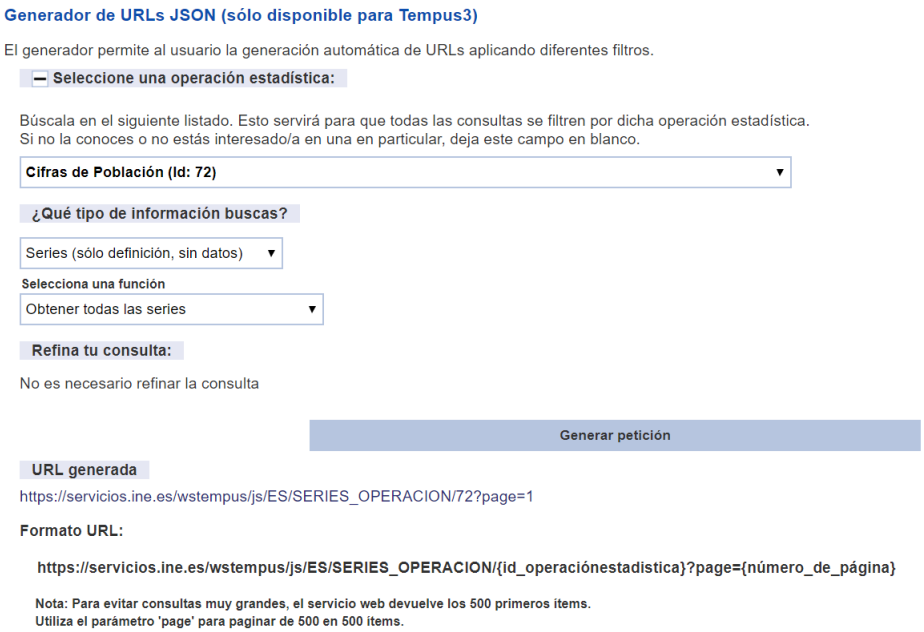

Figure 1: Elección de la serie.

Luego tenemos que fijarnos en el campo "COD" de la tabla en la que estemos interesados:

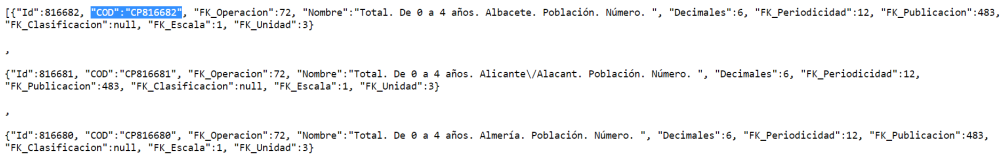

Figure 2: Busqueda del parámetro COD.

3 Demo IneApi

Una vez encontrado el código deseado ya podemos pasar a usar nuestra clase en python. Haremos una pequeña demo para demostrar que se puede hacer con la clase.

Primero de todo inicializaremos con los datos por defecto que vienen en la clase:

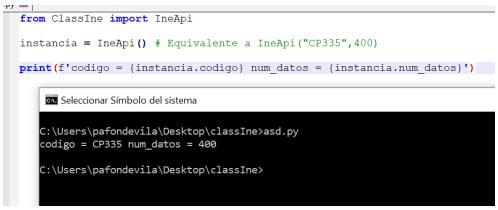

Figure 3: Inicializamos.

Descargamos los datos y extraemos la información de la tabla y pasamos a calcular la media y la desviación típica y a mostrar toda la información.

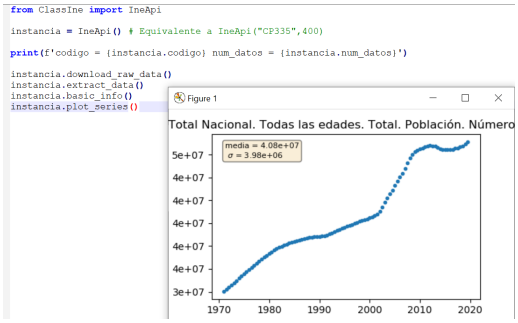

Figure 4: Pintamos la serie.

