APPENDIX 1

# Movie Recommendation System

END TERM REPORT

*by*

Kaushik

Section: -K18HV

Roll Number: 58

Vishal Naryal

Section: -K18HV

Roll Number: 35

Komal Singh

Section: -K18HV

Roll Number: 37

Theopan Simon

Section: -K18HV

Roll Number: 59


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

Kaushik Pandey

Roll number:58

Vishal Naryal

Roll Number: 35

Komal Singh

Roll Number: 37

Theopan Simon

Roll Number: 59

Place: Phagwara

Date:13 April 2020

APPENDIX 3

# Table of Contents

**Title** **Page**

APPENDIX 4

# BONAFIDE CERTIFICATE

Certified that this project report "**Movie Recommendation System**" is the bonafide work of "Kaushik Pandey", "Vishal Naryal"," Komal Singh"," Theopan Simon" who carried out the project work under my supervision.

<<Signature of the Supervisor>>

<<Name of supervisor>>

<<Academic Designation>>

<<ID of Supervisor>>

<<Department of Supervisor>>

# Background and objectives of the project assigned

A recommendation system is a type of information filtering system which attempts to predict the preferences of a user, and make suggests based on these preferences. There are a wide variety of applications for recommendation systems. These have become increasingly popular over the last few years and are now utilized in most online platforms that we use. The content of such platforms varies from movies, music, books and video, to friends and stories on social media platforms, to products on e-commerce websites, to people on professional and dating websites, to search results returned on Google. Often, these systems are able to collect information about a user's choices, and can use this information to improve their suggestions in the future. For example, Facebook can monitor your interaction with various stories on your feed in order to learn what types of stories appeal to you. Sometimes, the recommender systems can make improvements based on the activities of a large number of people. For example, if Amazon observes that a large number of customers who buy the latest Apple MacBook also buy a USB-C-to USB Adapter, they can recommend the Adapter to a new user who has just added a MacBook to his cart. Due to the advances in recommender systems, users constantly expect good recommendations. They have a low threshold for services that are not able to make appropriate suggestions. If a music streaming app is not able to predict and play music that the user likes, then the user will simply stop using it. This has led to a high emphasis by tech companies on improving their recommendation systems. However, the problem is more complex than it seems. Every user has different preferences and likes. In addition, even the taste of a single user can vary depending on a large number of factors, such as mood, season, or type of activity the user is doing. For example, the type of music one would like to hear while exercising differs greatly from the type of music he'd listen to when cooking dinner. The main approaches are widely used for recommender systems is content-based filtering, where we try to profile the user's interests using information collected, and recommend items based on that profile. The other is collaborative filtering, where we try to group similar users together and use information about the group to make recommendations to the user.

Our project entitled "Movie Recommendation System" aims to suggest or recommend the various users, the movie they might like, by intake of their ratings, comments and history. The system proposed is a kind of collaborative-based filtering system which finally recommends the likable movie to the users using K-means clustering. This will extract vital information and recommend the users according to user's preferences, interest, or history about movies. Our system is to use dataset which are to be thoroughly filtered in order to gain user's idea for movies. This system is to be implemented with web services (on server side) and desktop application (on client side). This filtering method matches content resources to user characteristics, base their predictions on user's information. It relies heavily on the ratings of different users.

# Algorithm were used for recommendation system are

## Content based algorithm

Content based system recommend items similar to that user has liked or purposed in the past. If any items are liked, similar items will be recommended. It is based on properties of each items to measure the similarity. The point of content based is that we have to know the content of both the user and items. Recommendations are based on the content of items rather than other user's opinion.

## Collaborative filtering

Collaborative filtering method finds a subset of user who have similar test and preferences to the target user and use this subset for offering recommendations. In this method user with similar interest have common preferences. If a person A likes items 1, 2, 3 and B likes 2, 3, 4 then they have similar interest and A should like item 4 and B should like item 1. It is entirely based on the past behaviour and not on the present context. It is not dependent on any additional information. It is used by 'Amazon'.

## Item-Item collaboration

It uses user-item ratings matrix and makes item-to-item correlations. It finds items that are highly correlated and recommend items with highest correlation. The advantages of this method no knowledge about item features is needed. It has better stability because correlation between limited number of items instead of very large number of users. It reduces sparsely problem.

## K-means clustering algorithm

K-means is one of the simple unsupervised learning algorithms that solves the well-known clustering problems. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priority. The main idea is to define k centres, one for each cluster. These centres should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centre. When no point is pending, the first step is completed and early group age is done. At this point we need to re-calculate k new centroids as barycentre of clusters resulting from the previous step.

# Description of project

## Modules: -

```
import urllib.request
import zipfile
import os
import pandas as pd
import numpy as np
from scipy import spatial
```

### PANDAS: -

Pandas is probably the most powerful library in Data analysis. It provides high performance tools for data manipulation and analysis. Furthermore, it is very effective at converting data formats and querying data out of databases.

### Urllib.request:-

urllib.request is a Python module for fetching URLs (Uniform Resource Locators). It offers a very simple interface, in the form of the urlopen function. This is capable of fetching URLs using a variety of different protocols.

### Zipfile: -

ZipFile is a class of zipfile module for reading and writing zip files. Here we import only class ZipFile from zipfile module.

### **OS**: -

The OS module in python provides functions for interacting with the operating system. OS, comes under Python's standard utility modules. This module provides a portable way of using operating system dependent functionality. The *os* and *os.path* modules include many functions to interact with the file system.

