پروژه بازیابی اطلاعات

زهرا احمدی

**سوال 1:**

* در سوال 1 قسمت اول با استفاده از روش زیر crawler نوشتم:
* ابتدا HTMLSession , csv , nest\_asyncio را IMPORT میکنیم.
* سپس در قسمت ارکایو سایت عصر ایران تاریخ را فیلتر کرده و از URL آن در searchURL استفاده میکنیم.
* سپس (.news\_nav.news\_id\_c و div.header\_pdate و h1.title a[title] و div.news\_path a و div.subtitle) را از طریق کد :

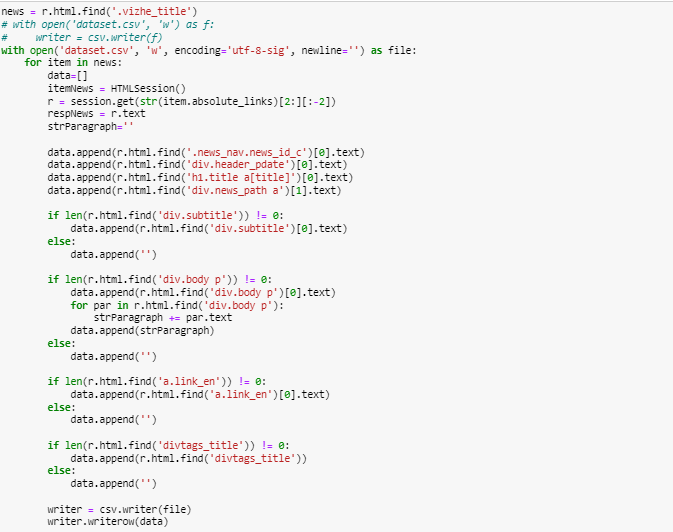
پیدا میکنیم و سپس از طریق روش زیر آن را داخل یک فایل CSV میریزیم:

* writer = csv.writer(file)



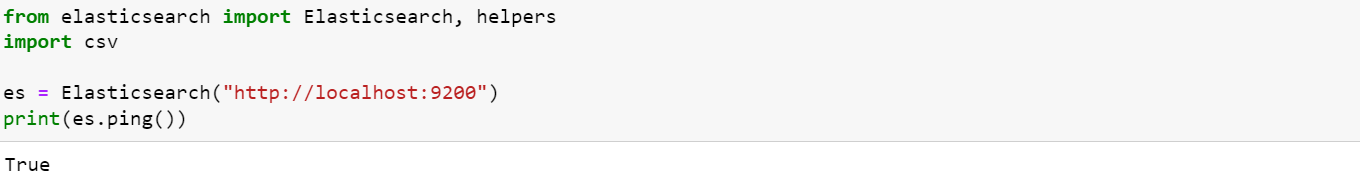
Asyinc ، text کل html است و با with open فقط اطلاعات هر خبرو میخونیم و با r=session.get… لینک مربوط به هر خبر و میگیرم و باز هم html است.

و بعد از html اون فیلدهایی که میخوایم و میکشیم بیرون و بعدش write میکنیم.



پس از نوشتن کرولر و استخراج اطلاعات سوالات را پاسخ میدهیم

در سوال اول ابتدا Elastic Search را Load میکنیم. سپس ping میگیریم که مطئن بشیم متصل است.



در سوال 1 قسمت دوم (سوال 4) با استفاده از روش زیر dataرا در elasticsearch، index کردم و روی آن query زدم:

سپس ParsiAnalyzer-7.13.1 را هم نصب کرده و سپس از طریق کد زیر فایل csv ای که از مرحله قبل استخراج نموده ایم را index میکنیم:

برای قسمت index کردن فایل csv از کد زیر استفاده میشود:



برای سوال 1 قسمت ب : سپس برای سوال اول که جایگاه کلمات در متن را بدهد به شرح زیر عمل کردم:

ابتدا از کاربر یک کلمه میگیریم و سپس با کد زیر کلمه را داخل متن خبر سرچ میکنیم:

