Movie Recommendation System Using Machine Learning

1. Manideep Prathipati
Computer Science
700726314
University of Central Missouri
mxp63140@ucmo.edu

2. Chouturi Praneeth babu
Computer Science
700742707
University of Central Missouri
pxc27070@ucmo.edu

3.Shaik Shameer
Computer Science
700740584
University of Central Missouri
sxs05840@ucmo.edu

4. Akhila Boddu
Computer Science
700742171
University of Central Missouri
axb21710@ucmo.edu

Abstract: Since digital platforms and streaming services have become more popular, movie recommendation systems have become more useful in guiding users through vast amounts of content.Machine learning techniques are used to analyze users' interests, preferences, and viewing history in order to provide recommended content based on their behavior, preferences, and viewing history.To provide user with accurate and relevant movie recommendations, we present a movie recommendation system based on machine learning algorithms.

To generate movie recommendations, we combine collaborative and content-driven filtering techniques. An algorithm that uses collaborative filtering identifies patterns in user behavior and preferences and makes recommendations based on those patterns. To make recommendations based on the user's preferences, content-based filtering analyzes attributes such as genre, cast, director, and plot of movies.

Our model was trained using a large dataset of movie review data and user ratings. Cleaning and transforming the data into a machine-learning-friendly format was part of the preprocessing. In addition to performing precision, recall, and F1 scores on the training set, we compared the model's performance to various metrics in the testing set.

The performance evaluation of our movie recommendation system indicates that it surpasses several baseline models in terms of accuracy in predicting user ratings. To assess the system's effectiveness in a real-world scenario, we conducted a user study. Users were requested to rate the relevance of the movie recommendations presented by our system. According to the study's results, our recommendations were considered highly relevant and useful by the users.

Compared to existing movie recommendation systems, our system has several benefits. Firstly, it is scalable and capable of handling vast datasets comprising millions of user ratings and movies. Secondly, our system incorporates both collaborative and content-based filtering techniques, which increases the accuracy and relevance of recommendations. Finally, our system is created to be adaptable and flexible to changes in user behavior and preferences.

Keywords: Movie recommendation, Random forest, Gaussian NB,algorithms, personalized, data, Regression, Machine learning.

Identify applicable funding agency here. If none, delete this.

I. Introduction

In today's fast-paced world, people are constantly busy and have limited time to accomplish their tasks. As a result, recommendation systems are becoming increasingly important as they help individuals make informed choices without having to expend their cognitive resources.

The primary purpose of a recommendation system is to search for and identify content that would be of interest to a particular individual. This involves several factors that are used to create personalized lists of relevant and useful content specific to each user. Artificial Intelligence-based algorithms are used to sift through all available options and generate a customized list of items that are interesting and relevant to an individual. These recommendations are based on the user's profile, browsing and search history, what other people with similar traits or demographics are watching, and how likely the user is to watch those movies. This is accomplished through predictive modeling and heuristics using available data.

The movie Unscripted dramas are expanding step by step in the current ages. There are various ways of finding the movie Rating Point. First and foremost, the crude information is taken in light Individuals' Meter and the no of perspectives will be counted from that. Then, at that point, we really want to isolate the entire informational collection into bunches in light of various movies. Here the informational index comprises of movies. Select the specific movie and take the view count. Contingent on the quantity of perspectives, rate the movie or show appropriately, as on the off chance that the view count is more than 10,000, and allocate 10 rating to that specific show. If any new information is available, add it simultaneously, and afterward the entire cycle begins once more. With the assistance of proposed calculation, we can refresh, add new sections in the process moreover.

In light of the quantity of perspectives we will rate that the specific movie programs likewise with the most elevated Rating. The rating can measure up among various shows and be seen in reference diagrams, pie outlines, histograms. Regression algorithm and calculations to analyze the rating. The correlation between the two calculations is extremely

clear on histograms. It is the simplest approach to foreseeing Television program examination. Assuming the information is incorrect it Might result to blame qualities.

A recommendation system is a type of information filtering system that aims to determine a user's preferences and make suggestions based on them. Such systems have a wide range of applications, from movies, podcasts, books, and videos, to colleagues and stories on social media, to products on e-commerce websites, to individuals on commercial and dating websites. These systems can collect and filter data on a user's preferences and use it to improve their suggestions. For example, Twitter can analyze a user's engagement with various stories on their feed to understand their preferences.

