به نام خدا

محمدحسین ارسلان تمرین دوم کدی هوش مصنوعی و سیستم های خبره 98243005

این فایل گزارشی از نحوه کار من برای حل و تکمیل نوتبوک این سری تمرین می باشد، به اختصار به توضیح هر بخش (سوال های Q1 تا Q4 که در فایل موجود است) می پردازیم بهمراه قطعه کد هر بخش.

بخش های ابتدایی که حاوی توضیحات و خواندن دیتاست و ساخت parquet فایل ها بود را با مطالعه سپری کردیم :)

Question 1

در این سوال ابتدا دیتافریم را براساس clientهای منحصر به فرد دسته بندی کرده و سپس تعداد هر دسته را با کمک aggregate فانکشن محاسبه کرده ایم سپس نام ستون بدست آمده را تعیین کرده ایم.

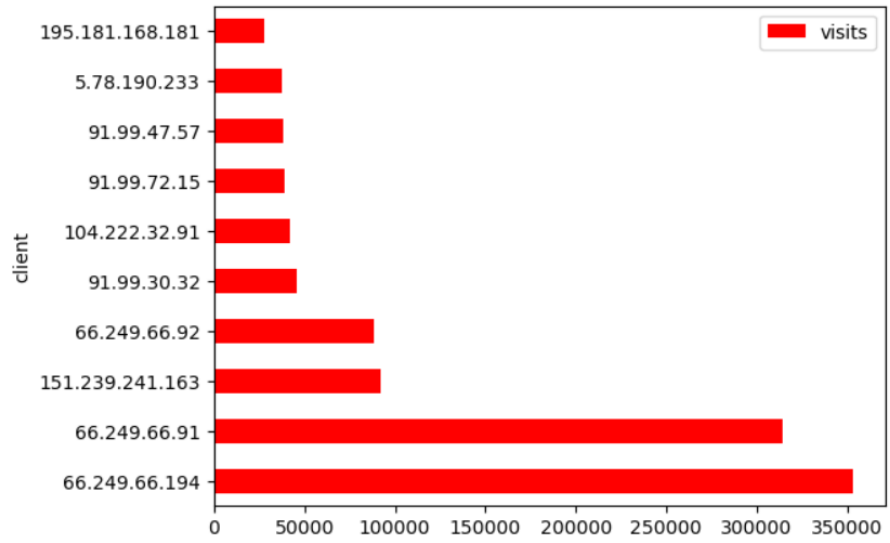
```
df.groupby('client').agg({"client":'count'}).rename(columns={'client':'visits'})
```

مشکلی که برای نمایش نمودار به کمک داکیومنتیشن ارائه شده در نوتبوک داشتم این بود که نمی توانستم برای هر سطر نمودار IP مختص هر کلاینت را قرار دهم و از دستور زیر برای رسم نمودار استفاده کردم.

```
top_n_visited_clients.plot.barh(color = {"visits":"red"})
```

پیش از این دستور هم به تعداد n تا از head دیتافریم سورت شده ایجادی (تعداد بازدید هر کلاینت) در دیتافریم بازگشتی این تابع قرار دادم.

client	visits
66.249.66.194	353483
66.249.66.91	314522
151.239.241.163	92473
66.249.66.92	88332
91.99.30.32	45973
104.222.32.91	42058
91.99.72.15	38694
91.99.47.57	38609
5.78.190.233	37203
195.181.168.181	27979



