INTRODUCTION TO WEB SCIENCES: Assignment 8

Babitha Bokka

15 November 2014

Contents

1 Question 1:

What 5 movies have the highest average ratings? Show the movies and their ratings sorted by their average ratings.

1.1 Approach

1. To solve this question the straight forward approach is to read u.data file get movie id, corresponding movie ratings and read u.item file to get movie id and corresponding movie name.
2. Extract the movie id and ratings. For each movie id build a dictionary that has all the ratings.
3. Now, for each movie id calculate the average using sum() and len() functions.
4. For each movie id get the movie name and append it to the dictionary which has average.
5. Now, sort the dictionary and print the top 5 movies with highest average ratings.
6. Program averageMovies.py produces the top 5 movies with highest average ratings. The output is based on a condition which can be changed accordingly.
7. Figure 1 shows the results of the program.

1.2 Source Code

1.2.1 averageMovies.py

```python
#!/usr/bin/env python

import sys

def main():
    # Create a dictionary
    movie_id_ratings    = {}
    movie_id_avg        = {}
    count = 1
    # Read the input files.
    readData = open('/home/bbokka/cs594/A8/u.data', 'r')
    readItem = open('/home/bbokka/cs594/A8/u.item','r')
    # reading the u.data file for item and ratings.
    for line in readData:
        split_input_line = line.strip().split('\t')
        movie_id         =  split_input_line[1]
        movie_rating     =  float(split_input_line[2])
        try:
            movie_id_ratings[movie_id].append(movie_rating)
        except KeyError:
            movie_id_ratings[movie_id] = list()
            movie_id_ratings[movie_id].append(movie_rating)
    readData.close()
    # Calculating the average.
    for key in movie_id_ratings:
        avg = sum(movie_id_ratings[key])/len(movie_id_ratings[key])
        movie_id_avg[key] = [float(avg)]

    # Reading the u.item file for movie name.
    for each_line in readItem:
        split_each_line  = each_line.strip().split('|')
        movie_item_id    = split_each_line[0]
        movie_name_split = split_each_line[1].split('(1')
        movie_name       = movie_name_split[0]

        movie_id_avg[movie_item_id].append(movie_name)
    readItem.close()
    print "*" * 60
    print "Top 5 movies having the highest average ratings"
    print "*" * 60
    print "Movie Name\t\t\t\t\t","Avg Rating"
    print "-" * 60
    # sorting the movies from highest to lowest based on the average value.
    for key, value in sorted(movie_id_avg.items(), key=lambda e: e[1], reverse=True):
        if (count <= 30):
            print '{:<50}{:<0.1f} '  .format(value[1],value[0])
            count += 1

if __name__ == "__main__":
    try:
        main()
    except KeyboardInterrupt:
        sys.exit(1)
```

1.3 Input Files

1.3.1 u.data

```
userid itemid rating timestamp

196     242     3   881250949
186     302     3   891717742
22      377     1   878887116
244     51      2   880606923
166     346     1   886397596
298     474     4   884182806
115     265     2   881171488
```

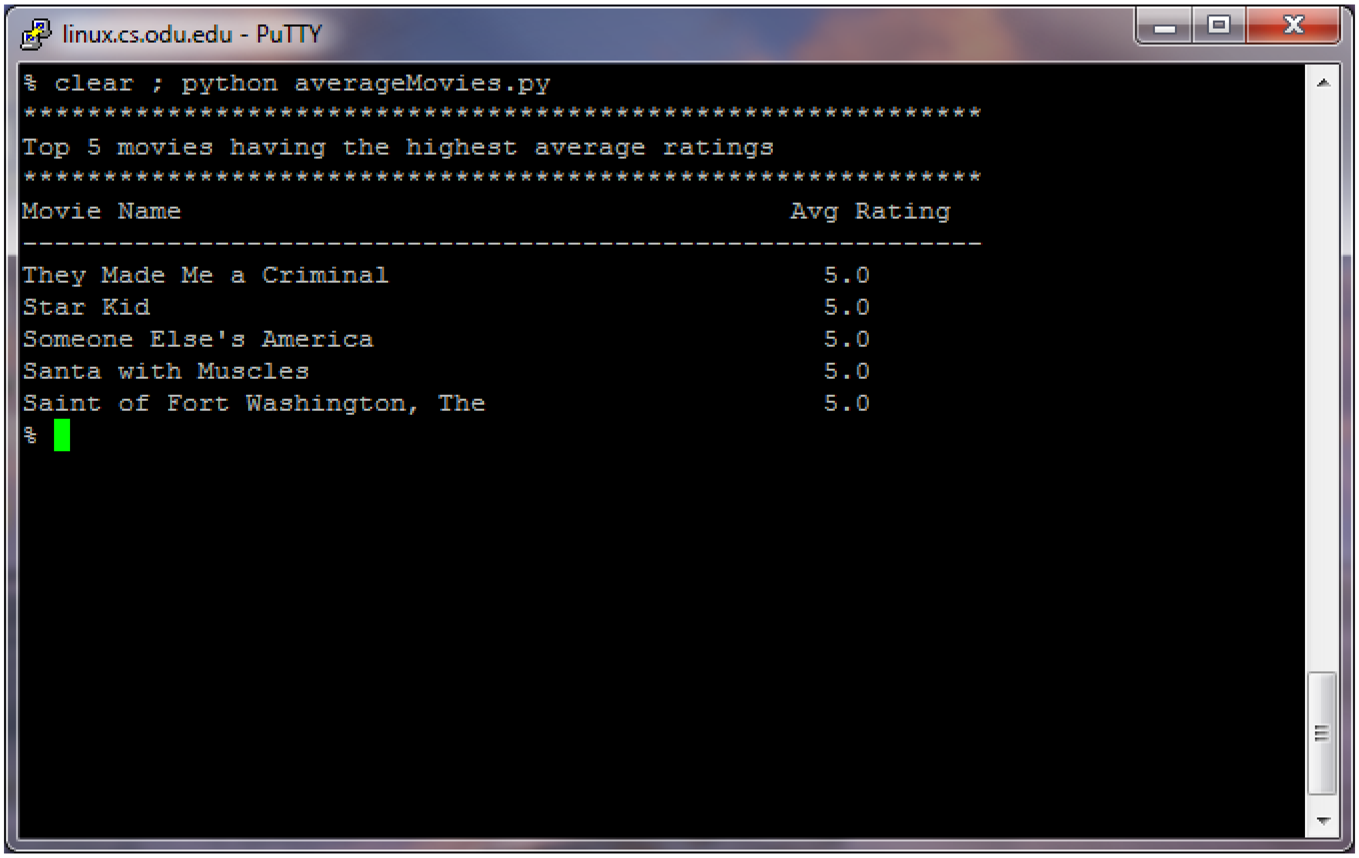
1.3.2 u.item

```
movie id | movie title | release date | video release date | IMDb URL | unknown |
    Action | Adventure | Animation |Children's | Comedy | Crime | Documentary | Drama
    | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller |
     War | Western |

1|Toy Story (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)
    |0|0|0|1|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0
2|GoldenEye (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?GoldenEye%20(1995)
    |0|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0
3|Four Rooms (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Four%20Rooms
    %20(1995)|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0
4|Get Shorty (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Get%20Shorty
    %20(1995)|0|1|0|0|0|1|0|0|1|0|0|0|0|0|0|0|0|0|0
5|Copycat (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Copycat%20(1995)
    |0|0|0|0|0|0|1|0|1|0|0|0|0|0|0|0|1|0|0
6|Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)|01-Jan-1995||http://us.imdb.com
    /Title?Yao+a+yao+yao+dao+waipo+qiao+(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0
7|Twelve Monkeys (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Twelve%20Monkeys
    %20(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|1|0|0|0
```

1.4 Output Files

1.4.1 averageMovies.png



```
% clear ; python averageMovies.py
************************************************************
Top 5 movies having the highest average ratings
************************************************************
Movie Name                                        Avg Rating
------------------------------------------------------------
They Made Me a Criminal                             5.0
Star Kid                                            5.0
Someone Else's America                              5.0
Santa with Muscles                                  5.0
Saint of Fort Washington, The                       5.0
%
```

Figure 1: Top 5 movies with highest average ratings.

1.4.2 averageMoviesMore.txt

Movies with highest average ratings. The above output shows only top 5 but the below output shows all the movies with the same average ratings.

```
% clear ; python averageMovies.py
************************************************************
Top 5 movies having the highest average ratings
************************************************************
Movie Name                                          Avg Rating
------------------------------------------------------------
They Made Me a Criminal                               5.0
Star Kid                                              5.0
Someone Else's America                                5.0
Santa with Muscles                                    5.0
Saint of Fort Washington, The                         5.0
Prefontaine                                           5.0
Marlene Dietrich: Shadow and Light                    5.0
Great Day in Harlem, A                                5.0
Entertaining Angels: The Dorothy Day Story            5.0
Aiqing wansui                                         5.0
Pather Panchali                                       4.6
Some Mother's Son                                     4.5
```

2 Question 2:

What 5 movies received the most ratings? Show the movies and the number of ratings sorted by number of ratings.

2.1 Approach

1. The approach is same as question 1.
2. The only difference is we count the number of ratings for each movie and display the corresponding movie name and number of ratings for each movie.
3. The output displays only top 5 movies and their number of ratings.
4. Figure 2 is the output for the program.

2.2 Source Code

2.2.1 mostRatingMovies.py

```python
#!/usr/bin/env python

import sys

def main():
    # Create a dictionary
    movie_id_ratings    = {}
    movie_id_avg        = {}
    count = 1
    # Read the input files.
    readData = open('/home/bbokka/cs594/A8/u.data', 'r')
    readItem = open('/home/bbokka/cs594/A8/u.item','r')
    # reading the u.data file for item and ratings.
    for line in readData:
        split_input_line = line.strip().split('\t')
        movie_id         =  split_input_line[1]
        movie_rating     =  int(split_input_line[2])
        try:
            movie_id_ratings[movie_id].append(movie_rating)
        except KeyError:
            movie_id_ratings[movie_id] = list()
            movie_id_ratings[movie_id].append(movie_rating)
    readData.close()
    # Calculating the average.
    for key in movie_id_ratings:
        length = len(movie_id_ratings[key])
        movie_id_avg[key] = [int(length)]

    # Reading the u.item file for movie name.
    for each_line in readItem:
        split_each_line  = each_line.strip().split('|')
        movie_item_id    = split_each_line[0]
        movie_name_split = split_each_line[1].split('(1')
        movie_name       = movie_name_split[0]

        movie_id_avg[movie_item_id].append(movie_name)
    readItem.close()
    print '*' * 60
    print "Top 5 movies received the most ratings"
    print "*" * 60
    print "Movie Name\t\t\t\t\t","Number of ratings"
    print "-" * 60
    # sorting the movies from highest to lowest based on the average value.
    for key, value in sorted(movie_id_avg.items(), key=lambda e: e[1], reverse=True):
        if (count <= 5):
            print '{:<50}{} '  .format(value[1],value[0])
            count += 1

if __name__ == "__main__":
    try:
        main()
    except KeyboardInterrupt:
        sys.exit(1)
```

2.3 Input Files

2.3.1 u.data

```
userid itemid rating timestamp

196    242    3   881250949
186    302    3   891717742
22     377    1   878887116
244    51     2   880606923
166    346    1   886397596
298    474    4   884182806
115    265    2   881171488
```

