

Project Report on

Movie Recommendation

Based on Machine Learning using Python

Submitted in the fulfilment of our training program

By:

Harsh Kumar Jha (11900220010) (TEAM LEADER)

Rishika Sharma (11900220002)

Harsh Sharma (11900220009)

Anshika Sahai(11900220036)

Trisha Dutta (11900220014)

Shubham Divyanshu (11900220008)

5th Semester students of

Siliguri Institute of Technology

Under

Maulana Abul Kalam Azad University of Technology

Under the supervision of

Mr. ARPAN SAMANTA

ABSTRACT

The principal objective of this project is to perform full analysis for the Movie of the employees of the organisation and detect and predict the salaries of the employees using Machine Learning. In this project the concept of Machine Learning using Python has been used to its fullest extent. The data has been collected through a company for the study and implementation of it through Machine Learning. The data collected so far is of experience years and the salary. We know that to bring out the best out of employee an organisation must set some parameters through which a productivity of an employee can be measured. One such metrics is the number of years a person has been in the field.

Salary Prediction based on experience using ML: In this project, we have worked on an end-to-end case study to understand the different stages of Model building using the Machine Learning concept. This will deal with "data manipulation" with pandas and Numpy, and "data visualisation" with Matplotlib library with the Salary dataset. After Data manipulation, Data visualisation will be performed using graphs.

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and participation of a large number of individuals to this attempt. Our project report has been structured under the valued suggestion, support and guidance of Mr. Arpan Samanta. Under his guidance we have accomplished the challenging task in a very short time. Finally, we would express our sincere thankfulness to our family members for inspiring us all throughout and always encouraging us.

INTRODUCTION

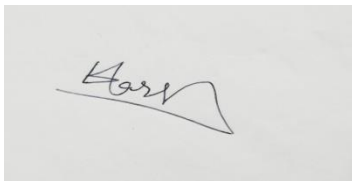
Have you ever wondered how YouTube recommends content, or how Facebook recommends you, new friends? Perhaps you've noticed similar recommendations with LinkedIn connections, or how Amazon will recommend similar products while you're browsing. All of these recommendations are made possible by the implementation of recommender systems.

Recommender systems encompass a class of techniques and algorithms that can suggest “relevant” items to users. They predict future behavior based on past data through a multitude of techniques including matrix factorization.

In this article, I'll look at why we need recommender systems and the different types of users online. Then, I'll show you how to build your own movie recommendation system using an open-source dataset.

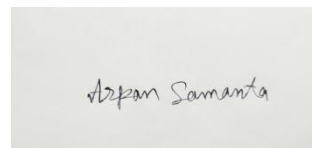
Certificate approval

The foregoing project is hereby approved as a creditable study for B.Tech and presented in a manner of satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorsed or approved any statement made, opinion express or conclusion therein but approve this project only for the purpose for which it is submitted.



