
WEB SCRAPING

NIVEL 1: UNA SOLA PÁGINA ESTÁTICA CON REQUESTS Y BEAUTIFUL SOUP

EDUARD LARA

1. INDICE

1. Introduccion Beautiful soup
2. Extraccion StaBeautiful soup

1. INTRODUCCION

- Accederemos a datos de la web StackOverflow, usando
 - **Requests**, para el requerimiento
 - **Beautiful Soup**, para el parseo del árbol.
- Beautiful Soup funciona de manera parecida a LXML y provee funciones para encontrar elementos por Id, por clase...
- En **StackOverflow** podemos hacer preguntas acerca de programación y la comunidad de usuarios nos responde a nuestras preguntas

2. EXTRACCION STACKOVERFLOW

Paso 1. De la web <https://stackoverflow.com/questions> extraeremos el título y la descripción de cada una de las preguntas, pero sólo de la página principal.

0 votes


0 answers

2 views

tuning hyperparameters using baesian optimisation, optimiser not recognised

I am using Baesian optimisation to tune my model however however I am getting an error for all the optimisers. here is my...

keras deep-learning neural-network hyperparameters bayesian-deep-learning

 Beginner_coder 45 asked 17 secs ago

0 votes


0 answers

3 views

Android Live Edit pauses everytime I change even it's automatic and gives error message

whenever I change my kotlin code while the live edit is automatic. It Pauses and gives me this message: Compilation...

MD android kotlin android-jetpack-compose

 oguztemizel98 1 asked 47 secs ago

No entraremos en el tema de la paginación, que corresponde con el nivel 2

2. EXTRACCION STACKOVERFLOW

Paso 2. Importamos la librería **requests** para hacer los requerimientos. Utilizaremos el encabezado user-agent para identificarnos ante el servidor emulando que somos un navegador haciendo requerimientos. Si no puede detectar que somos un script de extracción de datos y banee la IP. También definimos la URL semilla para acceder a la página.

```
import requests

# USER AGENT PARA PROTEGERNOS DE BANEOS
headers = {
    "user-agent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, "+
    "like Gecko) Ubuntu Chromium/71.0.3578.80 Chrome/71.0.3578.80 Safari/537.36"
}

# URL SEMILLA
url = 'https://stackoverflow.com/questions'
```

2. EXTRACCION STACKOVERFLOW

Paso 3. Hacemos el requerimiento de la pagina web, lo guardamos en la variable respuesta y la imprimimos para ver todo el árbol HTML que nos devuelve.

```
# REQUERIMIENTO AL SERVIDOR
respuesta = requests.get(url, headers=headers)
print(respuesta.text)
```

```
StackExchange.ga.setDimension('dimension7', "1709884361.684374197");
```

```
StackExchange.ga.trackPageView();
```

```
});
```

```
</script>
```

```
<script src="https://cdn.cookie law.org/scripttemplates/otSDKStub.js" charset="UTF-8" data-document
language="false" data-domain-script="c3d9f1e3-55f3-4eba-b268-46cee4c6789c"></script>
```

```
</body>
```

```
</html>
```

2. EXTRACCION STACKOVERFLOW

Paso 4. Importamos la nueva librería BeautifulSoup para hacer el parseo del árbol HTML

```
from bs4 import BeautifulSoup
```

Se basa en un concepto de que tenemos una sopa de tags. Para parsear el árbol de la respuesta empezamos con:

```
# PARSEO DEL ARBOL CON BEAUTIFUL SOUP  
soup = BeautifulSoup(respuesta.text)
```

Como parámetro de entrada usamos la cadena de la respuesta y la almacenamos en la variable soup

A partir de esta variable, empezaremos a buscar los diferentes elementos

2. EXTRACCION STACKOVERFLOW

Paso 5. Inspeccionamos la pagina y miramos el título de las preguntas. Hacemos clic derecho inspeccionar elemento y se nos abre la consola de elementos.

```
▼<div id="questions" class=" flush-left">
  ▼<div id="question-summary-78147685" class="s-post-summary js-post-summary" data-post-id="78147685" data-post-type-id="1"> flex
    ▶<div class="s-post-summary--stats js-post-summary-stats">...</div> flex
    ▼<div class="s-post-summary--content">
      ▼<h3 class="s-post-summary--content-title">
        <a href="/questions/78147685/set-values-based-on-variable-keys-in-a-python-dictionary" class="s-link">Set values based on variable keys in a python dictionary</a> == $0
      </h3>
      ▶<div class="s-post-summary--content-excerpt">...</div>
      ▶<div class="s-post-summary--meta">...</div> flex
    </div>
  </div>
```

Vemos que el primer titulo esta dentro de un tag <a>, a su vez dentro de un <h3>, dentro de un div que se llama **s-post-summary--content**. Luego tenemos un div hermano **s-post-summary--stats** y un div padre más externo que tiene como id **question summary**.