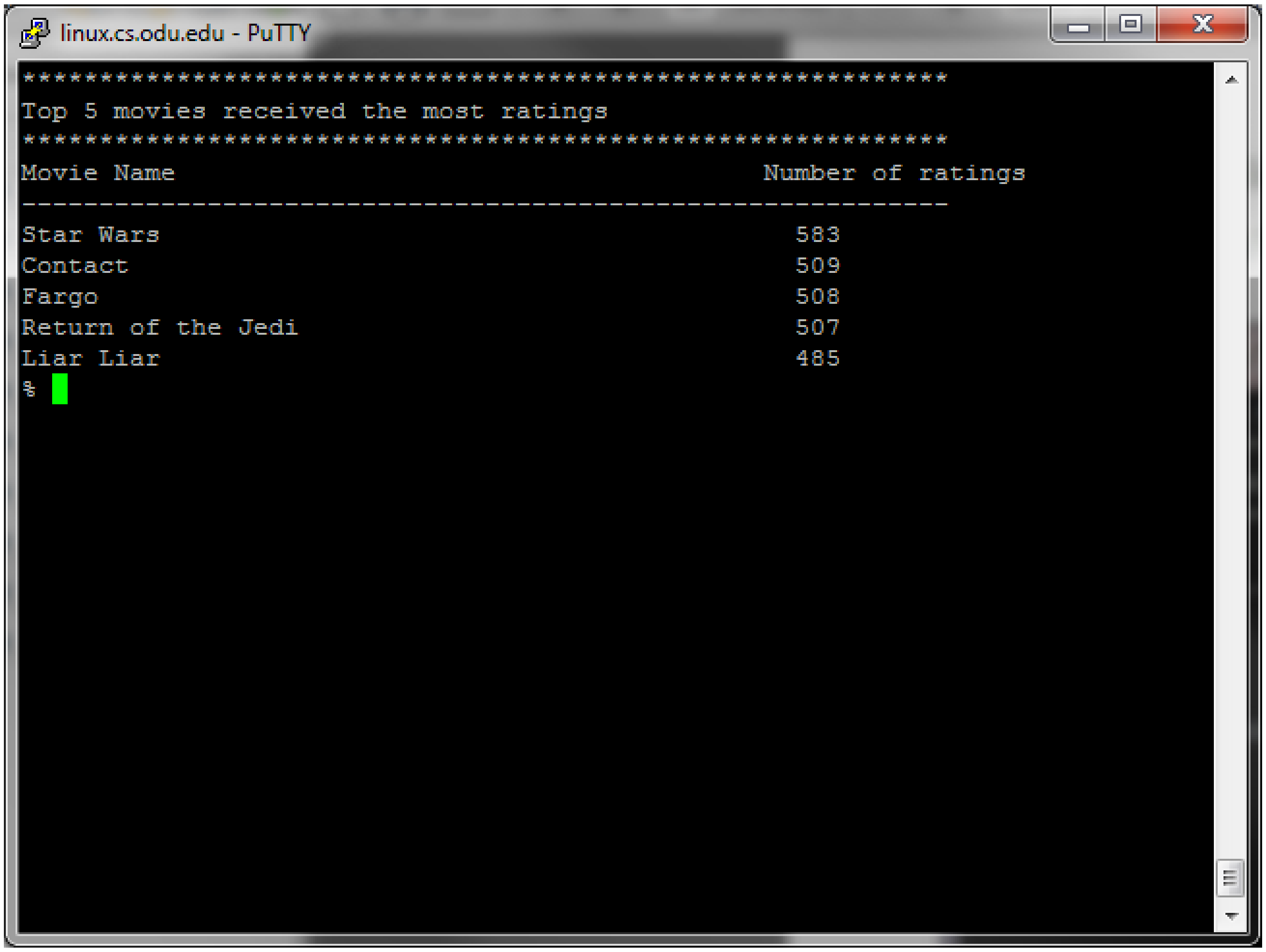
2.3.2 u.item

```
movie id | movie title | release date | video release date | IMDb URL | unknown |
    Action | Adventure | Animation |Children's | Comedy | Crime | Documentary | Drama
    | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller |
     War | Western |

1|Toy Story (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)
    |0|0|0|1|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0
2|GoldenEye (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?GoldenEye%20(1995)
    |0|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0
3|Four Rooms (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Four%20Rooms
    %20(1995)|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0
4|Get Shorty (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Get%20Shorty
    %20(1995)|0|1|0|0|0|1|0|0|1|0|0|0|0|0|0|0|0|0|0
5|Copycat (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Copycat%20(1995)
    |0|0|0|0|0|0|1|0|1|0|0|0|0|0|0|0|1|0|0
6|Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)|01-Jan-1995||http://us.imdb.com
    /Title?Yao+a+yao+yao+dao+waipo+qiao+(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0
7|Twelve Monkeys (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Twelve%20Monkeys
    %20(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|1|0|0|0
```

2.4 Output Files

2.4.1 mostRatingMovies.png



```
************************************************************
Top 5 movies received the most ratings
************************************************************
Movie Name                                        Number of ratings
------------------------------------------------------------
Star Wars                                           583
Contact                                             509
Fargo                                               508
Return of the Jedi                                  507
Liar Liar                                           485
%
```

Figure 2: Top 5 movies received the most ratings.

3 Question 3:

What 5 movies were rated the highest on average by women? Show the movies and their ratings sorted by ratings.

3.1 Approach

1. To solve this question the straight forward approach is to read u.user file get the user id, gender. Read u.data file get movie id, corresponding movie ratings. Read u.item file to get movie id and corresponding movie name.

2. From u.user file extract the user id and gender store them in a dictionary.

3. Now, Extract the movie id and ratings. For each user id build a dictionary that has all the ratings for the respective gender.

4. Now, for each movie id calculate the average a using sum() and len() functions.

5. For each movie id get the movie name and append it to the dictionary which has average.

6. Now, sort the dictionary and print the top 5 movies rated the highest on average by women.

7. Program averageGender.py is designed in such a way that it can get the highest averages by men and women just by changing the arguments to F or M while executing the program.

8. Figure 3 is the output showing top 5 movies rated the highest on average by women. The results of the program are limited to top 5 it can be modified to get more average ratings.

3.2 Source Code

3.2.1 averageMovies.py

```
#!/usr/bin/env python


# program to get the highest average ratings by men and women
import sys
import operator
def main():
    # take the arguments from the command line
    numOfArgs=len(sys.argv)
    if numOfArgs<2 or numOfArgs>2:

        print'Usage: averageGender.py <F or M> '
        print'e.g. : averageGender.py F'
        sys.exit(1)
    gender           = sys.argv[1]
    # Create a dictionary
    movie_userid_avg = {}
    movie_id_ratings = {}
    movie_userid_avg = {}
    users            = {}
    count            = 1

    readData = open('/home/bbokka/cs594/A8/u.data', 'r')
    readItem = open('/home/bbokka/cs594/A8/u.item','r')
    readUser = open('/home/bbokka/cs594/A8/u.user','r')

    # Reading u.user file to get the male or female
    for line in readUser:
        split_input_line = line.strip().split('|')
        user_id_from_user= split_input_line[0]
        user_gender      = split_input_line[2]

        if(gender.upper() == user_gender):
            users[user_id_from_user] = [user_gender]

    readUser.close()

    # reading the u.data file for item and ratings.
    for line in readData:
        split_input_line  = line.strip().split('\t')
        user_id_from_data = split_input_line[0]
        movie_id_from_data=  split_input_line[1]
        movie_rating      =  float(split_input_line[2])

        if user_id_from_data in users:
            try:
                movie_id_ratings[movie_id_from_data].append(movie_rating)
            except KeyError:
                movie_id_ratings[movie_id_from_data] = list()
                movie_id_ratings[movie_id_from_data].append(movie_rating)
    readData.close()

    # Calculating the average.
    for key in movie_id_ratings:
        avg = sum(movie_id_ratings[key])/len(movie_id_ratings[key])
        movie_userid_avg[key] = [float(avg)]

    # Reading the u.item file for movie name.
    for each_line in readItem:
        split_each_line  = each_line.strip().split('|')
        movie_item_id    = split_each_line[0]
        movie_name_split = split_each_line[1].split('(1')
        movie_name       = movie_name_split[0]

        try:
            movie_userid_avg[movie_item_id].append(movie_name)
        except KeyError:
            pass
    readItem.close()
    print "*" * 60
    print "Top 5 movies rated the highest on average by women"
    print "*" * 60
    print "Movie Name\t\t\t\t","Avg Rating"
    print "-" * 60
    # sorting the movies from highest to lowest based on the average value.
    for key, value in sorted(movie_userid_avg.items(), key=lambda e: e[1], reverse=
True):
        if (count <=20 ):
            print '{:<50}{:<0.1f} '  .format(value[1],value[0])
            count += 1

if __name__ == "__main__":
    try:
        main()
    except KeyboardInterrupt:
        sys.exit(1)
```

3.3 Input Files

3.3.1 u.data

```
userid  itemid  rating  timestamp

196      242      3    881250949
186      302      3    891717742
22       377      1    878887116
244      51       2    880606923
166      346      1    886397596
298      474      4    884182806
115      265      2    881171488
```

3.3.2 u.item

```
movie id | movie title | release date | video release date | IMDb URL | unknown |
    Action | Adventure | Animation |Children's | Comedy | Crime | Documentary | Drama
    | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller |
     War | Western |

1|Toy Story (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)
    |0|0|0|1|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0
2|GoldenEye (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?GoldenEye%20(1995)
    |0|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0
3|Four Rooms (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Four%20Rooms
    %20(1995)|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0
4|Get Shorty (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Get%20Shorty
    %20(1995)|0|1|0|0|0|1|0|0|1|0|0|0|0|0|0|0|0|0|0
5|Copycat (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Copycat%20(1995)
    |0|0|0|0|0|0|1|0|1|0|0|0|0|0|0|0|1|0|0
6|Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)|01-Jan-1995||http://us.imdb.com
    /Title?Yao+a+yao+yao+dao+waipo+qiao+(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0
7|Twelve Monkeys (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Twelve%20Monkeys
    %20(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|1|0|0|0
```

