Question 1

1. After reading the file and processing the date in Spark, the corresponding day of the week can be obtained and counted with the count() action.

   The results are the following:

Day	Avg Requests
Monday	74450.25
Tuesday	70164.25
Wednesday	78148.25
Thursday	92002.25
Friday	70264.25
Saturday	44455.25
Sunday	43444.00

2. When graphing these numbers we can see a clear trend in the data. The requests start to increase from Monday until Thursday and then fall abruptly on Friday and the weekend, probably because people are not accessing the website outside work hours.

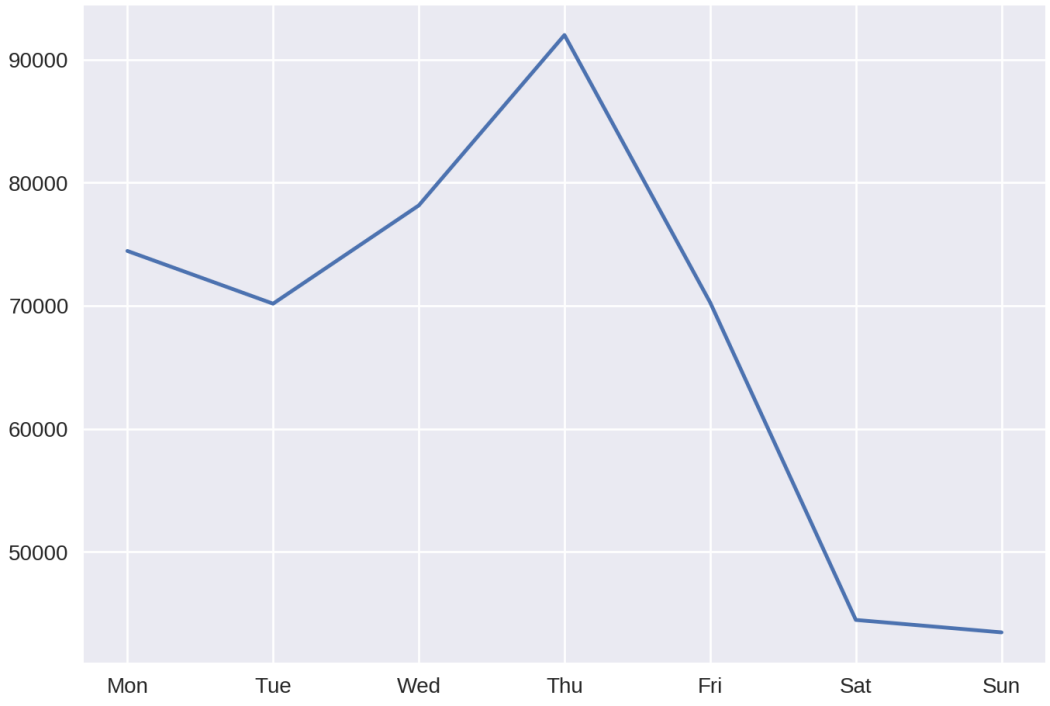


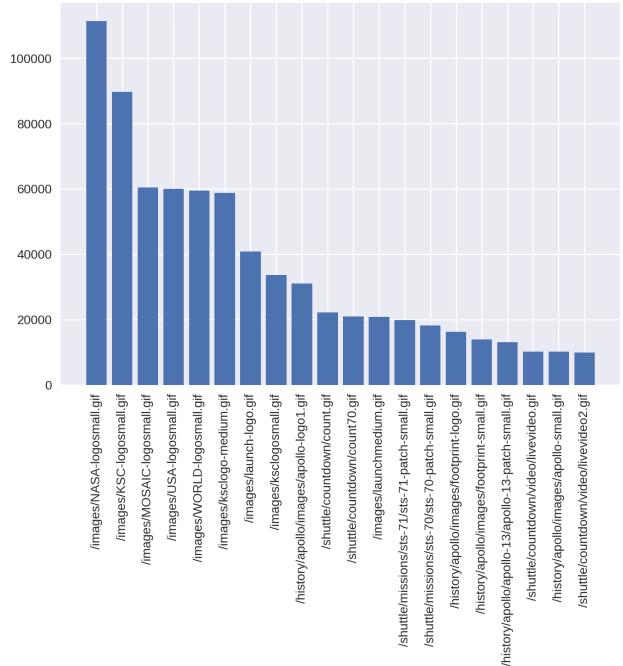
Figure 1: Average requests per day

3. To count the amount of gifs, the data paths that contained '.gif' ending were filtered and then counted.

   The results are the following:

