VERSIÓN 1.0

26/01/2021

SCRAPING PROJECT

PROYECTO PARA LA ASIGNATURA AII, UNIVERSIDAD DE SEVILLA

FRANCISCO JOSÉ QUINTELA VELA

SCRAPING PROJECT

OBJETIVO

Este proyecto tiene como objetivo la experimentación con las tecnologías versadas durante la asignatura. Estas son: BeautifulSoup, Woosh, Django y Sistemas de Recomendación.

Para ello, intentaremos desarrollar una plataforma de recopilación y tratamiento de datos obtenidos de la empresa Zalando.

La idea desarrollada se basa en recopilar datos de los productos en ofertas de la web de Zalando para mostrar la información de forma más accesible, además de permitir funcionalidades como ordenar por tipo de producto, valoraciones y funcionalidades propias como recomendaciones basadas en usuarios.

El verdadero potencial de la idea reside en que podría ser escalable para recopilar información de múltiples webs de ofertas para ofrecer un catálogo unificado de productos en oferta que nos facilite encontrar el producto deseado al mejor precio.

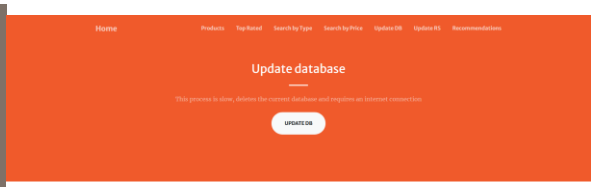
FUNCIONALIDADES Y PARTES DEL PROYECTO

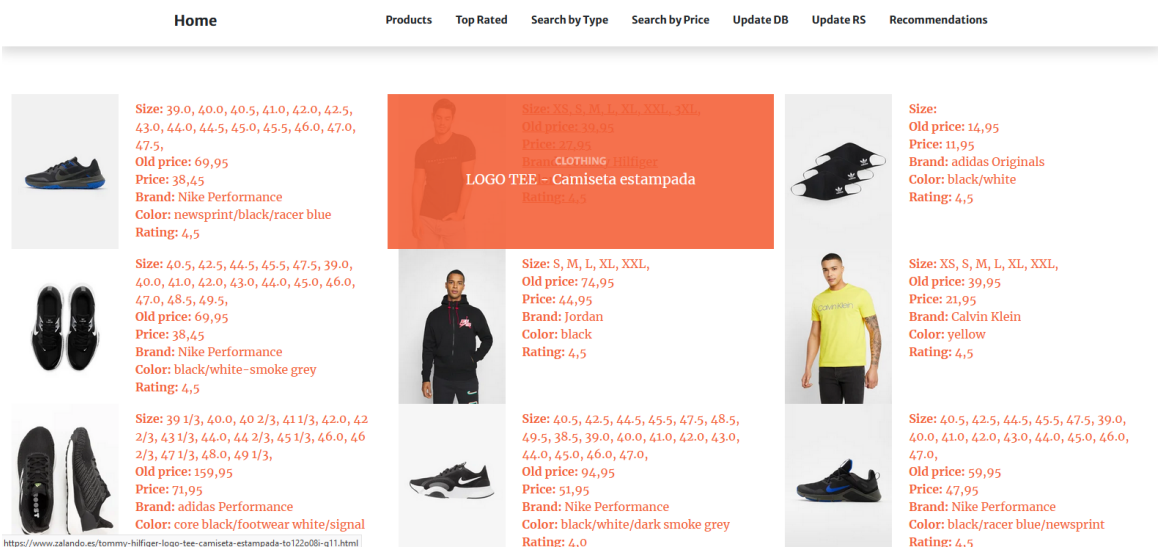
Herramienta	Funcionalidad
BeautifulSoup	Recopilación de información
Woosh	Almacenamiento de información y búsqueda con filtrado
Django	Sistema e interfaz web, tratamiento de información
Sistemas de recomendación	Recomendaciones basadas en valoraciones de usuarios

MANUAL DE USO

Para que la aplicación funcione se debe instalar Selenium con “pip install selenium” y cambiar la dirección del driver especificada en views.py por la propia del sistema en el que se ejecute. Posteriormente se ejecuta la aplicación con el comando “python ./manage.py runserver”

Una vez desplegada se carga la base de datos desde Update DB y el sistema de recomendación desde Update RS y ya estaría todo listo par probar las funcionalidades implementadas.