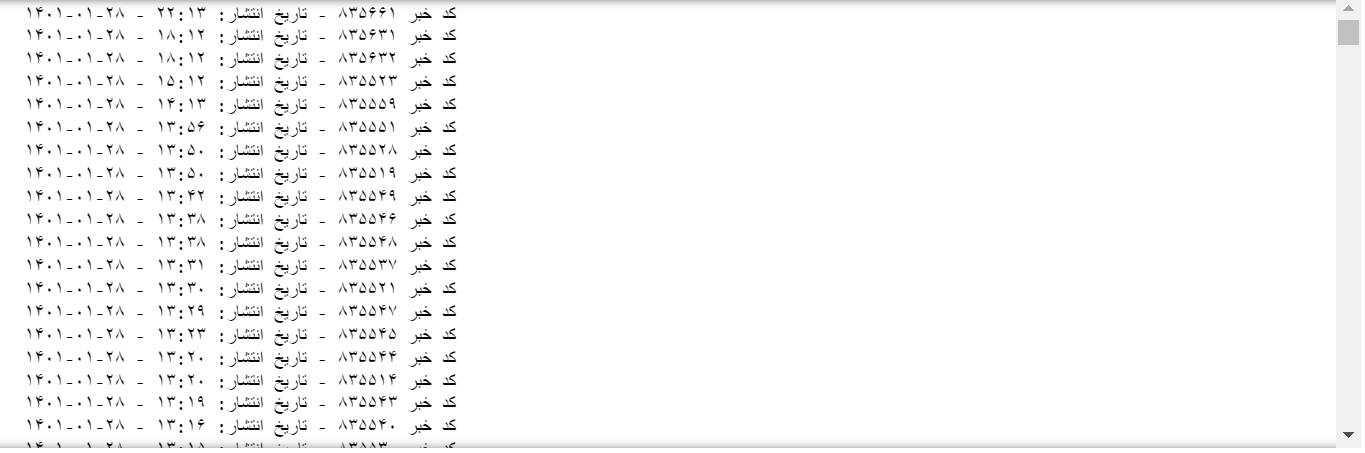


سپس برای سوال دوم که باید کد خبر های دارای تارخ 12-13 را بدهد :

با persian.convert… تمام فیلد های عددی را به انگلیسی تبدیل میکنیم تا بتوانیم روش سرچ بزنیم و بعد میایم روش find میزنیم.



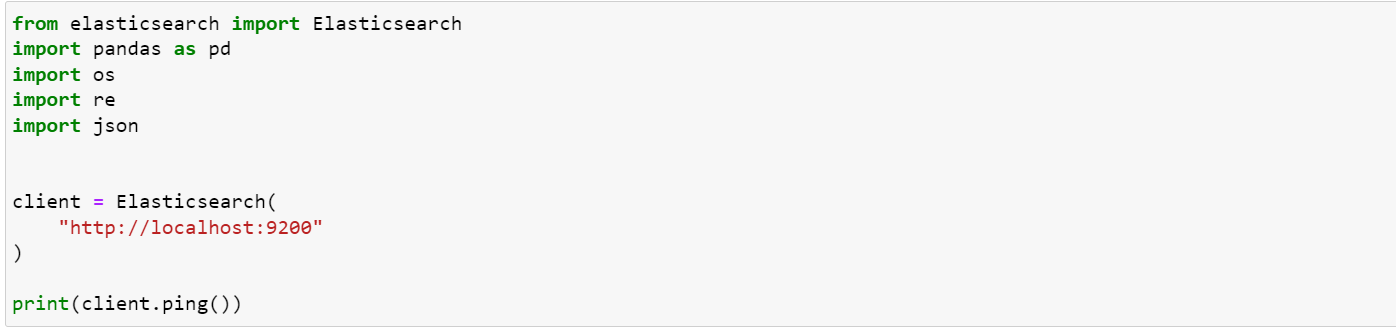
که خروجی زیر را میدهد:



