Hadoop Map Reduce

Part Two: Movie Similarities

The objective of this part is to use a large corpus of movie data and provide recommendations of similar movies based on ratings by using statistical correlation and cosine similarity. Movies.csv and ratings.csv files are downloaded for this purpose.

Movies.csv file contains “Movie ID”, “Movie Title”, “Genre” The ratings.csv file contains “User ID”, “Movie ID”, “Rating”, “Timestamp”.

Following is the source code for all movie computations

I have used 1 mapper and 4 reducers in the source code

Description of each step written below with # symbol prior to the reducer and mapper.

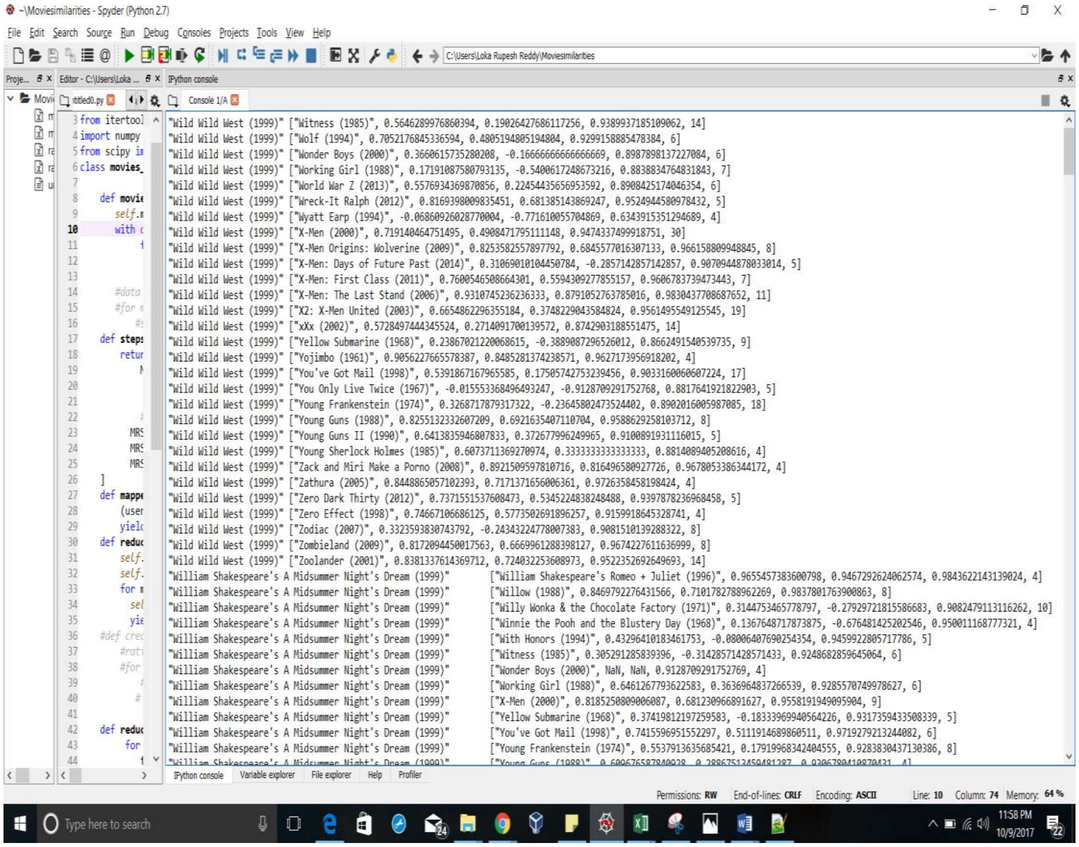
# Installed respective packages and libraries required for computation

