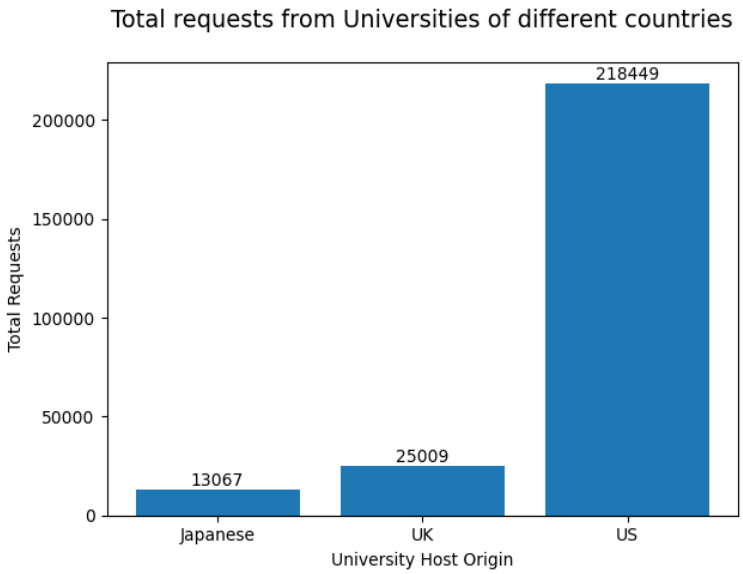
COM6012 - Assignment 1

Question 1. Log Mining and Analysis

A. Total number of requests from all hosts from Japan, UK and US are presented in the table below. A bar plot which visualises the quantity of the requests from the universities of the three countries is also shown below.

Host University origin country	Total number of requests
Japan ('.ac.jp')	13067
UK ('.ac.uk')	25009
US ('.edu')	**218449**



B.

1. Top 9 most frequent universities (hosts) from Japan are:

   '**u-tokyo.ac.jp**', 'nagoya-u.ac.jp', 'ritsumei.ac.jp', 'osaka-u.ac.jp', 'ims.ac.jp', 'tohoku.ac.jp', 'shizuoka.ac.jp', 'kyoto-u.ac.jp', 'jaist.ac.jp'.

   Top 9 most frequent universities (hosts) from UK are:

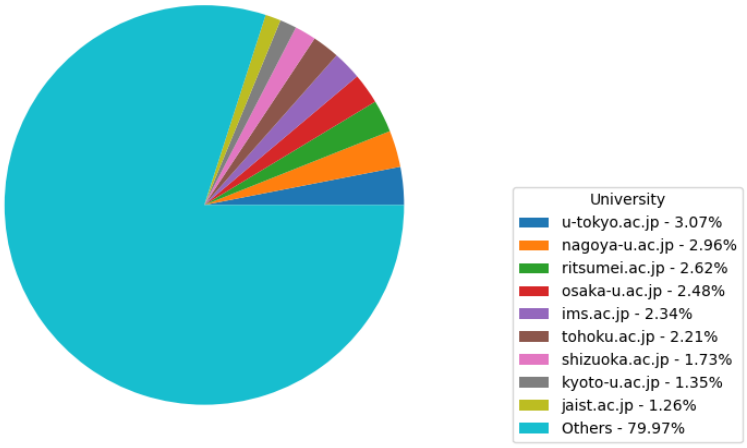
   '**hensa.ac.uk**', 'ucl.ac.uk', 'rl.ac.uk', 'man.ac.uk', 'bton.ac.uk', 'dundee.ac.uk', 'brunel.ac.uk', 'hw.ac.uk', 'soton.ac.uk'.

   Top 9 most frequent universities (hosts) from US are:

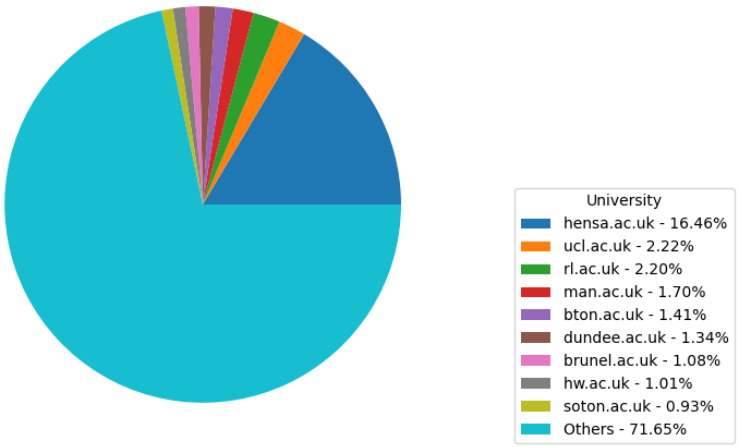
   '**msstate.edu**', 'berkeley.edu', 'nwu.edu', 'cmu.edu', 'caltech.edu', 'usf.edu', 'washington.edu', 'mit.edu', 'caltech.edu'.