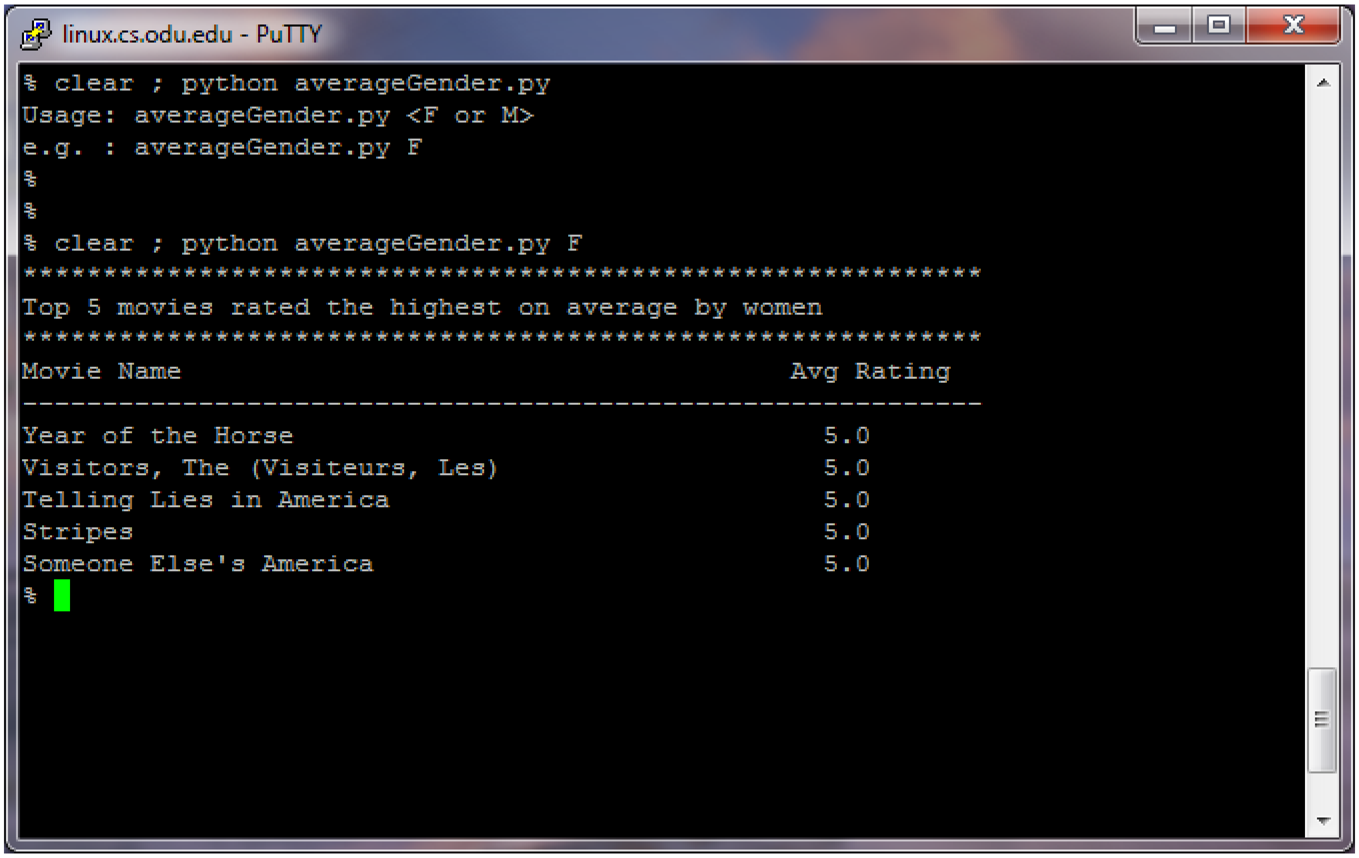
3.3.3 u.user

```
userid   age    gender     occupation     zip code

1       24         M             technician          85711
2       53         F      other               94043
3       23         M      writer              32067
4       24         M      technician          43537
5       33         F      other               15213
6       42         M      executive           98101
7       57         M      administrator       91344
```

3.4 Output Files

3.4.1 averageGenderF.png



```
% clear ; python averageGender.py
Usage: averageGender.py <F or M>
e.g. : averageGender.py F
%
%
% clear ; python averageGender.py F
************************************************************
Top 5 movies rated the highest on average by women
************************************************************
Movie Name                                      Avg Rating
------------------------------------------------------------
Year of the Horse                                 5.0
Visitors, The (Visiteurs, Les)                    5.0
Telling Lies in America                           5.0
Stripes                                           5.0
Someone Else's America                            5.0
%
```

Figure 3: Top 5 movies rated the highest on average by women.

3.4.2 averageGenderFull.txt

Showing all the movies with same average ratings.

```
% clear ; python averageGender.py
Usage: averageGender.py <F or M>
e.g. : averageGender.py F
%
%
%
% clear ; python averageGender.py F
***********************************************************
Top 5 movies rated the highest on average by women
***********************************************************
Movie Name                                          Avg Rating
-----------------------------------------------------------
Year of the Horse                                     5.0
Visitors, The (Visiteurs, Les)                        5.0
Telling Lies in America                               5.0
Stripes                                               5.0
Someone Else's America                                5.0
Prefontaine                                           5.0
Mina Tannenbaum                                       5.0
Maya Lin: A Strong Clear Vision                       5.0
Foreign Correspondent                                 5.0
Faster Pussycat! Kill! Kill!                          5.0
Everest                                               5.0
Schindler's List                                      4.6
Close Shave, A                                        4.6
Shawshank Redemption, The                             4.6
Wallace & Gromit: The Best of Aardman Animation       4.5
Shall We Dance?                                       4.5
Women, The                                            4.5
Some Mother's Son                                     4.5
Friday                                                4.5
Band Wagon, The                                       4.5
```

4 Question 4:

What 5 movies were rated the highest on average by men? Show the movies and their ratings sorted by ratings.

4.1 Approach

1. The approach is same as question 3.
2. The only difference is changing the command line argument which can be seen in the output produced.
3. The output displays only top 5 movies rated the highest on average by men.
4. Figure 4 is the output for the program.

4.2 Source Code

4.2.1 averageMovies.py

```
#!/usr/bin/env python


# program to get the highest average ratings by men and women
import sys
import operator
def main():
    # take the arguments from the command line
    numOfArgs=len(sys.argv)
    if numOfArgs<2 or numOfArgs>2:

        print'Usage: averageGender.py <F or M> '
        print'e.g. : averageGender.py F'
        sys.exit(1)
    gender           = sys.argv[1]
    # Create a dictionary
    movie_userid_avg = {}
    movie_id_ratings = {}
    movie_userid_avg = {}
    users            = {}
    count            = 1

    readData = open('/home/bbokka/cs594/A8/u.data', 'r')
    readItem = open('/home/bbokka/cs594/A8/u.item','r')
    readUser = open('/home/bbokka/cs594/A8/u.user','r')

    # Reading u.user file to get the male or female
    for line in readUser:
        split_input_line = line.strip().split('|')
        user_id_from_user= split_input_line[0]
        user_gender      = split_input_line[2]

        if(gender.upper() == user_gender):
            users[user_id_from_user] = [user_gender]

    readUser.close()

    # reading the u.data file for item and ratings.
    for line in readData:
        split_input_line  = line.strip().split('\t')
        user_id_from_data = split_input_line[0]
        movie_id_from_data=  split_input_line[1]
        movie_rating      =  float(split_input_line[2])

        if user_id_from_data in users:
            try:
                movie_id_ratings[movie_id_from_data].append(movie_rating)
            except KeyError:
                movie_id_ratings[movie_id_from_data] = list()
                movie_id_ratings[movie_id_from_data].append(movie_rating)
    readData.close()

    # Calculating the average.
    for key in movie_id_ratings:
        avg = sum(movie_id_ratings[key])/len(movie_id_ratings[key])
        movie_userid_avg[key] = [float(avg)]

    # Reading the u.item file for movie name.
    for each_line in readItem:
        split_each_line  = each_line.strip().split('|')
        movie_item_id    = split_each_line[0]
        movie_name_split = split_each_line[1].split('(1')
        movie_name       = movie_name_split[0]

        try:
            movie_userid_avg[movie_item_id].append(movie_name)
        except KeyError:
            pass
    readItem.close()
    print "*" * 60
    print "Top 5 movies rated the highest on average by women"
    print "*" * 60
    print "Movie Name\t\t\t\t","Avg Rating"
    print "-" * 60
    # sorting the movies from highest to lowest based on the average value.
    for key, value in sorted(movie_userid_avg.items(), key=lambda e: e[1], reverse=
   True):
        if (count <=20 ):
            print '{:<50}{:<0.1f} '  .format(value[1],value[0])
            count += 1

if __name__ == "__main__":
    try:
        main()
    except KeyboardInterrupt:
        sys.exit(1)
```

4.3 Input Files

4.3.1 u.data

```
userid itemid rating timestamp

196     242     3   881250949
186     302     3   891717742
22      377     1   878887116
244     51      2   880606923
166     346     1   886397596
298     474     4   884182806
115     265     2   881171488
```

4.3.2 u.item

```
movie id | movie title | release date | video release date | IMDb URL | unknown |
    Action | Adventure | Animation |Children's | Comedy | Crime | Documentary | Drama
    | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller |
     War | Western |

1|Toy Story (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)
    |0|0|0|1|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0
2|GoldenEye (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?GoldenEye%20(1995)
    |0|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0
3|Four Rooms (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Four%20Rooms
    %20(1995)|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0
4|Get Shorty (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Get%20Shorty
    %20(1995)|0|1|0|0|0|1|0|0|1|0|0|0|0|0|0|0|0|0|0
5|Copycat (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Copycat%20(1995)
    |0|0|0|0|0|0|1|0|1|0|0|0|0|0|0|0|1|0|0
6|Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)|01-Jan-1995||http://us.imdb.com
    /Title?Yao+a+yao+yao+dao+waipo+qiao+(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0
7|Twelve Monkeys (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Twelve%20Monkeys
    %20(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|1|0|0|0|0
```

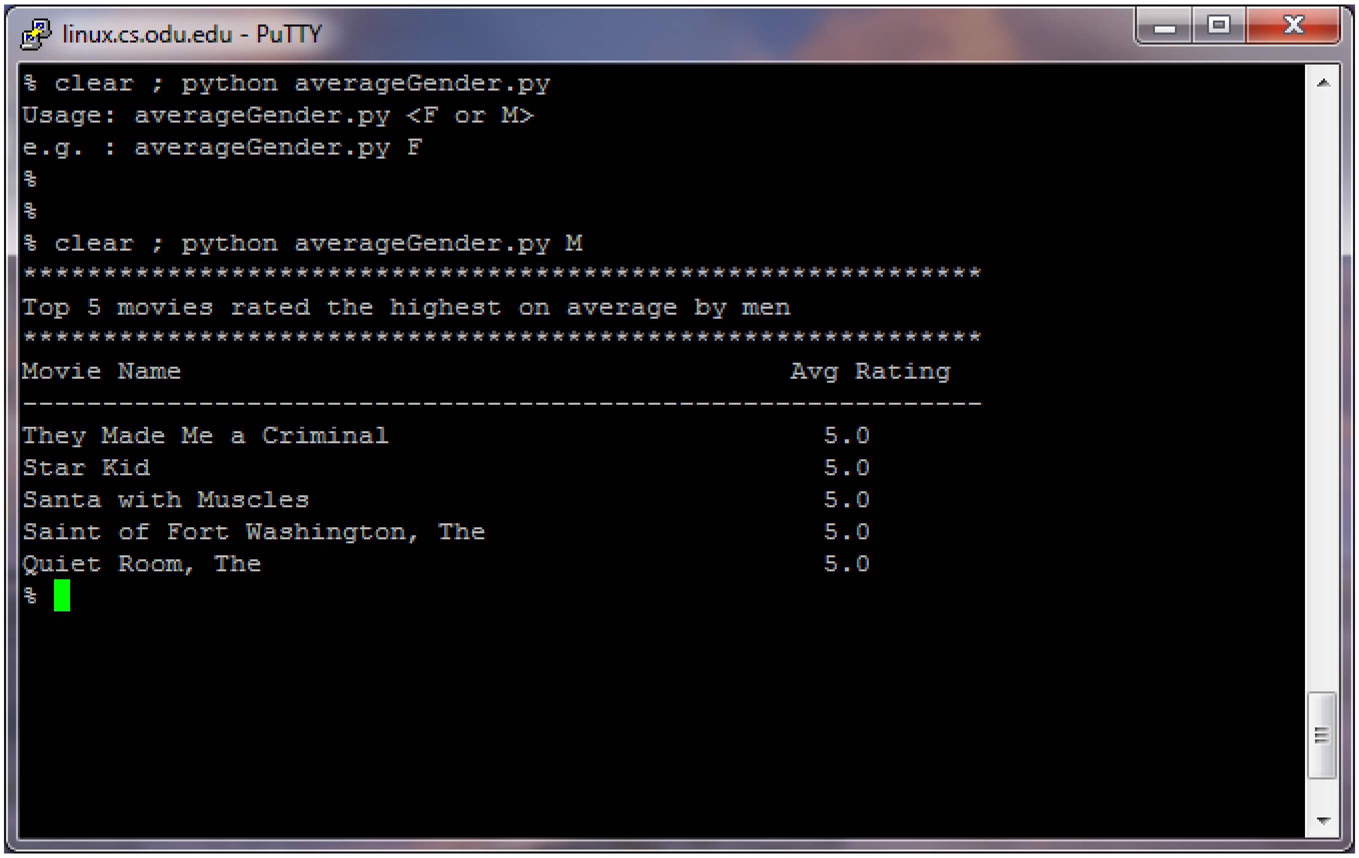
4.3.3 u.user

```
userid  age   gender   occupation    zip code

1       24     M             technician        85711
2       53     F       other             94043
3       23     M       writer            32067
4       24     M       technician        43537
5       33     F       other             15213
6       42     M       executive         98101
7       57     M       administrator     91344
```