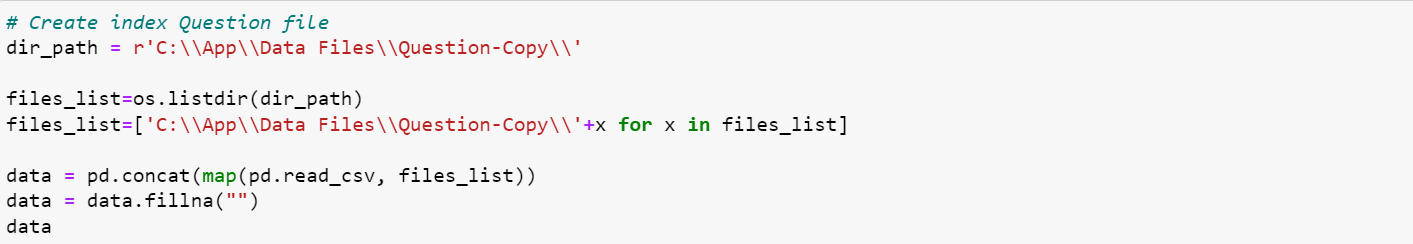
**سوال2:**

ابتدا elasticsearch را متصل میکنیم و ping میگیریم که مطمئن بشیم وصله یا نه.

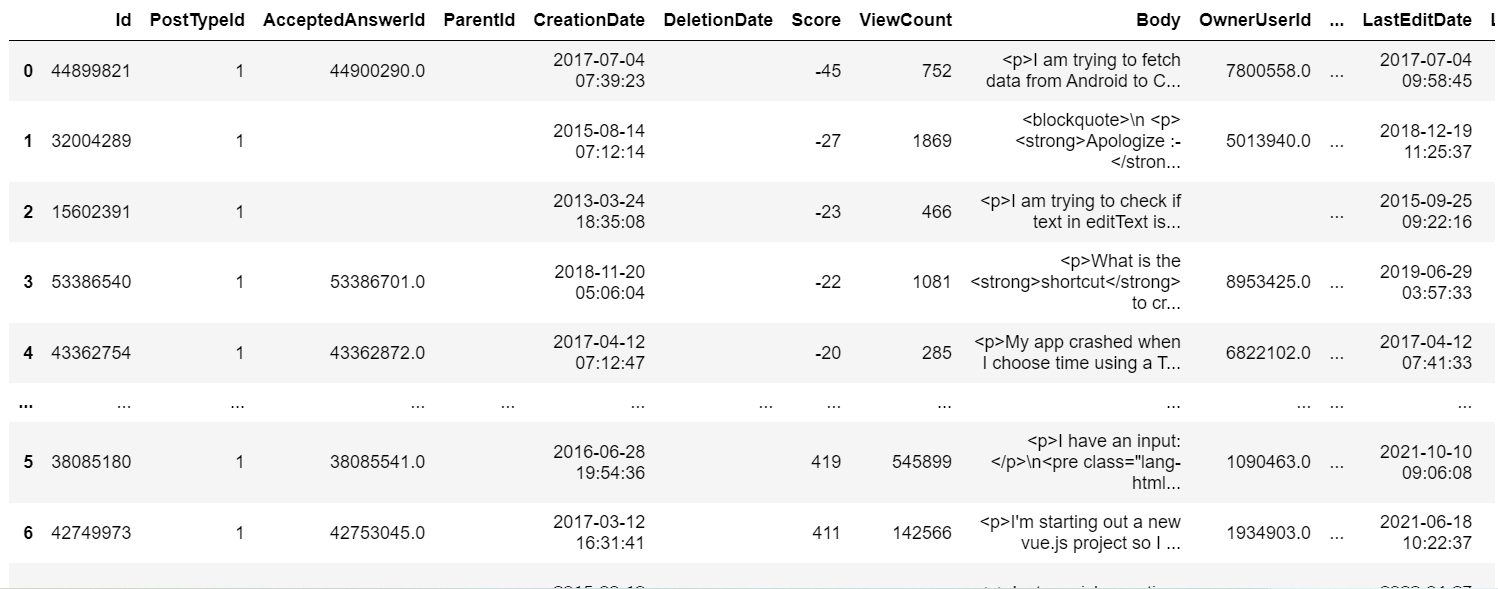
سپس ادرس اصلی فایل ها را میدهیم و سپس با لیستشون و در میارم بعدش با اضافه کردن مسیر اصلی به اول هر فایل لیست فایل ها رو در میاریم.