HARSH SHARMA

IT DEPARTMENT ,SIT

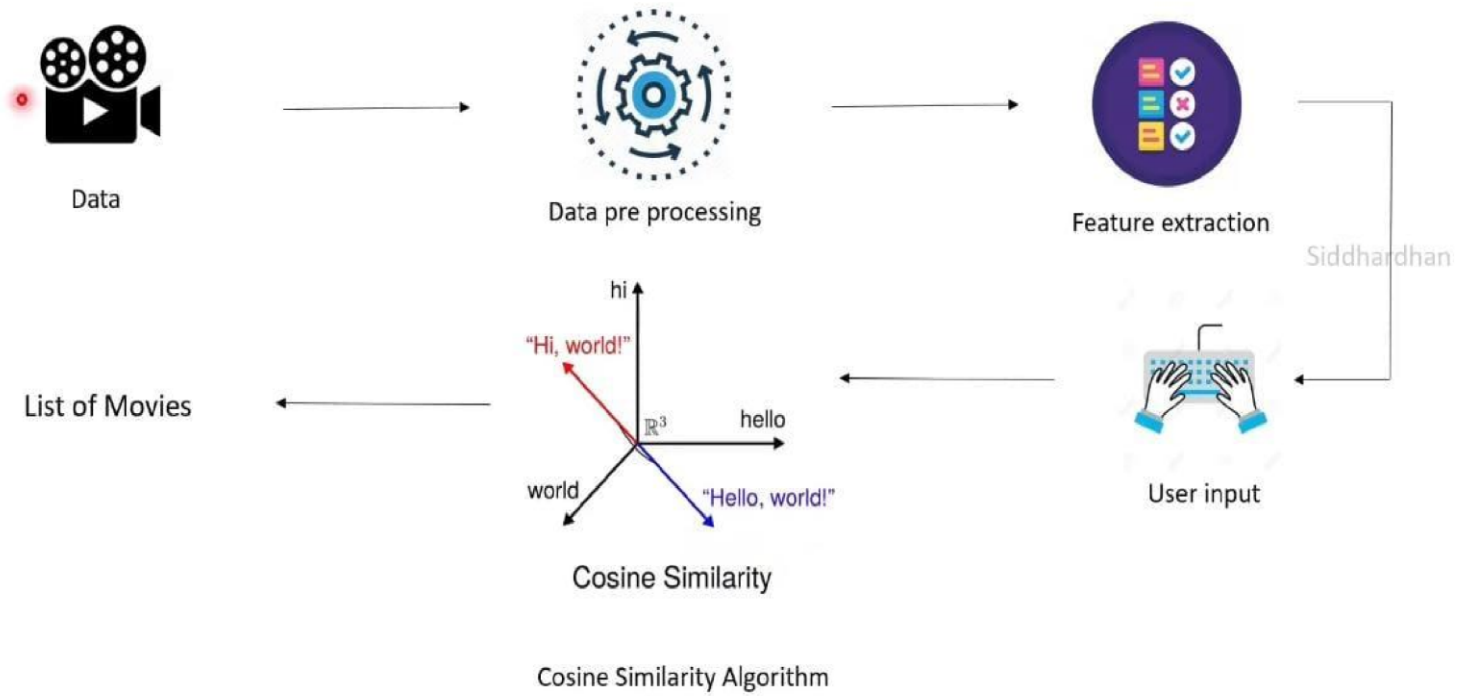


SIGNATURE

ARPAN SAMANTA

SIKHARTHY INFOTECH PVT.LTD

Work Flow



MOVIE RECOMMENDATION USING MACHINE LEARNING:

The screenshot displays a Google Colab environment with the following components:

- Browser Tabs:** New Tab, Movie Recommendation System, How To Start a Presentation (With), (9) WhatsApp.
- Address Bar:** colab.research.google.com/drive/11gxCTw6pX8FOWhe10imGo_FGgtK7CvLY#scrollTo=tu788kxcamrj
- Colab Interface:**
 - File Edit View Insert Runtime Tools Help** menu bar.
 - Movie Recommendation System.ipynb** title bar with a star icon.
 - Comment Share** buttons and a user profile icon.
 - RAM Disk** status bar.
 - Editing** mode indicator.
- Code Editor:**
 - PROJECT** section header.
 - MOVIE RECOMMENDATION** title.
 - Team members** section with the text: Harsh kumar jha , Rishika sharma , Anshika sahai ,Trisha dutta,Harsh sharma ,Shubham Divyanshu,Rajen gupta .
 - Code Cell [12]:**

```
# importing the dependencies
# to support array
import numpy as np
# covert array data into structure table or dataframe
import pandas as pd
# find closest match to user input
import difflib
#convert text file into numerical values / vectors
from sklearn.feature_extraction.text import TfidfVectorizer
# find similar data to the user
from sklearn.metrics.pairwise import cosine_similarity
```
- Bottom Bar:**
 - 3m 57s completed at 12:10 PM
 - System tray: 33°C Cloudy, Windows taskbar icons, ENG IN, 13:08 15-09-2022.

New TabMovie Recommendation SystemHow To Start a Presentation (With) (9) WhatsApp

colab.research.google.com/drive/11gxCTw6pX8FOWhe10imGo_FGgtK7CvLY#scrollTo=tu788kxcamrj

AppsGoogleWhatsAppOnline Tuition Teac...GmailYouTubeMaps

Movie Recommendation System.ipynb

FileEditViewInsertRuntimeToolsHelpAll changes saved

+ Code+ Text

RAMDisk

Editing

[12] # find similar data to the user
from sklearn.metrics.pairwise import cosine_similarity

[13] # loading the data from the csv file to apandas dataframe
movies_data = pd.read_csv('/content/movies.csv')