Question 2

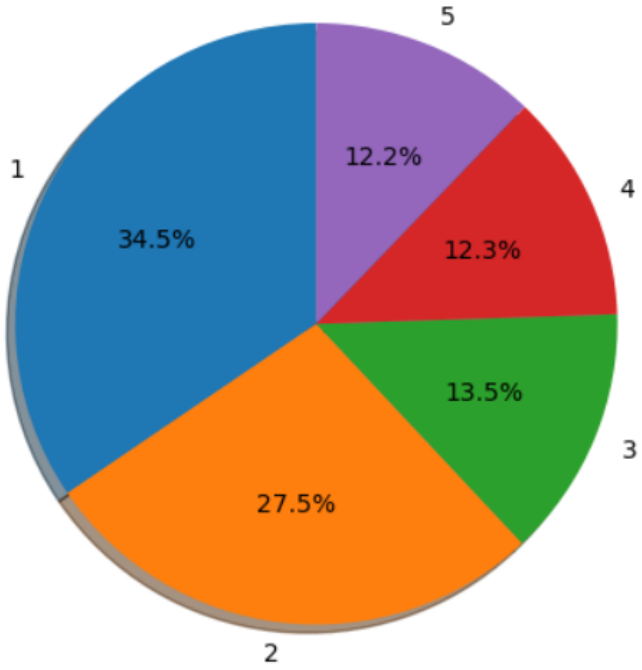
برای سوال دوم از ما خواسته شده بود که روی دسته بندی requestها پردازش کنیم و منتخب request ها را دریابیم و با کمک نمودار دایره رسم کنیم، تا قبل از مرحله رسم نمودار عینا سوال یک پیش رفتیم منتهی این بار دسته بندی به requestها وابسته بود.

سپس به کمک داکیمونتیشن ارائه شده متغیرهای موردنیاز نمودار دایره ای را ساختیم و سپس رسم کردیم.

```
labels = list(range(1, n+1))
    sizes = _list
    fig1, ax1 = plt.subplots()
    ax1.pie(sizes, labels=labels, autopct='%1.1f%%',
    shadow=True, startangle=90)
    ax1.axis('equal')
```

لیبل ها به تعداد nتا است و سایرتنظیمات مانند داکیومنتیشن می باشد.

request	number of requests
/settings/logo	352047
/static/css/font/wyekan/font.woff	280176
/static/images/guarantees/bestPrice.png	138010
/static/images/guarantees/fastDelivery.png	125689
/static/images/guarantees/warranty.png	124127



Question 3

در این سوال باید یک سری تمیزکاری روی داده های درخواستی توسط کلاینت ها انجام دهیم که با یکسری ریجکس مشابه خط زیر پاکسازی موردنظر صورت سوال را بر روی فایل هایی که پسوند دلخواه را دارند انجام دادم.

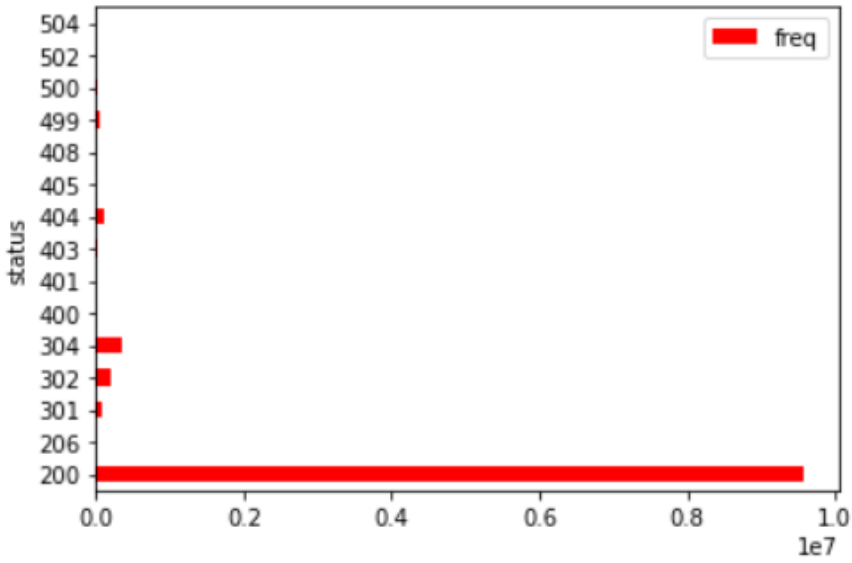
```
logs_df = logs_df.replace(to_replace='\?.*.jpg', value='.jpg', regex=True)
logs_df = logs_df.replace(to_replace='.jpg.*', value='.jpg', regex=True)
```

خط اول تا پیش از فرمت و بعد علامت سوال را و خط دوم عبارات پرت بعد از فرمت را پاک می کند. سپس با یک ریجکس دیگر و تابع count و sum تعداد دفعات دیده شدن فرمت موردنظر را شمرده و جمع میزنیم و تقسیم بر تعداد request ها می کنیم.