Source Code:

```
from mrjob.job import MRJob

from mrjob.step import MRStep

from itertools import combinations

import numpy

from scipy import spatial

class movies_count(MRJob):

 #  steps function determine the sequence of operations

    def steps(self):

        return [

            MRStep(mapper=self.moviedatasplit,

                reducer=self.joinfilereducer),

          MRStep(reducer=self.reducer_moviepairs),

          MRStep(reducer=self.reducer_pairs),
        MRStep(reducer=self.movie_similarity)
    ]

 # Passing two files (movies.csv and ratings.csv) to the first mapper
    def moviedatasplit(self, _, line):
            dsplit = line.split(",")
            if (len(dsplit) == 3): # movie data
                  yield dsplit[0], dsplit[1]
            else: # rating data
                  yield dsplit[1], (dsplit[0], dsplit[2])
# generating user id as key and movie title, movierating as values with the help of first reducer
    def joinfilereducer(self, _, values):
            movielist = list(values)
            movietitle = movielist[0]
            tuplevalue = movielist[1:]
            for val in tuplevalue:
                 userid = val[0]
                 movierating = val[1]
                 yield userid, (movietitle, movierating)
# generating combination of two movies as key and their respective ratings as value for each user id with the second reducer
    def reducer_moviepairs(self,userid,values):
        for pair1, pair2 in combinations(values,2):
          title1=pair1[0]
          rating1=pair1[1]
          title2=pair2[0]
          rating2=pair2[1]
```

```
            yield (title1,title2),(rating1,rating2)
# combining all the ratings for each movie pair by different users with the third reducer
    def reducer_pairs(self,titles,ratings):
        rating=[]
        for r in ratings:
            rating.append(r)
        yield titles,rating

# finding similarity between movies using statistical coorelation and cosine similarity
    def movie_similarity(self,titles,ratings):
        rating =list(ratings)
        for ratings in rating:
            n=len(ratings)
        q1=[]
        q2=[]
        for r1 in ratings:
            q1.append((float(r1[0])))
            q2.append((float(r1[1])))

        if(n>3):
                cor = numpy.corrcoef(q1,q2)[0,1]
                cos_cor = 1-spatial.distance.cosine(q1,q2)
                avg_cor = 0.5*(cor+cos_cor)
                yield titles[0], (titles[1],avg_cor,cor,cos_cor,n)
# main function
if __name__ == '__main__':
            movies_count.run()
```

- I have computer similarities for all movies, below is the screenshot that reinforces it. (with the condition being shared similarity>3), since for shared similarities <3, I could see Nan values, which happens in cases like num/0, inf/inf or 0/inf


~\Moviesimilarities - Spyder (Python 2.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - C:\Users\Loka ...
IPython console
Console 1/A
IPython console
Variable explorer
File explorer
Help
Profiler
Type here to search