2. EXTRACCION STACKOVERFLOW

Paso 6. A partir de aquí vamos viendo un patrón: un div con id questions, que tiene dentro todas las cuestiones con id question-summary.

```
▼ <div id="questions" class=" flush-left">
  ▶ <div id="question-summary-78147685" class="s-post-summary js-post-summary" data-post-id="78147685" data-post-type-id="1">... </div> flex == $0
  ▶ <div id="question-summary-78147684" class="s-post-summary js-post-summary" data-post-id="78147684" data-post-type-id="1">... </div> flex
  ▶ <div id="question-summary-78147683" class="s-post-summary js-post-summary" data-post-id="78147683" data-post-type-id="1">... </div> flex
  ▶ <div id="question-summary-78147679" class="s-post-summary js-post-summary" data-post-id="78147679" data-post-type-id="1">... </div> flex
  ▶ <div id="question-summary-78147674" class="s-post-summary js-post-summary" data-post-id="78147674" data-post-type-id="1">... </div> flex
  ▶ <div id="question-summary-78147673" class="s-post-summary js-post-summary" data-post-id="78147673" data-post-type-id="1">... </div> flex
```

2. EXTRACCION STACKOVERFLOW

Paso 7. Desde BeautifulSoup primero vamos a obtener el contenedor de todas las preguntas, con el div questions

```
contenedor_preguntas = soup.find(id="questions")  
#contenedor_preguntas = soup.find('div', id="questions")
```

Buscar por id es una estrategia muy segura, ya que el atributo id es único y no se puede repetir en ningún otro tag HTML a lo largo del árbol.

Nota: En el primer parámetro se puede poner el tag por el cual se quiere buscar, para asegurar la búsqueda (podemos poner el tag div o no)

2. EXTRACCION STACKOVERFLOW

Paso 8. A partir del contenedor de preguntas, vamos a obtener todos los hijos, cuya clase sea **s-post-summary**

```
lista_preguntas = contenedor_preguntas.find_all('div', class_="s-post-summary")
```

find_all nos va a encontrar todas las preguntas que estarán en un div con clase s-post-summary

2. EXTRACCION STACKOVERFLOW

Paso 9. Iteramos la lista de preguntas

```
for pregunta in lista_preguntas:
```

En cada una de estas iteraciones la variable pregunta va a ser un elemento HTML con una pregunta de Stackoverflow

```
<div id="questions" class=" flush-left">
  <div id="question-summary-78147685" class="s-post-summary js-post-summary" data-post-id="78147685"
    data-post-type-id="1"> flex == $0
    <div class="s-post-summary--stats js-post-summary-stats">...</div> flex
    <div class="s-post-summary--content">
      <h3 class="s-post-summary--content-title">
        <a href="/questions/78147685/set-values-based-on-variable-keys-in-a-python-dictionary" class="s-link">Set values based on variable keys in a python dictionary</a>
      </h3>
      <div class="s-post-summary--content-excerpt">...</div>
      <div class="s-post-summary--meta">...</div> flex
    </div>
  </div>
```

El titulo está dentro de un <a>, que está dentro de un tag <h3> que no se repite en ninguno de los otros tags div, a diferencia del tag <a> que si se repite