4.4 Output Files

4.4.1 averageGenderM.png



```
% clear ; python averageGender.py
Usage: averageGender.py <F or M>
e.g. : averageGender.py F
%
%
% clear ; python averageGender.py M
*************************************************************
Top 5 movies rated the highest on average by men
*************************************************************
Movie Name                                          Avg Rating
-------------------------------------------------------------
They Made Me a Criminal                               5.0
Star Kid                                              5.0
Santa with Muscles                                    5.0
Saint of Fort Washington, The                         5.0
Quiet Room, The                                       5.0
%
```

Figure 4: Top 5 movies rated the highest on average by men.

4.4.2 averageGenderFull.txt

Showing all the movies with same average ratings.

```
% clear ; python averageGender.py
Usage: averageGender.py <F or M>
e.g. : averageGender.py F
%
%
% clear ; python averageGender.py M
***********************************************************
Top 5 movies rated the highest on average by women
***********************************************************
Movie Name                                          Avg Rating
-----------------------------------------------------------
They Made Me a Criminal                               5.0
Star Kid                                              5.0
Santa with Muscles                                    5.0
Saint of Fort Washington, The                         5.0
Quiet Room, The                                       5.0
Prefontaine                                           5.0
Marlene Dietrich: Shadow and Light                    5.0
Love Serenade                                         5.0
Little City                                           5.0
Letter From Death Row, A                              5.0
Leading Man, The                                      5.0
Hugo Pool                                             5.0
Great Day in Harlem, A                                5.0
Entertaining Angels: The Dorothy Day Story            5.0
Delta of Venus                                        5.0
Aiqing wansui                                         5.0
Two or Three Things I Know About Her                  4.7
Little Princess, The                                  4.7
Pather Panchali                                       4.6
Sliding Doors                                         4.5
```

5 Question 5:

What movie received ratings most like Top Gun? Which movie received ratings that were least like Top Gun (negative correlation)?

5.1 Approach

5.2 Approach

1. To solve question 5 we need recommendation.py.
2. To get the desired output I modified calculateSimilarItems()and loadMovieLens() functions.
3. The file ratingsLikeTopGun.txt shows the output for movies having ratings most like Top Gun.
4. The file ratingsLeastTopGun.txt shows the output for movies having ratings least like Top Gun.
5. To see more results run ratingsTopGun.py.

5.3 Source Code

5.3.1 recommendations.py

```
1  def calculateSimilarItems(prefs,n=10):
2    # Create a dictionary of items showing which other items they
3    # are most similar to.
4    result={}
5    item1 = 'Top Gun (1986)'
6    # Invert the preference matrix to be item-centric
7    itemPrefs=transformPrefs(prefs)
8    c=0
9    for item in itemPrefs:
10     # Status updates for large datasets
11     # c+=1
12     # if c%100==0: print "%d / %d" % (c,len(itemPrefs))
13     # Find the most similar items to this one
14     scores=topMatches(itemPrefs,item1,n=n,similarity=sim_pearson)
15     result[item]=scores
16   return result
17
18
19 def loadMovieLens():
20   # Get movie titles
21   readData = open('/home/bbokka/cs594/A8/u.data', 'r')
22   readItem = open('/home/bbokka/cs594/A8/u.item','r')
23   movies={}
24   for line in readItem:
25     (id,title)=line.split('|')[0:2]
26     movies[id]=title
27   # Load data
28   prefs={}
29   for line in readData:
30     (user,movieid,rating,ts)=line.split('\t')
31     prefs.setdefault(user,{})
32     prefs[user][movies[movieid]]=float(rating)
33
34   return prefs
35
36   # To get the negative corelation line 42 is commented.
37   def topMatches(prefs,person,n=5,similarity=sim_pearson):
38   scores=[(similarity(prefs,person,other),other)
39                   for other in prefs if other!=person]
40   scores.sort()
41   #scores.reverse()
42   return scores[0:n]
```

5.3.2 ratingsTopGun.py

```
#!/usr/bin/env python

import sys
import recommendations

def main():
    count = 'Top Gun (1986)'
    resultsOfloadMovieLens          = recommendations.loadMovieLens()
    resultsOfcalculateSimilarItems = recommendations.calculateSimilarItems(
   resultsOfloadMovieLens,n=80)
    print "*" * 60
    print "Movies received ratings most like or least like Top Gun"
    print "*" * 60
    print "value\t\t\t\t\t","Movie Name"
    print "-" * 60

    for key, value in sorted(resultsOfcalculateSimilarItems.items(), key=lambda e: e
    [1], reverse=True):
        variable = key
        if (count == variable):
            for value , movie in value:
                print  value , movie

if __name__ == "__main__":
    try:
        main()
    except KeyboardInterrupt:
        sys.exit(1)
```

5.4 Input Files

5.4.1 u.data

```
userid  itemid  rating  timestamp

196      242      3    881250949
186      302      3    891717742
22       377      1    878887116
244      51       2    880606923
166      346      1    886397596
298      474      4    884182806
115      265      2    881171488
```

5.4.2 u.item

```
movie id | movie title | release date | video release date | IMDb URL | unknown |
    Action | Adventure | Animation |Children's | Comedy | Crime | Documentary | Drama
    | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller |
     War | Western |

1|Toy Story (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)
    |0|0|0|1|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0
2|GoldenEye (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?GoldenEye%20(1995)
    |0|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0
3|Four Rooms (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Four%20Rooms
    %20(1995)|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0
4|Get Shorty (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Get%20Shorty
    %20(1995)|0|1|0|0|0|1|0|0|1|0|0|0|0|0|0|0|0|0|0
5|Copycat (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Copycat%20(1995)
    |0|0|0|0|0|0|1|0|1|0|0|0|0|0|0|0|1|0|0
6|Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)|01-Jan-1995||http://us.imdb.com
    /Title?Yao+a+yao+yao+dao+waipo+qiao+(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0
7|Twelve Monkeys (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Twelve%20Monkeys
    %20(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|1|0|0|0
```

5.5 Output Files

5.5.1 ratingsLikeTopGun.txt

```
% clear ; python ratingsTopGun.py
***********************************************************
Movies received ratings most like Top Gun
***********************************************************
value                                         Movie Name
-----------------------------------------------------------
1.0 Shiloh (1997)
1.0 King of the Hill (1993)
1.0 Bhaji on the Beach (1993)
1.0 Wild America (1997)
1.0 Wedding Gift, The (1994)
1.0 Underground (1995)
1.0 Two or Three Things I Know About Her (1966)
1.0 Two Bits (1995)
1.0 Total Eclipse (1995)
1.0 The Innocent (1994)
1.0 That Old Feeling (1997)
1.0 Stars Fell on Henrietta, The (1995)
1.0 Stalker (1979)
1.0 Spirits of the Dead (Tre passi nel delirio) (1968)
1.0 Show, The (1995)
1.0 Shooter, The (1995)
1.0 Selena (1997)
1.0 Schizopolis (1996)
1.0 Scarlet Letter, The (1926)
1.0 Run of the Country, The (1995)
1.0 Ponette (1996)
1.0 Perfect Candidate, A (1996)
1.0 Outlaw, The (1943)
1.0 Old Lady Who Walked in the Sea, The (Vieille qui marchait dans la mer, La) (1991)
1.0 Nothing to Lose (1994)
1.0 New Jersey Drive (1995)
1.0 Mr. Jones (1993)
1.0 Metisse (Caf  au Lait) (1993)
1.0 Maybe, Maybe Not (Bewegte Mann, Der) (1994)
1.0 Manny & Lo (1996)
1.0 Man of the Year (1995)
1.0 Love Serenade (1996)
1.0 Last Time I Saw Paris, The (1954)
1.0 Killer (Bulletproof Heart) (1994)
1.0 Jerky Boys, The (1994)
1.0 I Like It Like That (1994)
1.0 Horse Whisperer, The (1998)
1.0 Hear My Song (1991)
1.0 Grosse Fatigue (1994)
1.0 Gone Fishin' (1997)
1.0 Glass Shield, The (1994)
1.0 Germinal (1993)
1.0 Gabbeh (1996)
1.0 Four Days in September (1997)
1.0 Flower of My Secret, The (Flor de mi secreto, La) (1995)
1.0 Fausto (1993)
1.0 Even Cowgirls Get the Blues (1993)
```

```
1.0 Enfer, L' (1994)
1.0 Dream With the Fishes (1997)
1.0 Dream Man (1995)
1.0 Dangerous Ground (1997)
1.0 Collectionneuse, La (1967)
1.0 Clean Slate (Coup de Torchon) (1981)
1.0 Calendar Girl (1993)
1.0 Blood For Dracula (Andy Warhol's Dracula) (1974)
1.0 Bliss (1997)
1.0 Best Men (1997)
1.0 American Dream (1990)
1.0 Albino Alligator (1996)
1.0 8 Seconds (1994)
1.0 Aparajito (1956)
0.995870594886 Scarlet Letter, The (1995)
0.986440050416 Kundun (1997)
0.981980506062 Traveller (1997)
```

