```
#!/usr/bin/python
import numpy as np
data_base = []
hosts = []
freq_final = []
freq_200_final = []
freq_302_final = []
freq_404_final = []
perc_succ_req = []

def feature_extraction():
    #Loading the file
    file_obj = open('sample.txt','r+')

    i = 0
    for line in file_obj:
        record = {}
        list_1 = line.split(' - - ')
        list_2 = list_1[1].split("")
        list_2[0] = list_2[0].strip()
        list_2[2] = list_2[2].strip()
        list_3 = list_2[2].split()
        list_2.remove(list_2[2])
        list_1.remove(list_1[1])
        list_info = list_1 + list_2 + list_3
        i = i + 1
        record['serial_no'] = i
        record['host'] = list_info[0]
        record['timestamp'] = list_info[1]
        record['request'] = list_info[2]
        record['HTTP_reply_code'] = int(list_info[3])
        record['Size'] = list_info[4]
        data_base.append(record)
    file_obj.close()

def host_seperation():
    #Seperating hosts

    for i in range(len(data_base)):
        hosts.append(data_base[i]['host'])


def freq_of_HTTP_req():
    #Frequencies

    freq_set = set()

    for i in range(len(hosts)):
        freq_set.add((hosts[i],hosts.count(hosts[i])))
```

```
    freq_set = list(freq_set)




    for i in range(len(freq_set)):
        freq_final.append(freq_set[i][1])


def freq_unique_url():
    #Freq unique URL
    URL_requests = []
    uniq_url = []
    for i in range(len(data_base)):
        URL_requests.append(data_base[i]['request'])
    for i in range(len(URL_requests)):
        if URL_requests.count(URL_requests[i]) == 1:
            uniq_url.append(URL_requests[i])
    freq_uniq_URL = len(uniq_url)
    return (freq_uniq_URL)

def freq_200_resp():
    #hosts whose reply code was 200
    freq_200 = []
    for i in range(len(data_base)):
        if data_base[i]['HTTP_reply_code'] == 200:
            freq_200.append([data_base[i]['host'],data_base[i]['HTTP_reply_code']])
    freq_200_set = set()
    for i in range(len(freq_200)):
        freq_200_set.add((freq_200[i][0], freq_200.count(freq_200[i])))
    freq_200_set = list(freq_200_set)

    for i in range(len(freq_200_set)):
        freq_200_final.append(freq_200_set[i][1])


def freq_302_resp():
    #hosts whose reply code was 302
    freq_302 = []
    for i in range(len(data_base)):
        if data_base[i]['HTTP_reply_code'] == 302:
            freq_302.append([data_base[i]['host'],data_base[i]['HTTP_reply_code']])
    freq_302_set = set()
    for i in range(len(freq_302)):
        freq_302_set.add((freq_302[i][0], freq_302.count(freq_302[i])))
    freq_302_set = list(freq_302_set)
```

```
    for i in range(len(freq_302_set)):
        freq_302_final.append(freq_302_set[i][1])

def freq_404_resp():
    #hosts whose reply code was 404
    freq_404 = []
    for i in range(len(data_base)):
        if data_base[i]['HTTP_reply_code'] == 404:
            freq_404.append([data_base[i]['host'],data_base[i]['HTTP_reply_code']])
    freq_404_set = set()
    for i in range(len(freq_404)):
        freq_404_set.add((freq_404[i][0], freq_404.count(freq_404[i])))
    freq_404_set = list(freq_404_set)

    for i in range(len(freq_404_set)):
        freq_404_final.append(freq_404_set[i][1])

def Percentage_diff_req():
    #Percentage of different request made
    total_freq = 0
    for i in range(len(hosts)):
        temp = {}
        temp[hosts[i]] = hosts.count(hosts[i])
        total_freq += hosts.count(hosts[i])
    perc_diff_req = round((((freq_unique_url()/total_freq)*100)),2)
    return (perc_diff_req)


def Percentage_succ_resp():
    #Percentage of successful response received by a user
    freq_200 = []
    for i in range(len(data_base)):
        if data_base[i]['HTTP_reply_code'] == 200:
            freq_200.append([data_base[i]['host'],data_base[i]['HTTP_reply_code']])
    freq_200_set = set()
    for i in range(len(freq_200)):
        freq_200_set.add((freq_200[i][0], freq_200.count(freq_200[i])))
    freq_200_set = list(freq_200_set)
    temp_freq = set()
    for i in range(len(data_base)):
        if data_base[i]['HTTP_reply_code'] == 200:
            temp_freq.add((hosts[i],hosts.count(hosts[i])))
    temp_freq = list(temp_freq)

    for i in range(len(temp_freq)):
        for j in range(len(temp_freq)):
            if (temp_freq[i][0] == freq_200_set[j][0]):
                perc_succ_req.append(round(((freq_200_set[j][1]/temp_freq[i][1])*100),2))
l=len(freq_final)
mat=np.zeros((35,5))
```

```
def output_display():
    print ('Features extracted -- Host, Timestamp, Request, HTTP_reply_code, Size in bytes\n')
    for i in range(len(data_base)):
        print (data_base[i])
    print ('\nExtra features\n')
    print ('Freq of HTTP requests --\n')
    for i in range(len(freq_final)):
        print (freq_final[i])
        mat[i,0]=freq_final[i]
    print ('\nFreq of unique HTTP request -- ', freq_unique_url())
    print ('\nFreq of 200 code response --\n')
    for i in range(len(freq_200_final)):
        print (freq_200_final[i])
        mat[i,1]=freq_200_final[i]
    print ('\nFreq of 302 code response --\n')
    for i in range(len(freq_302_final)):
      print (freq_302_final[i])
      mat[i,2]=freq_302_final[i]
    print ('\nFreq of 404 code response --\n')
    for i in range(len(freq_404_final)):
        print (freq_404_final[i])
        mat[i,3]=freq_404_final[i]
    print ('\nPercentage of different requests made -- ', Percentage_diff_req())
    print ('\nPercentage of successful requests --\n')
    for i in range(len(perc_succ_req)):
        print (perc_succ_req[i])
        mat[i,4]=perc_succ_req[i]

def main():
    feature_extraction()
    host_seperation()
    freq_of_HTTP_req()
    freq_unique_url()
    freq_200_resp()
    freq_302_resp()
    freq_404_resp()
    Percentage_diff_req()
    Percentage_succ_resp()
    output_display()
    print ('matrix is \n \n')
    return mat
```