سپس فایل های گرفته شده را index میکنیم



سپس خروجی به شکل زیر است:



سپس از طریق کد زیر از طریق doc یه object میسازیم .

سپس با iterrow میایم رکورد به رکورد اطلاعات و میخونیم و یه doc میسازیم برای هر سربرگ و اطلاعات آنها.

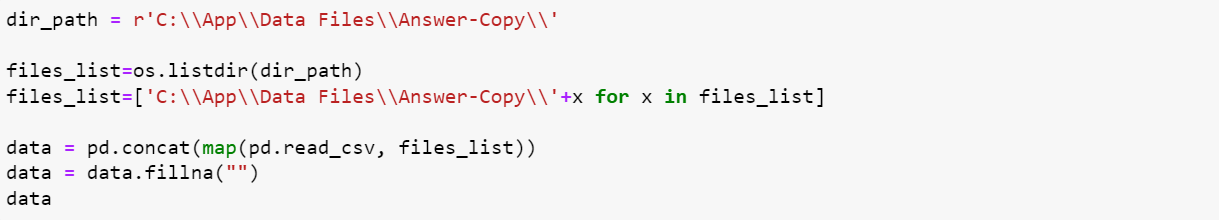
سپس اطلاعات id , row و میریزیم تو index ها

سپس با map(pd.read\_csv, files\_list) میایم لیست و میخونیم و بعد از طریق تابع read\_csv بریزیم تو data

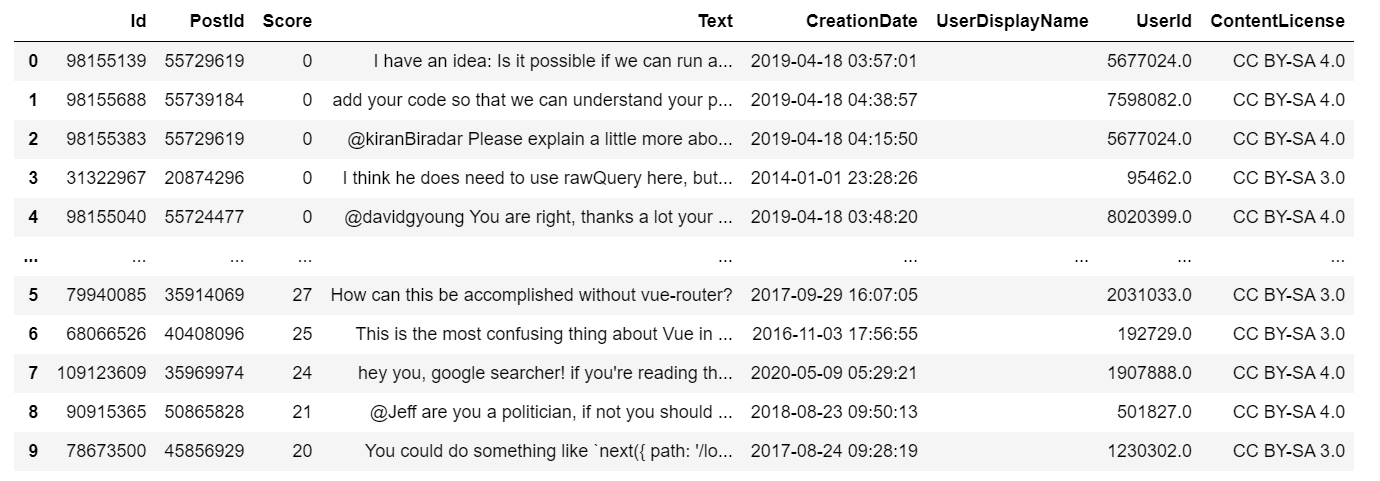
سپس میایم یه syntax list در میاریم از مواردی که تو برنامه نویسی استفاده میشه و میایم کوئری میزنیم .



