Data Science assignment

1) **Replace the NaN values with the correct value. And justify why you have chosen the same**

To Find NaN values in Python: dataset.isnull().sum()

Handling NaN Values - Four common ways to handle NaN
1. Replace with mean, median, or mode
2. Replace with 0 or a constant value
3. Drop rows with NaN values: Simple, but can result in data loss.
4. Use machine learning to predict missing values

No NaN values found.

2) **How many of them are Prioritizer(1) by the Load Balancer?**

There are 617 tasks that were Prioritized, as determined by analyzing the `‘Priority(0/1)’`column of the dataset.

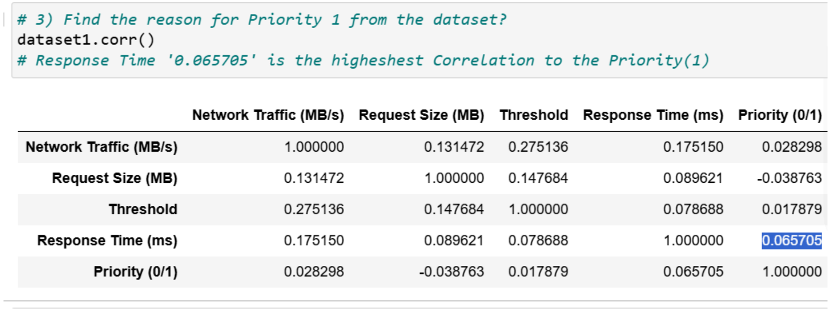
# Frequency of ‘Prioritized(1)’ tasks

PrioritizedTaskCount =dataset1['Priority (0/1)'].value_counts().get(1, 0)
print('PrioritizedTaskCount:', PrioritizedTaskCount, ' where total tasks: ', dataset1['Priority (0/1)'].size)

PrioritizedTaskCount: 617 where total tasks: 1254

3) **Find the reason for non placement from the dataset?**

`Response Time` stands out as the most critical factor. Tasks with lower Response time are significantly less likely to be a Priority 1 task.

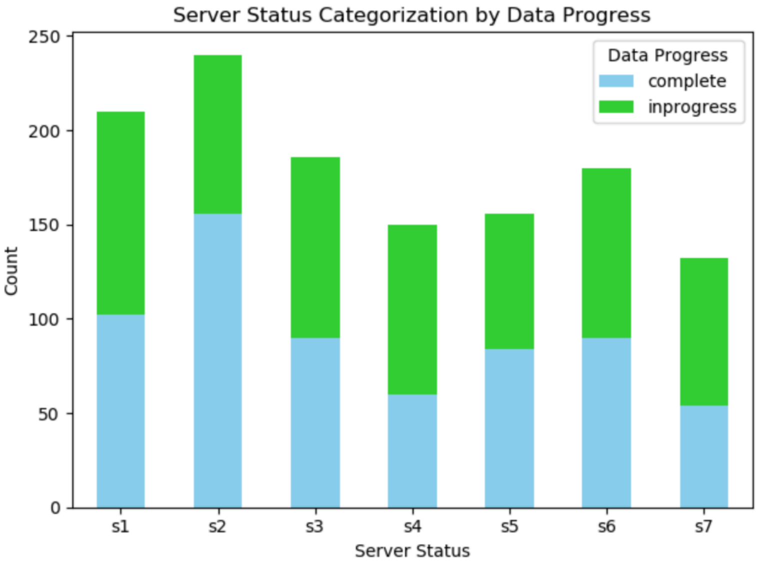


```
# 3) Find the reason for Priority 1 from the dataset?
dataset1.corr()
# Response Time '0.065705' is the higheshest Correlation to the Priority(1)
```