As the popularity of recommendation systems has grown, they have become ubiquitous across online platforms. Users now have high expectations for such systems and are quick to abandon those that fail to make appropriate recommendations. To meet these demands, technical companies have focused on improving their recommendation structures. However, this is not a simple task.

Each user has unique likes and dislikes, which can vary depending on a variety of factors, such as mood, season, or activity. For instance, the type of music one prefers during exercise may differ significantly from what they enjoy while cooking. To make accurate recommendations, these systems must continually expand their understanding of the user's preferences while still utilizing what they already know.

Recommender systems use two primary methods: content-based filtering and collaborative filtering. In content-based filtering, data is collected to create a user profile, which is then used to suggest items. Collaborative filtering, on the other hand, clusters users with similar preferences together and uses data from the group to make recommendations to individual users. These methods are critical in the development of efficient and accurate recommendation systems.

II. Motivation

A movie recommendation system can be motivated by a desire to improve the user experience for movie lovers. Many people enjoy watching movies but may struggle with finding movies that suit their preferences. A recommendation system can help address this challenge by suggesting movies that are likely to be of interest to the user based on their past viewing history, ratings, and other factors.

The driving force behind the creation of a movie recommendation system utilizing machine learning is the desire to enhance the user's browsing and movie selection experience. With an overwhelming number of movies to choose from, users often find it challenging to select a movie to watch, and personalized recommendations can help ease this process. Since users have varying preferences, the need for personalized recommendations is essential, and machine learning algorithms can analyze user data to provide tailored recommendations.

Moreover, utilizing machine learning algorithms for movie recommendations is a complex research problem that has generated significant interest from the scientific community. The development of such systems requires a multidisciplinary approach that includes expertise in statistics, data analysis, machine learning, and psychology.

In addition to improving the user experience and satisfaction, a movie recommendation system can increase user engagement and retention. Furthermore, the development of these systems can contribute to the advancement of the field by addressing complex research problems and generating new knowledge.

Overall, the motivation for creating a movie recommendation system using machine learning is to enhance the user experience, increase engagement and retention, address complex research problems, and contribute to the advancement of the field.

Additionally, movie recommendation systems can benefit movie streaming platforms by increasing user engagement and retention. When users find movies they enjoy, they are more likely to continue using the platform and recommend it to others, leading to increased revenue for the platform. Overall, a movie recommendation system can provide value to both users and businesses, making it a compelling area of research and development.

III. Main Contributions Objectives

- Development of personalized recommendations based on user preferences and viewing history
- Improvement of user experience and satisfaction
- Increase in user engagement and retention
- Advancement of the field through interdisciplinary research and application of machine learning algorithms
- Provide users with personalized and relevant movie recommendations
- Utilize vast amounts of user data to continuously improve the recommendation system
- Enhance the user browsing and selection experience
- Increase user engagement and retention on the movie platform
- Tackle complex research problems in machine learning and data analysis
- Contribute to the advancement of the field through the development of new techniques and methodologies for movie recommendation systems.

IV. Related Work

Movie recommendation rating systems with machine learning have gained significant attention in recent years due to the increasing popularity of online streaming platforms such as Netflix, Hulu, and Amazon Prime. These platforms use recommendation systems to provide personalized recommendations to users based on their preferences and viewing history, increasing user engagement and satisfaction. One of the most used techniques in movie recommendation systems is collaborative filtering, which analyzes the behavior of similar users to generate recommendations. The algorithm identifies users

who have similar movie preferences and recommends movies that those users have watched and rated highly. Collaborative filtering has been shown to be effective in generating accurate recommendations and has been used by platforms such as Netflix and Amazon. Content-based filtering is another technique used in movie recommendation systems. It involves analyzing the characteristics of the movies, such as genre, plot, and cast, to generate recommendations. Content-based filtering can be particularly useful in situations where user behavior data is limited or not available. Hybrid recommendation systems that combine collaborative and content-based filtering techniques have also been explored. These systems aim to provide more accurate and personalized recommendations[2] by combining the strengths of both techniques. Machine learning algorithms such as k-Nearest Neighbors (k-NN), Singular Value Decomposition (SVD), and Matrix Factorization (MF) have been used in movie recommendation systems. These algorithms analyze user behavior data and movie features to generate recommendations. Deep learning techniques such as neural networks have also been explored in movie recommendation systems, but they are typically more computationally intensive and require more data.