#sample data
movies_data.head(1)

	index	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	...	runtime	spoken_
0	0	237000000	Action Adventure Fantasy Science Fiction	http://www.avatarmovie.com/	19995	culture clash future space war space colony so...	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.437577	...	162.0	[{"iso_6:

1 rows x 24 columns

3m 57s completed at 12:10 PM

33°C Cloudy

ENG IN13:09 15-09-2022

New Tab x Movie Recommendation System. x How To Start a Presentation (Wit x (9) WhatsApp x

colab.research.google.com/drive/11gxCTw6pX8FOWhe10imGo_FGgtK7CvLY#scrollTo=tu788kxcamrj

Apps Google WhatsApp Online Tuition Teac... Gmail YouTube Maps

Movie Recommendation System.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk

Editing

```
[18] # selecting the relevant features for recommendation
selected_features = ['genres','keywords','tagline','cast','director']

[19] # replacing the null values with null string

for feature in selected_features:
    movies_data[feature] = movies_data[feature].fillna('')

[6] # combining all the 5 selected features
combined_features = movies_data['genres']+' '+movies_data['keywords']+' '+movies_data['tagline']+' '+movies_data['cast']+' '+movies_data['director']

# converting the text data to feature vectors
vectorizer = TfidfVectorizer()

[8] feature_vectors = vectorizer.fit_transform(combined_features)

[9] # getting the similarity scores using cosine similarity
similarity = cosine_similarity(feature_vectors)
```

3m 57s completed at 12:10 PM

33°C Cloudy

ENG IN 13:09 15-09-2022

New Tab x Movie Recommendation System. x How To Start a Presentation (Wit x (9) WhatsApp x

colab.research.google.com/drive/11gxCTw6pX8FOWhe10imGo_FGgtK7CvLY#scrollTo=tu788kxcamrj

Apps Google WhatsApp Online Tuition Teac... Gmail YouTube Maps

Movie Recommendation System.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk

Editing

Movie Recommendation Sytem

```
print("""click
    0 for Movie
    1 for exit """)
a= int(input("Choose what u want : "))
if a==0:
    c=0
    while c==0:
        # getting the movie name from the user
        movie_name = input(' Enter your favourite movie name : ')

        # creating a list with all the movie names given in the dataset
        list_of_all_titles = movies_data['title'].tolist()

        # finding the close match for the movie name given by the user
        find_close_match = difflib.get_close_matches(movie_name, list_of_all_titles)
        close_match = find_close_match[0]

        # finding the index of the movie with title
        index_of_the_movie = movies_data[movies_data.title == close_match]['index'].values[0]
```

3m 57s completed at 12:10 PM

33°C Cloudy

ENG IN 13:09 15-09-2022

Movie Recommendation System.ipynb

```
# getting a list of similar movies
similarity_score = list(enumerate(similarity[index_of_the_movie]))

# sorting the movies based on their similarity score
sorted_similar_movies = sorted(similarity_score, key = lambda x:x[1], reverse = True)

# print the name of similar movies based on the index
print('Movies suggested for you : \n')

i = 1

for movie in sorted_similar_movies:
    index = movie[0]
    title_from_index = movies_data[movies_data.index==index]['title'].values[0]
    if (i<30):
        print(i, '.',title_from_index)
        i+=1
print("""          click
0 for Search more movie
1 for exit """)
b= int(input("Choose what u want : "))
```

3m 57s completed at 12:10 PM

Movie Recommendation System.ipynb

```
index = movie[0]
title_from_index = movies_data[movies_data.index==index]['title'].values[0]
if (i<30):
    print(i, '.',title_from_index)
    i+=1
print("""          click
0 for Search more movie
1 for exit """)
b= int(input("Choose what u want : "))
c=b
if(b==1) :
    print(" visit again Thankyou ! ")
elif(b==0):
    print("")
else:
    print("WRONG OPTION OPTED")

# main loop elif
elif(a==1) :
    print(" visit again Thankyou ! ")
else:
    print(" WRONG OPTION OPTED")
```

3m 57s completed at 12:10 PM

CONCLUSION

We have fully completed this project based on Machine Learning using Python to recommend movies to the user by their own experience. We by know that Machine Learning has a huge and vast scope in such genuine problem. It is like that we can improve productivity by getting the knowledge on the basis of the experience a certain employee has over time.