Question 4

برای این سوال باید ابتدا Status ها را مشخص می کردیم و سپس با یک گروه بندی روی statusها مشخص می کردیم پرتکرارترین Status ها چه بوده است که اینکار را با گروه بندی بر اساس Statusها و سپس تبدیل کردن این گروه بندی به یک List و سپس تبدیل لیست به Set یک مجموعه داشتیم از Status های منحصر به فرد. سپس رسم نمودار را داشتیم که به سبک و سیاق سوال اول این کار را هم پیش بردیم.

```
[200, 206, 301, 302, 304, 400, 401, 403, 404, 405, 408, 499, 500, 502, 504]
```



در سوال دوم هدف جداکردن Status هایی بود که حاوی پیام های ارائه خطا به کاربر بود. گفته شده بود که کدهایی که با 4 و یا 5 آغاز می شوند بیانگر ارور می باشند. ابتدا به کمک تابع to_datetime ساعات داده شده در دیتاست را به یک ساعت خاص تبدیل کردیم. سپس یک دیتافریم به نام four_xx ساختیم که در آن استاتوس های بیشتر از 400 را فیلتر کردیم و قرار دادیم و سپس فیلتر کمتر از 500 را قرار دادیم. حال همه ارورهایی که با 4 شروع می شوند را داریم.

همین کار را با دیتافریم five_xx انجام دادیم. سپس یک گروه بندی براساس ساعت ها روی این دیتافریم ها داشتیم. و سپس یک دیتافریم ساختیم که ستون اول آن ساعت ها هستند و در ستون دوم و ستون سوم یک لیست نمایش دادیم که تعداد ارور ها به ترتیب ساعت ها هستند.

	4xx	5xx
00	6021	17
01	4215	10
02	2394	8
03	1388	26
04	1871	3
05	1683	10
06	2036	3
07	2923	6
08	4985	76
09	7574	220
10	9117	124
11	10102	255
12	10395	429
13	10095	528
14	10584	597
15	10513	581
16	9823	621
17	9351	202
18	8979	4816
19	8302	6550
20	7758	36
21	7019	6
22	7342	22
23	7786	21

Question 5

صرفا یک set_index صورت گرفته):

```
                                                              datetime  \
client         user_agent
37.152.163.59  Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0...       12
               Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0...       12
85.9.73.119    Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleW...       12
37.152.163.59  Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0...       12
85.9.73.119    Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleW...       12
...                                                                    ...
86.104.110.254 Mozilla/5.0 (iPhone; CPU iPhone OS 12_1 like Ma...       16
5.125.254.169  Mozilla/5.0 (iPhone; CPU iPhone OS 12_1_2 like ...       16
65.49.68.192   Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64...       16
5.125.254.169  Mozilla/5.0 (iPhone; CPU iPhone OS 12_1_2 like ...       16
65.49.68.192   Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64...       16

                                                              method  \
client         user_agent
37.152.163.59  Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0...    GET
               Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0...    GET
85.9.73.119    Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleW...    GET
37.152.163.59  Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0...    GET
85.9.73.119    Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleW...    GET
...                                                                  ...
86.104.110.254 Mozilla/5.0 (iPhone; CPU iPhone OS 12_1 like Ma...    GET
5.125.254.169  Mozilla/5.0 (iPhone; CPU iPhone OS 12_1_2 like ...    GET
65.49.68.192   Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64...    GET
5.125.254.169  Mozilla/5.0 (iPhone; CPU iPhone OS 12_1_2 like ...    GET
65.49.68.192   Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64...    GET
```