SIT742 Modern Data Science

Assignment 1

Student name: Mirnalini Arivalagan

Student ID: 220142881

Date: 17th April 2021

Table of Contents

Part I - Tulip Hotel Web Logs Exploratory Data Analysis

Hotel TULIP a five-star hotel located at Deakin University, and its CIO Dr Bear Guts has asked the Team-SIT742 team to analyse the weblogs files. As an employee for Hotel Tulip, working in the Information Technology Division, it is required to prepare a set of documentation for Team-SIT742 to allow them to understand the data being dealt with. Throughout this report, some source codes are to explore the weblog, which afterwards the information is presented to Dr Bear Guts in the format of a report.

1. Data ETL

1.1. Data Loading

Fill the DataDictionary.xlsx with discovery from the result of 1.1 Data Loading from your notebook.

1.1.1. Dataset Description

Please add a screenshot of Dataset Description from your DataDictionary.xlsx.

DATA SET NAME	*HTWebLog_p1*
DATA SIZE	*8438928 rows & 15 columns*
DATE OF RELEASE	*22/03/2021*
NO. OF FILES	*120.00*
NO. OF ATTRIBUTES	*15*
NO. OF DATA RECORDS	*8438928*
DATA SOURCE PROVIDER	*Deakin University*
DATA PRIVACY	*For internal use only*
NOTES	
Prepared by:	*Mirnalini Arivalagan*
Point of Contact:	*Mirnalini Arivalagan, arivalaganm@deakin.edu.au*
Team Members:	*[list team members if applicable]*

Data Type Name Convention	Main Type	Subtype
MD	Metric Discrete	BIN - Binary
MC	Metric Continous	DATE - Date/time
		CURR - Currency
CO	Categorical Ordinal	
CN	Categorical Nominal	DI - Dichotomous
		STR - Free String
		ID - Identification
		URL - links such as URLs
		ADDR - Address

1.1.2. Attribute Dictionary

Please add a screenshot of Attribute Dictionary from your DataDictionary.xlsx.

Attribute Name	Data Type	Data Subtype	Description	Examples	Additional Notes
Date	Metric Continous	DATE - Date/time	Request Date	1/11/2006	Converted from object to datetime64[ns] in python
time	Metric Continous	DATE - Date/time	Request Time	12:00:08 AM	Converted from object to datetime64[ns] in python
s-sitename	Categorical Nominal	ID-Identification	Internet Service and instance number accessed	W3SVC1	Python type = object
s-ip	Categorical Nominal	ADDR-Address	Server IP Address	127.0.0.1	Python type = object
cs-method	Categorical Nominal	ID-Identification	Action client was trying to perform at request	GET	Python type = object
cs-uri-stem	Categorical Nominal	URL-links such as URLs	Target of the client action	/Tulip/common/en-us/images/top_logo.gif	Python type = object
cs-uri-query	Categorical Nominal	URL-links such as URLs	The query the client tried to perform	jobPageId=97&lang=zh-hk	Python type = object
s-port	Metric Discrete	ID - Identification	Server port number configured to the service	80	Python type = int64
cs-username	Categorical Nominal	ID-Identification	Name of user who accessed sever. Anonymous users is coded as a hyphen (-)	- (Anonymous)	Python type = object
c-ip	Categorical Nominal	ADDR-Address	Client IP Address that accessed server	70.80.84.76	Python type = object
cs(User-Agent)	Categorical Nominal	STR-Free String	Browser type used by client	Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0)	Python type = object
cs(Referer)	Categorical Nominal	URL-links such as URLs	Previous site visited by client. This site provides a link to current site	http://www.hotelTulip.com.hk/Tulip/home/en-us/home_index.aspx	Python type = object
sc-status	Categorical Ordinal	ID-Identification	The HTTP status code	200	Python type = float64
sc-substatus	Categorical Ordinal	ID-Identification	The sub status error code	14	Python type = float64
sc-win32-status	Categorical Ordinal	ID-Identification	The Windows status code	64	Python type = float64

1.2. Data Cleaning

Please add description of the following contents by yourself.