A. Existing Solutions:

There are several existing movie recommendation rating systems with machine learning that have been implemented by various online streaming platforms. Here are a few examples:

1) ***Netflix****: :* Netflix's recommendation system is one of the most well-known and widely used movie recommendation systems. It uses a combination of collaborative filtering, content-based filtering, and hybrid recommendation techniques to generate personalized recommendations for users. Netflix uses a variety of machine learning algorithms, including Matrix Factorization and Restricted Boltzmann Machines, to analyze user behavior data and movie features.

2) ***Amazon Prime:****:* Amazon Prime's recommendation system uses collaborative filtering and content-based filtering techniques to generate recommendations. It also utilizes machine learning algorithms such as Factorization Machines and Multi-layer Perceptron (MLP) models [3] to analyze user behavior data and movie features.

3) ***Hulu: :*** Hulu's recommendation system uses a combination of collaborative filtering and content-based filtering techniques to generate recommendations. It also incorporates a hybrid recommendation model that combines the strengths of both techniques. Hulu uses machine learning algorithms such as Matrix Factorization and k-Nearest Neighbors to analyze user behavior data and movie features.

4) ***YouTube:****:* YouTube's recommendation system is focused on videos, but it can also recommend movies based on a user's viewing history and preferences. It uses a combination of collaborative filtering and content-based filtering techniques to generate recommendations. YouTube uses machine learning algorithms such as Long Short-Term Memory (LSTM) networks and Neural Collaborative Filtering to analyze user behavior data and video features.

B. Challenges in Existing Solutions:

1) ***Data quality and quantity: :*** The success of a recommendation system largely depends on the quality and quantity of data. Existing solutions needs to collect large amounts of data on user behavior, such as watching history, ratings, and search queries. However, data quality can be affected by various factors, such as incomplete data, inaccurate data, and biased data.

2) ***Cold start problem:****:* The cold start problem refers to the difficulty of making accurate recommendations for new users or items with little to no historical data. Existing solutions needs to find ways to make accurate recommendations for new users who have not yet interacted with the platform or for new movies that have not yet been watched by many users.

3) ***Diversity and novelty****::* One of the challenges for recommendation systems is to balance between recommending popular and relevant items and recommending diverse and novel items. This is important to avoid recommendation fatigue and to provide users with a variety of choices.

4) ***User privacy:****:* Existing solutions needs to balance between collecting enough data to make accurate recommendations and protecting user privacy. Collecting too much data or sharing user data with third parties can lead to privacy concerns and may harm user trust.

5) ***Interpretability: :*** Users often want to know why a recommendation was made in order to understand the reasoning behind the suggestion. Existing solutions needs to develop recommendation models that are transparent and interpretable to build user trust and satisfaction.

6) ***Scalability: :*** As the number of users and items grows, Existing solutions needs to ensure that its recommendation system can scale efficiently and effectively to continue to provide accurate and timely recommendations to its users.

V. *Proposed Framework*

The objective of movie rating analysis is to improve the movie rating in the real-world applications to find user opinion to improve the movie programs. In this work, the dataset containing the movie dataset will be taken into consideration. The pre-processing will be applied into the dataset and the noisy and null value data will be removed from the dataset. After the data will be analyzed and visualized for further processing. The machine learning algorithm will be chosen to make the prediction. The dataset will be divided into two parts. The first part of dataset is 70A movie recommendation system is a type of machine learning application that recommends movies to users based on their preferences. The proposed framework for building a movie recommendation system includes several key steps.

The first step is data collection. This involves gathering data on movies, including their attributes such as genre, cast, director, plot, and rating. This data can be collected from various sources, such as movie databases, IMDB, or user reviews.

The next step is data preprocessing. This involves cleaning the data, removing any irrelevant or duplicate information,

and preparing it for use in the recommendation system. This may also involve feature engineering, which involves creating new features that can help improve the accuracy of the recommendation system.