5.5.2 ratingsLeastTopGun.txt

```
% clear ; python ratingsTopGun.py
**********************************************************
Movies received ratings most Least like Top Gun
**********************************************************
value                                         Movie Name
----------------------------------------------------------
-1.0 Babysitter, The (1995)
-1.0 Telling Lies in America (1997)
-1.0 Bad Moon (1996)
-1.0 Beat the Devil (1954)
-1.0 Bewegte Mann, Der (1994)
-1.0 Bitter Sugar (Azucar Amargo) (1996)
-1.0 Broken English (1996)
-1.0 Caro Diario (Dear Diary) (1994)
-1.0 Carpool (1996)
-1.0 Carried Away (1996)
-1.0 Everest (1998)
-1.0 Frisk (1995)
-1.0 Heidi Fleiss: Hollywood Madam (1995)
-1.0 Joy Luck Club, The (1993)
-1.0 Lamerica (1994)
-1.0 Loch Ness (1995)
-1.0 Love and Death on Long Island (1997)
-1.0 Lover's Knot (1996)
-1.0 Meet Wally Sparks (1997)
-1.0 Midnight Dancers (Sibak) (1994)
-1.0 Naked in New York (1994)
-1.0 Nico Icon (1995)
-1.0 Nil By Mouth (1997)
-1.0 Romper Stomper (1992)
-1.0 Roseanna's Grave (For Roseanna) (1997)
-1.0 Safe Passage (1994)
-1.0 Switchback (1997)
-1.0 Tetsuo II: Body Hammer (1992)
-1.0 Two Much (1996)
-1.0 World of Apu, The (Apur Sansar) (1959)
-1.0 Year of the Horse (1997)
-0.946729262406 Alphaville (1965)
-0.944911182523 Denise Calls Up (1995)
-0.944911182523 Penny Serenade (1941)
-0.894427191 Awfully Big Adventure, An (1995)
-0.888431093553 Paradise Lost: The Child Murders at Robin Hood Hills (1996)
-0.880704845928 In the Realm of the Senses (Ai no corrida) (1976)
-0.866025403784 Chasers (1994)
-0.866025403784 Faces (1968)
-0.866025403784 Big Squeeze, The (1996)
-0.866025403784 Female Perversions (1996)
-0.816496580928 Theodore Rex (1995)
-0.777777777778 Half Baked (1998)
-0.777777777778 Wild Reeds (1994)
-0.770317441193 Cemetery Man (Dellamorte Dellamore) (1994)
-0.760885910253 Gang Related (1997)
-0.753266379297 Young Poisoner's Handbook, The (1995)
-0.751809411556 Ruby in Paradise (1993)
-0.734031784566 Funeral, The (1996)
```

```
-0.73029674334 Mark of Zorro, The (1940)
-0.73029674334 One Night Stand (1997)
-0.707106781187 Angel and the Badman (1947)
-0.688247201612 Switchblade Sisters (1975)
-0.683302297676 Apostle, The (1997)
```

6 Question 6:

Which 5 raters rated the most films? Show the raters' IDs and the number of films each rated.

6.1 Approach

1. The approach for this question is modify question 1 to read U.data file, get user id and for the corresponding user id store all the ratings .
2. Now, we count the number of ratings for each user by using the len() function.
3. The output displays only top 5 users and their number of ratings.
4. Figure 5 is the output for the program.

6.2 Source Code

6.2.1 ratersMostRatedMovies.py

```python
#!/usr/bin/env python

import sys

def main():
    # Create a dictionary
     movie_userid_ratings = {}
     movie_userid_len     = {}
     count                = 1
    # Read the input files.
    readData = open('/home/bbokka/cs594/A8/u.data', 'r')
     # reading the u.data file for item and ratings.
    for line in readData:
        split_input_line  = line.strip().split('\t')
        user_id_from_data = split_input_line[0]
        movie_rating      =  float(split_input_line[2])

        try:
            movie_userid_ratings[user_id_from_data].append(movie_rating)
        except KeyError:
            movie_userid_ratings[user_id_from_data] = list()
            movie_userid_ratings[user_id_from_data].append(movie_rating)
    readData.close()
    # Calculating the number of movies each user rated.
    for key in movie_userid_ratings:
        length = len(movie_userid_ratings[key])
        movie_userid_len[key] = int(length)


    print '*' * 60
    print "Top 5 raters rated movies."
    print "*" * 60
    print "User id\t\t","Number of movies rated"
    print "-" * 60
    # sorting the movies from highest to lowest based on the number of
    # ratings for each movie.
    for key, value in sorted(movie_userid_len.items(), key=lambda e: e[1], reverse=
    True):
        if (count <= 5):
            print '{:<20}{} '  .format(key,value)
            count += 1

if __name__ == "__main__":
    try:
        main()
    except KeyboardInterrupt:
        sys.exit(1)
```

6.3 Input Files

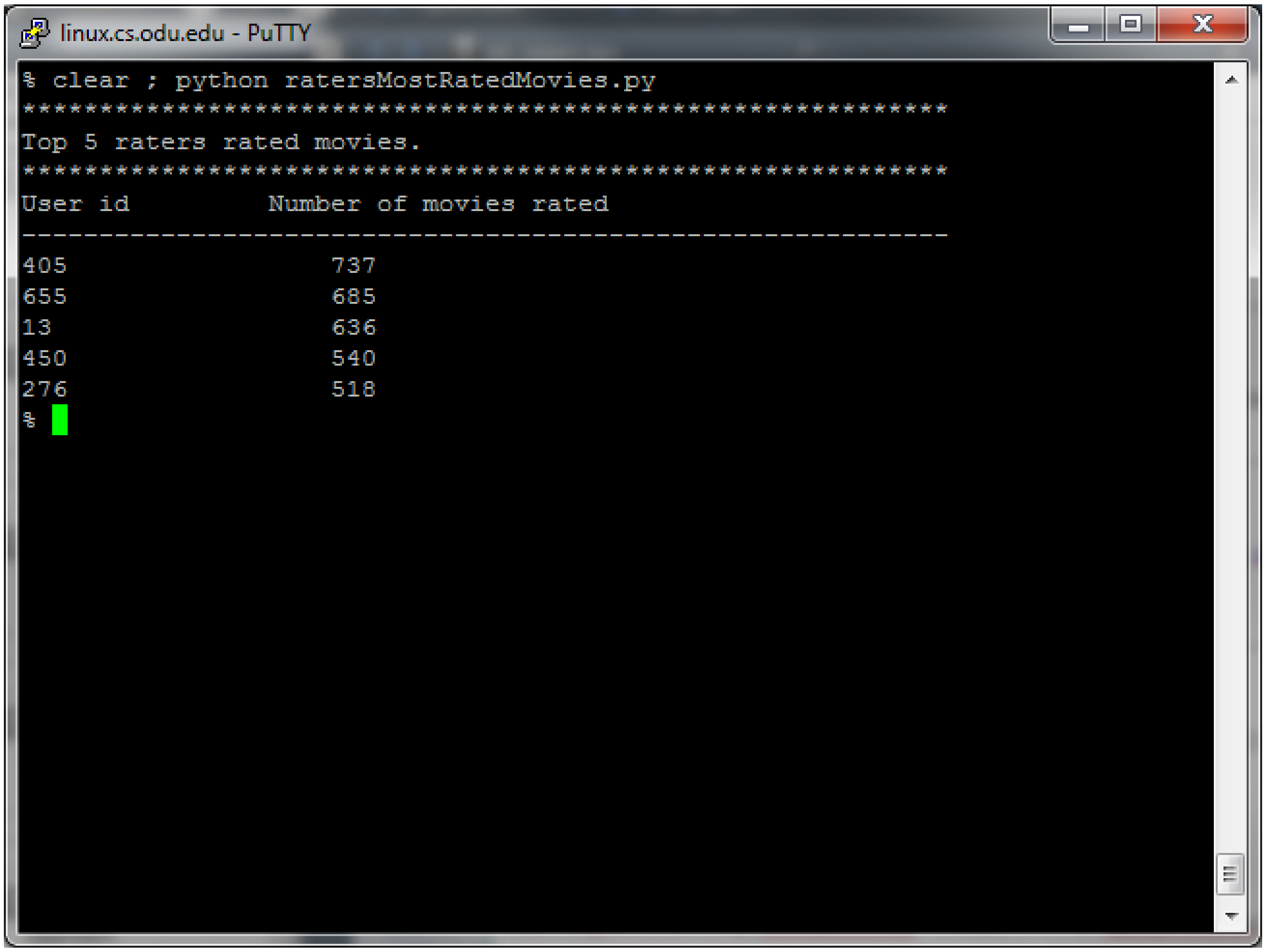
6.3.1 u.data

```
userid  itemid  rating  timestamp

196      242      3    881250949
186      302      3    891717742
22       377      1    878887116
244      51       2    880606923
166      346      1    886397596
298      474      4    884182806
115      265      2    881171488
```

6.4 Output Files

6.4.1 ratersMostRatedMovies.png



linux.cs.odu.edu - PuTTY

```
% clear ; python ratersMostRatedMovies.py
************************************************************
Top 5 raters rated movies.
************************************************************
User id          Number of movies rated
------------------------------------------------------------
405                     737
655                     685
13                      636
450                     540
276                     518
%
```

Figure 5: Top 5 raters rated movies.

7 Question 7:

Which 5 raters most agreed with each other? Show the raters' IDs and Pearson's r, sorted by r.

7.1 Approach

I couldn't complete this question.

7.2 Description of

7.3 Source Code

7.4 Output Files

8 Question 8:

Which 5 raters most disagreed with each other (negative correlation)? Show the raters' IDs and Pearson's r, sorted by r.

8.1 Approach

I couldn't complete this question.

8.2 Description of

8.3 Source Code

8.4 Output Files

9 Question 9:

What movie was rated highest on average by men over 40? By men under 40?

9.1 Approach

1. To solve this question, the straight forward approach is to read u.user file, get the user id, gender, age of each user. Read u.data file, get movie id, corresponding movie ratings. Read u.item file to get movie id and corresponding movie name.
2. Program averageOverUnderGender.py is designed in such a way that it can get the highest averages by men and women over and under 40 just by changing the arguments to F or M and "over" or "under" while executing the program.
3. From u.user file extract the user id, gender, age, based on arguments given from command line either men over 40 or men under 40 or women over 40 or women under 40 is stored into dictionary.
4. Now, Extract the movie id and ratings. For each user id build a dictionary that has all the ratings for the respective gender with over or under condition.
5. Now, for each movie id calculate the average a using sum() and len() functions.
6. For each movie id get the movie name and append it to the dictionary which has average.
7. Now, sort the dictionary and get the movies which are rated highest on average by men or women over 40 or under 40.
8. Figure 6 is the output showing how to execute the program and displays the top 5 movies which are rated highest on average by men "over" 40.
9. Figure 7 is the output showing how to execute the program and displays the top 5 movies which are rated highest on average by men "under" 40.
10. The file averageOverM.txt and averageUnderM.txt displays movies which have the same average ratings.