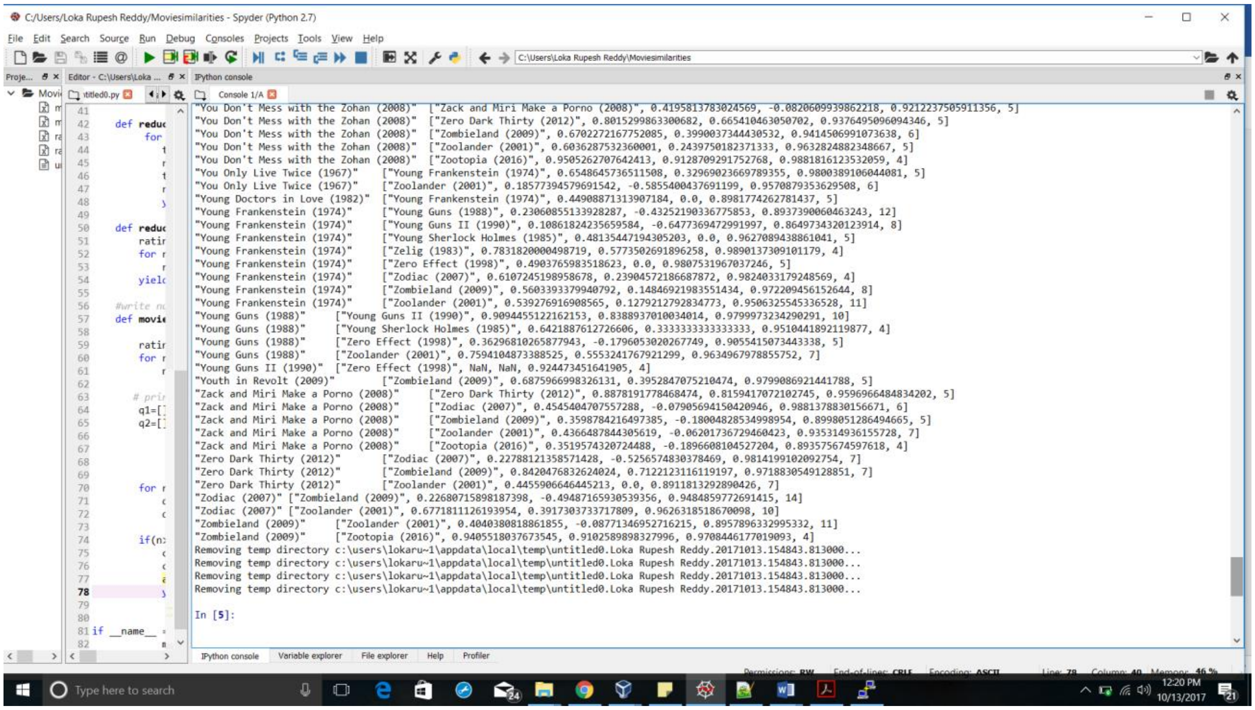
C:\Users\Loka Rupesh Reddy\Moviesimilarities

```
from itertool
import numpy
from scipy in
class movies_
```

```
"Wild Wild West (1999)" ["Witness (1985)", 0.5646289976860394, 0.19026427686117256, 0.9389937185109062, 14]
"Wild Wild West (1999)" ["Wolf (1994)", 0.7052176845336594, 0.4805194805194804, 0.9299158885478384, 6]
"Wild Wild West (1999)" ["Wonder Boys (2000)", 0.3660615735280208, -0.16666666666666669, 0.8987898137227084, 6]
"Wild Wild West (1999)" ["Working Girl (1988)", 0.17191087580793135, -0.5400617248673216, 0.8838834764831843, 7]
"Wild Wild West (1999)" ["World War Z (2013)", 0.5576934369870856, 0.22454435656953592, 0.8908425174046354, 6]
"Wild Wild West (1999)" ["Wreck-It Ralph (2012)", 0.8169398009835451, 0.681385143869247, 0.9524944580978432, 5]
"Wild Wild West (1999)" ["Wyatt Earp (1994)", -0.06860926028770004, -0.771610055704869, 0.634391535129468, 4]
"Wild Wild West (1999)" ["X-Men (2000)", 0.719140464751495, 0.4908471795111148, 0.9474337499918751, 30]
"Wild Wild West (1999)" ["X-Men Origins: Wolverine (2009)", 0.8253582557897792, 0.6845577016307133, 0.966158809948845, 8]
"Wild Wild West (1999)" ["X-Men: Days of Future Past (2014)", 0.31069010104450784, -0.2857142857142857, 0.9070944878033014, 5]
"Wild Wild West (1999)" ["X-Men: First Class (2011)", 0.7600546508664301, 0.5594309277855157, 0.9606783739473443, 7]
"Wild Wild West (1999)" ["X-Men: The Last Stand (2006)", 0.9310745236236333, 0.8791052763785016, 0.9830437708687652, 11]
"Wild Wild West (1999)" ["X2: X-Men United (2003)", 0.6654862296355184, 0.3748229043584824, 0.9561495549125545, 19]
"Wild Wild West (1999)" ["xXx (2002)", 0.5728497444345524, 0.271409170013957, 0.8742903188551475, 14]
"Wild Wild West (1999)" ["Yellow Submarine (1968)", 0.23867021220068615, -0.388908729652601, 0.8662491540539735, 9]
"Wild Wild West (1999)" ["Yojimbo (1961)", 0.9056227665578387, 0.8485281374238571, 0.9627173956918202, 4]
"Wild Wild West (1999)" ["You've Got Mail (1998)", 0.5391867167965585, 0.17505742753239456, 0.9033160060607224, 17]
"Wild Wild West (1999)" ["You Only Live Twice (1967)", -0.015553368496493247, -0.9128709291752768, 0.881764192182290, 5]
"Wild Wild West (1999)" ["Young Frankenstein (1974)", 0.3268717879317322, -0.23645802473524402, 0.8902016005987085, 18]
"Wild Wild West (1999)" ["Young Guns (1988)", 0.8255132332607209, 0.6921635407110704, 0.9588629258103712, 8]
"Wild Wild West (1999)" ["Young Guns II (1990)", 0.6413835946807833, 0.372677996249965, 0.9100891931116015, 5]
"Wild Wild West (1999)" ["Young Sherlock Holmes (1985)", 0.6073711369270974, 0.3333333333333333, 0.8814089405208616, 4]
"Wild Wild West (1999)" ["Zack and Miri Make a Porno (2008)", 0.8921509597810716, 0.816496580927726, 0.9678053386344172, 4]
