Question 1

Assume **MySQL** is being used
The query should be:
select count(*)
from
(select distinct uid from `piwik_track` where 1 and `time` >= '2017-04-01' and `time` < '2017-04-02' and event_name = 'FIRST_INSTALL')A join
(select distinct uid from `piwik_track` where 1 and `time` >= '2017-04-02' and `time` <= '2017-04-08' and event_name <> 'FIRST_INSTALL') B on A.uid = B.uid

Explanation:
The query is composed of two sub queries. One for detecting users with FIRST_INSTALL between 2017-04-01(inclusive) and 2017-04-02(exclusive). The second sub query detects users with non FIRST_INSTALL between 2017-04-02 and 2017-04-08.

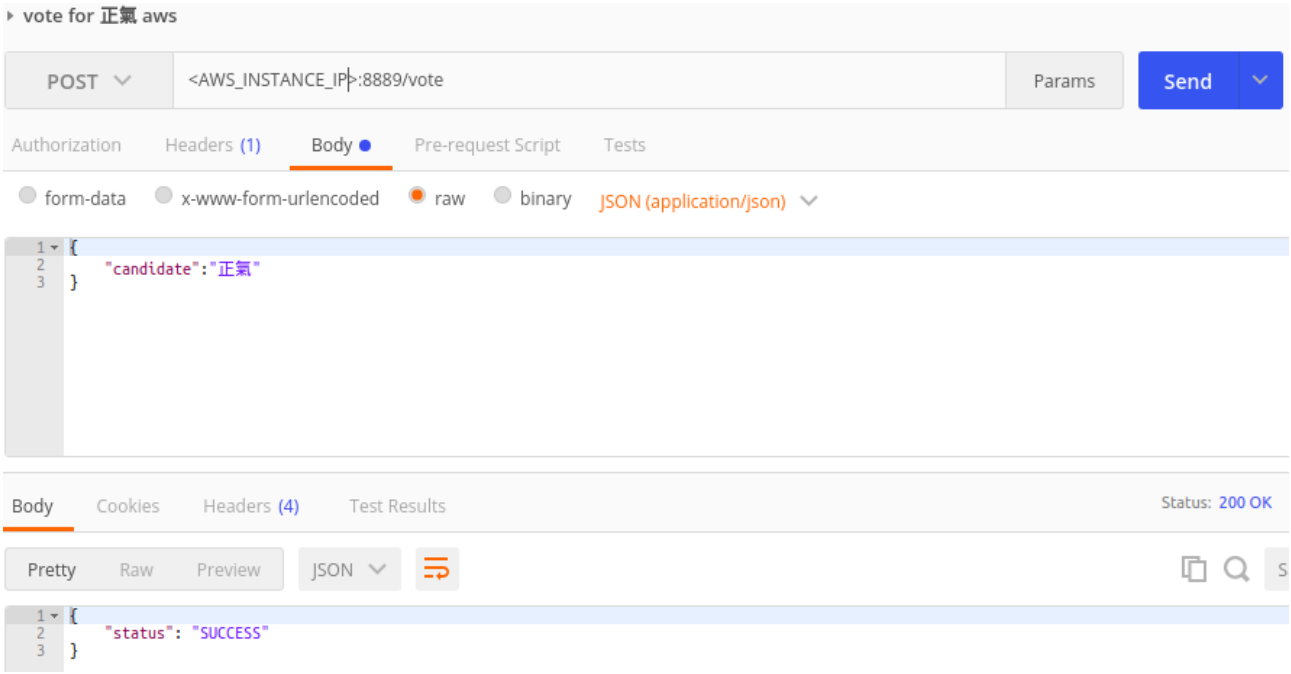
Question 2

Please refer to q2.ipynb. The program will read ./logfile and compute the sum of jpg images.