9.2 Source Code

9.2.1 averageOverUnderGender.py

```python
#!/usr/bin/env python


# program to get the highest average ratings by men under and over 40.
#
import sys
import operator
def main():
    # Take the arguments from the command line
    numOfArgs=len(sys.argv)
    if numOfArgs<3 or numOfArgs>3:

        print'Usage: averageOverUnderGender.py <F or M> <under or over>'
        print'e.g. : averageOverUnderGender.py F over'
        sys.exit(1)
    gender           = sys.argv[1]
    gender_under_over= sys.argv[2]

    # Create a dictionary
    movie_userid_avg = {}
    movie_id_ratings = {}
    movie_userid_avg = {}
    users            = {}
    count            = 1

    # Read the files
    readData = open('/home/bbokka/cs594/A8/u.data', 'r')
    readItem = open('/home/bbokka/cs594/A8/u.item','r')
    readUser = open('/home/bbokka/cs594/A8/u.user','r')

    # Reading u.user file to get the male or female
    for line in readUser:
        split_input_line = line.strip().split('|')
        user_id_from_user= split_input_line[0]
        user_age         = int(split_input_line[1])
        user_gender      = split_input_line[2]

        if gender_under_over == 'over' :
            if(gender.upper() == user_gender and user_age > 40):
                users[user_id_from_user] = [user_age]

        if gender_under_over == 'under' :
            if(gender.upper() == user_gender and user_age < 40):
                users[user_id_from_user] = [user_age]

    readUser.close()

    # reading the u.data file for item and ratings.
    for line in readData:
        split_input_line  = line.strip().split('\t')
        user_id_from_data = split_input_line[0]
        movie_id_from_data=  split_input_line[1]
        movie_rating      =  float(split_input_line[2])

        if user_id_from_data in users:
            try:
                movie_id_ratings[movie_id_from_data].append(movie_rating)
            except KeyError:
                movie_id_ratings[movie_id_from_data] = list()
                movie_id_ratings[movie_id_from_data].append(movie_rating)
    readData.close()

    # Calculating the average.
    for key in movie_id_ratings:
        avg = sum(movie_id_ratings[key])/len(movie_id_ratings[key])
        movie_userid_avg[key] = [float(avg)]

    # Reading the u.item file for movie name.
    for each_line in readItem:
        split_each_line  = each_line.strip().split('|')

        movie_item_id    = split_each_line[0]
        movie_name_split = split_each_line[1].split('(1')
        movie_name       = movie_name_split[0]

        try:
            movie_userid_avg[movie_item_id].append(movie_name)
        except KeyError:
            pass
    readItem.close()
    print '*' * 55
    print "Movies rated highest on average by men"
    print "*" * 55
    print "Movie Name\t\t\t\t\t","Avg Rating"
    print "-" * 55
    # sorting the movies from highest to lowest based on the average value.
    for key, value in sorted(movie_userid_avg.items(), key=lambda e: e[1], reverse=
   True):
        if (count <=30 ):
            print '{:<50}{:<0.1f} '  .format(value[1],value[0])
            count += 1

if __name__ == "__main__":
    try:
        main()
    except KeyboardInterrupt:
        sys.exit(1)
```

9.3 Input Files

9.3.1 u.data

```
userid  itemid  rating  timestamp

196      242      3    881250949
186      302      3    891717742
22       377      1    878887116
244      51       2    880606923
166      346      1    886397596
298      474      4    884182806
115      265      2    881171488
```

9.3.2 u.item

```
movie id | movie title | release date | video release date | IMDb URL | unknown |
    Action | Adventure | Animation |Children's | Comedy | Crime | Documentary | Drama
    | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller |
     War | Western |

1|Toy Story (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)
    |0|0|0|1|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0
2|GoldenEye (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?GoldenEye%20(1995)
    |0|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0
3|Four Rooms (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Four%20Rooms
    %20(1995)|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0
4|Get Shorty (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Get%20Shorty
    %20(1995)|0|1|0|0|0|1|0|0|1|0|0|0|0|0|0|0|0|0|0
5|Copycat (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Copycat%20(1995)
    |0|0|0|0|0|0|1|0|1|0|0|0|0|0|0|0|1|0|0
6|Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)|01-Jan-1995||http://us.imdb.com
    /Title?Yao+a+yao+yao+dao+waipo+qiao+(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0
7|Twelve Monkeys (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Twelve%20Monkeys
    %20(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|1|0|0|0
```

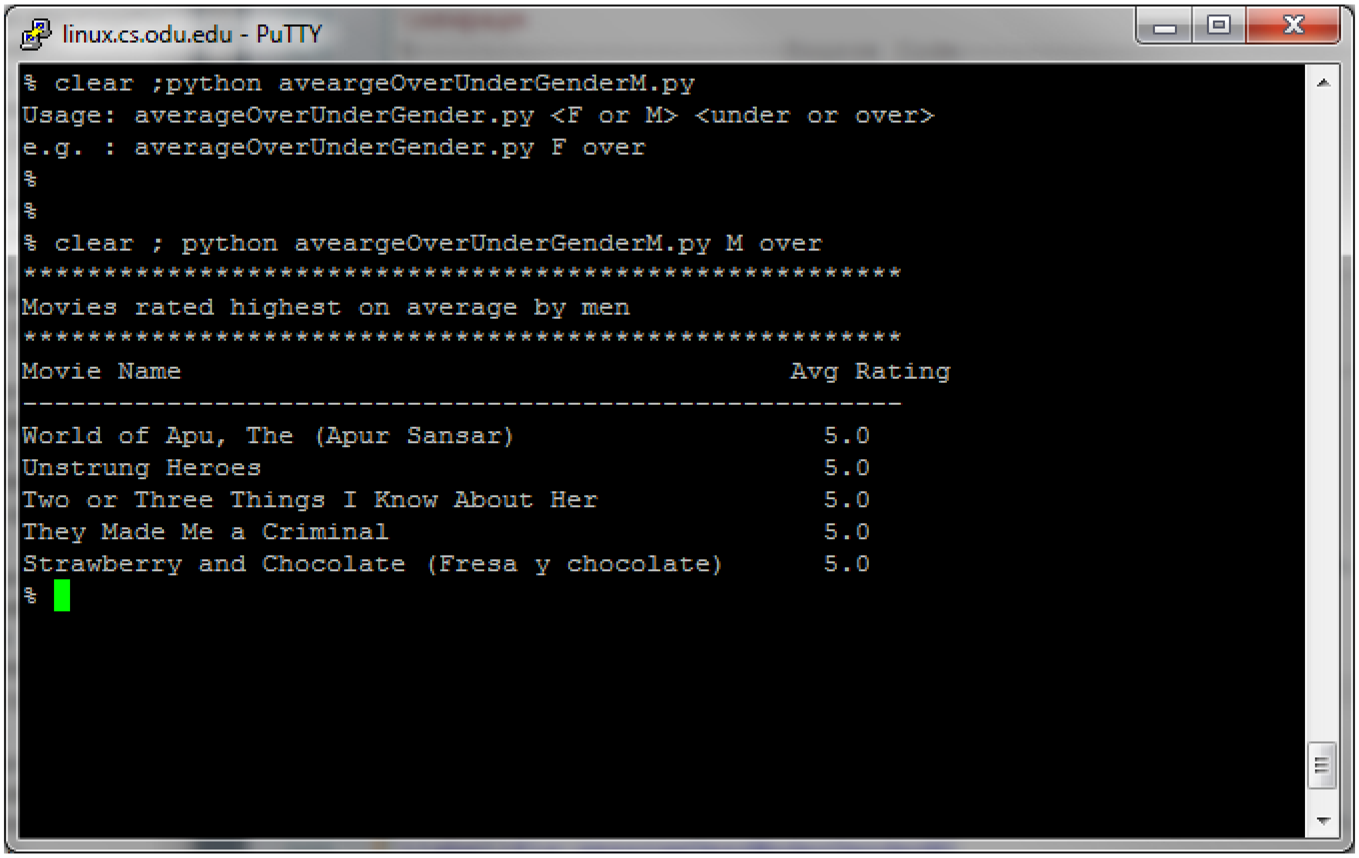
9.3.3 u.user

```
userid   age     gender     occupation     zip code

1       24         M             technician          85711
2       53         F      other                94043
3       23         M      writer               32067
4       24         M      technician           43537
5       33         F      other                15213
6       42         M      executive            98101
7       57         M      administrator        91344
```

9.4 Output Files

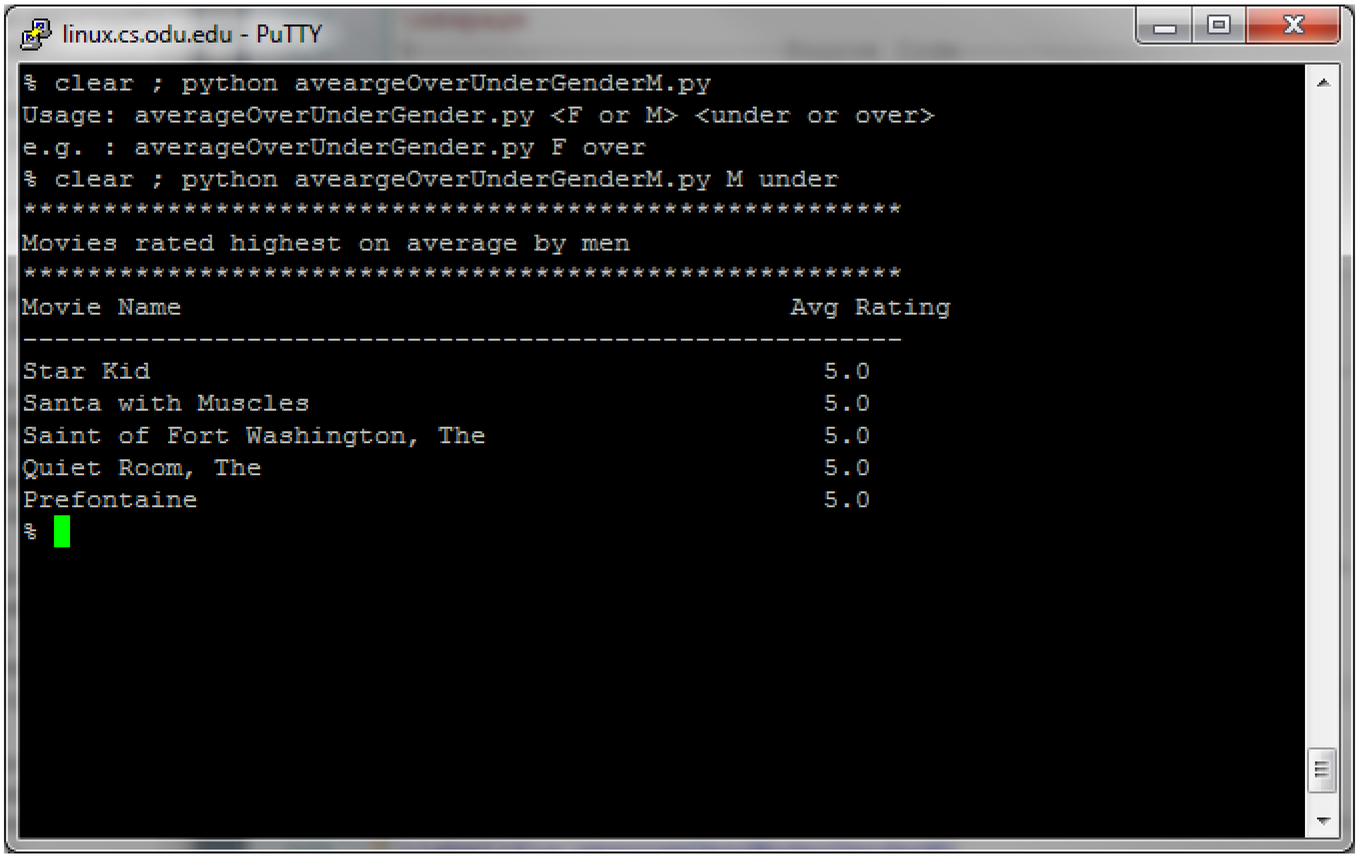
9.4.1 averageOverM.png



```
% clear ;python aveargeOverUnderGenderM.py
Usage: averageOverUnderGender.py <F or M> <under or over>
e.g. : averageOverUnderGender.py F over
%
%
% clear ; python aveargeOverUnderGenderM.py M over
*******************************************************
Movies rated highest on average by men
*******************************************************
Movie Name                                            Avg Rating
-------------------------------------------------------
World of Apu, The (Apur Sansar)                         5.0
Unstrung Heroes                                         5.0
Two or Three Things I Know About Her                    5.0
They Made Me a Criminal                                 5.0
Strawberry and Chocolate (Fresa y chocolate)            5.0
%
```

Figure 6: Movies rated highest on average by men over 40.