A. The number NAs for each column

Column Name	Number of Missing Values	% of Missing Values
cs-username	8,438,928	100%
cs-uri-query	7,886,532	93.45%
cs(Referer)	1,309,659	15.51%
cs(User-Agent)	3527	0.04%
sc-status	756	0.01%
sc-substatus	756	0.01%
Sc-win32-status	756	0.01%

B. The number of rows before removal NAs

- The number of rows before removing NAs is: **8438928**

C. The number of rows after removal NAs

- The number of rows after removing NAs is: **8434645**

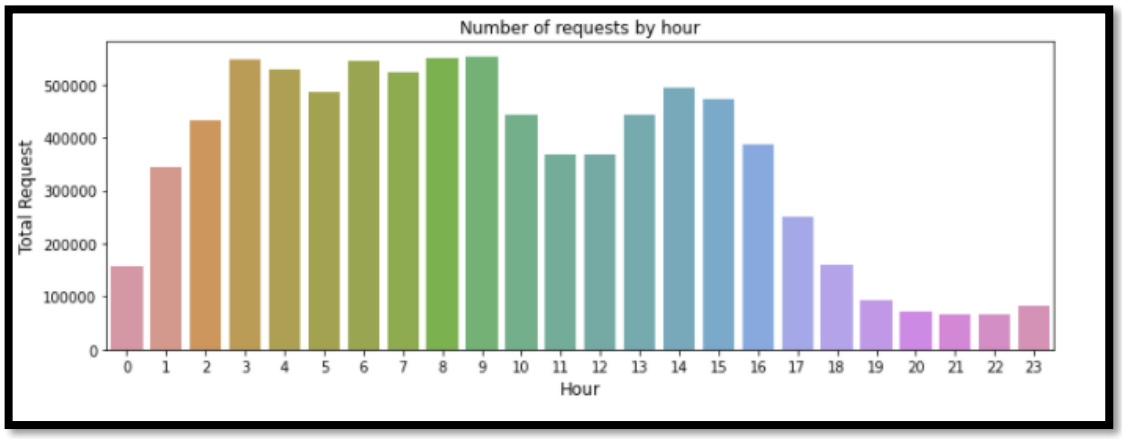
2. Data Statistics Description

2.1. Traffic Analysis

Please add description of the following contents by yourself.

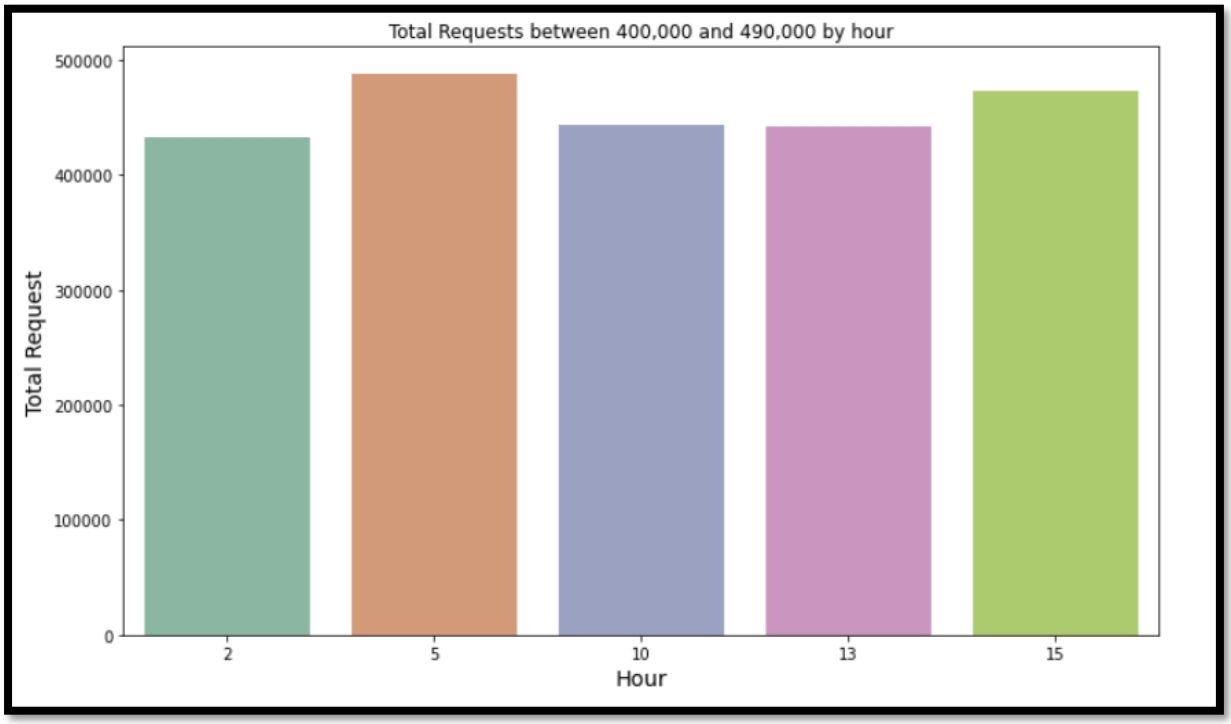
A. *Please add a figure of Hourly Requests Bar Chart from your Notebook and elaborate the findings from the figure.*

- The number of hourly requests is at its peak in the early hours of the morning, between 3am to 9am. And it drops off quite significantly from 10am onwards and runs through at a lower number of requests during the day, this can be attributed to the fact that most users are either at work or school during these hours, hence leisure web browsing activities will drop off. The lowest number of requests are between 7pm and 12am, this can be attributed perhaps that most people are either spending some quality with their loved ones winding down from the day working or schooling and getting to bed.