2. EXTRACCION STACKOVERFLOW

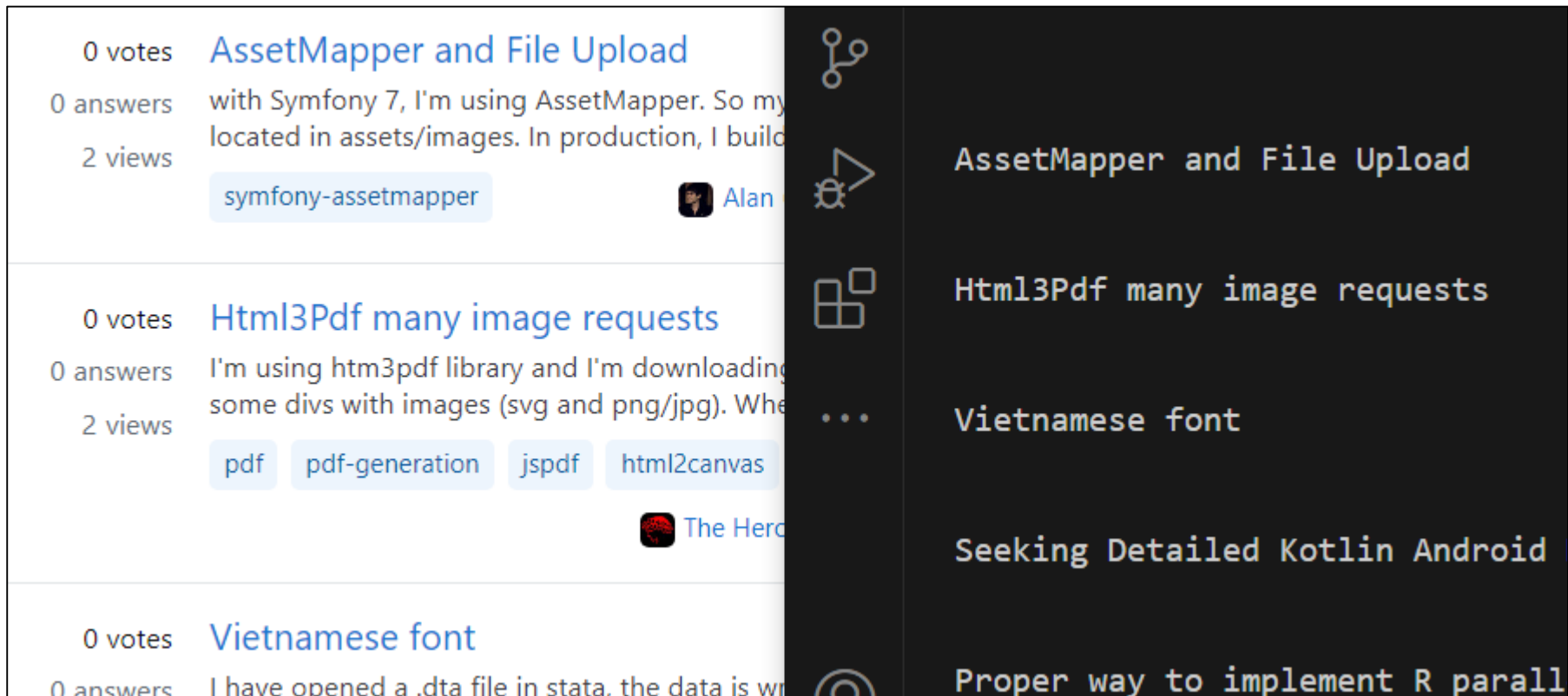
Paso 10. Para para sacar el título de la pregunta buscaremos el texto del tag h3. Esto devuelve un elemento HTML, pero si queremos encontrar el texto, tenemos que llamar a la función text para obtener el texto de un elemento.

```
for pregunta in lista_preguntas:
    texto_pregunta = pregunta.find('h3').text
    print (texto_pregunta)
```

Lo guardamos en la variable texto_pregunta y la imprimimos

2. EXTRACCION STACKOVERFLOW

Paso 11. Ejecutamos el código, y después de un warning, obtenemos los títulos de las preguntas de la primera pagina que se están actualizando constantemente



The image shows a web browser interface with a list of Stack Overflow questions on the left and a sidebar on the right. The questions on the left are:

- AssetMapper and File Upload**
0 votes, 0 answers, 2 views
with Symfony 7, I'm using AssetMapper. So my assets are located in assets/images. In production, I build...
symfony-assetmapper
Alan
- Html3Pdf many image requests**
0 votes, 0 answers, 2 views
I'm using htm3pdf library and I'm downloading... some divs with images (svg and png/jpg). Whe...
pdf, pdf-generation, jspdf, html2canvas
The Hero
- Vietnamese font**
0 votes, 0 answers
I have opened a .dta file in stata. the data is wi...