9.4.2 averageUnderM.png



linux.cs.odu.edu - PuTTY

```
% clear ; python aveargeOverUnderGenderM.py
Usage: averageOverUnderGender.py <F or M> <under or over>
e.g. : averageOverUnderGender.py F over
% clear ; python aveargeOverUnderGenderM.py M under
*******************************************************
Movies rated highest on average by men
*******************************************************
Movie Name                                              Avg Rating
-------------------------------------------------------
Star Kid                                                  5.0
Santa with Muscles                                        5.0
Saint of Fort Washington, The                             5.0
Quiet Room, The                                           5.0
Prefontaine                                               5.0
%
```

Figure 7: Movies rated highest on average by men under 40.

9.4.3 averageOverM.txt

```
% clear ; python averageOverUnderGender.py
Usage: averageOverUnderGender.py <F or M> <under or over>
e.g. : averageOverUnderGender.py F over
%
%
% clear ; python averageOverUnderGender.py M over
******************************************************
Movies rated highest on average by men
******************************************************
Movie Name                                          Avg Rating
------------------------------------------------------
World of Apu, The (Apur Sansar)                        5.0
Unstrung Heroes                                        5.0
Two or Three Things I Know About Her                   5.0
They Made Me a Criminal                                5.0
Strawberry and Chocolate (Fresa y chocolate)           5.0
Star Kid                                               5.0
Spice World                                            5.0
Solo                                                   5.0
Rendezvous in Paris (Rendez-vous de Paris, Les)        5.0
Prefontaine                                            5.0
Poison Ivy II                                          5.0
Marlene Dietrich: Shadow and Light                     5.0
Little Princess, The                                   5.0
Little City                                            5.0
Leading Man, The                                       5.0
Late Bloomers                                          5.0
Indian Summer                                          5.0
Hearts and Minds                                       5.0
Great Day in Harlem, A                                 5.0
Grateful Dead                                          5.0
Faithful                                               5.0
Double Happiness                                       5.0
Boxing Helena                                          5.0
Aparajito                                              5.0
Ace Ventura: When Nature Calls                         5.0
Pather Panchali                                        4.8
Whole Wide World, The                                  4.7
A Chef in Love                                         4.7
Close Shave, A                                         4.6
Shanghai Triad (Yao a yao yao dao waipo qiao)          4.6
```

9.4.4 averageUnderM.txt

```
% clear ; python averageOverUnderGender.py
Usage: averageOverUnderGender.py <F or M> <under or over>
e.g. : averageOverUnderGender.py F over
%
%
% clear ; python averageOverUnderGender.py M under
*****************************************************
Movies rated highest on average by men
*****************************************************
Movie Name                                          Avg Rating
-----------------------------------------------------
Star Kid                                              5.0
Santa with Muscles                                    5.0
Saint of Fort Washington, The                         5.0
Quiet Room, The                                       5.0
Prefontaine                                           5.0
Perfect Candidate, A                                  5.0
Maya Lin: A Strong Clear Vision                       5.0
Magic Hour, The                                       5.0
Love in the Afternoon                                 5.0
Love Serenade                                         5.0
Letter From Death Row, A                              5.0
Leading Man, The                                      5.0
Hugo Pool                                             5.0
Entertaining Angels: The Dorothy Day Story            5.0
Delta of Venus                                        5.0
Crossfire                                             5.0
Angel Baby                                            5.0
Aiqing wansui                                         5.0
Winter Guest, The                                     4.5
Two or Three Things I Know About Her                  4.5
Sum of Us, The                                        4.5
Sliding Doors                                         4.5
Man of No Importance, A                               4.5
Little Princess, The                                  4.5
Innocents, The                                        4.5
Grosse Fatigue                                        4.5
Fille seule, La (A Single Girl)                       4.5
Boy's Life 2                                          4.5
Anna                                                  4.5
Wallace & Gromit: The Best of Aardman Animation       4.5
```

10 Question 10:

What movie was rated highest on average by women over 40? By women under 40?

10.1 Approach

1. The approach is same as question 9.
2. To get the corresponding output for this question it will still use averageOverUnderGender.py but the only difference is changing the arguments.
3. Figure 8 is the output showing how to execute the program and displays the top 5 movies which are rated highest on average by men “over” 40.
4. Figure 9 is the output showing how to execute the program and displays the top 5 movies which are rated highest on average by men “under” 40.
5. The file averageOverF.txt and averageUnderF.txt displays movies which have the same average ratings.

10.2 Source Code

10.2.1 averageOverUnderGender.py

```
#!/usr/bin/env python


# program to get the highest average ratings by women under and over 40.
import sys
import operator
def main():
    # Take the arguments from the command line
    numOfArgs=len(sys.argv)
    if numOfArgs<3 or numOfArgs>3:

        print'Usage: averageOverUnderGender.py <F or M> <under or over>'
        print'e.g. : averageOverUnderGender.py F over'
        sys.exit(1)
    gender           = sys.argv[1]
    gender_under_over= sys.argv[2]

    # Create a dictionary
    movie_userid_avg = {}
    movie_id_ratings = {}
    movie_userid_avg = {}
    users            = {}
    count            = 1

    # Read the files
    readData = open('/home/bbokka/cs594/A8/u.data', 'r')
    readItem = open('/home/bbokka/cs594/A8/u.item','r')
    readUser = open('/home/bbokka/cs594/A8/u.user','r')

    # Reading u.user file to get the male or female
    for line in readUser:
        split_input_line = line.strip().split('|')
        user_id_from_user= split_input_line[0]
        user_age         = int(split_input_line[1])
        user_gender      = split_input_line[2]

        if gender_under_over == 'over' :
            if(gender.upper() == user_gender and user_age > 40):
                users[user_id_from_user] = [user_age]

        if gender_under_over == 'under' :
            if(gender.upper() == user_gender and user_age < 40):
                users[user_id_from_user] = [user_age]

    readUser.close()

    # reading the u.data file for item and ratings.
    for line in readData:
        split_input_line  = line.strip().split('\t')
        user_id_from_data = split_input_line[0]
        movie_id_from_data=  split_input_line[1]
        movie_rating      =  float(split_input_line[2])

```

```python
        if user_id_from_data in users:
            try:
                movie_id_ratings[movie_id_from_data].append(movie_rating)
            except KeyError:
                movie_id_ratings[movie_id_from_data] = list()
                movie_id_ratings[movie_id_from_data].append(movie_rating)
    readData.close()

    # Calculating the average.
    for key in movie_id_ratings:
        avg = sum(movie_id_ratings[key])/len(movie_id_ratings[key])
        movie_userid_avg[key] = [float(avg)]

    # Reading the u.item file for movie name.
    for each_line in readItem:
        split_each_line  = each_line.strip().split('|')

        movie_item_id    = split_each_line[0]
        movie_name_split = split_each_line[1].split('(1')
        movie_name       = movie_name_split[0]

        try:
            movie_userid_avg[movie_item_id].append(movie_name)
        except KeyError:
            pass
    readItem.close()
    print '*' * 55
    print "Movies rated highest on average by women"
    print "*" * 55
    print "Movie Name\t\t\t\t\t","Avg Rating"
    print "-" * 55
    # sorting the movies from highest to lowest based on the average value.
    for key, value in sorted(movie_userid_avg.items(), key=lambda e: e[1], reverse=True):
        if (count <=5 ):
            print '{:<50}{:<0.1f} '  .format(value[1],value[0])
            count += 1

if __name__ == "__main__":
    try:
        main()
    except KeyboardInterrupt:
        sys.exit(1)
```

10.3 Input Files

10.3.1 u.data

```
userid  itemid  rating  timestamp

196      242      3    881250949
186      302      3    891717742
22       377      1    878887116
244      51       2    880606923
166      346      1    886397596
298      474      4    884182806
115      265      2    881171488
```

10.3.2 u.item

```
movie id | movie title | release date | video release date | IMDb URL | unknown |
    Action | Adventure | Animation |Children's | Comedy | Crime | Documentary | Drama
    | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller |
     War | Western |

1|Toy Story (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)
    |0|0|0|1|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0
2|GoldenEye (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?GoldenEye%20(1995)
    |0|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0
3|Four Rooms (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Four%20Rooms
    %20(1995)|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0
4|Get Shorty (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Get%20Shorty
    %20(1995)|0|1|0|0|0|1|0|0|1|0|0|0|0|0|0|0|0|0|0
5|Copycat (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Copycat%20(1995)
    |0|0|0|0|0|0|1|0|1|0|0|0|0|0|0|0|1|0|0
6|Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)|01-Jan-1995||http://us.imdb.com
    /Title?Yao+a+yao+yao+dao+waipo+qiao+(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0
7|Twelve Monkeys (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Twelve%20Monkeys
    %20(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|1|0|0|0
```

