CSL 542 Project
Commercial use and visualisation of BrightKite dataset

Himanshu Dahiya(2015csb1013)
Keshav Garg (2015csb1016)

1. INTRODUCTION AND BACKGROUND

This project is based on Brightkite dataset. Brightkite was a location-based social networking website. The dataset consists of two files, the friendship network (Brightkite edges.txt) and the check-ins (Brightkite totalCheckins.txt). Brightkite edges.txt file contains friendship between the users while Brightkite totalCheckins.txt file contains node id (user who checked-in), check-in time by the user, check-in latitude, check-in longitude, and check-in location id values.

2. AIM

Our aim is to understand, mine and analyze the given dataset in order to discover the hidden and obscure information, and get an interesting insight of the data. We propose to do that in following ways.

3. APPROACH AND METHODOLOGY

3.1 Task 1

1. **Pattern of check-ins on a specific day**: User will be asked to input a day, and in return, he/she will be shown pattern of check-ins on all locations for that specific date. A list will be generated which will show the frequently visited location on that day. A support threshold must be provided by the user.

2. **Pattern of check-ins at a specific location**: User will be asked to input a location ID, and in return, he/she will be shown pattern of check-ins on that location for all days. Our algorithm will show most busiest days of that location based on the support threshold provided by the user. Another option is to see the timeline of number of visitors for that location.

3. **Check-ins on a specific day at a specific location**: User will be asked to input a location ID and a specific date, and in return, he/she will be shown all the check-ins on that location for that specific day.

4. **Analyses of above task** For analyzing check-ins on a particular date, y-axis will show number of check-ins, and x-axis will show the location ID. For analyzing check-ins on a particular location, y-axis will show number of check-ins, and x-axis will show the date.

3.2 Task 2

1. **Single user check-in patterns:** In this task, user is asked to input a userID. In return, the K-Means algorithm will give clusters for locations which the user frequently visits. K-Means is used for this task.

2. **Group check-in patterns:** In this task, user is asked to input a userID. Then we extract all the check-ins of this user and all his friends. In return, the K-Means algorithm will return the clusters for locations which the group of friends frequently visits.

3.3 Task 3

The following tasks were performed:

1. **Predict future check-ins at a location:** In this task, we have predicted the future check-in patterns of a user. For each user, we have predicted if he/she is going out in future and what can be possible check-ins on that day. To do this, we looked into his/her past check-in records, and used regression to predict the possible future check-ins. X-axis represent the time-line of user and Y-axis represent the number of check-ins he/she has at that particular time. Based on this, we got a check-in sequence for that user.

2. **Predict future check-ins of a user:** In this task, we have predicted the future number of check-ins at a particular location. For each location, we have predicted that how many possible check-ins can be there on any date in future. To do so, we looked into location's past check-in record, and used regression to predict the possible future number of check-ins on that location. X-axis represent the time-line of location and Y-axis represent the number of check-ins the location has at that particular time. Based on this, we got a check-in sequence for that particular location.

3.4 Task 4

1. **Show all the friends of a person**: User will be asked to input a userID, and in return, he/she will be shown all the friends of this userID.

2. **Find mutual friends of two people**: User will be asked to input userIDs of two people, and in return, he/she will be shown the userIDs of their mutual friends.

3. **Give friend suggestion for a person**: User will be asked to input a userID and a support value. Then the user will be suggested other userIDs, with whom the number of mutual friends surpasses the support value.

4. **Print friend list of all users to a file**: User will have an option to print the friend list of every user to a text file.

4. IMPLEMENTATION

4.1 Experimental Settings

Tasks 2 and 3 were performed on the following mentioned computer.

- Computer:MacBook Pro (13-inch, Mid 2012)
- OS: MacOS Mojave 10.14.5
- RAM : 8 GB 1600 MHz DDR3
- Processor: 2.5 GHz Intel Core i5
- Graphics: Intel HD Graphics 4000 1536 MB

Tasks 1 and 4 were performed on the following mentioned computer.