Por último vamos a hacer una regresión por mínimos cuadrádos, en este caso se hará sobre una recta pero se puede elegir cualquier polinomio con cualquier grado cambiando el parámetro grade de la función polinomial_regression.

Adicionalmente crearemos con esta regresión una previsión de 5 años vista, también se puede configurar los días que se quieren preveer con nuestra regresión actual.

```python
from ClassIne import IneApi

instancia = IneApi() # Equivalente a IneApi("CP335",400)

print(f'codigo = {instancia.codigo} num_datos = {instancia.num_datos}')

instancia.download_raw_data()
instancia.extract_data()
instancia.basic_info()
instancia.plot_series()

instancia.polinomial_regression(2)
instancia.prevision(365*5)
instancia.plot_series()
```

Figure 5: Mostramos la info

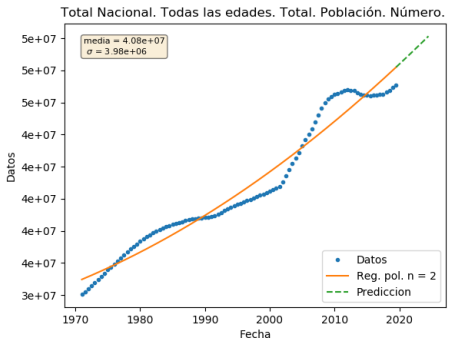

Figure 6: Mostramos toda la información generada.

4 Documentación

Pasamos a dar una explicación de las posibilidades que tenemos en la API:

- IneApi(codigo, num_datos)
  - Descripción: Inicializar una instancia.
  - Parámetros:
    - Opcional **codigo**: Código que hemos recuperado de la página del INE. Default = "CP335"
    - Opcional **num_dat**: Número de datos que se quiere descargar, descargará los últimos datos que se le indique. Si hay menos datos que num_dat se descargará toda la tabla. Default = 400
- download_raw_data()
  - Descripción: Realiza la descarga de los datos del INE con los datos previamente introducidos en **codigo** y **num_dat**.
  - Sin parámetros.
- print_raw_data()
  - Descripción: Por si se quiere inspeccionar el json directamente para mirar algún dato en bruto.
  - Sin parámetros.
- save_raw_data()
  - Descripción: Se usará para guardar los datos en bruto para su posterior uso. Se guardará en la carpeta actual con el nombre raw_data.txt
  - Sin parámetros.
- load_raw_data(filename)
  - Descripción: Se usará para cargar los datos en bruto.
  - Parámetros:
    - Obligatorio **filename**: Nombre del archivo que se va a cargar.
- extract_data(campo)
  - Descripción: Extraemos los datos y las fechas requeridas (se bajarán tantas fechas como diga num_dat).
  - Parámetros:
    - Opcional **campo**: Nombre en el json descargado del campo que queremos extraer. Default = 'Valor'
- basic_info()
  - Descripción: Calcula la media y la desviación típica.
  - Sin parámetros.

- get_mean()
  - Descripción: Devuelve la media calculada anteriormente por basic_info()
  - Sin parámetros.
- get_st_dev()
  - Descripción: Devuelve la desviacion típica anteriomente calculada por basic_info()
  - Sin parámetros.
- polinomial_regression(grade, points)
  - Descripción: Ajustaremos los datos por mínimos cuadrados al polinomio de grado **grade**.
  - Parámetros:
    - Opcional **grade**: Grado del polinomio al que se quiere ajustar la nube de puntos. Por defecto lo ajustará a una recta. Default = 1
    - Opcional **points**: Cantidad de puntos que se va a interpolar. Se crearán uniformemente en el rango de fechas recuperadas de la tabla del INE. Default = 1000
- prevision(days_into_future)
  - Descripción: A partir del polinomio calculado en polinomial_regression haremos una prediccion tan larga como **days_into_future** indique. Se realizará una predicción por día.
  - Parámetros:
    - Opcional **days_into_future**: Días que se quiere predecir a partir del último día de la serie descargada.

Parámetros:

- plot_series()
  - Descripción: Se pintará la información de la que se disponga en el momento de llamar a la función.
  - Sin parámetros.