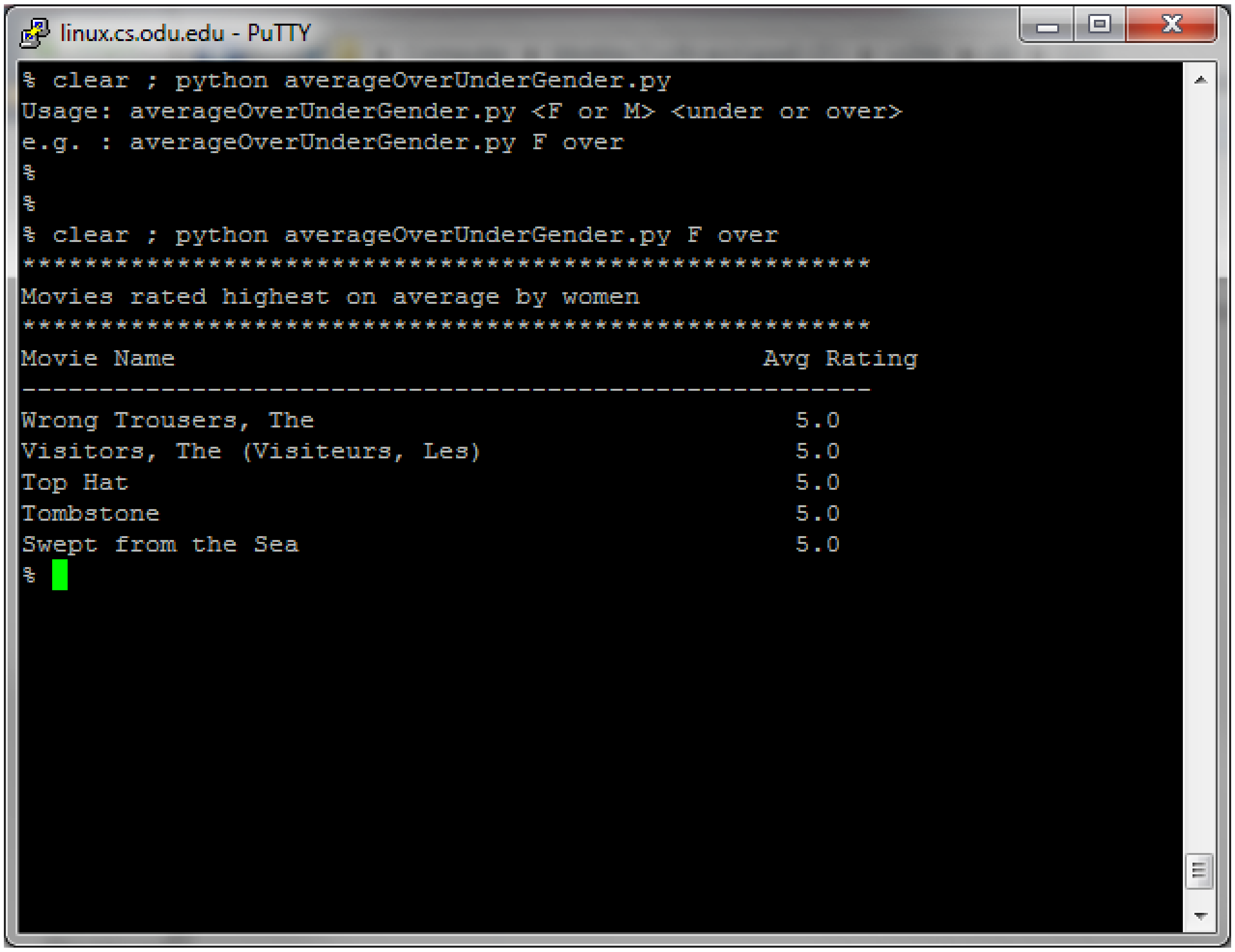
10.3.3 u.user

```
userid   age    gender    occupation    zip code

1       24        M             technician        85711
2       53        F       other             94043
3       23        M       writer            32067
4       24        M       technician        43537
5       33        F       other             15213
6       42        M       executive         98101
7       57        M       administrator     91344
```

10.4 Output Files

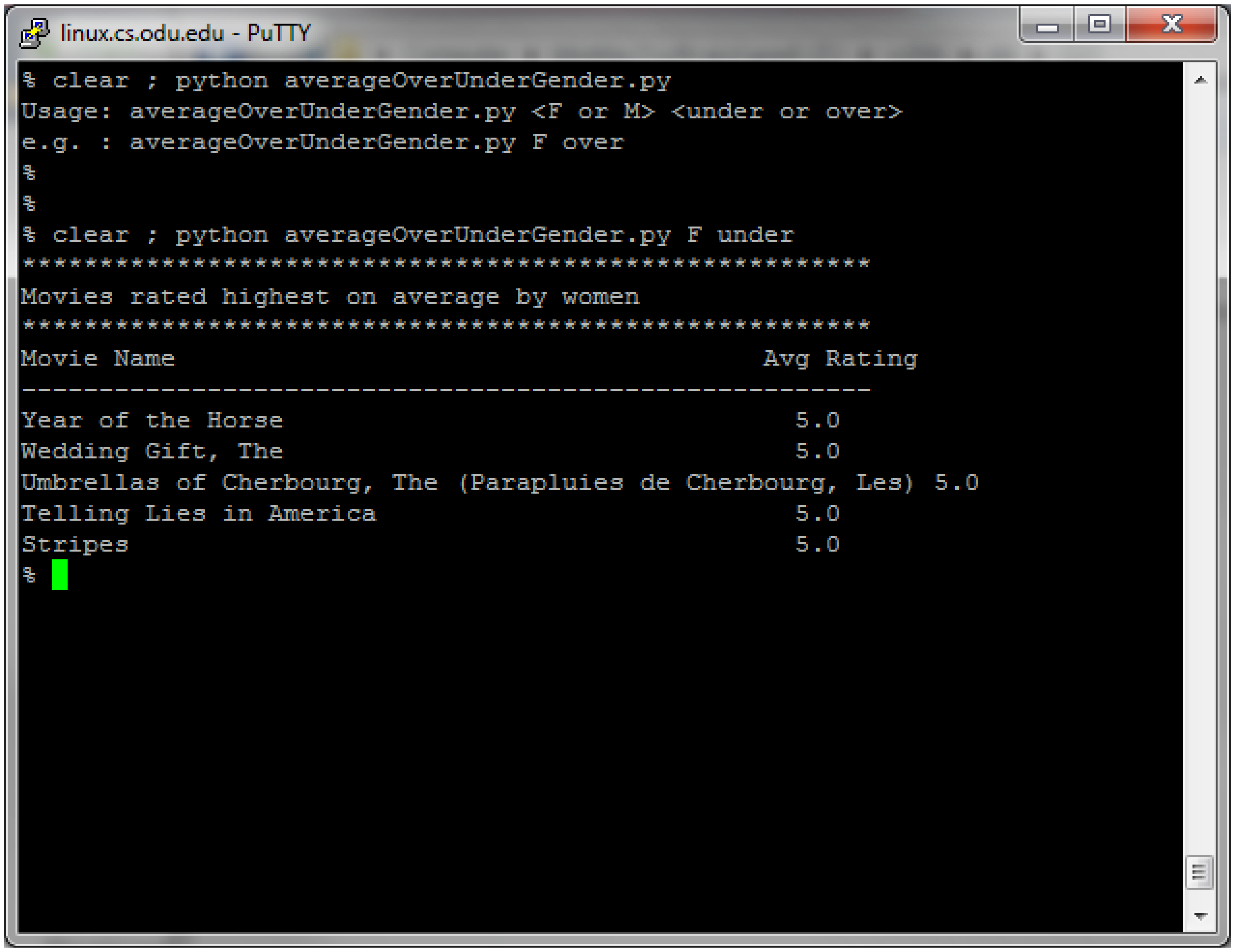
10.4.1 averageOverF.png



```
% clear ; python averageOverUnderGender.py
Usage: averageOverUnderGender.py <F or M> <under or over>
e.g. : averageOverUnderGender.py F over
%
%
% clear ; python averageOverUnderGender.py F over
*******************************************************
Movies rated highest on average by women
*******************************************************
Movie Name                                      Avg Rating
-------------------------------------------------------
Wrong Trousers, The                               5.0
Visitors, The (Visiteurs, Les)                    5.0
Top Hat                                           5.0
Tombstone                                         5.0
Swept from the Sea                                5.0
%
```

Figure 8: Movies rated highest on average by women over 40.

10.4.2 averageUnderF.png



linux.cs.odu.edu - PuTTY

```
% clear ; python averageOverUnderGender.py
Usage: averageOverUnderGender.py <F or M> <under or over>
e.g. : averageOverUnderGender.py F over
%
%
% clear ; python averageOverUnderGender.py F under
*******************************************************
Movies rated highest on average by women
*******************************************************
Movie Name                                      Avg Rating
-------------------------------------------------------
Year of the Horse                                 5.0
Wedding Gift, The                                 5.0
Umbrellas of Cherbourg, The (Parapluies de Cherbourg, Les) 5.0
Telling Lies in America                           5.0
Stripes                                           5.0
%
```

Figure 9: Movies rated highest on average by women under 40.

10.4.3 averageOverF.txt

```
% clear ; python averageOverUnderGender.py
Usage: averageOverUnderGender.py <F or M> <under or over>
e.g. : averageOverUnderGender.py F over
%
%
% clear ; python averageOverUnderGender.py F over
*****************************************************
Movies rated highest on average by women
*****************************************************
Movie Name                                          Avg Rating
-----------------------------------------------------------
Wrong Trousers, The                                    5.0
Visitors, The (Visiteurs, Les)                         5.0
Top Hat                                                5.0
Tombstone                                              5.0
Swept from the Sea                                     5.0
Shallow Grave                                          5.0
Shall We Dance?                                        5.0
Safe                                                   5.0
Quest, The                                             5.0
Pocahontas                                             5.0
Nightmare Before Christmas, The                        5.0
Mina Tannenbaum                                        5.0
Mary Shelley's Frankenstein                            5.0
Ma vie en rose (My Life in Pink)                       5.0
Letter From Death Row, A                               5.0
In the Bleak Midwinter                                 5.0
Great Dictator, The                                    5.0
Grand Day Out, A                                       5.0
Gold Diggers: The Secret of Bear Mountain              5.0
Funny Face                                             5.0
Foreign Correspondent                                  5.0
Bride of Frankenstein                                  5.0
Best Men                                               5.0
Band Wagon, The                                        5.0
Balto                                                  5.0
Angel Baby                                             5.0
Once Were Warriors                                     4.8
Fantasia                                               4.7
Last of the Mohicans, The                              4.7
Christmas Carol, A                                     4.6
```

10.4.4 averageUnderF.txt

```
% clear ; python averageOverUnderGender.py
Usage: averageOverUnderGender.py <F or M> <under or over>
e.g. : averageOverUnderGender.py F over
%
%
% clear ; python averageOverUnderGender.py F under
**************************************************
Movies rated highest on average by women
**************************************************
Movie Name                                          Avg Rating
--------------------------------------------------------------
Year of the Horse                                      5.0
Wedding Gift, The                                      5.0
Umbrellas of Cherbourg, The (Parapluies de Cherbourg, Les) 5.0
Telling Lies in America                                5.0
Stripes                                                5.0
Someone Else's America                                 5.0
Prefontaine                                            5.0
Nico Icon                                              5.0
Mina Tannenbaum                                        5.0
Maya Lin: A Strong Clear Vision                        5.0
Horseman on the Roof, The (Hussard sur le toit, Le) 5.0
Heaven's Prisoners                                     5.0
Grace of My Heart                                      5.0
Faster Pussycat! Kill! Kill!                           5.0
Everest                                                5.0
Don't Be a Menace to South Central While Drinking Your Juice in the Hood 5.0
Backbeat                                               5.0
Wallace & Gromit: The Best of Aardman Animation        4.8
Paradise Lost: The Child Murders at Robin Hood Hills 4.8
Anne Frank Remembered                                  4.8
Shawshank Redemption, The                              4.7
Shall We Dance?                                        4.7
Close Shave, A                                         4.7
Schindler's List                                       4.7
Margaret's Museum                                      4.7
Women, The                                             4.5
To Catch a Thief                                       4.5
Thin Man, The                                          4.5
Some Folks Call It a Sling Blade                       4.5
Primary Colors                                         4.5
```

References

[1] Data files used for assignment. http://files.grouplens.org/datasets/movielens/ml-100k/, April 1998. Data collected by GroupLens Research Project.

[2] K. Arthur Endsley. recommendations.py program. https://github.com/arthur-e/Programming-Collective-Intelligence/blob/master/chapter2/recommendations.py, December 2012.

[3] Frederick Hamidi. Python: Run function from the command line. http://stackoverflow.com/questions/3987041/python-run-function-from-the-command-line, October 2010.

[4] Devin Jeanpierre. Sort a python dictionary by value. http://stackoverflow.com/questions/613183/sort-a-python-dictionary-by-value, March 2009.

[5] Toby Segaran. *Programming Collective Intelligence*. O'Reilly Media, August 2007.

[]