Question 3a

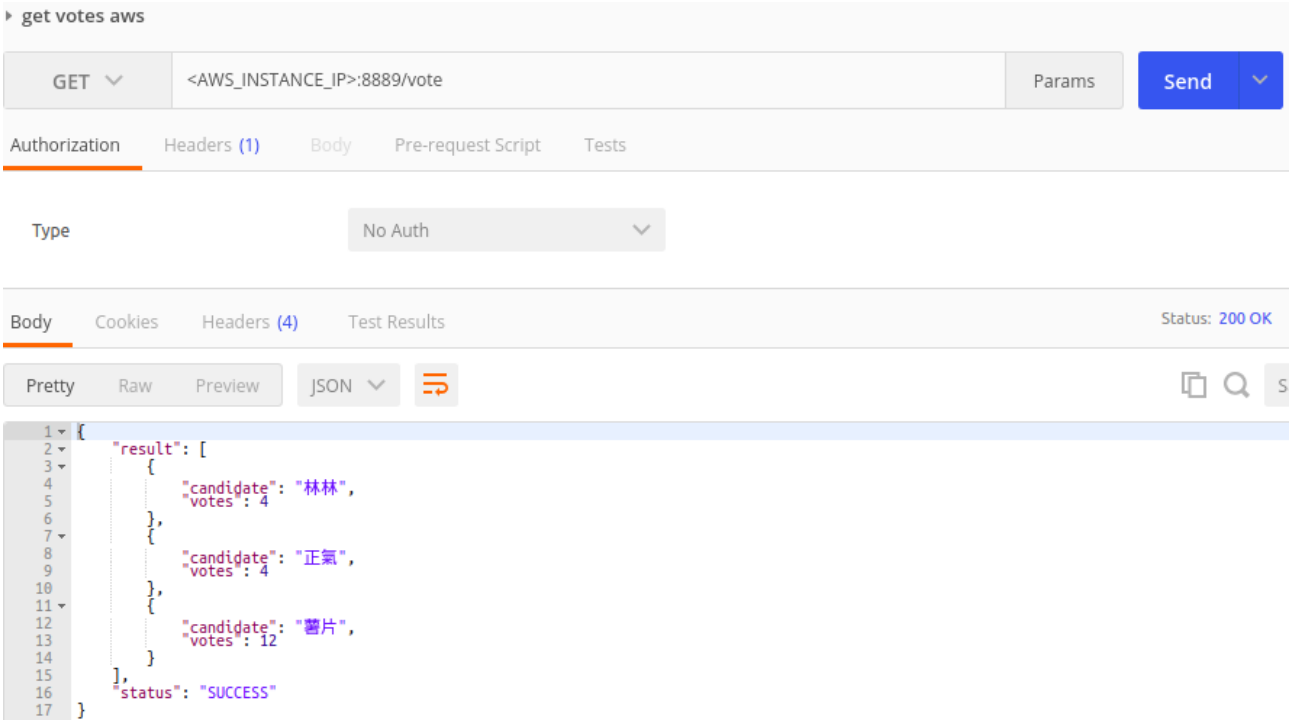
The code is located in q3a/q3a_backend. For simplicity, I implemented the endpoints using a sqlite database and Flask.

Sample API requests for voting:



If the request goes through, you should receive a json with SUCCESS status.

Sample API requests for voting



The <AWS_INSTANCE_IP> will be provided in the email.

Environment Setup:
You will find q3a_conda_env.yml and q3a_pip.txt in q3a/q3a_backend. You can recreate the anaconda environment using q3a_conda_env.yml by executing the following command:
conda env create -f q3a_conda_env.yml

After setting up the anaconda environment, you need to activate the anaconda environment by executing the following command:
source activate q3a

After that, you can startup the web service by running the following command in q3a/q3a_backend:
python run.py

Now, you can fire the above API requests. As a side note, the sqlite file is q3a.sqlite in q3a/q3a_backend.

Question 3b

Please refer to q3b.ipynb. The program requires jieba, pandas, numpy and scikit-learn to be installed. For jieba, it is only available in pip. To install it, please run the following command:
pip install jieba
For other libraries, please run the following command
conda install pandas numpy scikit-learn

Other than these libraries, I also wrote a helper method(located in stats_utils.py) which generates a confusion matrix and some useful statistics for evaluating the performance of the model.

I inspect the validation set (offsite-tagging-test-set (1).csv). It does not have any tags. Hence, I wrote a simple scrapper to store those tags in raw_validation_csv.csv.

1. The hit rate is 100% on the training set and the hit rate is 99.38% on validation set. Here are the confusion matrices.

Training Set		Actual		
		梁振英	美國大選	足球
Predicted	梁振英	929	0	0
	美國大選	0	842	0
	足球	0	0	2123

As you can see, all the non diagonal entries are 0 meaning that there are 0 misclassifications. Hence, the hit rate is 100%.

Validation Set		Actual		
		梁振英	美國大選	足球
Predicted	梁振英	245	2	0
	美國大選	1	214	0
	足球	3	0	509

For validation set, there are 2 + 1 + 3 = 6 misclassifications and there are 974 records. Hence, the hit rate is 1 – 6/974 = 0.9938 (99.38%).