- OS: Ubuntu 18.04.2 LTS (64-bit)
- RAM : 7.7 GiB
- Processor: IntelÂő CoreâĎć i5-5200U CPU @ 2.20GHz ÃŮ 4
- Graphics: Intel HD Graphics 5500 (Broadwell GT2)

4.2 Evaluation measures

4.2.1 Task 1:

Efficiency measure: Time of execution 2.615479999999998 seconds Method Used: Filtering, Frequent Pattern Mining

4.2.2 Task 2:

Efficiency measure: Time of execution 6.915384000000003 seconds for userID = 11543 and single user. Methods Used: K-Means

4.2.3 Task 3:

Efficiency measure: Time of execution 0.671289999999999 seconds Methods Used: Linear Regression and Polynomial Regression with deg=2

4.2.4 Task 4:

Efficiency measure: Time of execution 0.557564 seconds Method Used: Filtering and Statistics

4.3 Experimental Results and Discussion

4.3.1 Task 1

- **Figure 1** It shows the the number of check-ins on different location at a particular date ”2009-04-01”.
- **Figure 2** It shows the the number of check-ins at a specific location on different dates.
- **Figure 3** It shows the the number of check-ins at a specific location and on a particular date ”2009-04-01”. It also shows the time taken by our algorithm to run.

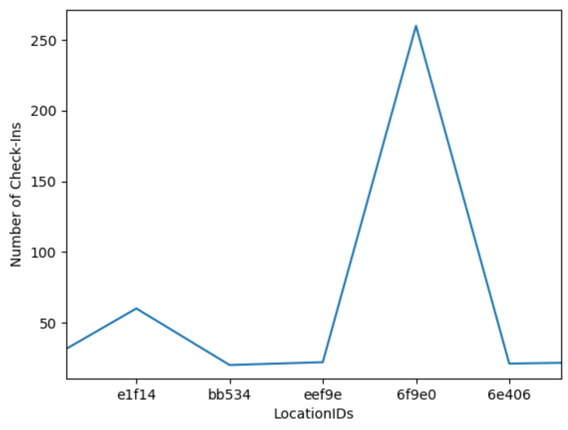


Figure 1: Number of check-ins vs Location ID

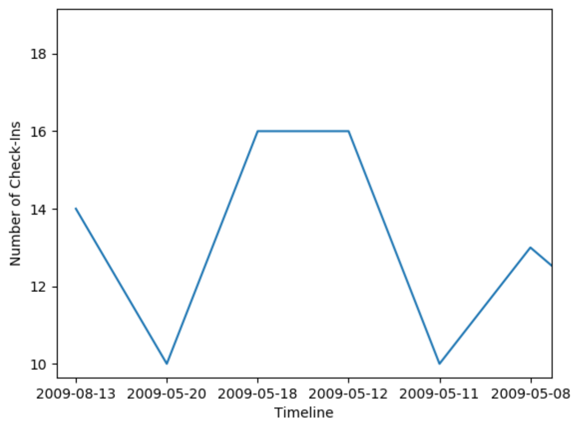


Figure 2: Number of check-ins vs Time-line

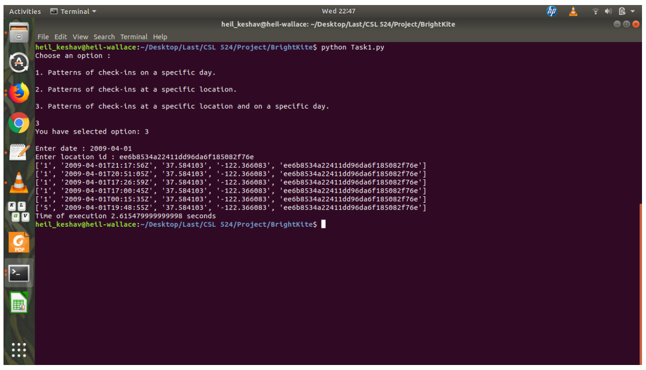
4.3.2 Task 2

- **Figure 4** It represents the cluster of locations a single user goes based on his/her check-ins. We experimented for userID=11543 for this task.
- **Figure 5** It represents the cluster of locations a single user and his friends goes based on their check-ins. We experimented for userID=11543 and extracted all the check-ins of this user and his/her friend’s check-ins.

4.3.3 Task 3

The results are for locationID = dd7cd3d264c2d063832db506fba8bf7 Similar results can also be generated for any userID by selecting option 2 while executing the code.

