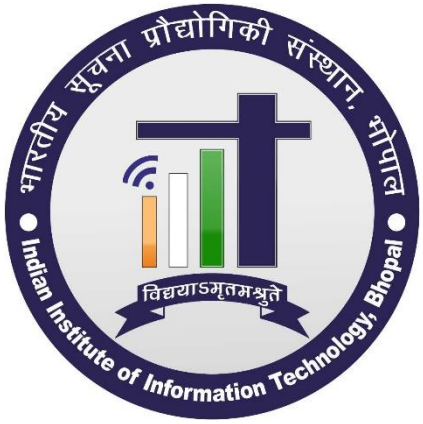
MINOR PROJECT REPORT

Submitted in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY
(Department of Computer Science and Engineering)

Submitted to

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY BHOPAL (M.P.)



Submitted by

Samarth Purandar – 20U02020
Himanshu Nageshwar – 20U02057

Under the supervision of

Dr. Nikhil Singh

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY BHOPAL (M.P.)

Certificate

This is to certify that the minor project report entitled "Friends Recommendation System using Machine Learning"is submitted by Samarth Purandar & Himanshu Nageshwar (IIIT Bhopal) in fulfilment of the requirements for the degree of Bachelor of Technology in Department of Computer Science and Engineering. This project is an authentic work done by them under my supervision and guidance.

This project has not been submitted to any other institution for the award of any degree.

Date: 19/04/2023

Dr. Nikhil Singh

CSE

IIIIT Bhopal

Minor Project Co-ordinator

Dr. Yatendra Sahu

Department of Computer Science and Engineering, IIIT Bhopal

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY BHOPAL (M.P.)

Student Declaration

I hereby declare, that the work presented in the project report entitled **"**Friends Recommendation System using Machine Learning**"** in partial fulfilment of the requirement for the award of degree of **"Bachelor of Engineering"** from **INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, BHOPAL** is record of my own work.

I, with this, declare that the facts mentioned above are true to the best of our knowledge. In case of any unlikely discrepancy that may occur, we will be the ones to take responsibility.

Date:19/04/2023

Place: IIIT Bhopal

Samarth Purandar
20U02020

Himanshu Nageshwar
20U02057

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY BHOPAL (M.P.)

Acknowledgement

With immense pleasure I, Mr. Samarth Purandar & Mr. Himanshu Nageshwarpresenting "Friends Recommendation System using Machine Learning" minor project report as a part of curriculum of "Bachelor of Engineering". I wish to thank all the people who gave me unending support.

I express my profound thanks to my project supervisor Dr. Nikhil Singhand all those who have indirectly guided and helped me in preparation of the report.

Samarth Purandar – 20U02020

Himanshu Nageshwar – 20U02057

Contents

ABSTRACT

- We will implement a friend recommendation system in python using collaborative filtering.

- We plan to use and analyze two methods for recommending friends to a user based on their commonfriends and influence of number of friends.

- To find the potential friends we will measure them using the ‘score’ metric where a higher score wouldmean that user is a better candidate for being friend.

- We will then test the recommendation system accuracy by removing the friendship edges of all friends in

- The data and then compute which method gives the most similar recommendations to that of theoriginal data.

- We would perform many trails of randomly chosen users from the data to compute accuracies.

- A subset map of friendship of the users will be displayed using edges and nodes in the form of graphs inpython.

INTRODUCTION

A recommender system or a recommendation system is a subclass of information filtering system that

seeks to predict the "rating" or "preference" that a user would give to an item. Recommender systems have

become increasingly popular in recent years and are utilized in a variety of areas including movies, music,

news, books, research articles, search queries, social tags, and products in general. There are also

recommender systems for experts, collaborators, jokes, restaurants, garments, financial services, life

insurance, online dating, and Twitter pages. Facebook suggests people you could be friends with. The

actual algorithms used by these companies are closely-guarded trade secrets. There are two generalapproaches: collaborative filtering and content-based filtering.

CONTENT BASED FILTERING:

- Idea: If you like an item then you will also like a “similar” item
- Based on similarity of the items being recommended
- It generally works well when it's easy to determine the context/properties of each item. For instance, whenwe are recommending the same kind of item like a movie recommendation or song recommendation.

COLLABORATIVE FILTERINIG:

- Idea: If a person A likes item 1, 2, 3 and B like 2,3,4 then they have similar interests and A should like item4 and B should like item 1.
- This algorithm is entirely based on the past behaviour and not on the context. This makes it one of themost commonly used algorithmas it is not dependent on any additional information.
- For instance: product recommendations by e-commerce player like Amazon and merchantrecommendations by banks like American Express.User-User Collaborative filtering,tem-ltemCollaborative filtering and Other simpler algorithms

RELATED WORK

ALGORITHM 1 - RECOMMEND BY NUMBER OF COMMON FRIENDS

Task definition:

One of the important factors while recommending new friends to a user is the number of mutual orcommon friends between them. As per the definition of collaborative filtering if two users have similar pastpreferences then their next maybe same as well, this makes common friends an important criteria. We willtake graph of friends as input and output recommendation list of new friends based on number of commonfriends.