"Wild Wild West (1999)" ["Zathura (2005)", 0.8448865057102393, 0.7171371656006361, 0.9726358458198424, 4]
"Wild Wild West (1999)" ["Zero Dark Thirty (2012)", 0.7371551537608473, 0.5345224838248488, 0.9397878236968458, 5]
"Wild Wild West (1999)" ["Zero Effect (1998)", 0.7466710668612, 0.5773502691896257, 0.9159918645328741, 4]
"Wild Wild West (1999)" ["Zodiac (2007)", 0.3323593830743792, -0.24343224778007383, 0.9081510139288322, 8]
"Wild Wild West (1999)" ["Zombieland (2009)", 0.8172094450017563, 0.6669961288398127, 0.9674227611636999, 8]
"Wild Wild West (1999)" ["Zoolander (2001)", 0.8381337614369712, 0.72403225360897, 0.9522352692649693, 14]
"William Shakespeare's A Midsummer Night's Dream (1999)"        ["William Shakespeare's Romeo + Juliet (1996)", 0.9655457383600798, 0.9467292624062574, 0.984362214313902, 4]
"William Shakespeare's A Midsummer Night's Dream (1999)"        ["Willow (1988)", 0.8469792276431566, 0.7101782788962269, 0.983780176390086, 8]
"William Shakespeare's A Midsummer Night's Dream (1999)"        ["Willy Wonka & the Chocolate Factory (1971)", 0.3144753465778797, -0.27929721815586683, 0.9082479113116262, 10]
"William Shakespeare's A Midsummer Night's Dream (1999)"        ["Winnie the Pooh and the Blustery Day (1968)", 0.1367648717873875, -0.676481425202546, 0.950011168777321, 4]
"William Shakespeare's A Midsummer Night's Dream (1999)"        ["With Honors (1994)", 0.43296410183461753, -0.08006407690254354, 0.9459922805717786, 5]
"William Shakespeare's A Midsummer Night's Dream (1999)"        ["Witness (1985)", 0.305291285839396, -0.3142857142857143, 0.9248682859645064, 6]
"William Shakespeare's A Midsummer Night's Dream (1999)"        ["Wonder Boys (2000)", NaN, NaN, 0.9128709291752769, 4]
"William Shakespeare's A Midsummer Night's Dream (1999)"        ["Working Girl (1988)", 0.646126779362258, 0.3636964837266539, 0.9285570749978627, 6]
"William Shakespeare's A Midsummer Night's Dream (1999)"        ["X-Men (2000)", 0.8185250809006087, 0.681230966891627, 0.9558191949095904, 9]
"William Shakespeare's A Midsummer Night's Dream (1999)"        ["Yellow Submarine (1968)", 0.3741981219725958, -0.18333969940564226, 0.9317359433508339, 5]
"William Shakespeare's A Midsummer Night's Dream (1999)"        ["You've Got Mail (1998)", 0.7415596951552297, 0.5111914689860511, 0.9719279213244082, 6]
"William Shakespeare's A Midsummer Night's Dream (1999)"        ["Young Frankenstein (1974)", 0.5537913635685421, 0.17919968342404555, 0.9283830437130386, 8]
"William Shakespeare's A Midsummer Night's Dream (1999)"        ["Young Guns (1988)", 0.609676587840928, 0.2886751345948128, 0.930678041087043, 4]
```