	Network Traffic (MB/s)	Request Size (MB)	Threshold	Response Time (ms)	Priority (0/1)
Network Traffic (MB/s)	1.000000	0.131472	0.275136	0.175150	0.028298
Request Size (MB)	0.131472	1.000000	0.147684	0.089621	-0.038763
Threshold	0.275136	0.147684	1.000000	0.078688	0.017879
Response Time (ms)	0.175150	0.089621	0.078688	1.000000	0.065705
Priority (0/1)	0.028298	-0.038763	0.017879	0.065705	1.000000

4) **What kind of relation between Request Size(MB) and Response Time(ms)**

They are positively correlated(>0) with a weaker degree. When the Request Size (MB) increases there is a very small tendency for increase in Response Time (ms).
Correlation coefficient between Request Size (MB) and Response Time (ms) = 0.175

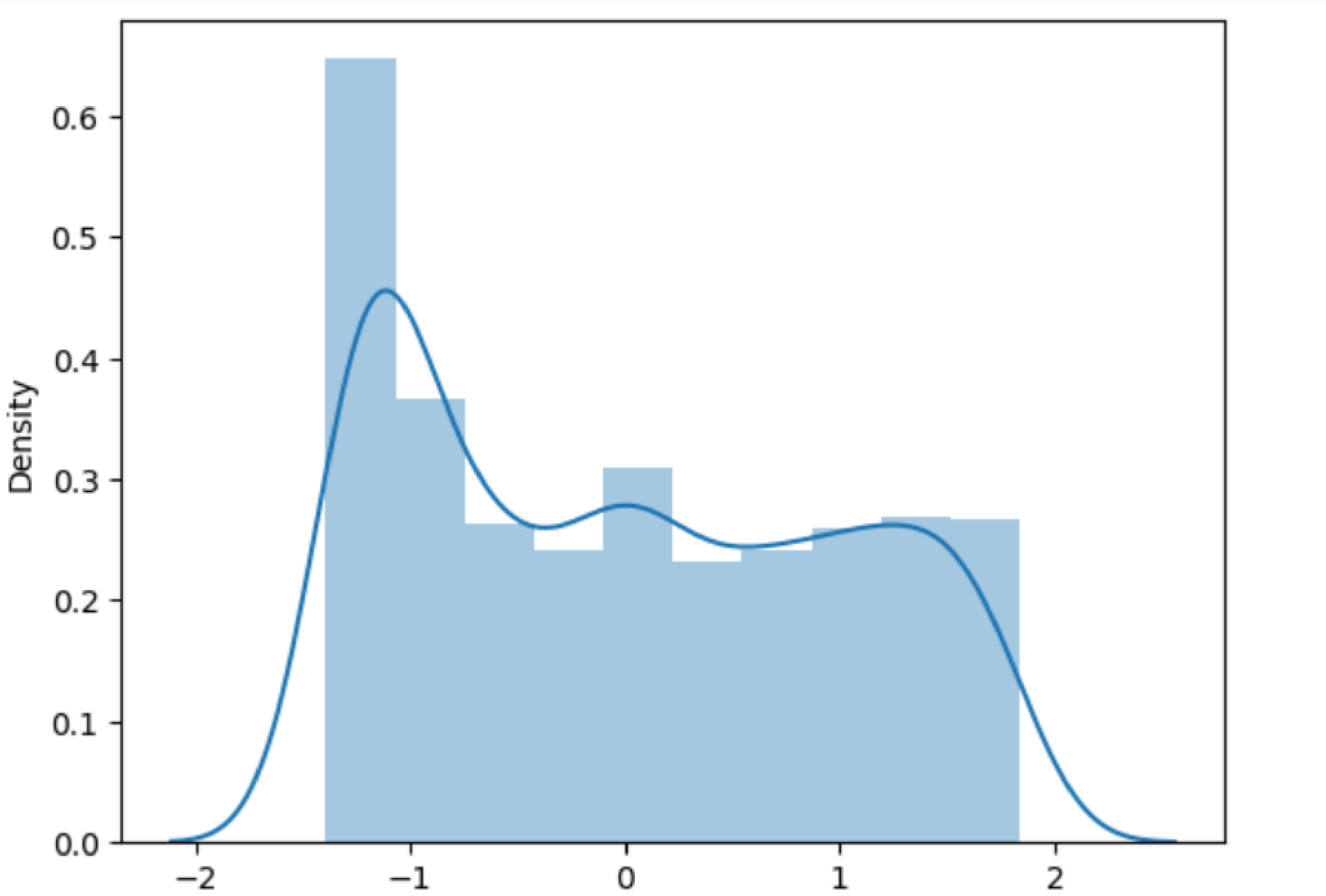
	dataset1[['Request Size (MB)','Response Time (ms)']].corr()
5)	**Which Server Status completes the most?**
	S2 has the most completed tasks This is found by plotting the histogram 
6)	**How many of the tasks are above Network Traffic of 788 MB/s**
	11 tasks were having a Network Traffic of greater than 788 MB/s This is found using Frequency - value_counts() (dataset1['Network Traffic (MB/s)']> 788).value_counts() False 1243 True 11
7)	**Test the Analysis of Variance between Request Size (MB) and Response Time (ms) at significance level 5%.(Make decision using Hypothesis Testing)**
	P value= 0.4305 Since the p-value (0.4305) is greater than the significance level (0.05), you fail to reject the null hypothesis. This means there is no statistically significant difference between and at the 5% significance level. Used One-way classification: stats.f_oneway(dataset1['Request Size (MB)'],dataset['Response Time (ms)'])
8)	**Test the similarity between the server Load(high) and Data Progress(inprogress) with respect to Priority at significance level of 5%.(Make decision using Hypothesis Testing)**