ALGORITHM 2 - RECOMMEND BY INFLUENCE

Task definition:

Consider the following hypothetical situation.

Two of your friends are X and Y

X has only two friends (you and one other person).

Y has 7 billion friends.

X and Y have no friends in common (besides you).

Since X is highly selective in terms of friendship, and is a friend of yours, you are likely to have a lot incommon with J.D's other friend. On the other hand, Y is indiscriminate and there is little reason to believethat you should be friendly with any one of Y's other friends.

Incorporate the above idea into your friend recommendation algorithm. Here is the concrete way that youwill do so. We call the technique “influence scoring”.

METHODOLOGY

K – Nearest Neighbours Algorithm

The k-nearest neighbours algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.

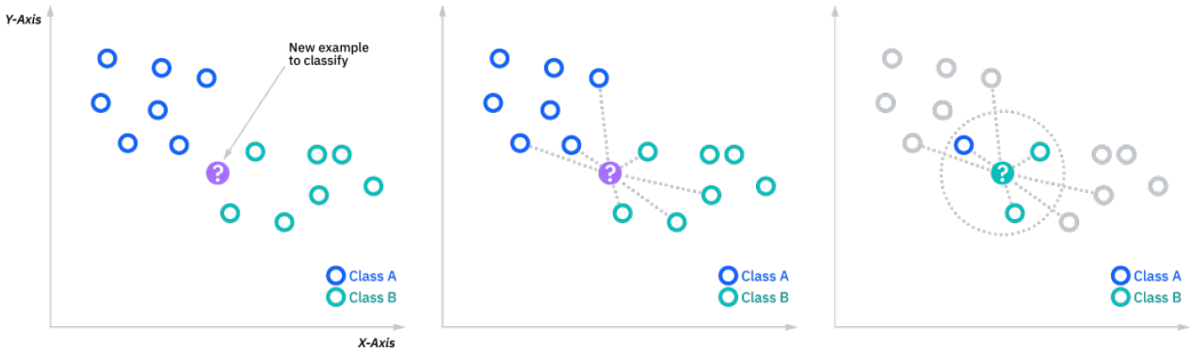


Fig. 1- KNN Diagram

DEFINING K

The k value in the K-NN algorithm defines how many neighbours will be checked to determine the classification of a specific query point. For example, if k=1, the instance will be assigned to the same class as its single nearest neighbour. Defining k can be a balancing act as different values can lead to overfitting or underfitting. Lower values of k can have high variance, but low bias, and larger values of k may lead to high bias and lower variance. The choice of k will largely depend on the input data as data with more outliers or noise will likely perform better with higher values of k. Overall, it is recommended to have an odd number for k to avoid ties in classification, and cross-validation tactics can help you choose the optimal k for your dataset.

Module Used

SURPRISE

Surprise is a Python scikit for building and analyzing recommender systems that deal with explicit rating data.

NUMPY

NumPy is a Python library used for working with arrays.

It also has functions for working in domain of linear algebra, fourier transform, and matrices.

NumPy stands for Numerical Python.

KNN Basic

SOURCE CODE - https://github.com/SH3R1FF/Friends-Recommendation-System

D: > minor project > friend.py > FriendRecommender

```python
from surprise import KNNBasic
import pandas as pd
from surprise import Dataset, SVD
from surprise import Reader


movies = ['Star wars', 'Star wars', 'GOT' , 'GOT' ,'South park', 'South park','Harry potter', 'Harry potter']
rating = [1,5,1,1,1,5,3,2]
users = ['Kim','Tim', 'John','Jimmy', 'Julia', 'Kim','Jimmy', 'Kim' ]

rating_dict = {'users' : users,
               'item' : movies,
               'rating': rating}

def FriendRecommender(user):
    df = pd.DataFrame(rating_dict)
    reader = Reader(rating_scale = (1,5))
    data  = Dataset.load_from_df(df[['users','item','rating']],reader)
    trainset = data.build_full_trainset()
    sim_options = {
        'name':'cosine',
        'user_based':True
    }

    algo = KNNBasic(sim_options)
    algo.fit(trainset)

    uid = trainset.to_inner_uid(user)
    pred = algo.get_neighbors(uid,3)

    for i in pred:
        print(trainset.to_raw_uid(i))

FriendRecommender('Kim')
```

Fig. 2 – Source Code

Conclusion

After testing our recommendation algorithms on the dataset, we can see that K nearest neighbour algorithm gives the best recommendations for the user. Tofurther enhance our friend recommendation system we can also use the time at which the friendship wasformed between the two users, this is the future scope of our project.

References

Knn Algorithm Module - https://pypi.org/project/basic-knn/

Sci Surprise - https://surpriselib.com/

Numpy - https://numpy.org/

Github - https://github.com/SH3R1FF/Friends-Recommendation-System

List of Figures