2. My model is rather simple. The parameters are features to include in the model and whether or not to treat each feature as binary or frequency. I choose my parameters based on the training and validation set hit rate. The better hit rate, the better the parameters.

3. I think that the performance of the model is dependent on many factors but the key factors are the features. Therefore, I focus mainly on extracting useful features from the articles. My final version has 450 features.
   To find these features, I first organized my training set into 3 groups.

Group Name	**Description**
soccer_news	Articles with tags = 足球
cy_news	Articles with tags = 梁振英
election_news	Articles with tags = 美國大選

For each group, I concat all articles together and end up having 3 giantic documents.

Document Name	**Description**
soccer_doc	Articles with tags = 足球
cy_doc	Articles with tags = 梁振英
election_doc	Articles with tags = 美國大選

For each document, I used jieba to tokenize the documents and removed all stopwords and punctuations. After that, I compute the frequencies for each token. Then, I compute the tfidf for each token relative to soccer_doc,cy_doc and election_doc. The tfidf matrix will look like this:

	soccer_doc	**cy_doc**	**election_doc**
包圍	0.000729	0.001258	0.001170
包埋	0.000000	0.000000	0.000117
包容	0.000260	0.000719	0.001515
...	...	...	...

For each document, I sort the tokens by the tfidf and select the top 150 tokens as features. As a side note, I play around with this value (tried 20, 30, 50, 100, 150) and find that 150 tokens are enough.

Each article in the training set will be represented as a vector. Each vector entry indicates whether or not the token exists. Then, I feed those vectors with tags to the logistic regression algorithm.

Here is my list of predicted tags for the validation set:

```
['足球', '梁振英', '足球', '足球', '梁振英', '梁振英', '梁振英', '足球', '足球', '足
球', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '梁振英', '梁振英',
 '梁振英', '美國大選', '梁振英', '梁振英', '梁振英', '梁振英', '梁振英', '足球', '梁
振英', '足球', '梁振英', '足球', '梁振英', '足球', '足球', '足球', '足球', '梁振英',
'美國大選', '足球', '美國大選', '足球', '梁振英', '足球', '足球', '足球', '足球',
 '足球', '足球', '足球', '足球', '足球', '美國大選', '足球', '足球', '美國大選', '足
球', '足球', '足球', '足球', '足球', '足球', '美國大選', '美國大選', '美國大選', '美
國大選', '足球', '足球', '美國大選', '足球', '足球', '足球', '足球', '足球', '梁振
英', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '美國大選', '梁振英',
 '美國大選', '美國大選', '美國大選', '足球', '梁振英', '足球', '足球', '足球', '美國
大選', '梁振英', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '美國大選',
'足球', '足球', '足球', '足球', '梁振英', '足球', '足球', '足球', '梁振英', '美國大
選', '梁振英', '足球', '足球', '足球', '梁振英', '梁振英', '美國大選', '美國大選',
 '美國大選', '美國大選', '足球', '足球', '美國大選', '梁振英', '美國大選', '梁振英',
 '美國大選', '足球', '足球', '足球', '足球', '梁振英', '足球', '美國大選', '足球',
 '足球', '足球', '足球', '足球', '足球', '足球', '美國大選', '美國大選', '美國大選',
'足球', '足球', '足球', '梁振英', '足球', '足球', '足球', '足球', '足球', '足球',
 '梁振英', '足球', '美國大選', '足球', '美國大選', '足球', '足球', '足球', '足球',
 '足球', '足球', '足球', '足球', '足球', '梁振英', '足球', '足球', '足球', '梁振英',
 '美國大選', '梁振英', '梁振英', '梁振英', '梁振英', '足球', '足球', '梁振英', '梁振
英', '足球', '美國大選', '梁振英', '梁振英', '梁振英', '足球', '梁振英', '梁振英',
 '足球', '足球', '梁振英', '足球', '足球', '足球', '梁振英', '梁振英', '足球', '足
球', '梁振英', '梁振英', '梁振英', '梁振英', '梁振英', '足球', '梁振英', '梁振英',
 '梁振英', '美國大選', '足球', '美國大選', '美國大選', '梁振英', '足球', '美國大選',
 '梁振英', '足球', '足球', '足球', '足球', '梁振英', '足球', '足球', '足球', '足球',
 '足球', '美國大選', '美國大選', '美國大選', '美國大選', '足球', '足球', '足球', '足
```