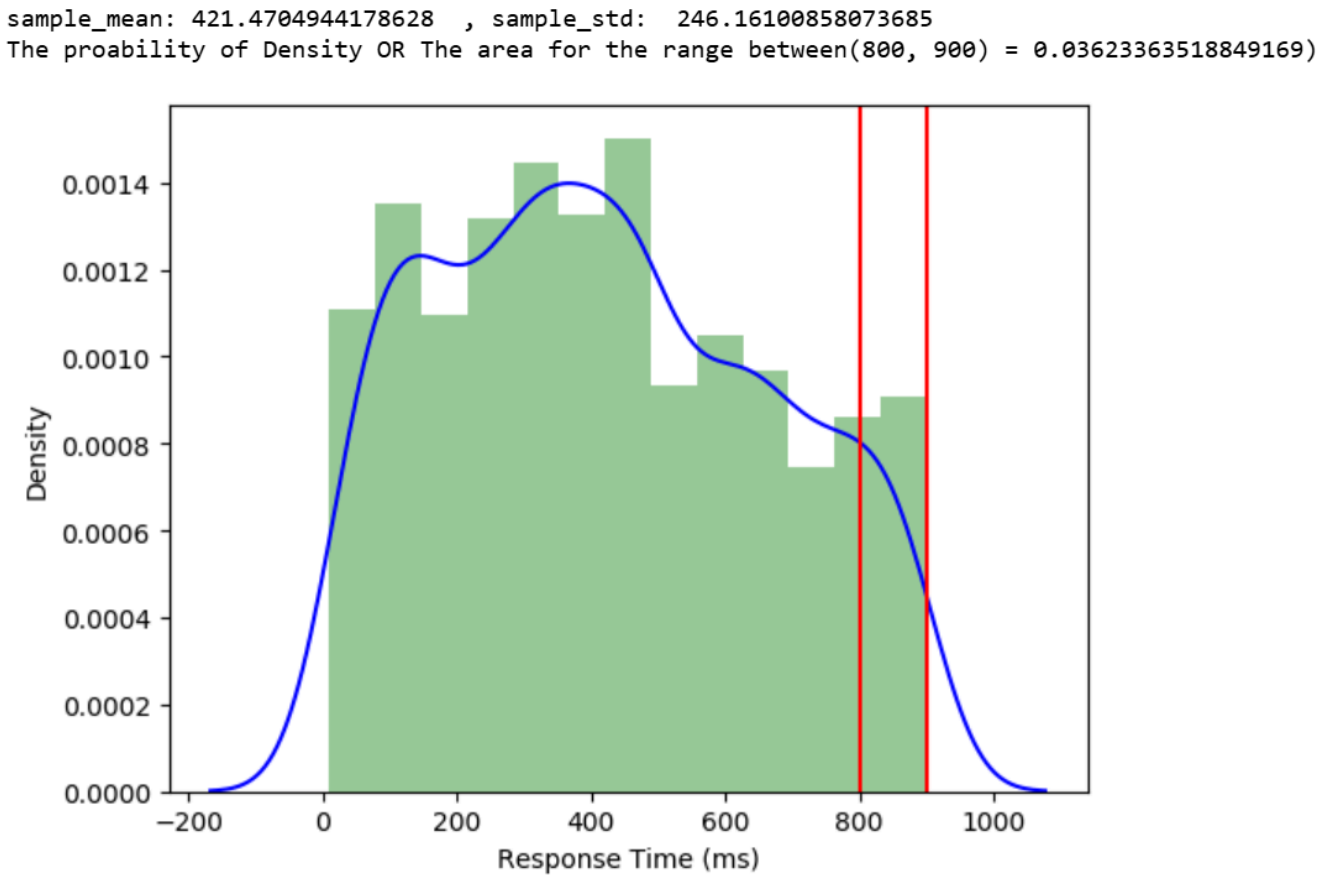
With respect to **Priority**	**P Value**	**Conclusion**
C(Server_Load)	0.456098	The p-value(0.45) is greater than 0.05, so there is no significant effect of Server_Load on Priority.
C(Data_Progress)	0.003725	The p-value(0.003) is less than 0.05, so Data_Progress significantly affects Priority. This means Priority changes depending on whether Data Progress is "inprogress" or "complete."
C(Server_Load):C(Data_Progress)	0.393861	The p-value is greater than 0.05, so there is no significant interaction between Server_Load and Data_Progress. Server_Load does not influence how Data_Progress affects Priority.