- **Figure 6** It represents the number of check-ins on a given date. The points are the actual number of check-ins and the curve is the regression fitted curve with degree=1, which means linear regression.
- **Figure 7** It represents the number of check-ins on a given date. The points are the actual number of check-



```
heil_keshav@heil-wallace:~/Desktop/Last/CSL 524/Project/BrightKite$ python Task1.py
Choose an option :

1. Patterns of check-ins on a specific day.

2. Patterns of check-ins at a specific location.

3. Patterns of check-ins at a specific location and on a specific day.

3
You have selected option: 3

Enter date : 2009-04-01
Enter location id : ee6b8534a22411dd96da6f185082f76e
['1', '2009-04-01T21:17:56Z', '37.584103', '-122.366083', 'ee6b8534a22411dd96da6f185082f76e']
['1', '2009-04-01T20:51:05Z', '37.584103', '-122.366083', 'ee6b8534a22411dd96da6f185082f76e']
['1', '2009-04-01T17:26:59Z', '37.584103', '-122.366083', 'ee6b8534a22411dd96da6f185082f76e']
['1', '2009-04-01T17:00:45Z', '37.584103', '-122.366083', 'ee6b8534a22411dd96da6f185082f76e']
['1', '2009-04-01T00:15:35Z', '37.584103', '-122.366083', 'ee6b8534a22411dd96da6f185082f76e']
['5', '2009-04-01T19:48:55Z', '37.584103', '-122.366083', 'ee6b8534a22411dd96da6f185082f76e']
Time of execution 2.615479999999998 seconds
heil_keshav@heil-wallace:~/Desktop/Last/CSL 524/Project/BrightKite$
```

Figure 3: Check-Ins on a fix date and location

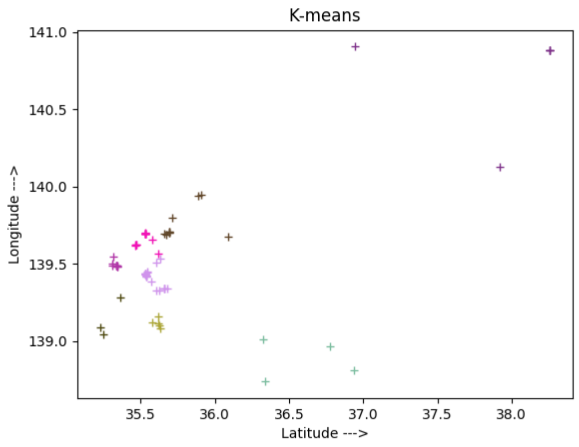


Figure 4: K Means clustering on single user with userID = 11543

ins and the curve is the regression fitted curve with degree=2, which means quadratic regression.

- **Figure 8** It shows a terminal output which has the linear regression coefficient values, date wise actual and predicted values, and at the end predicted value for a particular given date.

- **Figure 9** It shows a terminal output which has the quadratic regression coefficient values, date wise actual and predicted values, and at the end predicted value for a particular given date.

4.3.4 Task 4

- **Figure 10** Given a UserID, it shows all the userIDs of friends of this person.

- **Figure 11** Given UserIDs of two people, it shows the userIDs of their mutual friends.

- **Figure 12** Given a UserID and a support value, it shows all the userIDs of suggested friends in sorted order.

- **Figure 13** It generates a text file, where each entry represents the friend list of a user.

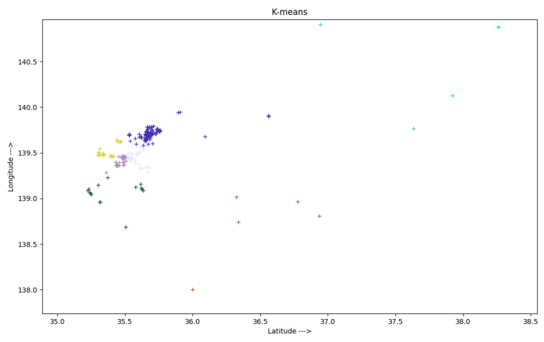


Figure 5: K Means clustering on group of friends of a user with userID = 11543

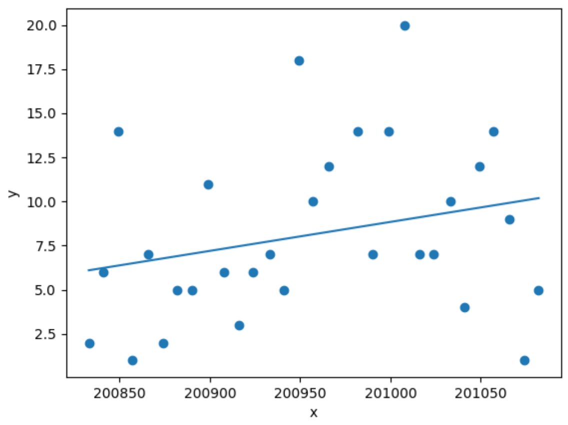


Figure 6: Linear regression plot for locationID = dd7cd3d264c2d063832db506fba8bf79 and degree = 1

5. CONCLUSION

After implementing the above algorithms on different tasks, We found some interesting frequent patterns in the given dataset like groups of people that frequently check-in together, users which happens to check-in on similar locations, prediction of future time and location of check-ins by a user. Based on the check-in patterns of a user, we can refer other locations to the user with better value proposal (commercially beneficial). We can also make communities of user who visits specific locations often. This also has commercial value, say in case if both person are visiting there from same location, they can share the transportation costs by having a common ride.