FRAMEWORK, VISTAS Y DESPLIEGUE DE APLICACIÓN WEB

Para la implementación del resto de funcionalidades en una aplicación web funcional haremos uso de Django, esta herramienta nos permitirá diseñal un proyecto haciendo uso de un modelo MVC:

- Modelos para almacenar la información
- Controladores que mapean las llamadas a urls y las procesan con la información requerida para su correcto funcionamiento
- Vistas que muestran la información deseada y procesada en los controladores.

RECOPILACIÓN DE INFORMACIÓN

Para recopilar la información de los productos utilizaremos la url zalando.es/hombre-rebajas/?sale=true y realizaremos la búsqueda de datos usando BeautifulSoup.

Dentro de dicha url conseguiremos los enlaces propios de cada producto e iteraremos por todos ellos entrando en la pagina individual de cada uno y consiguiendo sus datos.

Como Zalando no ofrece toda la información de sus productos en el código fuente básico de los productos necesitaremos hacer uso de la herramienta Selenium para simular la interacción con la web. Esto nos permitirá mostrar aquellos datos que no están visibles como son las tallas o las valoraciones de clientes.

Una vez hemos almacenado los datos de los productos creamos los objetos asociados a los modelos definidos para Products, Sizes, Ratings y UserInformation.

Posteriormente, utilizaremos Woosh para almacenar la información deseada dentro un documento generado con el esquema de datos necesario para las futuras funcionalidades.

Desde la web se accede usando la sección Update DB

El código puede consultarse dentro del proyecto adjunto a esta entrega en los archivos:

- views.py -> carga()

- views.py -> populateDB()

- views.py -> save_data()

- models.py

LISTADO DE INFORMACIÓN

Una vez disponemos de la información de los productos disponemos de dos funcionalidades simples para mostrarlos: listado simple y listado agrupado por valoración general.

Para el listado simple recogemos la información almacenada en Products y la mostramos iterando sobre los productos.

Para el listado agrupado recogemos Products y lo pasamos a la vista ordenados por valoraciones, una vez en la vista agrupamos la información y la mostramos.

Desde la web se accede usando la sección Products y Top Rated

El código puede consultarse dentro del proyecto adjunto a esta entrega en los archivos:

- views.py -> list_products()

- views.py -> list_products_top_rated()

TRATAMIENTO DE LA INFORMACIÓN

Hacemos uso de Woosh para dos métodos de filtrado de información.

Primero tenemos la opción de buscar por tipo de producto, lo que nos permite diferenciar entre productos de tipo ropa, zapato e indefinidos. El tipo de producto es obtenido tras clasificarlo en función de las tallas ofrecidas ya que la web Zalando no disponía de este atributo.

Para hacer uso de esta funcionalidad primero se procesa un formulario que lista los tipos de producto existentes y filtra la información almacenada por Woosh en función de la entrada introducida.

La otra funcionalidad consiste en buscar por rango de precio, estableciendo umbral menor y umbral mayor y filtrando la búsqueda para productos cuyo precio actual se encuentre dentro de esos umbrales. El funcionamiento es similar al anterior pero el formulario a procesar permite introducir un rango de precios con el formato “XX YY”.

Desde la web se accede usando la sección Search by Type y Search by Price.

El código puede consultarse dentro del proyecto adjunto a esta entrega en los archivos:

- views.py -> search_products_by_type()

- views.py -> search_products_by_price_interval()

SISTEMAS DE RECOMENDACIÓN

Además las funcionalides anteriores se ha implementado un sistema de recomendación colaborativo basado en las valoraciones que los distintos usuarios de la web han realizado sobre los productos.

Estas valoraciones se guardan asociando el valor numérico de la valoración, el producto valorado y el usuario que deja la valoración. Con esta información cargamos un diccionario que guarde esta relación de valoraciones.

Con esta información podemos ofrecer una funcionalidad que permite al usuario introducir el id de un usuario almacenado y recibir recomendaciones basadas en usuarios con valoraciones similares a a las suyas.

Como la web Zalando no contaba con suficientes datos de valoraciones para asegurar el correcto funcionamiento del sistema de recomendación hemos utilizado un usuario de prueba que deja valoraciones en todos los productos y así poder favorecer las coincidencias.

Desde la web se accede usando la sección Update RS (para cargar el diccionario) y Recommendations.

El código puede consultarse dentro del proyecto adjunto a esta entrega en los archivos:

- views.py -> loadRS()

- views.py -> loadDict()

- views.py -> recommendedProductUser()

-recomendations.py -> getRecommendations()