- Data Progress has a significant affect on Priority.
- There is no significant effect of:

Decision: Fail to reject Null Hypothesis(H0). Data Progress has a significant effect on Priority

9) **Convert the normal distribution to standard normal distribution for Network MB/s column**

	 Using Z-score with the column’s mean and std deviation values we can convert to a standard normal deviation.
10)	**What is the probability Density Function of the Response Time (ms) range from 800 ms to 900 ms?**



```
sample_mean: 421.4704944178628  , sample_std:  246.16100858073685
The proability of Density OR The area for the range between(800, 900) = 0.03623363518849169)
```

About 3.62% from the total tasks has the probability to be in the range of 800-900

11) **Test the similarity between the Server Status(s2)with respect to Request Size (MB) and Threshold at significance level of 5%.(Make decision using Hypothesis Testing)**

There is a significant difference between etest_p and mba_p for candidates with degree_t = Sci&Tech.
1 group value,
Paired T-test Dependent Sample
s2_RequestSize = dataset[dataset['Server Status']=='s2']['Request Size (MB)']
s2_Threshold = dataset[dataset['Server Status']=='s2']['Threshold']

1. Hypothesis Statements
   - Null Hypothesis (H0H_0): There is no significant difference between the means of Request Size and Threshold for server status s2.
   - Alternative Hypothesis (HaH_a): There is a significant difference between the means of Request Size and Threshold for tasks with server status= s2.
2. Significance Level: The significance level (α\alpha) is 5% (0.05).
3. Paired T-Test: The paired t-test is used as the two datasets (Request Size and Threshold) are related (collected from the same candidates).

s2_RequestSize = dataset[dataset['Server Status']=='s2']['Request Size (MB)']
s2_Threshold = dataset[dataset['Server Status']=='s2']['Threshold']