The third step is model selection. There are various machine learning algorithms that can be used for building a movie recommendation system, including collaborative filtering, content-based filtering, and hybrid filtering. The choice of algorithm will depend on the type of data available and the specific requirements of the recommendation system.

Once the model has been selected, the next step is to train the model using the prepared data. This involves fitting the model to the data and optimizing its parameters to improve its performance.

Finally, the model is deployed and used to make recommendations to users. The recommendation system may be integrated into a web application or mobile app, where users can input their preferences and receive personalized movie recommendations. Ongoing monitoring and maintenance of the system may also be required to ensure that it continues to provide accurate recommendations over time.

A. Milestone Division

1) ***Data collection:*** *:* Gathering a large and diverse dataset of movies, along with relevant metadata such as ratings, reviews, genres, and release dates.

2) ***Data preprocessing:*** *:* Cleaning and organizing the data, removing duplicates and missing values, and transforming it into a format suitable for analysis and modeling.

3) Exploratory data analysis:: Exploring the dataset to identify patterns and relationships, and gain insights into user preferences, popular genres, and other relevant features.

4) ***Model selection:*** *:* Choosing an appropriate machine learning model or combination of models that can effectively predict user preferences based on the available features.

5) ***Implementation****::* Integrating the trained model(s) into a production system or application and testing their performance in real-world scenarios.

6) ***Monitoring and maintenance:*** *:* Monitoring the performance of the deployed system, updating the model and features as needed, and addressing any issues or bugs that arise.

B. Dataset:

Most of the columns in a dataset are noisy and contain lots of information. But with feature engineering do, will get more good results. The first step is to import the libraries and load data. After that will take a basic understanding of data like its shape, sample, is there are any NULL values present in the dataset. Understanding the data is an important step for prediction or any machine learning project. It is good that there are no NULL values. And we need little changes in weight and Ram column to convert them to numeric by removing the unit written after value.

C. Detailed Design of Features:

This movie dataset contains the fields needed for the analyzing of the movie ratings. Movie dataset with ratings are stored in a separate dataset file. Exploratory examination is a cycle to investigate and comprehend the information and information relationship in a total profundity with the goal that it makes highlight designing and machine learning demonstrating steps smooth and smoothed out for expectation. Exploratory examination assists with validating our presumptions or misleading.

D. Analysis of Movie Recommendation System:

It will begin from the principal segment and investigate every section and comprehend what influence it makes on the objective segment. At the necessary step, we will likewise perform preprocessing and include designing undertakings. The point in acting top to bottom exploratory examination is to get ready and clean information for better machine learning demonstrating to accomplish elite execution and summed up models. So it should begin with breaking down and setting up the dataset for expectation.

E. Modules

1) Dataset collection
2) Data cleaning
3) Exploratory Data Analysis
4) Machine learning Modeling
5) Report
: **1) Dataset collection:** The information about the movies with different types of attributes are collected. The dataset stores the information, rating of the movie.
2) Data Cleaning: The large dataset contains more noisy and improper data which have to be pre-processed to produce the quality dataset for further pruning. The data is cleaned and processed with initial stage of removing the null values.
3) Exploratory Data Analysis Exploratory analysis is a process to explore and understand the data and data relationship in a complete depth so that it makes feature engineering and machine learning modeling steps smooth and streamlined for prediction. EDA involves Univariate, Bivariate, or Multivariate analysis. EDA helps to prove our assumptions true or false. In other words, it helps to perform hypothesis testing.
4) Machine learning Modeling Machine learning modeling helps to find the best algorithm with the best hyper parameters to achieve maximum accuracy.the dataset is split into 2 variants. 70percent of records are taken as training data and used to train the machine learning algorithm. The remaining 30percent of dataset is applied to testing which helps to predict the process.
5) Report: The Data is visualized based on the output of the machine learning algorithm and the data is mapped with different types of graphs to analyze and visualize the exact data to the user for the prediction. Matplot libraries are implemented to map the results based on the user requirements.

F. Implementation

The dataset once loaded into the python Colab, the initial pre-processing is done to remove the noisy data. Must clean the dataset as this information may be fragmented and it can't be sent straightforwardly to the model. So will make a capability of cleaning which does the accompanying system to clean the information and returns the cleaned words:

• **Eliminate numbers**, Alphanumeric words for example words which contain the two letter sets and numbers for example hello123.