B. *Please add a table of filter result (hourly_request_amount >= 400000 & hourly_request_amount <= 490000)*

```
Table of hours with number of requests >=400,000 and <= 490,000
hour
2     432289
5     487306
10    443413
13    442414
15    472843
Name: hour, dtype: int64
```



2.2. Server Analysis

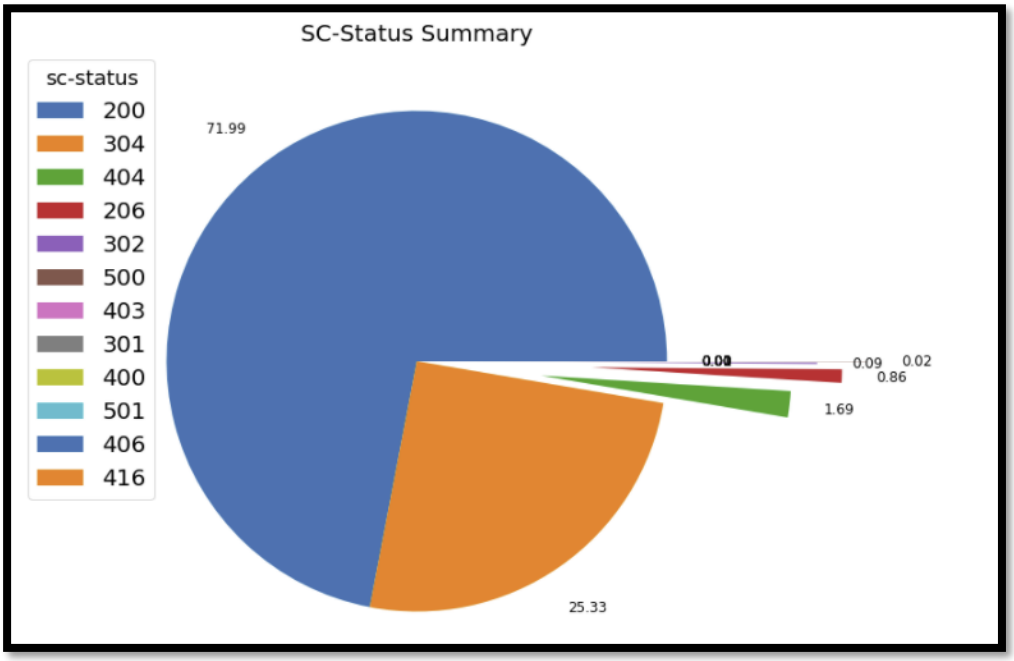
Please add description of the following contents by yourself.

A. *Please elaborate how many types of reported server status.*

- There are 12 types of server status, the top 3 server status were 200, 304 & 404 with a total of request of 6,071,931, 2,136,775 & 142,578. The server status that received the least amount of requests were 501, 406 & 416 with a request of 113, 54 & 3.

B. *Please add a figure of Server Error Pie Chart from your Notebook, and elaborate the findings from the figure.*

- The majority of the requests has the sc-status of 200 (72%) which means most of the requests received were successful. The 2nd most common status is 304 (25%), which means the website the user is trying to access has not been updated since the last time the user accessed it. The third most common status is 404(1.69%), this status means that the request is valid, but the page being requested cannot be found on the server, these could be dead links that do not have a URL redirection set up. Looking at these numbers, we can confidently say that a vast majority of web browsing requests is successful.



