APPENDIX 1

MOVIE RECOMMENDATION SYSTEM

END TERM REPORT

by

SNO	NAME	REG NO	ROLL NO
1	M.GANGA THARUN	11802393	21
2	P.DORA BABU	11802429	32
3	K.SUSMITH KUMAR	11814586	33
4	DURGAJANARDHANA SUBRAHMANYAM	11802437	08

SECTION: K18HV

Department of Intelligent Systems

School of Computer Science Engineering

Lovely Professional University, Jalandhar

APRIL-2020

APPENDIX 2

Student Declaration

This is to declare that this report has been written by us. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged. We aver that if any part of the report is found to be copied, we are shall take full responsibility for it.

M.GANGA THARUN

Roll :21

P.DORA BABU

Roll :32

K.SUSMITH KUMAR

Roll : 33

DURGAJANARDHANA SUBRAHMANYAM

Roll: 08

Place: Jalandhar

Date: 10/04/20

APPENDIX 3

TABLE OF CONTENTS

TITLE **PAGE NO.**

APPENDIX 4

BONAFIDE CERTIFICATE

Certified that this project report "MOVIE RECOMMENDATION SYSTEM is the bonafide work of "M.GANGA THARUN, P.DORA BABU, K.SUSMITH KUMAR, DURGAJANARDHANA SUBRAHMANYAM" who carried out the project work under my supervision.

INTRODUCTION

To build a simple recommender system in python, we need to know what is a recommender. A recommender is a service provider that suggests based on the users taste. In this project we created a movie recommender system which recommends or suggests new movies based on their viewing history.

OBJECTIVES OF THE PROJECT ASSIGNED :

The main objectives of the movie recommender project are

1.To create a system that suggests new movies based on the users view history

2.To acknowledge the students how simple intelligent systems work.

Recommender Systems:

A recommender system is a system that makes suggestions based on the users surfing history or interests. For example, when you continuously browse the same product in any shopping websites like Amazon, flipkart they recommend the same type of objects or things that you have already gone through online.So next time Amazon suggests you a product, or Netflix recommends you a tv show or medium display a great post on your feed, understand that there is a recommendation system working under the hood.

There are two types of recommender systems. They are

1. Content based
2. Collaborative

Content based recommender system:

It works on the generated data of a user. There are two ways in which data is generated, either explicitly or implicitly. A user profile is created using the data generated. It contains the meta data of the items the user interacted. The accuracy of the system or engine depends on how much amount of data it receives.

Collaborative Recommender System

This system makes recommendation based on how many users liked the same item in a similar way. Using item similarity, it can also perform collaborative filtering (like 'Users who liked this item X also liked Y').

Packages used :

Lightfm:

LightFM is a Python implementation of a number of popular recommendation algorithms. LightFM includes implementations of BPR and WARP ranking losses(A **loss function** is a measure of how good a prediction model does in terms of being able to predict the expected outcome.).

BPR:

Bayesian Personalised Ranking pairwise loss: It maximizes the prediction difference between a positive example and a randomly chosen negative example. It is useful when only positive interactions are present.

WARP: Weighted Approximate-Rank Pairwise loss:

Maximises the rank of positive examples by repeatedly sampling negative examples until rank violating one is found.

Code:

```
import numpy as np

from lightfm.datasets import fetch_movielens
```

```
from lightfm import LightFM
#fetch data from model
data = fetch_movielens(min_rating = 4.0)

#print training and testing data
print(repr(data['train']))
print(repr(data['test']))

#create model
model = LightFM(loss = 'warp')

#train mode
model.fit(data['train'], epochs=30, num_threads=2)

#recommender fucntion
def sample_recommendation(model, data, user_ids):
    #number of users and movies in training data
    n_users, n_items = data['train'].shape
    for user_id in user_ids:
        #movies they already like
        known_positives = data['item_labels'][data['train'].tocsr()[user_id].indices]

        #movies our model predicts they will like
        scores = model.predict(user_id, np.arange(n_items))
        #sort them in order of most liked to least
        top_items = data['item_labels'][np.argsort(-scores)]
        #print out the results
```

```
    print("User %s" % user_id)
    print("     Known positives:")

    for x in known_positives[:3]:
        print("         %s" % x)

    print("     Recommended:")

    for x in top_items[:3]:
        print("         %s" % x)

sample_recommendation(model, data, [3, 25, 451])
```

GITHUB LINK: https://github.com/Durgajanardhana/ai-project

Snapshots: