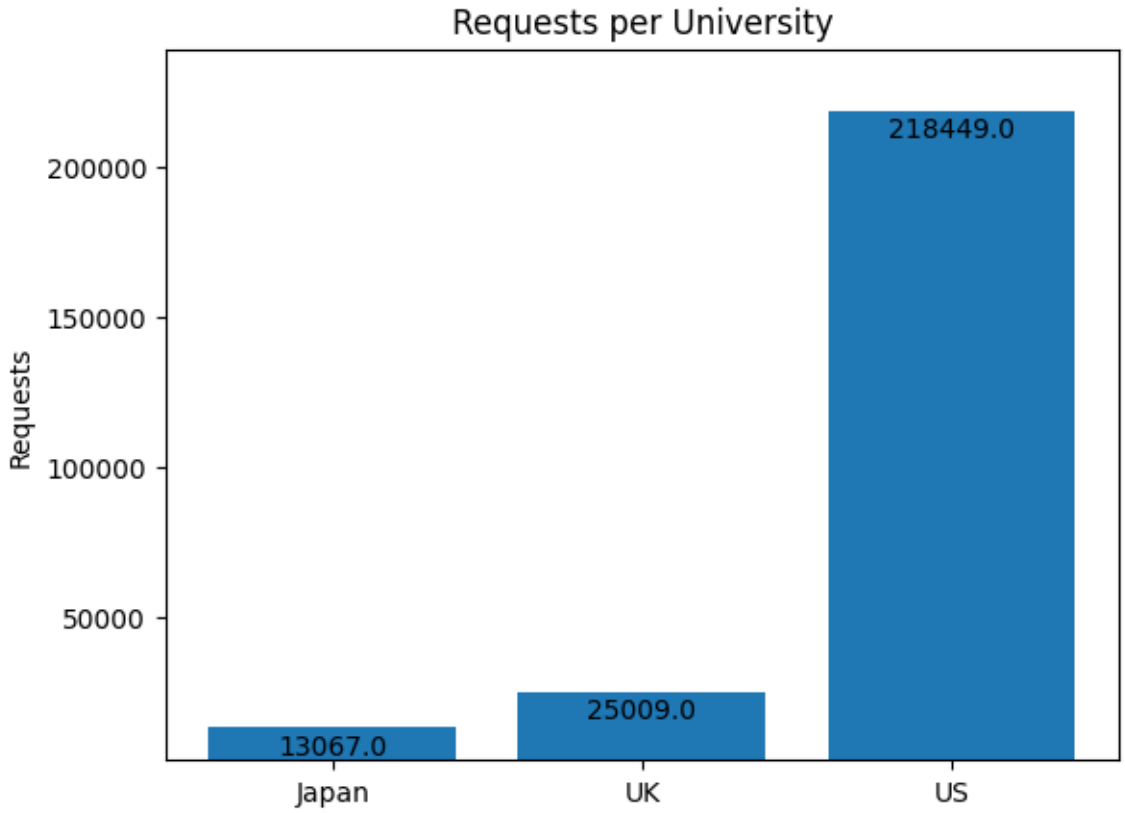
Assignment 1:

Question 1. Log Mining and Analysis

A. 1) Japanese universities ending with "**.ac.jp**" : 13067
   2) UK universities ending with "**.ac.uk**" : 25009
   3) US universities ending with "**.edu**" : 218449

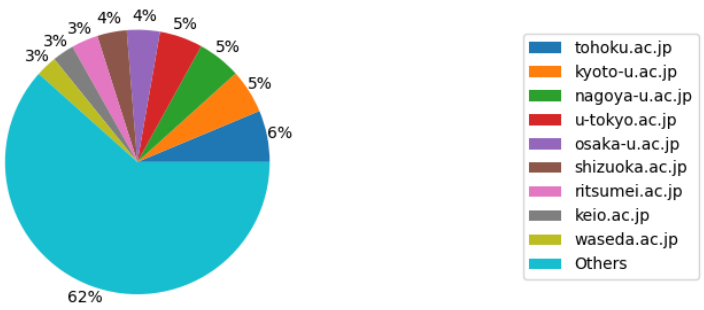


B. **1) Below are the 9 most frequent universities according to the host domain for each of the three countries:**

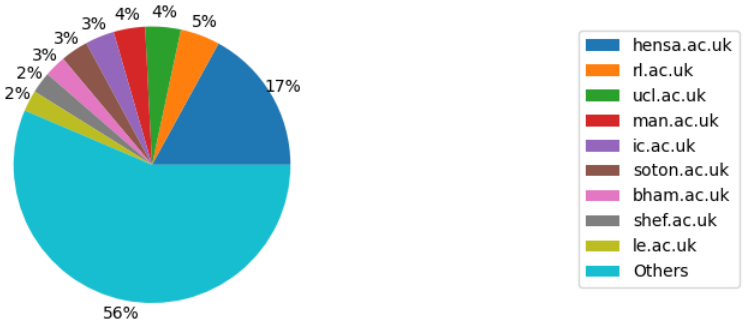
Japan		UK		US	
University	**Count**	**University**	**Count**	**University**	**Count**
tohoku.ac.jp	824	hensa.ac.uk	4257	tamu.edu	6062
kyoto-u.ac.jp	703	rl.ac.uk	1158	berkeley.edu	5439
nagoya-u.ac.jp	692	ucl.ac.uk	1036	fsu.edu	4418
u-tokyo.ac.jp	689	man.ac.uk	921	umn.edu	4404
osaka-u.ac.jp	527	ic.ac.uk	851	mit.edu	3966
shizuoka.ac.jp	472	soton.ac.uk	808	washington.edu	3893
ritsumei.ac.jp	426	bham.ac.uk	629	uiuc.edu	3750
keio.ac.jp	346	shef.ac.uk	623	utexas.edu	3665
waseda.ac.jp	337	le.ac.uk	616	cmu.edu	3244

2) The pie charts below represent the percentage of requests by each of the top 9 most frequent universities and the rest in each region.

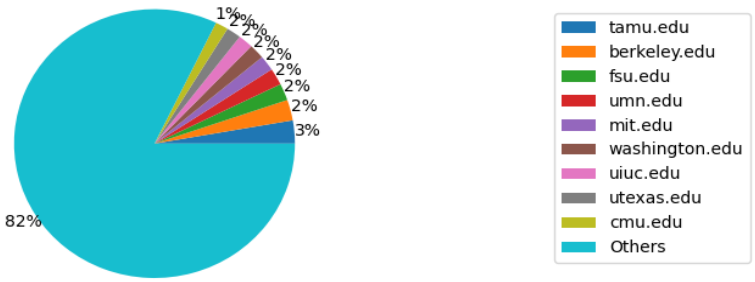
Japan



UK

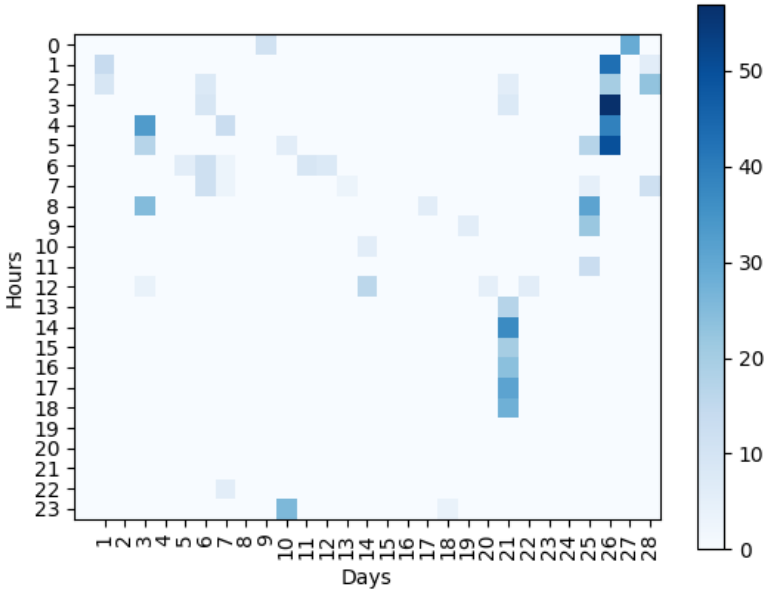


US

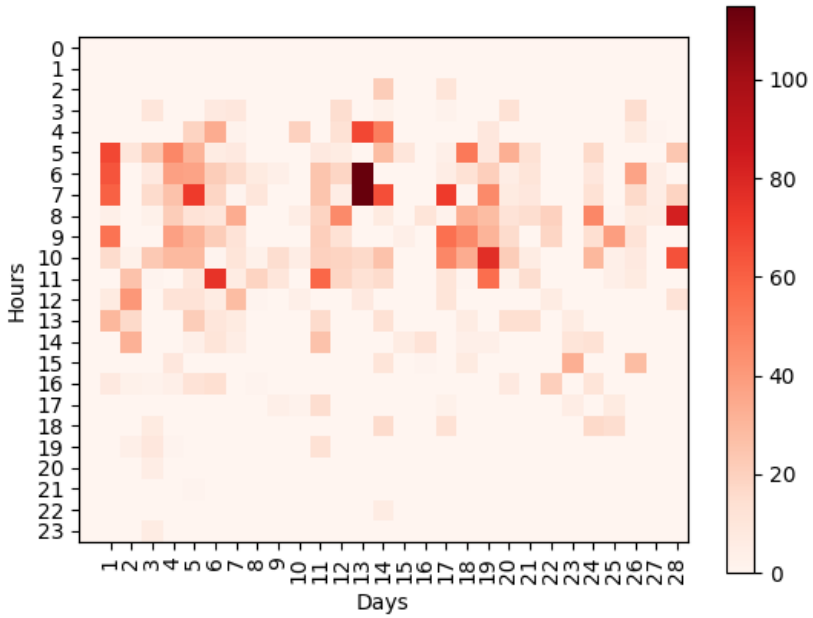


C. Heatmaps for the most frequent university from each country

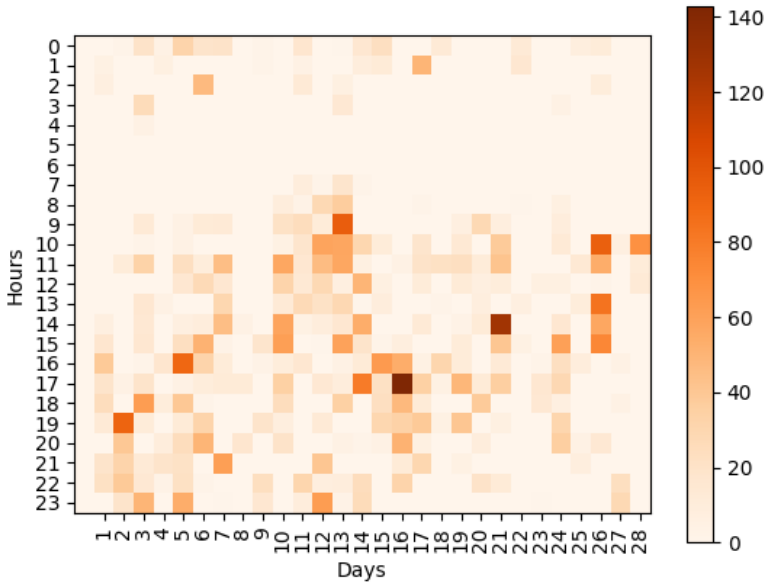
Japan



UK



US



D. Observations:

- The log analysis tells us that NASA receives the highest request from the US universities in comparison with the universities from Japan and the UK.
- 9 of the total 13067 universities in Japan generate ~40% of the requests.
- Unlike Japan, where the traffic has been more towards the end of the month universities, from both the UK and US have traffic generated spanned over the entire month. The only difference between the two regions is that the traffic in the US begins after around 10 am while that in the UK starts at dawn, time difference between the two regions could be the possible reason.

Data for NASA, as like any other organisation is of utmost importance for their own sustainability. The observations stated above can be leveraged by NASA to run some predictive analytics like predicting the traffic coming from a region or at any particular time of day and then using it to evaluate and if necessary, alter their infrastructure capability.

Question 2. Movie Recommendation and Analysis:

A. Time-split Recommendation:

1) To perform time split recommendation using ALS recommendation where data is split with respect to the timestamp i.e. earlier time for training and later time for test, we have made use of partitionBy() method deploying percent_rank(). This helps us to split the data in a sequential manner. It first sorts

the data on the field of our choice in our case, timestamp, and return a relative rank of each row. These ranks can now be used to make the desired split in terms of the ranks(percentages).

2) For each of the three splits, we have trained 3 ALS models:
   - ALS 1 : using parameter values as used in the lab
   - ALS 2 :
     a) Rank = 25, we have increased the number of latent features from default 10 as they would directly contribute to modelling the user features which would be used to form clusters in the next part of the assignment.
     b) maxIter = 10, we have chosen this default value to be not too high or too low but sufficient enough for the model to learn in combination of the high number of features.
   - ALS 3:
     a) Rank = 15, we have now decreased the number of latent features to 15 from the above model to try and not overburden the model with too many features to work with and also for the reason that the higher rank didn't much improve the model's performance.
     b) maxIter = 5, we have chosen this value as we have decreased the feature count.
     c) regParam = 0.001, in order to avoid overfitting our model, we have chosen a low value for the regularization parameter.

   The performance of the model was fairly good with the default setting than with the other two models.

3) Below are the model metrics:

Training- Test	Model	RMSE	MSE	MAE
50-50	**ALS 1**	0.79	0.6241	0.6007
	ALS 2	0.7904	0.6248	0.6029
	ALS 3	0.9057	0.8203	0.6523
65-35	**ALS 1**	0.8093	0.6549	0.6096
	ALS 2	0.8081	0.6531	0.6107
	ALS 3	1.0655	1.1353	0.6975
80-20	**ALS 1**	0.8597	0.7391	0.6445
	ALS 2	0.8571	0.7347	0.6448
	ALS 3	1.2808	1.6405	0.7927

B. User Analysis:

1) From the userFactors obtained from the ALS model for each of the three splits, we create clusters for our data using Kmeans keeping k=20 for our case and we get the below largest clusters:

	Training		Test	
Training- Test	**Cluster Number**	**Count**	**Cluster Number**	**Count**
50-50	11	12387	19	15014
	1	10959	6	13557
	14	10955	9	12173
65-35	18	16185	16	10598
	6	15550	12	8533
	7	14456	11	8272
80-20	8	22255	6	4797
	16	19914	3	4507
	3	17934	13	4266

2) Using the top cluster from each split, we retrieve all the users belonging to those respective clusters. Then by fetching the 'genres' from 'movies.csv' and we get the top 5 genres for all the movies which were rated 4 and above in that cluster, below are the results:

	Training		Test	
Training- Test	**Top Genres**	**Count**	**Top Genres**	**Count**
50-50	Drama	180291	Drama	495325
	Comedy	112333	Action	279918
	Thriller	102554	Thriller	274747
	Romance	80177	Comedy	233288
	Action	72729	Crime	231472
65-35	Drama	627143	Drama	344020
	Comedy	406695	Thriller	198739
	Thriller	321864	Action	182085
	Action	277924	Crime	165733
	Crime	252902	Comedy	137967
80-20	Drama	798895	Drama	199117
	Comedy	622831	Comedy	99041
	Thriller	403101	Thriller	96616
	Action	381233	Action	87141
	Adventure	328023	Crime	78308

C. Observations:

- ALS model performance has been the best with the default parameters.
- 'Drama' has been the top choice of users and is evident in both the training and test data. This could be because the number of movies tagged under the genre 'Drama' is higher.

Netflix could very well benefit from this and analysis and recommendation system. Once a user logs in to the system, based on the ratings provided by the user at an earlier time, he would be assigned a cluster and since we already have the top genres in every cluster, the recommendations made to him would be more tailored and to his liking.