Permissions: RW   End-of-lines: CRLF   Encoding: ASCII   Line: 10   Column: 74   Memory: 64 %

11:58 PM
10/9/2017


C:/Users/Loka Rupesh Reddy/Moviesimilarities - Spyder (Python 2.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Loka Rupesh Reddy\Moviesimilarities
IPython console
Console 1/A
IPython console Variable explorer File explorer Help Profiler

```
"You Don't Mess with the Zohan (2008)"  ["Zack and Miri Make a Porno (2008)", 0.4195813783024569, -0.0820609939862218, 0.9212237505911356, 5]
"You Don't Mess with the Zohan (2008)"  ["Zero Dark Thirty (2012)", 0.8015299863300682, 0.665410463050702, 0.9376495096094346, 5]
"You Don't Mess with the Zohan (2008)"  ["Zombieland (2009)", 0.6702272167752085, 0.3990037344430532, 0.9414506991073638, 6]
"You Don't Mess with the Zohan (2008)"  ["Zoolander (2001)", 0.6036287532360001, 0.2439750182371333, 0.9632824882348667, 5]
"You Don't Mess with the Zohan (2008)"  ["Zootopia (2016)", 0.9505262707642413, 0.9128709291752768, 0.9881816123532059, 4]
"You Only Live Twice (1967)"    ["Young Frankenstein (1974)", 0.6548645736511508, 0.32969023669789355, 0.9800389106044081, 5]
"You Only Live Twice (1967)"    ["Zoolander (2001)", 0.18577394579691542, -0.5855400437691199, 0.9570879353629508, 6]
"Young Doctors in Love (1982)"  ["Young Frankenstein (1974)", 0.44908871313907184, 0.0, 0.8981774262781437, 5]
"Young Frankenstein (1974)"     ["Young Guns (1988)", 0.23060855133928287, -0.43252190336775853, 0.8937390060463243, 12]
"Young Frankenstein (1974)"     ["Young Guns II (1990)", 0.10861824235659584, -0.6477369472991997, 0.8649734320123914, 8]
"Young Frankenstein (1974)"     ["Young Sherlock Holmes (1985)", 0.48135447194305203, 0.0, 0.9627089438861041, 5]
"Young Frankenstein (1974)"     ["Zelig (1983)", 0.7831820000498719, 0.5773502691896258, 0.9890137309101179, 4]
"Young Frankenstein (1974)"     ["Zero Effect (1998)", 0.4903765983518623, 0.0, 0.9807531967037246, 5]
"Young Frankenstein (1974)"     ["Zodiac (2007)", 0.6107245198958678, 0.23904572186687872, 0.9824033179248569, 4]
"Young Frankenstein (1974)"     ["Zombieland (2009)", 0.5603393379940792, 0.14846921983551434, 0.972209456152644, 8]
"Young Frankenstein (1974)"     ["Zoolander (2001)", 0.539276916908565, 0.1279212792834773, 0.9506325545336528, 11]
"Young Guns (1988)"     ["Young Guns II (1990)", 0.9094455122162153, 0.8388937010034014, 0.9799973234290291, 10]
"Young Guns (1988)"     ["Young Sherlock Holmes (1985)", 0.6421887612726606, 0.3333333333333333, 0.9510441892119877, 4]
"Young Guns (1988)"     ["Zero Effect (1998)", 0.36296810265877943, -0.1796053020267749, 0.9055415073443338, 5]
"Young Guns (1988)"     ["Zoolander (2001)", 0.7594104873388525, 0.5553241767921299, 0.9634967978855752, 7]
"Young Guns II (1990)"  ["Zero Effect (1998)", NaN, NaN, 0.924473451641905, 4]
"Youth in Revolt (2009)"        ["Zombieland (2009)", 0.6875966998326131, 0.3952847075210474, 0.9799086921441788, 5]
"Zack and Miri Make a Porno (2008)"     ["Zero Dark Thirty (2012)", 0.8878191778468474, 0.8159417072102745, 0.9596966484834202, 5]
"Zack and Miri Make a Porno (2008)"     ["Zodiac (2007)", 0.4545404707557288, -0.07905694150420946, 0.9881378830156671, 6]
"Zack and Miri Make a Porno (2008)"     ["Zombieland (2009)", 0.3598784216497385, -0.18004828534998954, 0.8998051286494665, 5]
"Zack and Miri Make a Porno (2008)"     ["Zoolander (2001)", 0.4366487844305619, -0.06201736729460423, 0.935314936155728, 7]
"Zack and Miri Make a Porno (2008)"     ["Zootopia (2016)", 0.3519574320724488, -0.1896608104527204, 0.893575674597618, 4]
"Zero Dark Thirty (2012)"       ["Zodiac (2007)", 0.22788121358571428, -0.5256574830378469, 0.9814199102092754, 7]
"Zero Dark Thirty (2012)"       ["Zombieland (2009)", 0.8420476832624024, 0.7122123116119197, 0.9718830549128851, 7]
"Zero Dark Thirty (2012)"       ["Zoolander (2001)", 0.4455906646445213, 0.0, 0.8911813292890426, 7]
"Zodiac (2007)" ["Zombieland (2009)", 0.22680715898187398, -0.49487165930539356, 0.9484859772691415, 14]
"Zodiac (2007)" ["Zoolander (2001)", 0.6771811126193954, 0.3917303733717809, 0.9626318518670098, 10]
"Zombieland (2009)"     ["Zoolander (2001)", 0.4040380818861855, -0.08771346952716215, 0.8957896332995332, 11]
"Zombieland (2009)"     ["Zootopia (2016)", 0.9405518037673545, 0.9102589898327996, 0.9708446177019093, 4]
Removing temp directory c:\users\lokaru~1\appdata\local\temp\untitled0.Loka Rupesh Reddy.20171013.154843.813000...
Removing temp directory c:\users\lokaru~1\appdata\local\temp\untitled0.Loka Rupesh Reddy.20171013.154843.813000...
Removing temp directory c:\users\lokaru~1\appdata\local\temp\untitled0.Loka Rupesh Reddy.20171013.154843.813000...
Removing temp directory c:\users\lokaru~1\appdata\local\temp\untitled0.Loka Rupesh Reddy.20171013.154843.813000...

In [5]:
```

Filter and format the output:

I have used 1 mapper and 4 reducers in the source code

Description of each step written below with # symbol prior to the reducer and mapper.