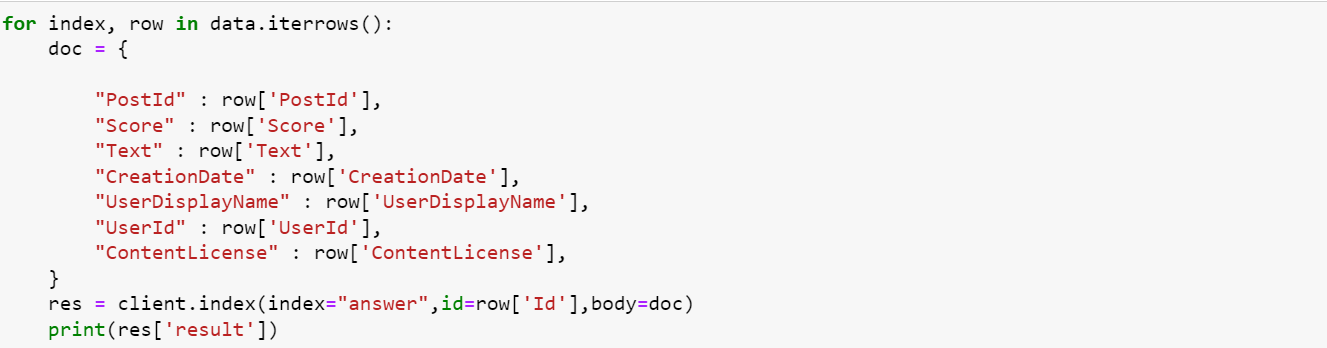
سپس از طریق کد زیر فایل های جدا جدا question را index میکینیم داخل الستیک سرچ:



سپس خروجی زیر را میدهد:



همچنین از طریق کد زیر فایل های جدا جدا answer را index میکینیم داخل الستیک سرچ:

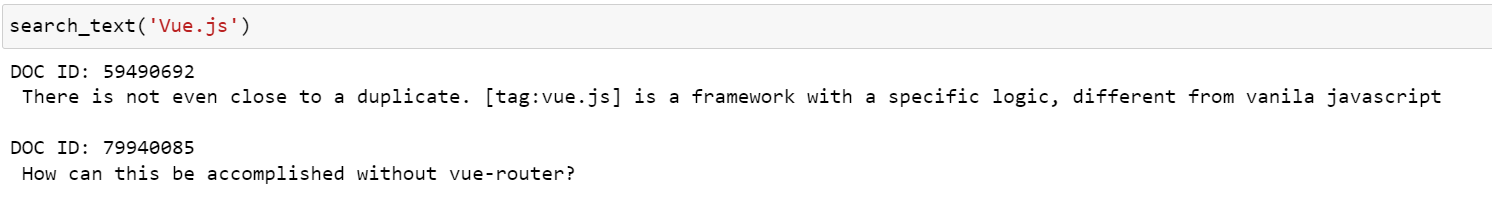


سپس برای کوئری هایی که ابتدا جواب های دارای کد و سایت و سپس فقط کد یا سایت و سپس هیچکدام را نشان دهد از کد زیر استفاده میشود که یک نمونه زده شد و خروجی زیر را داد:

Result[hits] [hits] رکورد هایی که پیدا شده را برمیگرداند.

بعد میندازیمش تو حلقه for،id هر رکوردی که به دست اورده و میریزه تو num





**سوال 3:**

در اینجا ابتدا 2 تا فایل tag , tagname ساختیم که یکیش تگای خالیه یکیش id , count , tagname ,… هست.

تو خط totaldata.fillna به رشته خالی تبدیل کردم.

بعد csvdata ردیف هایی که قراره بزاریم و تعریف میکنیم.

بعد یه for میزنیم که بیاد رکورد به رکورد دیتا فایل تگ و خوندیم اول اسمشو خونیدم و تگ والدی که تعریف کرئیم و تو for بعدی استفاده میکنیم.

حال اسمی که داریم و تو total tag سرچش میکنیم و id تک رکود و برمیگردونیم

توbaseurl از stackexchange.com یه api هست که اگه هر تگی و بهش بدیم تگ های مرتبط و باهاش و نشون میده

اومدیم ریکوست دادیم و json دریافت کردیم.

برای tagname اصلی id , name و در میاریم

و بعد از طریق parent id , name تمام related tag هاشو در میاریم.

Time.sleep به خاطر اینه که تند تند درخواست ندیم اینو گذاشتیم که هر یک دقیقه یک بار درخواست بده که ip , block نکنه.