The sidebar on the right displays the titles of the questions in a dark theme:

- AssetMapper and File Upload
- Html3Pdf many image requests
- Vietnamese font
- Seeking Detailed Kotlin Android
- Proper way to implement R parall...

2. EXTRACCION STACKOVERFLOW

Paso 12. Para obtener la descripción, hacemos clic botón derecho y seleccionamos Inspeccionar elemento.

La descripción está dentro de un div con una clase que se llama **s-post-summary--content-excerpt**

```
▼ <div id="questions" class=" flush-left">
  ▼ <div id="question-summary-78148582" class="s-post-summary js-post-summary" data-post-id="78148582" data-post-type-id="1"> flex
    ▶ <div class="s-post-summary--stats js-post-summary-stats">...</div> flex
    ▼ <div class="s-post-summary--content">
      ▶ <h3 class="s-post-summary--content-title">...</h3>
      ▼ <div class="s-post-summary--content-excerpt"> == $0
        " When I run the CustomTranslatorApiSamples solution in Visual Studio to upload combofile, I get following error but I have successfully created workspace, project with the sample code. https://
        "
      </div>
      ▶ <div class="s-post-summary--meta">...</div> flex
    </div>
  </div>
```

2. EXTRACCION STACKOVERFLOW

Paso 13. En la variable pregunta buscaremos el texto de cualquier tag cuya clase sea igual a **s-post-summary--content-excerpt**

```
for pregunta in lista_preguntas:
    texto_pregunta = pregunta.find('h3').text

    descripcion_pregunta = pregunta.find(class_='s-post-summary--content-excerpt').text

    print (texto_pregunta)
    print (descripcion_pregunta)
    print
```

Imprimos la pregunta - descripción y vamos a poner un print vacío para hacer un salto de línea.

2. EXTRACCION STACKOVERFLOW

Paso 14. Ejecutamos el script y empezamos a recibir toda la información: titulo de la pregunta y descripción

```
How to create an excel budget to track funds [closed]
```

```
salaries will draw from this amount. Each employee works 160hrs a month. How do I create a table in excel to determine the month of which the budget ...
```

```
return provider count as zero
```

```
How do I return both PROV_ID's from a query below.  
454 returns 8 but 323 doesn't return any value since it is not in the table. I'd like it to show 0. thanks.
```

```
SELECT PROV_ID  
,COUNT(PROV_ID) AS PROVS  
FROM ...
```

2. EXTRACCION STACKOVERFLOW

Paso 15. Vemos que la descripción tiene algunos espacios. La formatearemos reemplazando todos los enters, saltos de línea por un espacio vacío, para hacerlo mas legible. Mediante el método strip() eliminamos los tabuladores y/o espacios que hay antes o después en una cadena.

```
for pregunta in lista_preguntas:
    texto_pregunta = pregunta.find('h3').text
    descripcion_pregunta = pregunta.find(class_='s-post-summary--content-excerpt').text

    descripcion_pregunta = descripcion_pregunta.replace('\n', ' ').replace('\r', ' ').strip()

    print (texto_pregunta)
    print (descripcion_pregunta)
    print
```

2. EXTRACCION STACKOVERFLOW

Paso 16. Ejecutamos de nuevo el programa y obtenemos el titulo de la pregunta y la descripción pregunta más bonito

```
Selenium error: java.awt.AWTException: headless environment
```

```
Can someone assist how to solve this issue (if possible to be solved)?I am using DevTools for getting necessary token (in all automation tests). Tests are written in Java (Selenium 4.16.1)In order ...
```

```
Upload Geotiff or Shapefile to Snowflake through streamlit
```

```
I am uploading a geotiff file from my streamlit app using st.file_uploader. How to store this file to snowflake DB. Is it possible to do so? fo rnow its geotiff but file type can be shape file also....
```