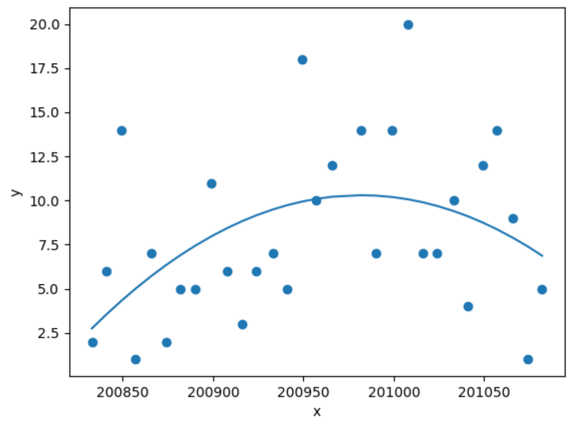
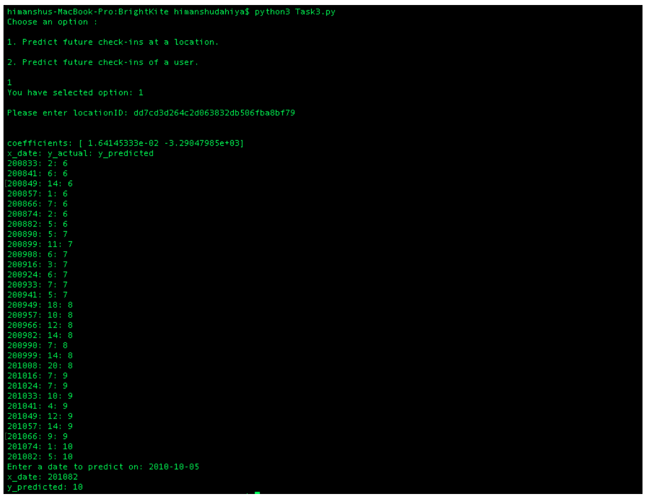
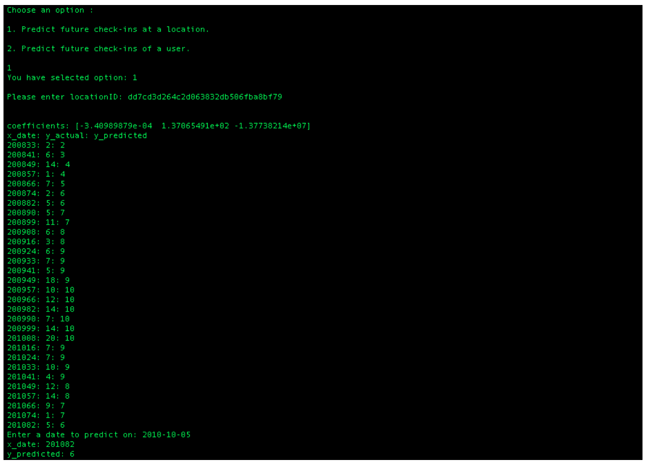


Figure 7: Quadratic regression plot for locationID = dd7cd3d264c2d063832db506fba8bf79 and degree = 2