path	amount	% of top 20
/images/NASA-logosmall.gif	111388	15.46%
/images/KSC-logosmall.gif	89639	12.44%
/images/MOSAIC-logosmall.gif	60468	8.39%
/images/USA-logosmall.gif	60014	8.33%
/images/WORLD-logosmall.gif	59489	8.26%
/images/ksclogo-medium.gif	58802	8.16%
/images/launch-logo.gif	40871	5.67%
/images/ksclogosmall.gif	33585	4.66%
/history/apollo/images/apollo-logo1.gif	31072	4.31%
/shuttle/countdown/count.gif	22216	3.08%
/shuttle/countdown/count70.gif	20957	2.91%
/images/launchmedium.gif	20812	2.89%
/shuttle/missions/sts-71/sts-71-patch-small.gif	19853	2.76%
/shuttle/missions/sts-70/sts-70-patch-small.gif	18159	2.52%
/history/apollo/images/footprint-logo.gif	16175	2.24%
/history/apollo/images/footprint-small.gif	13920	1.93%
/history/apollo/apollo-13/apollo-13-patch-small.gif	13014	1.81%
/shuttle/countdown/video/livevideo.gif	10150	1.41%
/history/apollo/images/apollo-small.gif	10095	1.40%
/shuttle/countdown/video/livevideo2.gif	9827	1.36%

4. As we can see in the graph below, our results show that the top5 gifs requested in July amount for 50% of the top 20 gifs. It is also notable that there are big differences, the 1st gif is requested 11 times more than the 20th gif. This is probably because logos are requested each time someone visits the homepage.



Question 2

1. A 5-fold cross validation was done for 2 ALS models, one with the default settings, and another one with the same settings except for a rank of 5

   RMSE scores:

	RMSE 1	RMSE 2	RMSE 3	RMSE 4	RMSE 5	AVG	STD
Defaults	0.8056854	0.8053100	0.8053078	0.8057383	0.8057797	0.8055642	0.0002106
Rank 5	0.8157172	0.8149439	0.8152193	0.8152857	0.8153856	0.8153103	0.0002507

   MAE scores:

	MAE 1	MAE 2	MAE 3	MAE 4	MAE 5	AVG	STD
Defaults	0.6256627	0.6252051	0.6254455	0.6256892	0.6258825	0.6255770	0.0002319
Rank 5	0.6333206	0.6327011	0.6332236	0.6330658	0.6331507	0.6330924	0.0002128

2. From the results above we can see that the model with rank 10 has a better RMSE and a better MAE than a rank 5, which corresponds with the intuition that a higher rank will be able to get better results (up to a certain point) but with more computing time.

3. For each model (with defaults) in part 1, a KNN(20) model was trained on the itemFactors of each split. The tag results for the 3 biggers clusters for each split are the following

	model 1	**model 2**	**model 3**	**model 4**	**model 5**
Cluster 1	original	original	original	original	original
	mentor	mentor	predictable	storytelling	predictable
	great ending	great ending	mentor	mentor	mentor
	dialogue	dialogue	great ending	great ending	great ending
	predictable	predictable	dialogue	good soundtrack	so bad it's funny
Cluster 2	original	original	original	original	original
	predictable	storytelling	mentor	mentor	mentor
	mentor	mentor	storytelling	dialogue	great ending
	great ending	good soundtrack	great ending	great ending	dialogue
	so bad it's funny	great ending	good soundtrack	catastrophe	catastrophe
Cluster 3	original	original	original	original	original
	mentor	great ending	great ending	predictable	great ending
	storytelling	mentor	mentor	mentor	mentor
	great ending	dialogue	dialogue	great ending	storytelling
	good soundtrack	good soundtrack	great	so bad it's funny	dialogue

4. As per the results of the previous table we can see that the algorithm predicts almost the same tags in the clusters for each split, this shows that the algorithm is pretty stable across the different datasets splits we trained it on. There are some cases where you can see that one cluster in one model is the same as other cluster in another model, for example the Cluster 1 in model 5, is the same as the Cluster 2 in model 1. This probably means that even if the data is different in the split, the clusters will be similar, even though not in the same order.