LOVELY PROFESSIONAL UNIVERSITY

INT-254

Submitted By: Aniket Chauhan, Abhinav Choudhary

2020-2024

Annexure-II: Student Declaration

To whom so ever it may concern I, ANIKET CHAUHAN, 12011156 and ABHINAV CHOUDHARY, 12015798, hereby declare that the work done by me on “Movie Recommendation System” from October, 2022 to November, 2022, for the course code of INT254: “Fundamental of Machine Learning” is a record of original work for the partial fulfillment of the requirements for the award of the degree, Bachelor of Technology.

ANIKET CHAUHAN (12011156)

ABHINAV CHOUDHARY (12015798)

Dated: 15th November,2022

ACKNOWLEDGEMENT

Primarily I would like to thank God for being able to learn a new technology. Then I would like to express my special thanks of gratitude to the teacher Dr.Pawan Kumar Mall (Registration Number: 28839) and who provide me the golden opportunity to learn a new technology from home. I would like to also thank my own college Lovely Professional University for offering such a course which not only improve my programming skill but also taught me other new technology. Then I would like to thank my parents and friends who have helped me with their valuable suggestions and guidance for choosing this course. Finally I would like to thank my all classmates who have helped me a lot.

Date: 15/11/2022

ANIKET CHAUHAN (12011156)

ABHINAV CHOUDHARY (12015798)

TABLE OF CONTENT

INTRODUCTION 5
The Intelligent Recommender System 5
What is a Recommendation System? 5
ALGORITHMS 6
LIBRARIES INCLUDED 7
DATASET USED 8
PROGRAMMING SCREENSHOTS 9

INTRODUCTION

The Intelligent Recommender System

Ever wondered how Netflix or Hotstar recommends new movies based on the watch history, how Amazon or Flipkart suggests new products based on your order or search history?

These suggestions or recommendations are done by a system called a recommendation system. This engine makes suggestions by learning and understanding the patterns in your watch history (let's say) and then applies those patterns and findings to make new suggestions.

What is a Recommendation System?

Before moving on to build a recommender engine for movies, let's discuss recommendation systems.

Recommendation systems are computer programs that suggest recommendations to users depending on a variety of criteria.
These systems estimate the most likely product that consumers will buy and that they will be interested in. Netflix, Amazon, and other companies use recommender systems to help their users find the right product or movie for them.

ALGORITHMS

Step 1: Perform Exploratory Data Analysis (EDA) on the data.

Step 2: Build the Movie Recommender System.

Step 3: Get recommendations for the movies.

LIBRARIES INCLUDED

Libraries included in our projects are as follows:

1. PANDAS

   Pandas is built on top of two core Python libraries—matplotlib for data visualization and NumPy for mathematical operations. Pandas acts as a wrapper over these libraries, allowing you to access many of matplotlib's and NumPy's methods with less code.

2. SCIPY

   SciPy in Python is an **open-source library used for solving mathematical, scientific, engineering, and technical problems**. It allows users to manipulate the data and visualize the data using a wide range of high-level Python commands. SciPy is built on the Python NumPy extention.

3. SKLEARN

   Sickie-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbors, and it also supports Python numerical and scientific libraries like NumPy and SciPy.

4. FUZZYWUZZY

   FuzzyWuzzy is a library of Python which is used for string matching. Fuzzy string matching is the process of finding strings that match a given pattern. Basically it uses Levenshtein Distance to calculate the differences between sequences.

DATASET USED

DATA OF SONG :

"C:\Users\1rajp\Desktop\songdata.csv"

PROGRAMMING SCREENSHOTS

1.

jupyter music recommender system Last Checkpoint: 7 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Run Code