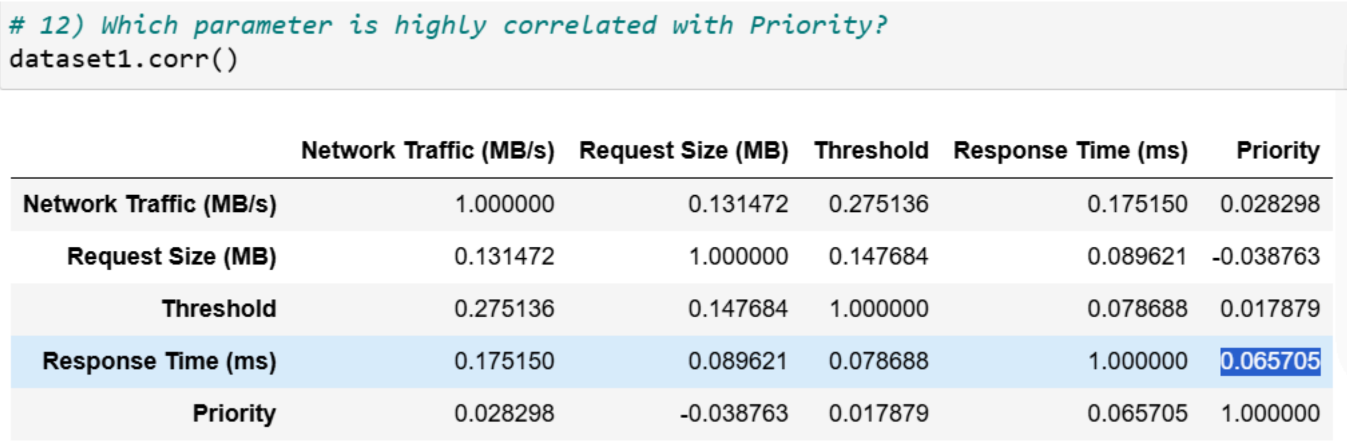
```
# Perform the paired t-test
t_stat, p_value = ttest_rel(s2_RequestSize , s2_Threshold )
```

4. Test Results: statistic=26.154854324915355, pvalue=4.4604225343314296e-72
5. Decision Rule
   - If p-value<0.05p, reject the null hypothesis (H0).
   - If p-value≥0.05p, fail to reject the null hypothesis (H0).
6. Conclusion
   A. The p-value = 0.000000004 is much smaller than 0.05.
   B. Thus, we reject the null hypothesis (H0H_0).
   C. There is sufficient evidence to conclude that there is a significant difference between Request Size and Threshold for tasks with server status= s2.

12) **Which parameter is highly correlated with Priority?**

There is not any highly correlated value with Priority

The positive correlated value for Priority Is 6.5% with Response Time (ms) but this is weak degree of positive correlation

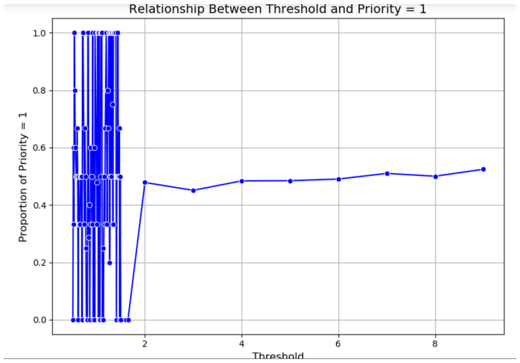


```
# 12) Which parameter is highly correlated with Priority?
dataset1.corr()
```

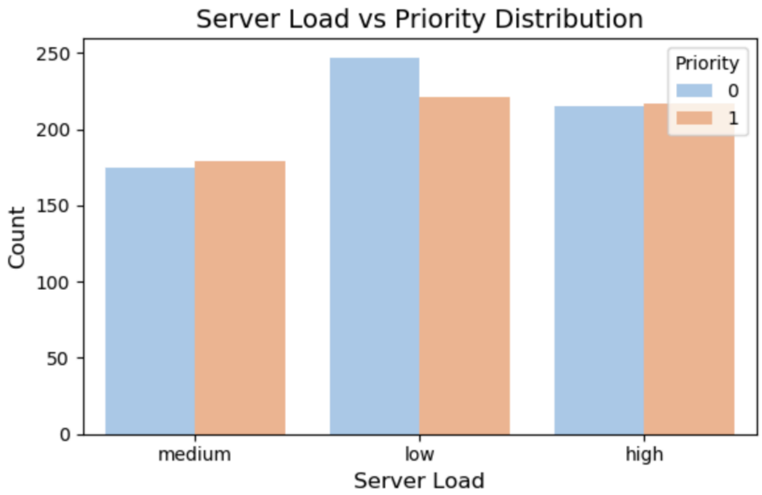
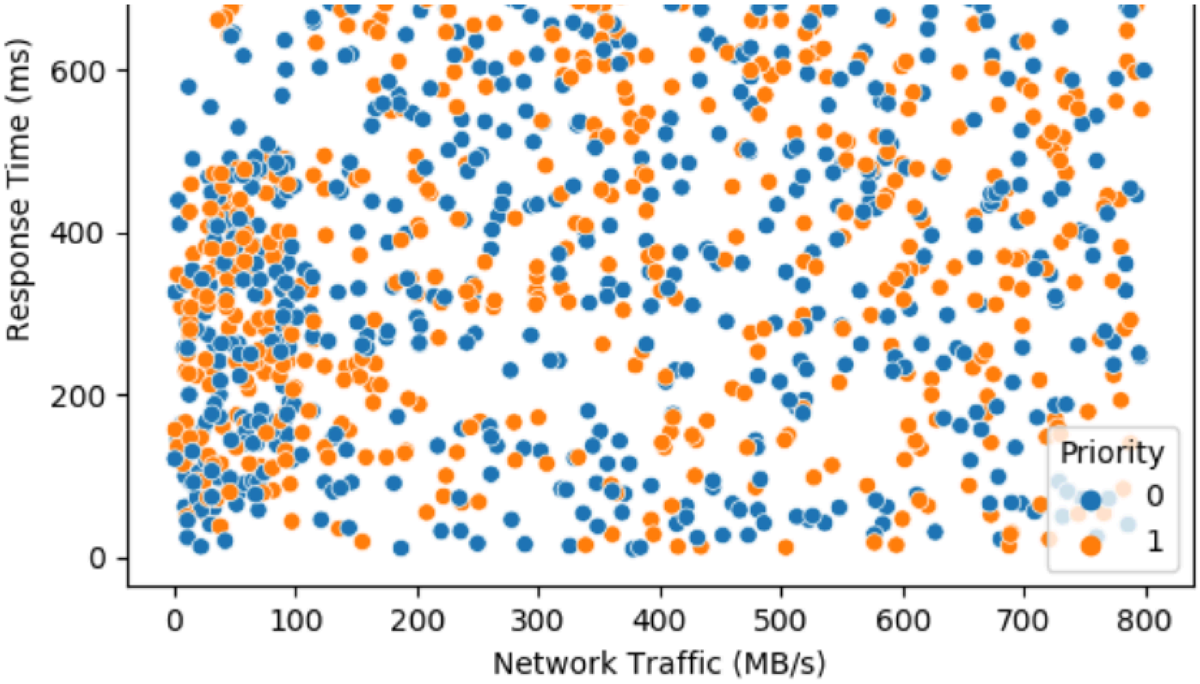
	Network Traffic (MB/s)	Request Size (MB)	Threshold	Response Time (ms)	Priority
Network Traffic (MB/s)	1.000000	0.131472	0.275136	0.175150	0.028298
Request Size (MB)	0.131472	1.000000	0.147684	0.089621	-0.038763
Threshold	0.275136	0.147684	1.000000	0.078688	0.017879
Response Time (ms)	0.175150	0.089621	0.078688	1.000000	0.065705
Priority	0.028298	-0.038763	0.017879	0.065705	1.000000