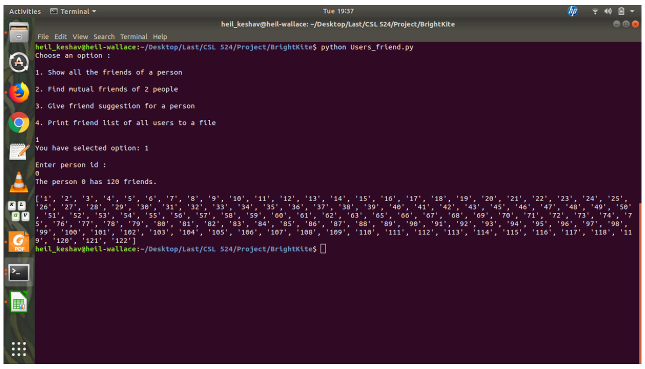
```
himanshus-MacBook-Pro:BrightKite himanshudahiya$ python3 Task3.py
Choose an option :

1. Predict future check-ins at a location.

2. Predict future check-ins of a user.

1
You have selected option: 1

Please enter locationID: dd7cd3d264c2d063832db506fba8bf79


coefficients: [ 1.64145333e-02 -3.29047985e+03]
x_date: y_actual: y_predicted
200833: 2: 6
200841: 6: 6
200849: 14: 6
200857: 1: 6
200866: 7: 6
200874: 2: 6
200882: 5: 6
200890: 5: 7
200899: 11: 7
200908: 6: 7
200916: 3: 7
200924: 6: 7
200933: 7: 7
200941: 5: 7
200949: 18: 8
200957: 10: 8
200966: 12: 8
200982: 14: 8
200990: 7: 8
200999: 14: 8
201008: 20: 8
201016: 7: 9
201024: 7: 9
201033: 10: 9
201041: 4: 9
201049: 12: 9
201057: 14: 9
201066: 9: 9
201074: 1: 10
201082: 5: 10
Enter a date to predict on: 2010-10-05
x_date: 201082
y_predicted: 10
```

Figure 8: Terminal output for linear regression with locationID = dd7cd3d264c2d063832db506fba8bf79 and degree = 1



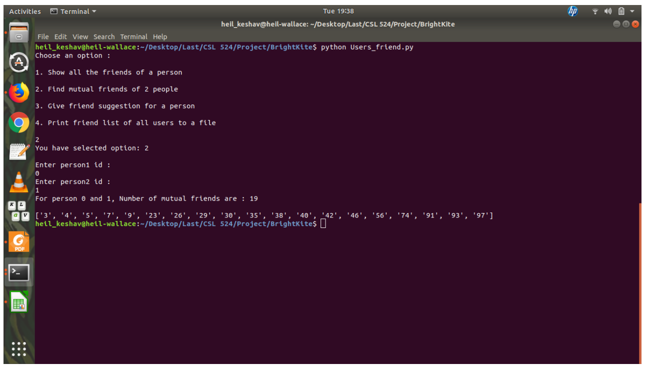
```
Choose an option :

1. Predict future check-ins at a location.

2. Predict future check-ins of a user.

1
You have selected option: 1

Please enter locationID: dd7cd3d264c2d063832db506fba8bf79


coefficients: [-3.40989879e-04  1.37065491e+02 -1.37738214e+07]
x_date: y_actual: y_predicted
200833: 2: 2
200841: 6: 3
200849: 14: 4
200857: 1: 4
200866: 7: 5
200874: 2: 6
200882: 5: 6
200890: 5: 7
200899: 11: 7
200908: 6: 8
200916: 3: 8
200924: 6: 9
200933: 7: 9
200941: 5: 9
200949: 18: 9
200957: 10: 10
200966: 12: 10
200982: 14: 10
200990: 7: 10
200999: 14: 10
201008: 20: 10
201016: 7: 9
201024: 7: 9
201033: 10: 9
201041: 4: 9
201049: 12: 8
201057: 14: 8
201066: 9: 7
201074: 1: 7
201082: 5: 6
Enter a date to predict on: 2010-10-05
x_date: 201082
y_predicted: 6
```

Figure 9: Terminal output for quadratic regression with locationID = dd7cd3d264c2d063832db506fba8bf79 and degree = 2


Activities
Terminal
Tue 19:37
heil_keshav@heil-wallace: ~/Desktop/Last/CSL 524/Project/BrightKite
File Edit View Search Terminal Help