• **Pre-processing:** The noisy data, empty values in the cell are pre-processed. The columns which are not needed for the evaluation of the model also removed using the drop function in the python.

• **Data Visualization:** visualized results will be the result of this part.

• **Evaluation of the model**: The random forest classifier is applied in the prediction of the movie ratings.

• The dataset is divided into training and testing data. The splitted dataset is passed into the different machine learning algorithm models and the accuracy levels were found.

• The dataset containing of 30 percentage data are split for testing and remaining 70 percentage records are applied for the training.

• The Gaussian NB is applied, the train and test data are divided, and the accuracy report is taken.

• Implementation with Logistic Regression model:

• The Logistic Regression algorithm is applied in the evaluation of the movie rating system.

Elimination of numbers: Eliminate numbers, Alphanumeric words for example words which contain the two letter sets and numbers for example hello123 .

Pre-processing: The noisy data, empty values in the cell are pre-processed. The columns which are not needed for the evaluation of the model also removed using th drop function in the python

G. Evaluation of Model

The randomforest classifier is applied in the prediction of the movie ratings. The dataset is divided into training and testing data. The splitted dataset is passed into the different machine learning algorithm models and the accuracy levels were found. The dataset containing of 30percent data are split for testing and remaining 70percent records are applied for the training. The accuracy of the testing data are displayed with the precision, recall, f1-score, support values. The GaussianNB is applied, the train and test data are divided and the accuracy report is taken.

H. Implementation with Logistic Regression model:

The Logistic Regression algorithm is applied in the evaluation of the movie rating system. The similar movies corrleation are displayed based on condition number of ratings greater than 100.

VI. Data Description

First step have to import the necessary packages to the application:

- import numpy as np
- import pandas as pd
- import matplotlib.pyplot as plt
- import seaborn as sns

Next the dataset would be connected from the Google cola. Initially the dataset is uploaded into the Google colab folder. Then the python file should connect to the path from the Google colab folder. The information has an extremely straightforward design with elements. Each column is related with interesting details, and it shows the details. The label has been set with different label description features of: The data gathered has the following details

Title: Movie Title. Certificate: Movie certificate. Runtime in mins: movie running time. Genres: Movie Genres in the list. Genres Types: film type Theme Strategy: story. cast: actor names

A Dissipate plot is utilized when both the sections are mathematical and it responds to our inquiry in a superior manner. From the underneath plot we can infer that there is a relationship yet not a solid connection between the cost and size segment. The dictionary shows the records displayed with head values of first 5 records from the dataset.

To create a movie recommendation system using machine learning, a comprehensive dataset comprising information on movies, users, and their interactions is necessary. The dataset should consist of several fields, including movie ID, title, release year, genres, cast, director, rating, synopsis, user ID, age, gender, occupation, rating provided by the user, and timestamp.

The dataset should be representative of the target audience and should be large enough to capture a diverse range of movies and users. It is essential to ensure that the dataset is free of errors, such as missing values or inconsistent data, to produce precise and beneficial recommendations. After gathering and organizing the dataset, machine learning algorithms can be utilized to examine user preferences and behavior and recommend movies based on similar user behavior or preferences. A well-structured movie recommendation system can enhance user engagement, boost user retention, and provide a satisfying overall user experience on the platform.

VII. Results

The results of a movie recommendation system using machine learning will depend on several factors, such as the quality of the data, the choice of algorithm, and the accuracy of the model.

The success of the system can be measured by its ability to accurately predict the movies that users are likely to enjoy based on their past viewing history and other preferences.

Metrics such as precision, recall, and F1 score can be used to evaluate the performance of the system.

In general, a well-designed movie recommendation system using machine learning can provide accurate and personalized recommendations to users, which can lead to increased user engagement and satisfaction. Users may be more likely to continue using the application or service if they feel that the recommendations are relevant and useful.

However, it's important to note that no recommendation system is perfect, and there will always be some degree of error or bias in the recommendations. Additionally, the system may need to be updated or adjusted over time as user preferences and the movie database change.