### **NumPy**: -

NumPy stands for "Numeric Python" or "Numerical Python" It is a Python module (open source) that provides a faster way of doing mathematical computation on arrays and matrices Arrays and matrices are very essential in most machine learning ecosystems, hence when NumPy is combined with other machine learning modules such as Scikit learn, Matplotlib, TensorFlow, Pandas and others, the machine learning ecosystem becomes complete It supports both 1 D Arrays(Vectors) and Multi-Dimensional Arrays(Matrices)

### SciPy Spatial: -

The scipy.spatial package can calculate Triangulation, Voronoi Diagram and Convex Hulls of a set of points, by leveraging the Qhull library. Likewise, it contains KDTree implementations for nearest-neighbor point queries and utilities for distance computations in various metrics.

## DATASET: -

A data set (or dataset) is a collection of data. In the case of tabular data, a data set corresponds to one or more database tables, where every column of a table represents a particular variable, and each row corresponds to a given record of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set. Each value is known as a datum. Data sets can also consist of a collection of documents or files.

| | user_id | item_id | Rating | Movie |
|---|---|---|---|---|
| **0** | 101 | 5 | 5 | Copycat (1995) |
| **1** | 107 | 5 | 2 | Copycat (1995) |
| **2** | 126 | 5 | 4 | Copycat (1995) |
| **3** | 129 | 5 | 2 | Copycat (1995) |
| **4** | 130 | 5 | 5 | Copycat (1995) |
| **5** | 147 | 5 | 3 | Copycat (1995) |
| **6** | 102 | 1 | 5 | Toy Story (1995) |
| **7** | 103 | 1 | 1 | Toy Story (1995) |
| **8** | 104 | 2 | 3 | GoldenEye (1995) |
| **9** | 110 | 2 | 2 | GoldenEye (1995) |
| **10** | 114 | 2 | 2 | GoldenEye (1995) |
| **11** | 119 | 2 | 3 | GoldenEye (1995) |
| **12** | 132 | 2 | 4 | GoldenEye (1995) |
| **13** | 133 | 2 | 5 | GoldenEye (1995) |
| **14** | 134 | 2 | 3 | GoldenEye (1995) |
| **15** | 149 | 2 | 5 | GoldenEye (1995) |
| **16** | 105 | 3 | 3 | Four Rooms (1995) |
| **17** | 106 | 3 | 1 | Four Rooms (1995) |

In this Dataset we have used MovieLens dataset in order to build movie recommendation engine.

## Types of Dataset: -

The full dataset: This dataset consists of 26,000,000 ratings and 750,000 tag applications applied to 45,000 movies by 270,000 users. Includes tag genome data with 12 million relevance scores across 1,100 tags.

The small dataset: This dataset comprises of 100,000 ratings and 1,300 tag applications applied to 9,000 movies by 700 users. We will build a simple Recommendation for movies using the full dataset.

## Data description

It contains 100004 ratings and 1296 tag applications across 9125 movies. These data were created by 671 users between January 09, 1995 and October 16, 2016. This dataset was generated on October 17, 2016.

Users were selected at random for inclusion. All selected users had rated at least 20 movies. No demographic information is included. Each user is represented by an id, and no other information is provided.

## The data are contained in the following files.

credits.csv

keywords.csv

links.csv

links_small.csv

movies_metadata.csv

ratings.csv

ratings_small.csv

```
# import pandas library
import pandas as pd

# Reading all the data about the movie from the file
column_names = ['user_id', 'item_id', 'Rating']

path = 'fffile.tsv'

df = pd.read_csv(path, sep='\t',names=column_names)

# Check the head of the data
df.head(150)

# Check out all the movies and their respective IDs
movie_names = pd.read_csv('Movie_Titles.csv')
movie_names.head(150)

data = pd.merge(df, movie_names, on='item_id')
data.head(150) #List of all movies
```

**DESCRIPTION OF ROLES AMONG STUDENTS: -**

1. **Kaushik Pandey (11814352**): - Studied about all the libraries required to make this project and coded the resume reading and parsing part in the code.

2. **Vishal Naryal (11815638)**: - Studied about Dataset and help Kaushik in coding.

3. **Komal Singh (11813322)**: - Studied about SciPy Spatial libraries He gave us an idea how to implement member function.

4. **Simon (11816041)**: - He wrote the report and help in searching information.

Technologies and Frameworks used in projects: -

1.Python idle or any python compiler

2.Dataset for movie credit, rating, user id, movie name, link, keyword.

## Dependencies

- Python >=3.5
- pandas
- numpy
- scipy

## SWOT Analysis achieved in project.

### Strengths:

1. By analysing the customer's current site usage and his previous browsing history, a recommendation engine can deliver relevant product recommendations as he shops.
2. Shoppers become more engaged in the site when personalized product recommendations are made.
3.They are able to delve more deeply into the product line without having to perform search after search.

### Weakness:

1. Perhaps the biggest issue facing recommender systems is that they need a lot of data to effectively make recommendations
2. Requires an active internet connection.
3.User Preferences changes after month by month.
4.Unpredictable Item Company doesn't know about the user preference and have to predict more.

### Threats:

1. User Preferences data may be leaked.
2. Security threats.

## Conclusion:

Movie Recommender Systems are software tools that and techniques providing suggestions for movies to be rated by a critic. The suggestions provided are aimed at supporting the critics in various decision-making processes, such as what movies to rate, what music from the movie users will like, or what new movies they have to rate. Recommender systems have proven to be necessary method for users to cope with the information bloom and have become most powerful and popular tools in e-commerce. Simultaneously, various algorithms for recommendation systems have been proposed over last decade, many of them have also been successfully implemented in commercial markets and some still suffer from demerits and waiting to be corrected. Recommender systems is a multi-disciplinary effort which involves expertise from various fields of Artificial intelligence, Human Computer Interaction, Information Technology, Data Mining, Statistics, Adaptive User Inter