Project Book Butler is an application that evaluates desired output based on past data. The application is designed to cater users like students who wish to read books, consumers that visit DVD stores and by new DVDs etc.

We have used graphs as the main technique in our program. Graphs are tools that represent information that are easy to read. It’s an easy tool to understand links between different elements such as examining the relation between users which is the central aim of our project. For example, if two students read the same book, an edge would be created between them. Now the weight of that edge would depend on their taste; whether or not they have the same opinion regarding the books they read. larger the weight, similar the opinion.

We created two main functions ‘Show\_top\_picks’ and ‘recommend\_genre’ which are the options available to the user. The other functions are created in order for these two main functions to operate.

These are some of the pre-algorithms we made that we’ll need for the main functions:

Function openbookfile(name):

1. Import csv
2. Open csv file
3. X:=read csv file
4. d:=()
5. F:=[]
6. for each x in X
7. d=(x)
8. add d to f
9. d:=()

bookdict:={}

for f in F:

bookdict[f[0]]:=[f[2].split(‘.’), f[1], f[3]]

return bookdict

the above function reads the csv file booklist.csv. as the name suggests that, the file contains the data of the books in the following order, book name, author genre and price. In the first step we import csv and in the second we read the file. d is taken to be an empty file and f an empty list. Then a loop is run over the file and every line is added to the tuple and then appended to the list. After that the tuple is made empty again so for that the next line can be added to it and so on the process goes on. then we take an empty dictionary; bookdict. Then a loop is run over the list, over every tuple. The first element of the tuple is made the key of the dictionary and all the other elements in the tuple are stored in list and then made its value. The order of the values are genre, author and price

at the end of this function, this dictionary is stored in some variable; book\_data

function addNodes(G, nodes):

G=:{}

a)for nodes in NODES:

b) G[nodes]:=[]

this function takes an empty dictionary and creates a graph by putting nodes in the graph

function userdataload(name):

1. import csv
2. x:=read csv file
3. d:=()
4. F:=[]
5. for x in X
6. d:=(x)
7. add d to f
8. d:=()
9. bookdict:={}
10. for f in F
11. if x ()
12. final:={}
13. for y in range(length(f))
14. if f[y]:=’’
15. if y0
16. M:=f[y].split(‘,’)
17. o:=M[1]
18. final[M[0]]:=eval(o)
19. bookdict2[f[0]]:=final
20. return bookdict2

in this function we read the csv file user\_data.csv. as the name suggests, it contains the information of the user; the name of the user followed by the books read alongside the preference of each book. Preference pertains to whether the reader liked the book or not; True or False respectively. So first we import csv and read the file. we take an empty tuple and list. And then run a loop over every line in the file, putting the line into a tuple and then appending it to a list. Then the tuple is set to empty so that the next line can be put into the tuple and appended to the list. After the loop has finished, we take an empty dictionary; bookdict2. A loop is run over the list, accessing each tuple. It first checks whether the tuple is empty or not. If so, creates an empty dictionary; final. Then another loop is run over the tuples, accessing every element of the tuple. If that value is an empty string, we break out of the loop. Now we’ll only deal with all the elements except for the 0th element. The element that we’re dealing is in the form ‘bookname, preference (True/False)’. We split this value