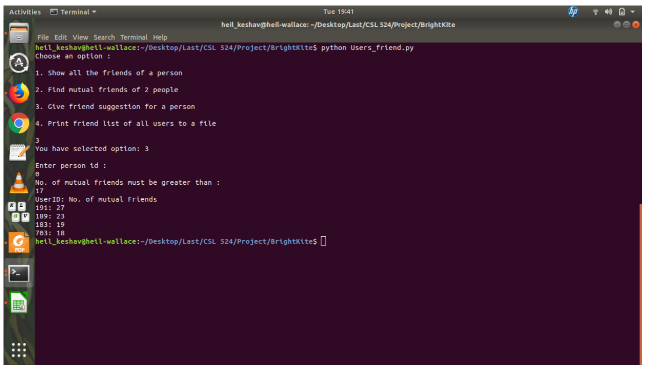
```
heil_keshav@heil-wallace:~/Desktop/Last/CSL 524/Project/BrightKite$ python Users_friend.py
Choose an option :

1. Show all the friends of a person

2. Find mutual friends of 2 people

3. Give friend suggestion for a person

4. Print friend list of all users to a file

1
You have selected option: 1

Enter person id :
0
The person 0 has 120 friends.

['1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20', '21', '22', '23', '24', '25',
'26', '27', '28', '29', '30', '31', '32', '33', '34', '35', '36', '37', '38', '39', '40', '41', '42', '43', '45', '46', '47', '48', '49', '50'
, '51', '52', '53', '54', '55', '56', '57', '58', '59', '60', '61', '62', '63', '65', '66', '67', '68', '69', '70', '71', '72', '73', '74', '7
5', '76', '77', '78', '79', '80', '81', '82', '83', '84', '85', '86', '87', '88', '89', '90', '91', '92', '93', '94', '95', '96', '97', '98',
'99', '100', '101', '102', '103', '104', '105', '106', '107', '108', '109', '110', '111', '112', '113', '114', '115', '116', '117', '118', '11
9', '120', '121', '122']
heil_keshav@heil-wallace:~/Desktop/Last/CSL 524/Project/BrightKite$ 
```

Figure 10: List of friends of user ”0”


Activities
Terminal
Tue 19:38
heil_keshav@heil-wallace: ~/Desktop/Last/CSL 524/Project/BrightKite
File Edit View Search Terminal Help

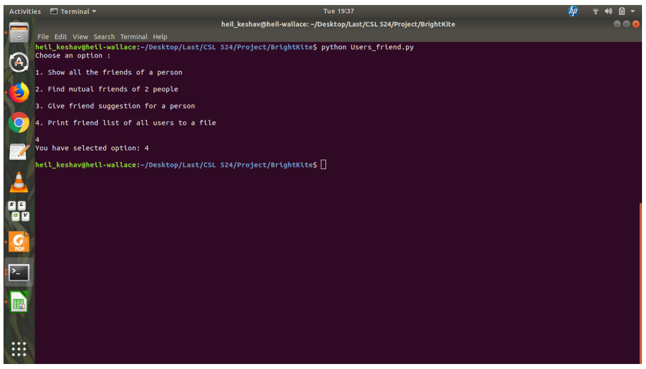
```
heil_keshav@heil-wallace:~/Desktop/Last/CSL 524/Project/BrightKite$ python Users_friend.py
Choose an option :

1. Show all the friends of a person

2. Find mutual friends of 2 people

3. Give friend suggestion for a person

4. Print friend list of all users to a file

2
You have selected option: 2

Enter person1 id :
0
Enter person2 id :
1
For person 0 and 1, Number of mutual friends are : 19

['3', '4', '5', '7', '9', '23', '26', '29', '30', '35', '38', '40', '42', '46', '56', '74', '91', '93', '97']
heil_keshav@heil-wallace:~/Desktop/Last/CSL 524/Project/BrightKite$ 
```

Figure 11: Mutual Friends of ”0” and ”1”


Activities
Terminal
Tue 19:41
heil_keshav@heil-wallace: ~/Desktop/Last/CSL 524/Project/BrightKite
File Edit View Search Terminal Help

```
heil_keshav@heil-wallace:~/Desktop/Last/CSL 524/Project/BrightKite$ python Users_friend.py
Choose an option :

1. Show all the friends of a person

2. Find mutual friends of 2 people

3. Give friend suggestion for a person

4. Print friend list of all users to a file

3
You have selected option: 3

Enter person id :
0
No. of mutual friends must be greater than :
17
UserID: No. of mutual Friends
191: 27
189: 23
183: 19
703: 18
heil_keshav@heil-wallace:~/Desktop/Last/CSL 524/Project/BrightKite$ 
```

Figure 12: Suggest friends for userID ”0”


Activities
Terminal
Tue 19:37
heil_keshav@heil-wallace: ~/Desktop/Last/CSL 524/Project/BrightKite
File Edit View Search Terminal Help

```
heil_keshav@heil-wallace:~/Desktop/Last/CSL 524/Project/BrightKite$ python Users_friend.py
Choose an option :

1. Show all the friends of a person

2. Find mutual friends of 2 people

3. Give friend suggestion for a person

4. Print friend list of all users to a file

4
You have selected option: 4

heil_keshav@heil-wallace:~/Desktop/Last/CSL 524/Project/BrightKite$ 
```

Figure 13: Generate friend lists