2. EXTRACCION STACKOVERFLOW

Resumen

- BeautifulSoup provee funciones muy útiles para parsear el árbol HTML y hacer consultas a los tags HTML
- Hemos buscado en todo el árbol cualquier tag que tenga el ID questions y lo hemos almacenado en un contenedor
- Luego hemos buscado dentro de ese contenedor, no en todo el árbol, los divs cuya clase sea **s-post-summary**
- Obtenemos una lista de preguntas, la cual iteramos
- En cada pregunta, buscamos el texto del tag h3 que es donde está el título, y el tag con clase **s-post-summary—content-excerpt** de donde obtengo la descripción
- Limpiamos la descripción (saltos de línea, espacios en blanco) para que se muestre de una manera más bonita₂₀

3. VENTAJA BEAUTIFULSOUP SOBRE LXML

Paso 1. Hasta ahora hemos hecho lo mismo con BeautifulSoup y Lxml: con funciones find hemos buscado y obtenido los elementos en uno y otro caso

¿Qué sucedería en el caso de que el div con la descripción no tuviera la clase **s-post-summary--content-excerpt**?

```
▼ <div id="question-summary-78148711" class="s-post-summary js-post-summary" data-post-id="78148711" data-post-type-id="1"> flex
  ▶ <div class="s-post-summary--stats js-post-summary-stats"> ... </div> flex
  ▼ <div class="s-post-summary--content">
    ▼ <h3 class="s-post-summary--content-title">
      <a href="/questions/78148711/keys-in-redis-instance-deleted-all-of-a-sudden-from-8gb-to-no-keys-at-all" class="s-link">Keys in Redis Instance deleted all of a sudden from 8gb to no keys at all</a>
    </h3>
    ▼ <div class="s-post-summary--content-excerpt"> == $0
      " We have Kubernetes cluster in which we have deployed a Redis instance using Helm chart version 9.4.3 , Disk is 8gb for persistence via AOF way. Image : docker.io/bitnami/redis:5.0.5-ubi8
      ... "
    </div>
    ▶ <div class="s-post-summary--meta"> ... </div> flex
  </div>
</div>
```

3. VENTAJA BEAUTIFULSOUP SOBRE LXML

Paso 2. Si este Div no hubiera tenido esa clase, una de las maneras para acceder a la descripción sería:

- Primero encontramos el tag H3
- A partir de él nos movemos hacia su tag hermano que es el div con clase **s-post-summary--content-excerpt**, cosa que BeautifulSoup permite hacer

3. VENTAJA BEAUTIFULSOUP SOBRE LXML

Paso 3. Quitamos el método `text()` de `find(h3)` para tener el contenedor de la pregunta para después poder buscar en esta variable y aplicar funciones.

En la variable `texto_pregunta` volvemos a dejar el texto del título

```
for pregunta in lista_preguntas:
    contenedor_pregunta = pregunta.find('h3')
    texto_pregunta = contenedor_pregunta.text
```

3. VENTAJA BEAUTIFULSOUP SOBRE LXML

Paso 4. A partir del contenedor de la pregunta, podemos movernos al siguiente tag hermano/primo donde esta la descripción, que se trata del div cuya clase es **s-post-summary--content-excerpt**. De aquí obtenemos `descripcion_pregunta` y de esto sacamos el texto. Con esto obtenemos exactamente el mismo resultado.

```
for pregunta in lista_preguntas:
    contenedor_pregunta = pregunta.find('h3')
    texto_pregunta = contenedor_pregunta.text

    descripcion_pregunta = contenedor_pregunta.find_next_sibling('div')

    texto_descripcion_pregunta = descripcion_pregunta.text
    texto_descripcion_pregunta = texto_descripcion_pregunta.replace('\n', '').replace('\t', '').strip()
    print (texto_pregunta)
    print (texto_descripcion_pregunta)
    print ()
```


3. VENTAJA BEAUTIFULSOUP SOBRE LXML

Paso 5. Lo ejecutamos y obtenemos el mismo resultado

How to use static library .lib in C++ VS?

We have open-source-project PrusiaSlicer github. It says: The slicing core is the libslic3r library, which can be built and used in a standalone way. I have built with CMake, and have libslic3r.lib .
..

Authentication state issues during hot reload in dart

After creation of a new user through admin profile (Only admin has rights to create profiles) after successfully storing in firebase if i do a hot reload the recently created user profile is logged in ...

Beautiful Soup permite recorrer el árbol, no sólo buscar por clase o por ID, sino también a partir de un elemento, ir hacia el siguiente tag hermano/primo directo de ese elemento. También permite ir al padre, ir al hijo, etc