Source Code:

# Imported packages and libraries required for computation

from mrjob.job import MRJob

from mrjob.step import MRStep

from itertools import combinations

import numpy

from scipy import spatial

class movies_count(MRJob):

  # Configure_options function is used to customize the output

    def configure_options(self):

        super(movies_count, self).configure_options()

        self.add_passthrough_option(

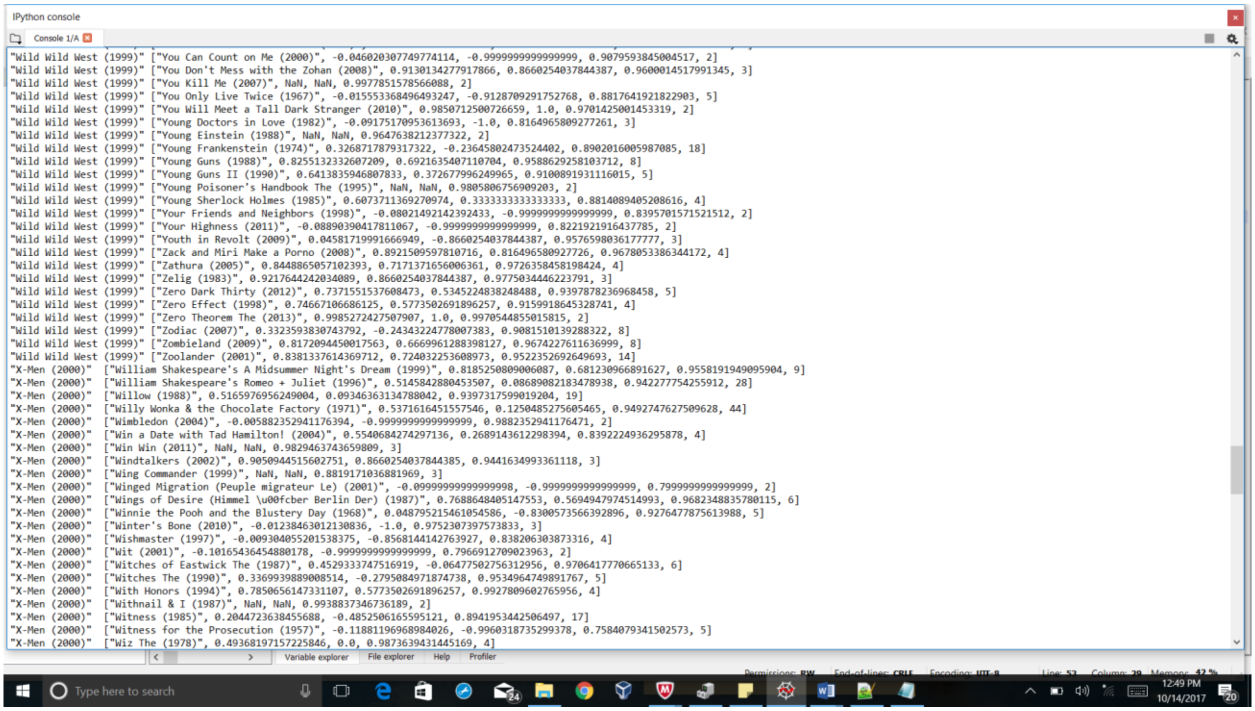
```
        '-m', '--moviename', action="append", type='str', default=[], help='Expressions to search for.')
    self.add_passthrough_option(
        '-p', '--rating_pairs', type='int', default=1, help='minimum rating pairs')
    self.add_passthrough_option(
        '-k', '--items', type='int', default=25, help='number of items to looks for')
    self.add_passthrough_option(
                  '-l', '-bound', type ='float', default=0.4, help ='similarity bound to look for')
  # steps determine the sequence of functions to be executed
  def steps(self):
    return [
      MRStep(mapper=self.moviedatasplit,
        reducer=self.joinfilereducer),
     MRStep(reducer=self.reducer_moviepairs),
     MRStep(reducer=self.reducer_pairs),
     MRStep(reducer=self.movie_similarity)
  ]

 # Passing two files (movies.csv and ratings.csv) to the first mapper
  def moviedatasplit(self, _, line):
        dsplit = line.split(",")
        if (len(dsplit) == 3): # movie data
             yield dsplit[0], dsplit[1]
        else: # rating data
             yield dsplit[1], (dsplit[0], dsplit[2])
# generating user id as key and movie title, movierating as values with the help of first reducer
  def joinfilereducer(self, _, values):
```

```
            movielist = list(values)

            movietitle = movielist[0]

            tuplevalue = movielist[1:]

            for val in tuplevalue:

                  userid = val[0]

                  movierating = val[1]


                  yield userid, (movietitle, movierating)

# generating combination of two movies as key and their respective ratings  as value for each user id with the second reducer

   def reducer_moviepairs(self,userid,values):

       for pair1,pair2 in combinations(values,2):

         title1=pair1[0]

         rating1=pair1[1]

         title2=pair2[0]

         rating2=pair2[1]

         yield (title1,title2),(rating1,rating2)

# combining all the ratings for each movie pair by different users with the third reducer

   def reducer_pairs(self,titles,ratings):

      rating=[]

      for r in ratings:

         rating.append(r)

      yield titles,rating


# finding similarity between movies using Cosine Similarity and Corelation

   def movie_similarity(self,titles,ratings):

      k= self.options.items
```

```
    rating =list(ratings)
    for ratings in rating:
      n=len(ratings)
    q1=[]
    q2=[]
    for r1 in ratings:
      q1.append((float(r1[0])))
      q2.append((float(r1[1])))

    if(n>self.options.rating_pairs):
        for movie in self.options.moviename:

          cor = numpy.corrcoef(q1,q2)[0,1]
          cos_cor = 1-spatial.distance.cosine(q1,q2)
          avg_cor = 0.5*(cor+cos_cor)
          while(k>0):
            if titles[0] == movie:
              yield titles[0],(titles[1],avg_cor,cor,cos_cor,n)
            elif titles[1]==movie:
              yield titles[1], (titles[0],avg_cor,cor,cos_cor,n)
            k=k-1
# Main function:
if __name__ == '__main__':
      movies_count.run()
```