2. Pie charts visualising the percentage (with respect to the total) of requests by each of the top 9 most frequent universities and the rest (represented as others), for three countries are as below.

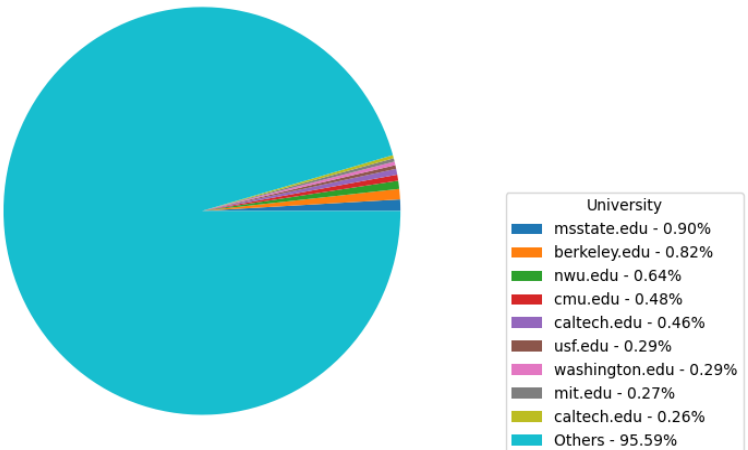
Proportion of requests among Japanese Universities



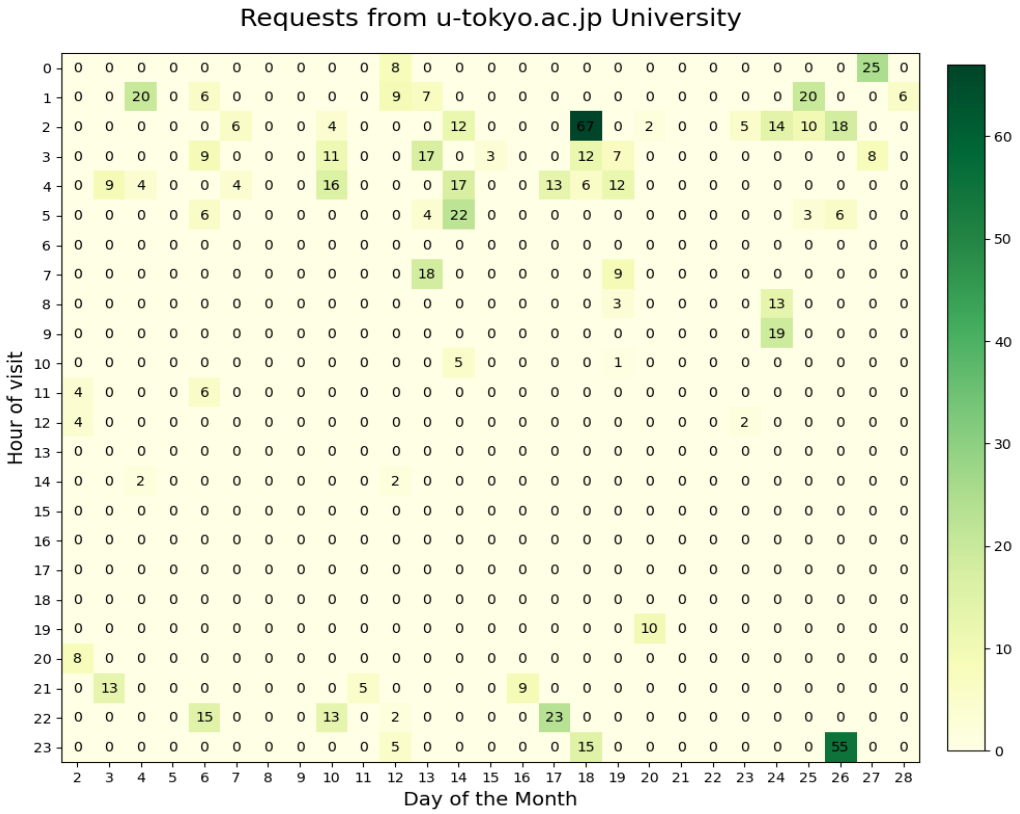
Proportion of requests among UK Universities