2.3. Geographics Analysis

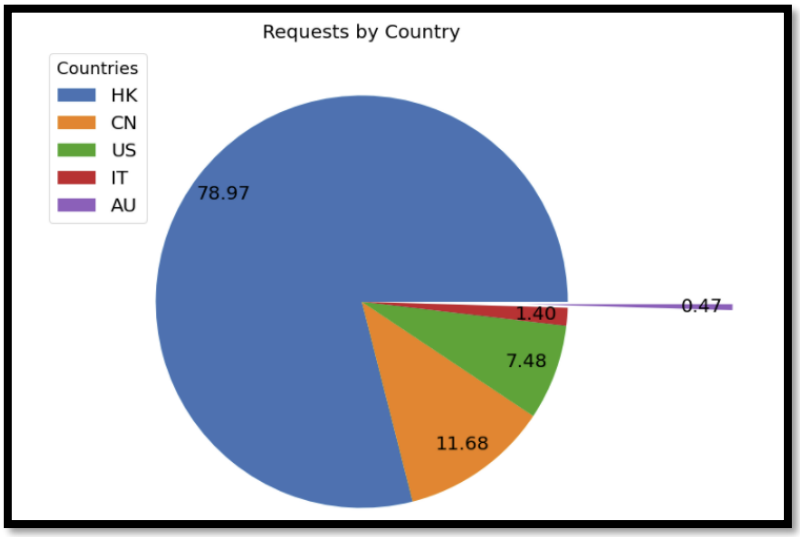
Please add description of the following contents by yourself.

A. Please add a figure of Country distribution and list top 3 with the number of requests.

- The top 3 countries with the most requests were Hong Kong (79%), followed by China (12%) and United States (7%).

```
Top 3 Countries

HK    169
CN     25
US     16
Name: country, dtype: int64
```

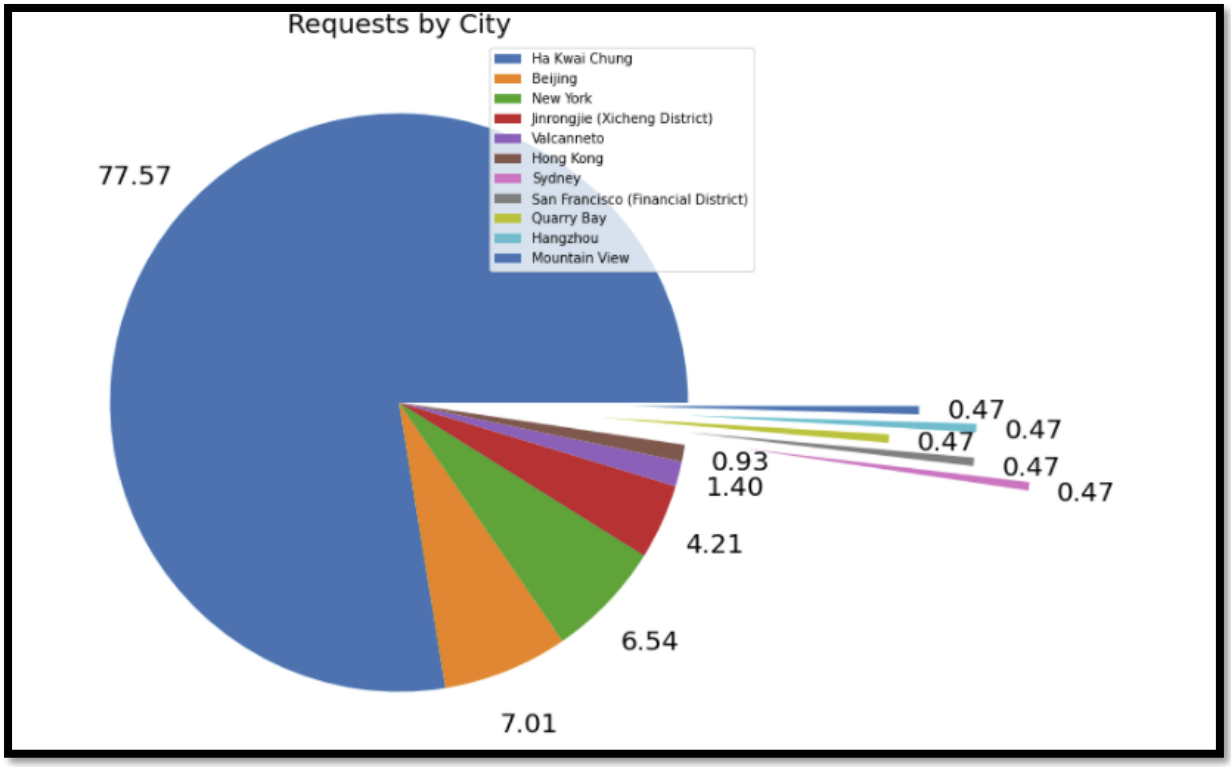


B. Please add a figure of City distribution and list top 3 with the number of requests.

- The top 3 cities with the most number of requests were Ha Kwai Chung (77%), Beijing (7%) and New York City (6.5%)

```
Top 3 Cities

Ha Kwai Chung    166
Beijing           15
New York          14
Name: city, dtype: int6
```



Part II - School of IT Professor Citation Information

To better introduce all the professors including the emeritus professor, the professor and also associate professor in Deakin University School of IT, faculty will need to know all the citation information on all professors. Google Scholar is a web search engine that freely indexes the metadata of articles on many authors. Majority of the professors choose to use google scholar to track their publications and research works. Therefore, the web crawling on google scholar will be able to have the citation information obtained across all the professors (who have the google scholar profile).