Overall, the results of a movie recommendation system using machine learning can be quite promising, but it's important to carefully evaluate the performance of the system and continually improve it to ensure that it meets the needs and preferences of its users.

```
[ ] data.isnull().sum()

Title                0
Year                 0
Genres               0
Genres_Types         0
Certificate          0
Runtime_in_mins      0
Theme_Strategy       7
Cast                 0
Directors            0
Studio               0
ReleaseDate          0
Budget_in_crores     6
Box_office_in_crores 6
Rating               6
Verdict              6
No_of_Votes          0
dtype: int64
```

Fig. 1. checking null values

First the null values inside the each attribute are taken care and sum of null values in each column is found.

```
[ ] data['Rating'].fillna(data['Rating'].median(),inplace=True)
    data['Budget_in_crores'].fillna(data['Budget_in_crores'].median(),inplace=True)
    data['Box_office_in_crores'].fillna(data['Box_office_in_crores'].median(),inplace=True)
    data['Verdict'].fillna(data['Verdict'].mode().iloc[0],inplace=True)
    data['Theme_Strategy'].fillna(data['Theme_Strategy'].mode().iloc[0],inplace=True)
```

Fig. 2. removing null values

```
[ ] df =data[['Genres', 'Year']].groupby(['Genres', 'Year']).size().reset_index()
    p= df.set_index(['Genres', 'Year']).unstack(level=-1).plot(kind='bar', stacked=True, figsize=(5,5))
    a.set_xlabel('Genres', fontsize=10)
    a.set_ylabel('count', fontsize=10)
    a.legend(data['Year'].unique())
    plt.show()
```

Fig. 3. based on genre, year

The pre-processing is done by removing the null values by fillna() function for the columns having null values.

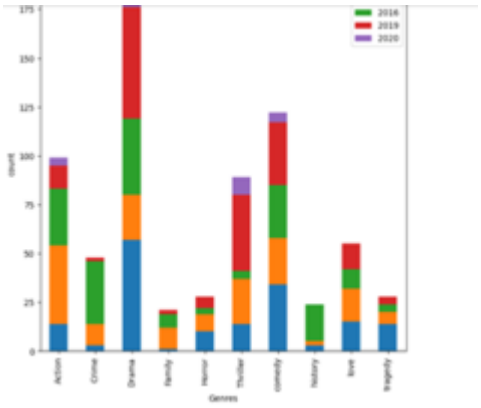

Fig. 4. genres, year

The attributes 'Genres','Year' are grouped and the bar chart is created to dispaly the contents.

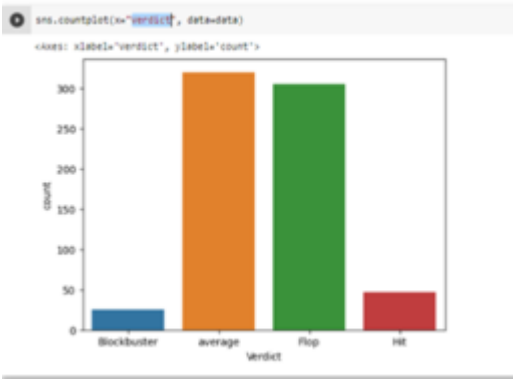

Fig. 5. bar chat representation of genres,year

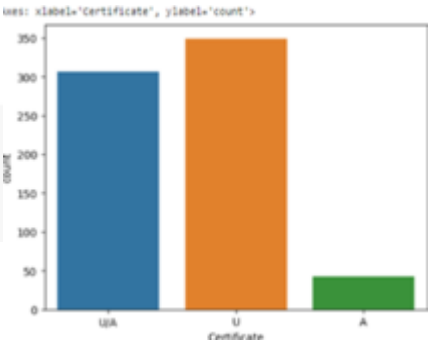

Fig. 6. bar chat of certification of movie result

REFERENCES

[1] Alexandra Fanca, Adela Puscasiu' Recommendation Systems with Machine Learning". 978-1-7281-1951-9/20/

[2] Nan Zhi-hong, Zhao Fei" Research on Semi-supervised Recommendation Algorithm Based on Hybrid Model

```
plt.title("Running time")
plt.xlabel("runtime of movies")
plt.ylabel("number of movies")
plt.hist(data["Runtime_in_mins"],bins= 30,color="y" )
plt.show()
```