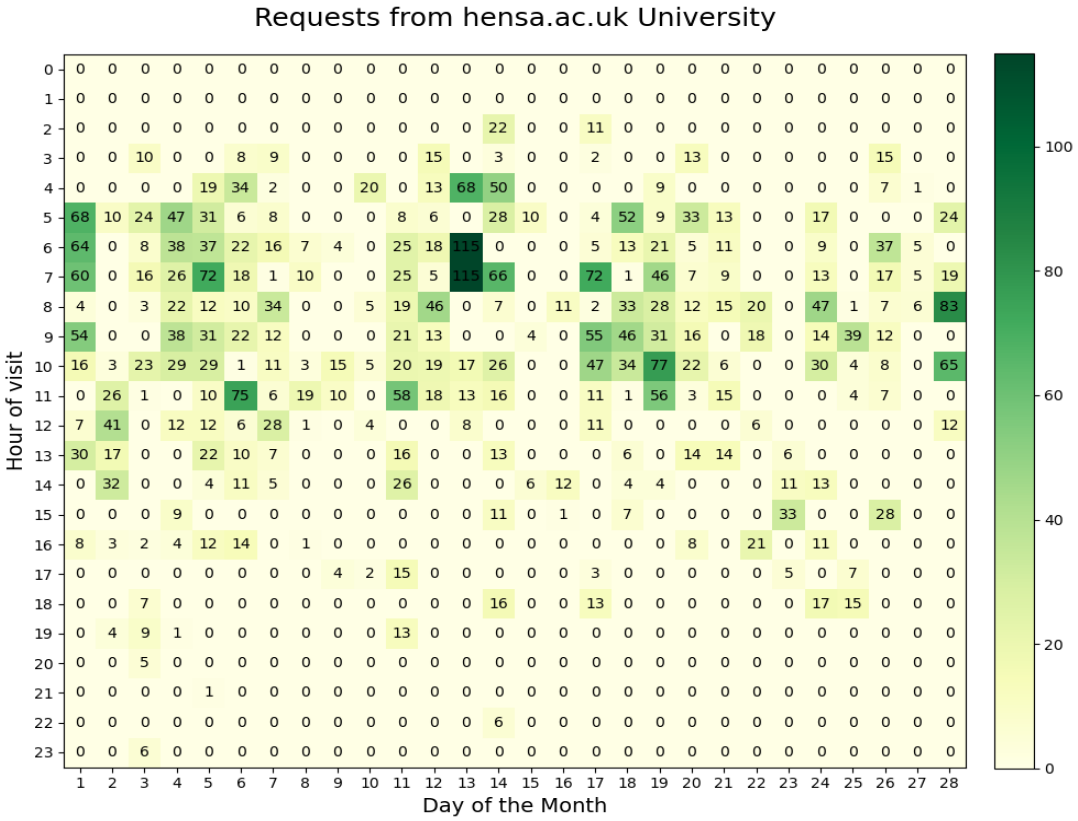


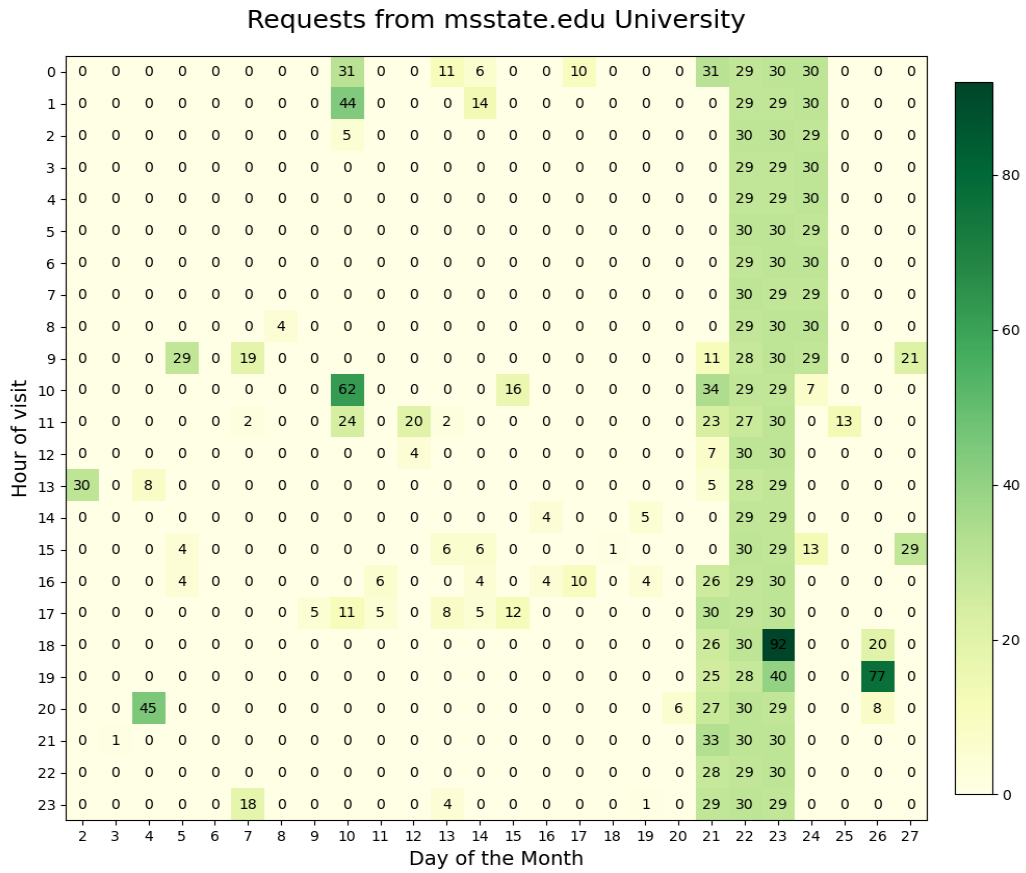
Proportion of requests among US Universities



C. Heat-map plot for the most frequent university from the three countries are presented below. The plots followed the guidelines provided for the axes (range of days on the x-axis and the hour of visit on the y-axis) and the extent of days along with annotation for the number of requests.







D. Observations:

1. Total number of requests from US universities are way too higher than UK and Japan and among US universities the top most frequently visited hosts are all accounted for less than 5% of the total requests which indicates that the traffic is distributed across more number of universities. Universities from US are indeed highly likely to use the services or request the information for the research purposes from the NASA and also the universities might be collaborated with NASA for any study. By understanding the most frequent visited universities and the country, NASA can better architect the servers to handle the traffic by implementing strategies like load balancing. Also NASA can strategically plan for the cloud storages or cache servers to serve the requests efficiently and reduce the latency.

2. The requests from the most visited university host from three countries show different patterns across the month, "**u-tokyo**" from Japan has visits mostly on the late hours to early hours (20:00 to 7:00) whereas the visits from "**Hensa**" (UK Mirror service) were mostly in the peak hours (5:00 to 18:00) and "**msstate**" has the maximum visits on particular time period (21st to 24th) and across whole day. This might be because of difference in the time zones and the working hours in different countries. In case of 'msstate' the visits would probably be due to a study at that particular time period and the research due toward the end of the month. By understanding the visiting hours across the month NASA can effectively plan for any server maintenance works or may be have an automate process of upscaling and downscaling for the efficient use of resources to ensure the maximum availability (up-time) for the end users as per the traffic needs and at the same time can cut-down the server costs.