In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
from typing import List, Dict
```

In [3]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

In [26]:
```
songs = pd.read_csv(r'C:\Users\1rajp\Desktop\songdata.csv')
```

In [27]:
```
songs.head()
```

Out[27]:

	artist	song	link	text
0	ABBA	Ahe's My Kind Of Girl	/a/abba/ahes+my+kind+of+girl_20598417.html	Look at her face, it's a wonderful face \nAnd...
1	ABBA	Andante, Andante	/a/abba/andante+andante_20002708.html	Take it easy with me, please \nTouch me gentl...
2	ABBA	As Good As New	/a/abba/as+good+as+new_20003033.html	I'll never know why I had to go \nWhy I had t...
3	ABBA	Bang	/a/abba/bang_20598415.html	Making somebody happy is a question of give an...
4	ABBA	Bang-A-Boomerang	/a/abba/bang+a+boomerang_20002668.html	Making somebody happy is a question of give an...

In [7]:
```
songs = songs.sample(n=5000).drop('link', axis=1).reset_index(drop=True)
```

In [28]:
```
songs['text'] = songs['text'].str.replace(r'\n', '')
```

2.

In [9]:
```
tfidf = TfidfVectorizer(analyzer='word', stop_words='english')
lyrics_matrix = tfidf.fit_transform(songs['text'])
```

In [10]:
```
cosine_similarities = cosine_similarity(lyrics_matrix)
```

In [11]:
```
similarities = {}
```

3.

```
In [29]: for i in range(len(cosine_similarities)):
             # Now we'll sort each element in cosine_similarities and get the indexes of the songs.
             similar_indices = cosine_similarities[i].argsort()[:-50:-1]
             # After that, we'll store in similarities each name of the 50 most similar songs.
             # Except the first one that is the same song.
             similarities[songs['song'].iloc[i]] = [(cosine_similarities[i][x], songs['song'][x], songs['artist'][x])
                                                    for x in similar_indices][1:]
```

4.

```
In [13]: class ContentBasedRecommender:
             def __init__(self, matrix):
                 self.matrix_similar = matrix

             def _print_message(self, song, recom_song):
                 rec_items = len(recom_song)

                 print(f'The {rec_items} recommended songs for {song} are:')
                 for i in range(rec_items):
                     print(f"Number {i+1}:")
                     print(f"{recom_song[i][1]} by {recom_song[i][2]} with {round(recom_song[i][0], 3)} similarity score")
                     print("--------------------")

             def recommend(self, recommendation):
                 # Get song to find recommendations for
                 song = recommendation['song']
                 # Get number of songs to recommend
                 number_songs = recommendation['number_songs']
                 # Get the number of songs most similars from matrix similarities
                 recom_song = self.matrix_similar[song][:number_songs]
                 # print each item
                 self._print_message(song=song, recom_song=recom_song)
```

\

5.

```
In [18]: recommedations = ContentBasedRecommender(similarities)
```

```
In [20]: recommendation = {
             "song": songs['song'].iloc[10],
             "number_songs": 4
         }
```

```
In [21]: recommedations.recommend(recommendation)
```

```
The 4 recommended songs for Plaisir D'amour are:
Number 1:
I Do Adore Her by Harry Belafonte with 0.217 similarity score
--------------------
Number 2:
La Vie En Rose by Donna Summer with 0.161 similarity score
--------------------
Number 3:
Outside My Window by Stevie Wonder with 0.155 similarity score
--------------------
Number 4:
People Need Love by ABBA with 0.151 similarity score
--------------------
```

6.

In [22]:
```
recommendation2 = {
    "song": songs['song'].iloc[120],
    "number_songs": 4
}
```

In [23]:
```
recommedations.recommend(recommendation2)
```

```
The 4 recommended songs for I Want Your Love Tonight are:
Number 1:
All Your Love by Bob Seger with 0.587 similarity score
--------------------
Number 2:
If I Can't Have You by Bee Gees with 0.485 similarity score
--------------------
Number 3:
I Want To Know What Love Is by Foreigner with 0.429 similarity score
--------------------
Number 4:
Again Tonight by John Mellencamp with 0.415 similarity score
--------------------
```

References to Websites:

1. https://www.geeksforgeeks.org/best-python-libraries-for-machine-learning/
2. https://towardsdatascience.com/how-to-build-a-movie-recommendation-system-67e321339109
3. https://data-flair.training/blogs/data-science-r-movie-recommendation