Fig. 7. Movie Run time based

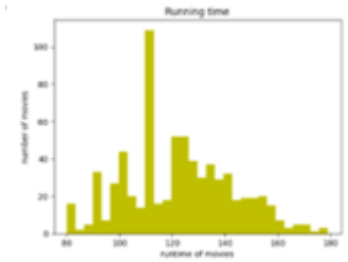

Fig. 8. run time chart

Fig. 9. random forest classifier

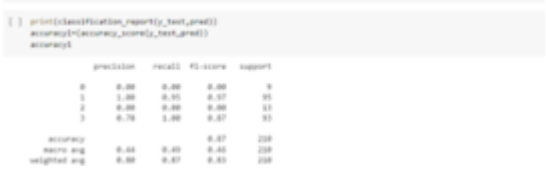

Fig. 10. The accuracy

Fig. 11. GaussianNB

```
print(classification_report(y_test,pred2))
accuracy2=accuracy_score(y_test,pred2)
accuracy2
```

	precision	recall	f1-score	support
0	1.00	0.33	0.50	9
1	0.64	0.98	0.77	95
2	0.20	0.08	0.11	13
3	0.79	0.48	0.60	93
accuracy			0.68	210
macro avg	0.66	0.47	0.50	210
weighted avg	0.70	0.68	0.64	210

0.6761904761904762

Fig. 12. Gaussian NB result

Fig. 13. Logistic Regression

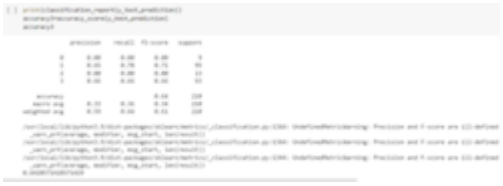

Fig. 14. Logistic Regression result

[3]]. Haili Wang, Nana Lou. "A Personalized Movie Recommendation System based on LSTM-CNN" , 2020 2nd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI).
[4] Gutiérrez a, Bobadilla J, Hernando a, Ortega F. Recommender systems survey. Knowledge Based System 2013; 46:109–32.
[5] Semeraro G, De Gemmis M, Lops P. Content-based recommender systems: state of the art and trends. Recommender System Handbook 2011:1–33.
[6] Billsus D, Pazzani MJ. Content-based recommendation systems 2007:325–41.
[7] W. Croft, T. Strohman, and D. Metzler. Search Engines: Information Retrieval in Practice. Addison Wesley, 2010.
[8] Zhang R, Wang Y, Bao H, Liu X, Sun H. Recommender systems based on ranking performance optimization. Front Computer Sci China 2015;1–11.
[9] Su X, Khoshgoftaar TM. A survey of collaborative filtering techniques. Adv Artificial Intelligence 2009; 2009:1–19.
[10] Zeng D, Huang Z, Chen H. A comparison of collaborative-filtering algorithms for e-commerce. IEEE Intelligent Systems 2007.
[11] Nikhil Rao, Inderjit S. Dhillon, Pradeep K. Ravikumar, Hsiang-Fu Yu. Collaborative filtering with graph information: Consistency and scalable methods. In C. Cortes, R. Garnett, N. D. Lawrence, M. Sugiyama, D. D. Lee, and editors, Advances in Neural Info Processing Syst 28, pages 2107–2115. Curran Associates, Inc., 2015.
[12] Kim BM. Clustering approach for hybrid recommender system. In: Proc IEEE/WIC Int Conf Web Intell (WI 2003). p. 33–8. [15] A. Y. Song, Elkahky, and X. He. A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems. In WWW, pages 278–288, 2015.
[13] https://www.nomidl.com/machine-learning
[14] https://colab.research.google.com/
[15] https://www.tutorialspoint.com
[16] https://www.codingforentrepreneurs.com/courses/python-google-colab-sheets-drive/
[17] https://www.researchgate.net/publication/
[18] https://ieeexplore.ieee.org/
[19] A Survey on Movie Recommendation Systems" by R. Arunkumar and R. Kavitha
[20] Netflix Recommendations: Beyond the 5 stars" by X. Amatriain, J. Lao, and N. Oliver
[21] Ma, H., Yang, H., Lyu, M.R. and King, I., 2011. Sorec: social recommendation using probabilistic matrix factorization. In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information