球', '梁振英', '足球', '足球', '足球', '梁振英', '美國大選', '梁振英', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '美國大選', '美國大選', '足球', '足球', '梁振英', '梁振英', '梁振英', '梁振英', '足球', '足球', '梁振英', '梁振英', '梁振英', '足球', '梁振英', '美國大選', '足球', '足球', '足球', '足球', '足球', '美國大選', '美國大選', '足球', '足球', '梁振英', '梁振英', '足球', '梁振英', '足球', '梁振英', '梁振英', '美國大選', '梁振英', '足球', '足球', '梁振英', '美國大選', '梁振英', '足球', '足球', '足球', '美國大選', '足球', '美國大選', '梁振英', '梁振英', '梁振英', '美國大選', '梁振英', '梁振英', '足球', '美國大選', '足球', '梁振英', '足球', '足球', '足球', '足球', '梁振英', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '美國大選', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '梁振英', '梁振英', '梁振英', '足球', '足球', '足球', '足球', '足球', '梁振英', '足球', '足球', '足球', '足球', '足球', '足球', '梁振英', '足球', '足球', '梁振英', '足球', '足球', '美國大選', '梁振英', '足球', '足球', '梁振英', '梁振英', '梁振英', '梁振英', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '梁振英', '梁振英', '梁振英', '足球', '梁振英', '足球', '足球', '足球', '梁振英', '足球', '足球', '足球', '足球', '美國大選', '足球', '美國大選', '足球', '美國大選', '美國大選', '足球', '足球', '足球', '足球', '足球', '美國大選', '足球', '美國大選', '足球', '梁振英', '梁振英', '足球', '足球', '足球', '美國大選', '足球', '美國大選', '足球', '梁振英', '梁振英', '美國大選', '足球', '足球', '足球', '梁振英', '美國大選', '足球', '足球', '足球', '美國大選', '足球', '足球', '足球', '梁振英', '梁振英', '足球', '梁振英', '梁振英', '梁振英', '梁振英', '足球', '足球', '足球', '美國大選', '梁振英', '足球', '梁振英', '美國大選', '足球', '足球', '美國大選', '美國大選', '美國大選', '梁振英', '足球', '梁振英', '美國大選', '梁振英', '梁振英', '梁振英', '美國大選', '梁振英', '梁振英', '梁振英', '美國大選', '足球', '足球', '美國大選', '梁振英', '美國大選', '梁振英', '美國大選', '美國大選', '美國大選', '梁振英', '美國大選', '美國大選', '美國大選', '足球', '梁振英', '美國大選', '美國大選', '足球', '梁振英', '美國大選', '美國大選', '足球', '足球', '足球', '美國大選', '美國大選', '美國大選', '足球', '足球', '足球', '美國大選', '足球', '美國大選', '足球', '美國大選', '梁振英', '梁振英', '美國大選', '美國大選', '梁振英', '美國大選', '梁振英', '梁振英', '梁振英', '美國大選', '梁振英', '梁振英', '梁振英', '美國大選', '足球', '足球', '梁振英', '足球', '美國大選', '梁振英', '美國大選', '美國大選', '足球', '足球', '美國大選', '美國大選', '美國大選', '梁振英', '梁振英', '足球', '足球', '美國大選', '美國大選', '美國大選', '美國大選', '美國大選', '美國大選', '美國大選', '美國大選', '美國大選', '美國大選', '美國大選', '美國大選', '美國大選', '梁振英', '梁振英', '美國大選', '美國大選', '足球', '美國大選', '美國大選', '梁振英', '美國大選', '梁振英', '梁振英', '梁振英', '美國大選', '梁振英', '梁振英', '梁振英', '美國大選', '足球', '足球', '梁振英', '美國大選', '美國大選', '美國大選', '美國大選', '美國大選', '足球', '梁振英', '美國大選', '美國大選', '美國大選', '美國大選', '美國大選', '足球', '美國大選', '美國大選', '美國大選', '美國大選', '美國大選', '梁振英', '足球', '美國大選', '美國大選', '美國大選', '梁振英', '美國大選', '美國大選', '美國大選', '美國大選', '足球', '美國大選', '美國大選', '美國大選', '美國大選', '美國大選', '美國大選', '美國大選', '梁振英', '美國大選', '美國大選', '美國大選', '美國大選', '美國大選', '美國大選', '足球', '美國大選', '足球', '美國大選', '足球', '足球', '美國大選', '美國大選', '足球', '美國大選', '美國大選',