Command used -

-m "Wild Wild West (1999)" -k 25 -m "X-Men (2000)"-k 25 movies.csv ratings.csv



```
"Wild Wild West (1999)" ["You Can Count on Me (2000)", -0.046020307749774114, -0.9999999999999999, 0.9079593845004517, 2]
"Wild Wild West (1999)" ["You Don't Mess with the Zohan (2008)", 0.9130134277917866, 0.8660254037844387, 0.9600014517991345, 3]
"Wild Wild West (1999)" ["You Kill Me (2007)", NaN, NaN, 0.9977851578566088, 2]
"Wild Wild West (1999)" ["You Only Live Twice (1967)", -0.015553368496493247, -0.9128709291752768, 0.8817641921822903, 5]
"Wild Wild West (1999)" ["You Will Meet a Tall Dark Stranger (2010)", 0.9850712500726659, 1.0, 0.9701425001453319, 2]
"Wild Wild West (1999)" ["Young Doctors in Love (1982)", -0.09175170953613693, -1.0, 0.8164965809277261, 3]
"Wild Wild West (1999)" ["Young Einstein (1988)", NaN, NaN, 0.9647638212377322, 2]
"Wild Wild West (1999)" ["Young Frankenstein (1974)", 0.3268717879317322, -0.23645802473524402, 0.8902016005987085, 18]
"Wild Wild West (1999)" ["Young Guns (1988)", 0.8255132332607209, 0.6921635407110704, 0.9588629258103712, 8]
"Wild Wild West (1999)" ["Young Guns II (1990)", 0.6413835946807833, 0.372677996249965, 0.9100891931116015, 5]
"Wild Wild West (1999)" ["Young Poisoner's Handbook The (1995)", NaN, NaN, 0.9805806756909203, 2]
"Wild Wild West (1999)" ["Young Sherlock Holmes (1985)", 0.6073711369270974, 0.3333333333333333, 0.8814089405208616, 4]
"Wild Wild West (1999)" ["Your Friends and Neighbors (1998)", -0.08021492142392433, -0.9999999999999999, 0.8395701571521512, 2]
"Wild Wild West (1999)" ["Your Highness (2011)", -0.08890390417811067, -0.9999999999999999, 0.8221921916437785, 2]
"Wild Wild West (1999)" ["Youth in Revolt (2009)", 0.04581719991666949, -0.8660254037844387, 0.9576598036177777, 3]
"Wild Wild West (1999)" ["Zack and Miri Make a Porno (2008)", 0.8921509597810716, 0.816496580927726, 0.9678053386344172, 4]
"Wild Wild West (1999)" ["Zathura (2005)", 0.8448865057102393, 0.7171371656006361, 0.9726358458198424, 4]
"Wild Wild West (1999)" ["Zelig (1983)", 0.9217644242034089, 0.8660254037844387, 0.9775034446223791, 3]
"Wild Wild West (1999)" ["Zero Dark Thirty (2012)", 0.7371551537608473, 0.5345224838248488, 0.9397878236968458, 5]
"Wild Wild West (1999)" ["Zero Effect (1998)", 0.74667106686125, 0.5773502691896257, 0.9159918645328741, 4]
"Wild Wild West (1999)" ["Zero Theorem The (2013)", 0.9985272427507907, 1.0, 0.9970544855015815, 2]
"Wild Wild West (1999)" ["Zodiac (2007)", 0.3323593830743792, -0.24343224778007383, 0.9081510139288322, 8]
"Wild Wild West (1999)" ["Zombieland (2009)", 0.8172094450017563, 0.6669961288398127, 0.9674227611636999, 8]
"Wild Wild West (1999)" ["Zoolander (2001)", 0.8381337614369712, 0.724032253608973, 0.9522352692649693, 14]
"X-Men (2000)"  ["William Shakespeare's A Midsummer Night's Dream (1999)", 0.8185250809006087, 0.681230966891627, 0.9558191949095904, 9]
"X-Men (2000)"  ["William Shakespeare's Romeo + Juliet (1996)", 0.5145842880453507, 0.08689082183478938, 0.942277754255912, 28]