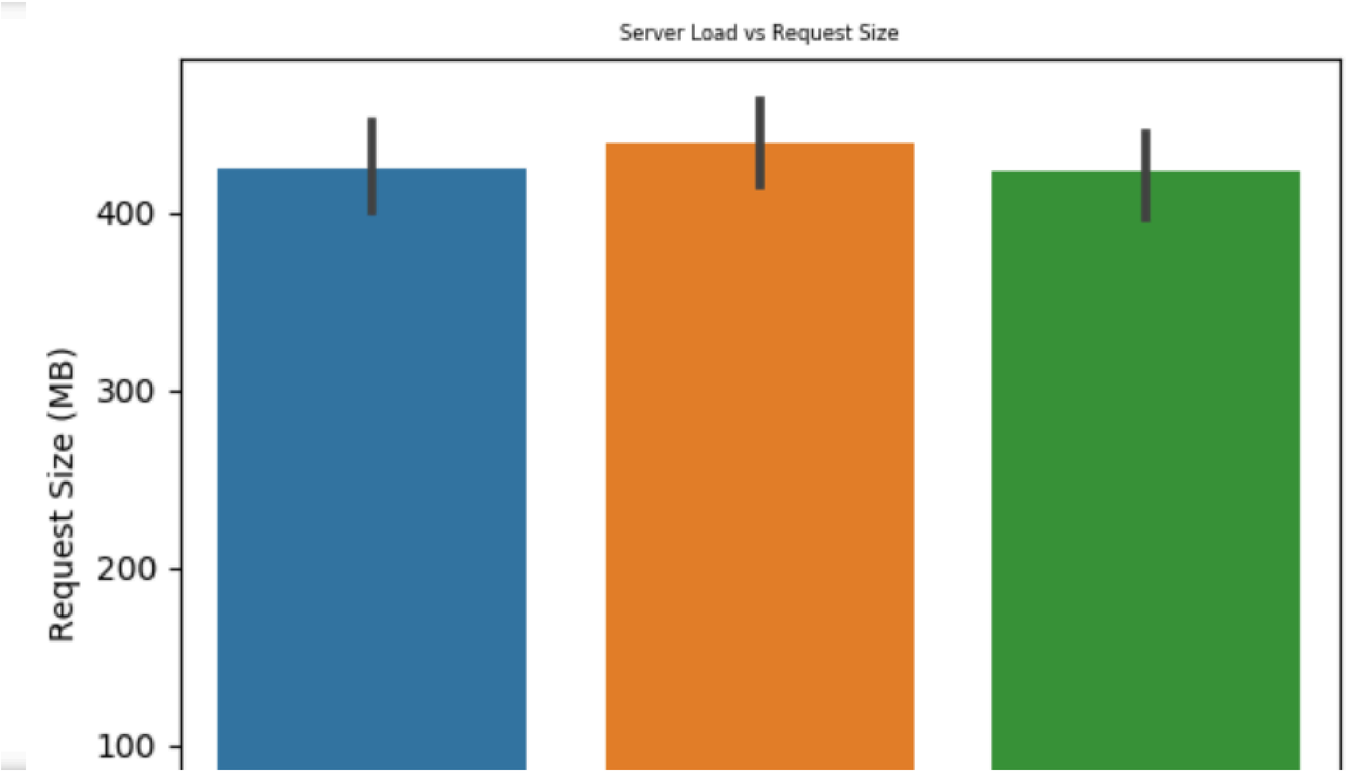
13 A) **Does higher threshold get more priority 1**

Yes. With a weaker degree, yes



13 B) **Do Task server load relate to Priority.**

	 No. It is not necessary that the task with higher get priority 1. They can almost get priority 0 as well.
13) C)	**Relation between Network Traffic (MB/s) and Response Time (ms)**
	When the Network traffic is less the task have a tendency to take less Response Time 
13) D)	Server Load vs Request Size relation



Even when a Request size is greater it can very well be of any Server Load, Low, med, high