'足球', '梁振英', '梁振英', '美國大選', '足球', '美國大選', '美國大選', '足球', '足球', '梁振英', '美國大選', '足球', '足球', '梁振英', '梁振英', '足球', '足球', '梁振英', '足球', '足球', '足球', '足球', '足球', '梁振英', '美國大選', '足球', '足球', '足球', '足球', '梁振英', '梁振英', '梁振英', '足球', '足球', '足球', '梁振英', '梁振英', '梁振英', '梁振英', '梁振英', '梁振英', '梁振英', '梁振英', '梁振英', '梁振英', '美國大選', '梁振英', '梁振英', '梁振英', '梁振英', '梁振英', '足球', '足球', '梁振英', '梁振英', '足球', '梁振英', '梁振英', '足球', '足球', '足球', '梁振英', '足球', '美國大選', '梁振英', '梁振英', '美國大選', '梁振英', '梁振英', '足球', '足球', '梁振英', '梁振英', '足球', '梁振英', '梁振英', '足球', '足球', '梁振英', '足球', '足球', '足球', '梁振英', '足球', '梁振英', '美國大選', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '美國大選', '美國大選', '梁振英', '足球', '美國大選', '梁振英', '梁振英', '足球', '足球', '足球', '足球', '足球', '梁振英', '梁振英', '足球', '足球', '梁振英', '足球', '美國大選', '足球', '足球', '足球', '梁振英', '足球', '足球', '足球', '足球', '美國大選', '梁振英', '足球', '足球', '梁振英', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '梁振英', '足球', '足球', '足球', '足球', '足球', '足球', '梁振英', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '梁振英', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '梁振英', '梁振英', '足球', '足球', '足球', '梁振英', '足球', '足球', '足球', '梁振英', '足球', '足球', '足球', '足球', '梁振英', '足球', '足球', '足球', '足球', '梁振英', '足球', '梁振英', '梁振英', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '梁振英', '足球', '梁振英', '梁振英', '足球', '梁振英', '足球', '足球', '梁振英', '足球', '足球', '足球', '足球', '足球', '足球', '梁振英', '梁振英', '足球', '梁振英', '足球', '梁振英', '梁振英', '梁振英', '梁振英', '足球', '足球', '足球', '足球', '足球', '足球', '梁振英', '足球', '足球', '足球', '梁振英', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '足球', '梁振英', '足球', '足球', '足球', '梁振英', '足球', '梁振英', '足球', '梁振英', '足球', '梁振英', '足球', '梁振英', '足球', '梁振英', '足球', '梁振英', '足球', '足球', '足球', '梁振英', '梁振英', '足球', '足球', '梁振英', '足球', '足球', '足球', '梁振英', '足球', '梁振英', '足球', '梁振英', '足球', '足球', '足球', '足球', '足球']

Question 3c

Please refer to q3c.ipynb. The program requires pandas and numpy library.

Design of Recommendation System:

My implementation of the recommendation system will return a list of artists for a given female user id. The first step is to extract to female users from usersha1-profile.tsv(I didnt include the original tsv files because the files are huge) and stored them into q3c_female_profiles.pkl. Then, I extract all listening history for these female users from usersha1-artmbid-artname-plays.tsv. After that, I need to prepare the rating matrix where the rows are artists and columns are the female user ids. Since there are too many artists, I only consider artists with accumulative play frequencies over 100000. After creating the rating matrix, I need to build the normalized rating matrix by substracting the average rating from the original rating for each artist. After that, I build a similarity matrix where the rows and columns are the popular artists. Each entry in the similarity matrix is calculated using normalized rating matrix and the cosine similarity. The similarity matrix is symmetric. Entry[i][j] is the cosine similarity between artist i and artist j. Everything required to build my recommendation engine is ready.

The engine will take a female user id. Then, it will find the list of artists that the female user already listened to and the list of artists that the female never listened to. At this point, the goal is to fill in the ratings for the unlistened artists. By using the similarity matrix, I can find the similar artists and I compute the rating by using the weighted average. After computing all missing ratings, I sort them and extract the top ratings as recommendations.

How to use the recommendation system?

Please run the very last cell of q3c.ipynb. This cell implements a function called recommendation_engine. It takes 3 parameters. The first parameter the female user id. The second parameter is the maximum number of recommendation. The third parameter is the maximum number of neighbors. The first parameter is compulsory and the others are optional. I also include an example of the recommendation_engine usage at the end of the cell.

Limitations:

Currently, the recommendation engine will only recommend popular artists (subset of artists).
In production, it should consider all artists. Producing the supporting matrices (rating matrix, normalized rating, similarity matrix) are extremely slow. In production, the rating matrix will be updated constantly. Hence, the similarity matrix needs to be recomputed and this computation is very time consuming. Therefore, the current implementation may not be efficient in production environment.