3. Professor List Generation

3.1. Import Web Crawling Library

Please fill this part with the screenshot of your code for import your own web crawling library.

```
!pip install beautifulsoup4
from bs4 import BeautifulSoup
import requests
```

3.2. Find all professors in School of IT

Please fill this part with the screenshot of your code for generating the professor name list csv. The screen shot will also include the results of the running on the code.

```
url = 'https://www.deakin.edu.au/information-technology/staff-listing' #assign website to variable URL

req = requests.get(url) #use requests function to call and access URL

soup = BeautifulSoup(req.text, 'lxml') #assign request from URL request as text to variable soup

tables = soup.select('table') #locate all tables from web scrape results

SIT_df =[] #create empty dataframe to add web scraping results

for i in tables:
    SIT_df.append(pd.concat(pd.read_html(i.prettify())))

SIT = pd.concat(SIT_df)
```

```
SIT #check data frame content
```

	Name
0	Emeritus Professor Lynn Batten
1	Emeritus Professor Andrzej Goscinski
0	Professor Jemal Abawajy
1	Professor Maia Angelova
2	Professor Gleb Beliakov
...	...
2	Jesse Mcmeikan
3	Thuy Nguyen
4	Michelle Yu
0	Nghia Dang
1	Justin Li

176 rows × 1 columns

```
SIT = SIT['Name'].str.extract(r'(.*Professor)?\s?(.+)') #extract anything before and including professor substring from string va

SIT.rename({0:'Title', 1:'Name'}, axis=1, inplace=True) #ammend data frame to edit headers

SIT['University'] = 'Deakin University' #append data frame with new column named University populated with Deakin University as v
```

```
SIT.head(10)#check titles have been extracted from name and new columns added
```

	Title	Name	University
0	Emeritus Professor	Lynn Batten	Deakin University
1	Emeritus Professor	Andrzej Goscinski	Deakin University
0	Professor	Jemal Abawajy	Deakin University
1	Professor	Maia Angelova	Deakin University
2	Professor	Gleb Beliakov	Deakin University
3	Professor	Terry Caelli	Deakin University
4	Professor	Jinho Choi	Deakin University
5	Professor	Chang-Tsun Li	Deakin University
6	Professor	Robin Doss	Deakin University
7	Professor	Peter Eklund	Deakin University

```
#filter dataframe to only include individuals that are either Emeritus Professor, Professor or Associate Professor
SIT = SIT[(SIT['Title'] == "Emeritus Professor") |(SIT['Title'] == "Professor") | (SIT['Title'] == "Associate Professor")]
```

```
SIT.to_csv('Professor-list.csv', index=False) #export data frame to csv
```

3.2.1. Professor Name List CSV

Please fill this part with the screenshot of your csv.

	A	B	C
1	Title	Name	University
2	Emeritus Professor	Lynn Batten	Deakin University
3	Emeritus Professor	Andrzej Goscinski	Deakin University
4	Professor	Jemal Abawajy	Deakin University
5	Professor	Maia Angelova	Deakin University
6	Professor	Gleb Beliakov	Deakin University
7	Professor	Terry Caelli	Deakin University
8	Professor	Jinho Choi	Deakin University
9	Professor	Chang-Tsun Li	Deakin University
10	Professor	Robin Doss	Deakin University
11	Professor	Peter Eklund	Deakin University
12	Professor	Seng Loke	Deakin University
13	Professor	Antonio Robles-Kelly	Deakin University
14	Professor	Jean-Guy Schneider	Deakin University
15	Professor	Yong Xiang	Deakin University
16	Professor	John Yearwood	Deakin University
17	Professor	Arkady Zaslavsky	Deakin University
18	Associate Professor	Mohamed Abdelrazek	Deakin University
19	Associate Professor	Andrew Cain	Deakin University
20	Associate Professor	Richard Dazeley	Deakin University
21	Associate Professor	Guangyan Huang	Deakin University
22	Associate Professor	Gang Li	Deakin University
23	Associate Professor	Jianxin Li	Deakin University
24	Associate Professor	Xiao Liu	Deakin University
25	Associate Professor	Vicky Mak	Deakin University
26	Associate Professor	Tim Wilkin	Deakin University
27	Professor	Abbas Kudrati	Deakin University
28			