Question 2. Movie Recommendation and Analysis

A. Time-split Recommendation

1. The ratings data is sorted by the timestamp and by using the percent rank approach the splits are created with the training data sizes 50%, 65% and 80% as specified. Please refer to the A.1 section of the Q2_Code.py file.

2. For each of the three splits three versions of ALS model as described below are built.

1. Default ALS parameters except for seed and coldStartStrategy which is set to 'drop' for avoiding NaN evaluation values by dropping the rows in the DataFrame of predictions that contain NaN values.
2. For the next settings, hyper parameter tuning is performed on the 80% split train data (maximum information) with a choice of values for rank, regularization parameter and max iterations. The second ALS model is chosen with the optimum values of rank and max iterations obtained from tuning. The rank (50) is increased which is the number of latent underlying factors (k) to be fit in the model and it is actually based on the size of the data usually higher rank is preferred until an optimum performance. Next the maxIter (50) is also increased so that more optimizations of lower rank user and item matrices will be performed alternatively to learn the factors.
3. For the third setting, rank, maxIter and regParam are updated. The number of latent factors value (rank - 75) is further increased to incorporate more factors but increasing the rank might lead to overfitting after certain number of latent factors, so regularisation parameter (0.05) is added which prevents overfitting by applying a small amount to the error which requires more time/iterations to fully minimize. And lastly maxIter (50) is kept same, as the optimum regParam is chosen from the hyper tuned model.

Please note that the values chosen for tuning are randomly selected and kept to relatively small values (for the given dataset) such that there will be less impact on the memory and computation time.

3. For each split as in A.1 and for each version of ALS as in A.2, the Root Mean Square Error (RMSE), Mean Square Error (MSE), and Mean Absolute Error (MAE) are computed and the results are tabulated as below.

Split - ALS configuration	RMSE	MSE	MAE
50% - ALS_1	0.79010	0.62425	0.60071
65% - ALS_1	0.80838	0.65348	0.60874
80% - ALS_1	0.85990	0.73943	0.64508
50% - ALS_2	0.78526	0.61664	0.59418
65% - ALS_2	0.80425	0.64682	0.60362
80% - ALS_2	0.85529	0.73152	0.64029
50% - ALS_3	**0.77379**	**0.59876**	**0.58229**
65% - ALS_3	0.79970	0.63952	0.59535
80% - ALS_3	0.85217	0.72620	0.63496

B. User Analysis

1. For each of the three time-splits, KMeans model is applied (with ***k=20***) to cluster the user factors learned from the ALS setting 1 (as in A.2.1 above). The top three largest user cluster sizes are extracted from the models and the results are as shown below.

Split - KMeans_ALS configuration	Size of Cluster_1	Size of Cluster_2	Size of Cluster_3
50% - Kmeans_ALS_1	12631	12200	10941
65% - Kmeans_ALS_1	16818	15998	15364
80% - Kmeans_ALS_1	21512	18625	17395

2. For each of the three splits, for all users in the largest user cluster, top five movie genres are extracted among movies with ratings 4 and above (as specified) in the training set and in the test set and the results are as shown in the table below. A function is written to extract the top n movie genres and the steps involved are clearly commented out, please refer to the B.2 section of the Q2_code.py file.

Split - KMeans_ALS - dataset configuration	Genre_1	Genre_2	Genre_3	Genre_4	Genre_5
50% - Kmeans_ALS_1_train	Drama	Comedy	Thriller	Action	Crime
50% - Kmeans_ALS_1_test	Drama	Comedy	Thriller	Action	Crime
65% - Kmeans_ALS_1_train	Drama	Comedy	Romance	Thriller	Crime
65% - Kmeans_ALS_1_test	Drama	Comedy	Thriller	Romance	Action
80% - Kmeans_ALS_1_train	Drama	Comedy	Romance	Thriller	Action
80% - Kmeans_ALS_1_test	Drama	Comedy	Action	Adventure	Thriller

C. Observations:

1. In case of time based (temporal global split) strategy the model performance is effected by the percentage of the split. It is observed that ALS model on the ml-latest data has produced better performance on the 50-50 % split for all configurations. Global time-split seems to be a realistic approach when it comes to the recommendation models as they need to primarily take into account the temporal behaviour of the users as well and the behaviour observed might be due to fact that the test size is more in the 50% split and the model does not pick up any future patterns of the user and not overfitting. Video streaming applications like Amazon, Netflix etc. can adapt the strategy of time-split to various models to understand the temporal behaviour of the user and leverage the recommendations based on changing user interests.

2. The users from the largest cluster (group that has given similar ratings) mostly like the movies that belong to drama and comedy genres. It is observed that for all three splits the top genres comes out to be drama and comedy followed by thriller, romance, and action, this pattern is most likely because the high proportion of movies made across the time period might belong to these genres. Movie website such as Netflix can take this intuition into account as a generic genre recommendation to the new users since the largest group of similar users have rated highly about or most like the movies of these genres over the period.

Conclusion:

The process of log data analysis, collaborative filtering and clustering has been understood practically through the implementation of the algorithms in Spark environment. All parts of the assignment are answered and the observations are discussed under each section.

References:

1. COM6012 - Labs 1, 2, 3 & 4.
2. https://towardsdatascience.com/scalable-log-analytics-with-apache-spark-a-comprehensive-case-study-2be3eb3be977
3. https://stackoverflow.com/questions/23577505/how-to-avoid-overlapping-of-labels-autopct-in-a-matplotlib-pie-chart
4. https://matplotlib.org/stable/gallery/images_contours_and_fields/image_annotated_heatmap.html
5. https://stackoverflow.com/questions/51772908/split-time-series-pyspark-data-frame-into-test-train-without-using-random-spli
6. https://medium.com/analytics-vidhya/movie-lens-data-analysis-using-pyspark-for-beginners-9c0f5f21eaf5