"X-Men (2000)"  ["Willow (1988)", 0.5165976956249004, 0.09346363134788042, 0.9397317599019204, 19]
"X-Men (2000)"  ["Willy Wonka & the Chocolate Factory (1971)", 0.5371616451557546, 0.1250485275605465, 0.9492747627509628, 44]
"X-Men (2000)"  ["Wimbledon (2004)", -0.005882352941176394, -0.9999999999999999, 0.9882352941176471, 2]
"X-Men (2000)"  ["Win a Date with Tad Hamilton! (2004)", 0.5540684274297136, 0.2689143612298394, 0.8392224936295878, 4]
"X-Men (2000)"  ["Win Win (2011)", NaN, NaN, 0.9829463743659809, 3]
"X-Men (2000)"  ["Windtalkers (2002)", 0.9050944515602751, 0.8660254037844385, 0.9441634993361118, 3]
"X-Men (2000)"  ["Wing Commander (1999)", NaN, NaN, 0.8819171036881969, 3]
"X-Men (2000)"  ["Winged Migration (Peuple migrateur Le) (2001)", -0.09999999999999998, -0.9999999999999999, 0.7999999999999999, 2]
"X-Men (2000)"  ["Wings of Desire (Himmel \u00fcber Berlin Der) (1987)", 0.7688648405147553, 0.5694947974514993, 0.9682348835780115, 6]
"X-Men (2000)"  ["Winnie the Pooh and the Blustery Day (1968)", 0.048795215461054586, -0.8300573566392896, 0.9276477875613988, 5]
"X-Men (2000)"  ["Winter's Bone (2010)", -0.01238463012130836, -1.0, 0.9752307397573833, 3]
"X-Men (2000)"  ["Wishmaster (1997)", -0.009304055201538375, -0.8568144142763927, 0.838206303873316, 4]
"X-Men (2000)"  ["Wit (2001)", -0.10165436454880178, -0.9999999999999999, 0.7966912709023963, 2]
"X-Men (2000)"  ["Witches of Eastwick The (1987)", 0.4529333747516919, -0.06477502756312956, 0.9706417770665133, 6]
"X-Men (2000)"  ["Witches The (1990)", 0.3369939889008514, -0.2795084971874738, 0.9534964749891767, 5]
"X-Men (2000)"  ["With Honors (1994)", 0.7850656147331107, 0.5773502691896257, 0.9927809602765956, 4]
"X-Men (2000)"  ["Withnail & I (1987)", NaN, NaN, 0.9938837346736189, 2]
"X-Men (2000)"  ["Witness (1985)", 0.2044723638455688, -0.4852506165595121, 0.8941953442506497, 17]
"X-Men (2000)"  ["Witness for the Prosecution (1957)", -0.11881196968984026, -0.9960318735299378, 0.7584079341502573, 5]
"X-Men (2000)"  ["Wiz The (1978)", 0.49368197157225846, 0.0, 0.9873639431445169, 4]
```

Ran the movie recommendation part 2 in amazon aws as well using EMR-(Elastic Map Reduce).

Set the environment variables aws_access_key_id , aws_secret_access_key in local system

Taken from the aws profile security options.

Procedure followed:

1) Kept the movies and ratings csv files, and python file in S3 bucket.
2) Created cluster using EMR with 1 Master Node and 9 slave nodes
3) Set the SSH key between master node and slave nodes
4) Connected to amazon aws using putty
5) Once connected, copied the files available in s3 bucket to the created instance
6) Executed the command from console

Commands used:

Performed the below step to copy each file one by one from s3 bucket to the instance created

1) aws s3 cp s3://moviesimilarities/movies.csv ./

Executed the program with the below command

1) python untitled.py movies.csv ratings.csv