“Movie genre Classification”

A project Report supported to

MOHAN BABU UNIVERSITY

In partial fulfillment of the Requirements for the Award of the degree of

BACHELOR OFTECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

BY

TANIKONDA AVINASH	**(22102A041096)**
KORADA MANMOHANASAI	**(22102A040879)**
SURINENI NITHIN KRISHNA	**(22102A040262)**

Under the Guidance of

Dr. J. Avanija

Professor

Dept. of AIML, School of Computing, MBU

Department of Artificial Intelligence and Machine Learning

MOHANBABUUNIVERSITY

Sree Sainath Nagar, Tirupathi- 517 102

2022-2026

MOHAN BABU UNIVERSITY

Sree Sainath Nagar, A. Rangampet, Tirupati- 517102, Chittoor Dist., A.P.

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

CERTIFICATE

This is to certify that the Project Work entitled

"Movie Genre Classification"

is the bonafide work done by

TANIKONDA AVINASH	**(22102A041096)**
KORADA MANMOHANASAI	**(22102A040879)**
SURINENI NITHIN KRISHNA	**(22102A040262)**

In the Department of Artificial Intelligence and Machine learning, Mohan Babu University, A. Rangampet. In partial fulfillment of the requirements for the award of Bachelor of Technology in Artificial Intelligence and Machine Learning during 2022-2026.

This is work has been carried out under my guidance and supervision.

The results embodied in this Project report have not been submitted in any University or Organization for award of any degree or diploma.

Internal Guide

Dr. J. Avanija
Professor
Dept of AIML
Mohan Babu University
Tirupathi

Head

Dr. B. Narendra Kumar Rao
Prof & Head
Dept of AIML
Mohan Babu University
Tirupathi

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

VISION

To become a Centre of Excellence in Artificial Intelligence and Machine Learning by imparting high quality education through teaching, training and research.

MISSION

- To impart quality education in Computer Science and Engineering with specializations in Artificial Intelligence and Machine Learning by disseminating knowledge through contemporary curriculum, competent faculty and effective teaching learning methodologies.
- Nurture research, innovation and entrepreneurial skills among students and faculty to contribute to the needs of industry and society.
- Inculcate professional attitude, ethical and social responsibility for prospective and promising Engineering profession.
- Encourage students to engage in lifelong learning by creating awareness of the contemporary developments in Computer Science and Engineering Artificial Intelligence and Machine Learning.

Program Educational Objectives (PEO's)

After few years of graduation, the graduates of B. Tech CSE (Artificial Intelligence and Machine Learning) will:

PEO1. Pursuing higher studies in core or allied areas of Computer Science, Artificial Intelligence, Machine Learning, Data Science or Management.

PEO2. Employed in reputed Computer and I.T organizations or Government, to have a globally competent professional career in Computer Science and Engineering, Artificial Intelligence Machine Learning, Data Science domain or be successful entrepreneurs.

PEO3. Able to demonstrate effective communication, engage in teamwork, exhibit leadership skills, ethical attitude, and achieve professional advancement through continuing education.

Program Outcomes (PO's)

On successful completion of the Program, the graduates of B.Tech. CSE (Artificial Intelligence and Machine Learning) Program will be able to:

PO1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

PO2. Problem analysis: Identify, formulate, review research literature, analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Course Outcomes

1. Analyze the process of machine learning modeling and evaluation to automatically infer a general description for a given learning problem.
2. Design and implement machine learning solutions for classification, and regression problems.
3. Design and implement efficient neural architecture to model patterns for a given learning problems.
4. Analyze intelligent solutions to solve societal problems related to computer vision information security, healthcare and other areas.
5. Analyze and apply clustering techniques for effective data analysis and pattern recognition.
6. Work independently to solve problems with effective communication.

CO-PO Mapping

	PO1	PO2	P03	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	**3**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**-**
CO2	**2**	**3**	**3**	**3**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**-**
CO3	**3**	**3**	**3**	**1**	**-**	**-**	**-**	**-**	-	**-**	**-**	**-**	**-**	**-**	**3**	**-**
CO4	**1**	**3**	**3**	**3**	**3**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**-**
CO5	**2**	**3**	**3**	**3**	**3**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**-**
CO6	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**3**	**-**	**-**	**-**	**-**	**3**	**-**
Course Correlation Mapping	**3**	**3**	**3**	**3**	**3**	**3**	**-**	**-**	**3**	**3**	**-**	**-**	**-**	**-**	**3**	**-**

(Correlation Levels: 3: High; 2: Medium; 1: Low)

DECLARATION

We hereby declare that this project report titled **"Movie Genre Classification"** is a genuine project work carried out by us, in **B. Tech (Artificial Intelligence and Machine Learning)** degree course of **Mohan Babu University** and has not been submitted to any other course or university for the award of any degree by us.

Signature of the student

1.

2.

3.

ACKNOWLEDGEMENT

We are extremely thankful to our beloved chairman and founder **Dr. M. Mohan Babu** who took keen interest to provide us the infrastructural facilities for carrying out the project work.

We are highly indebted to

We are very much obliged to **Dr. B. Narendra Kumar Rao**, professor & Head Department of AIML, for providing us the guidance and encouragement in completion of this project.

We would like to express our indebtedness to the project coordinator, **Dr. J. Avanija**, Professor, Department of AIML for her valuable guidance during the course of project work.

We would like to express our deep sense of gratitude to **Avanija**, Professor, Department of AIML, for the constant support and invaluable guidance provided for the successful completion of the project.

We are also thankful to all the faculty members of AIML Department, who have cooperated in carrying out our project. We would like to thank our parents and friends who have extended their help and encouragement either directly or indirectly in completion of our project work.

Abstract

Creating a model to predict movie genres and proposing movies based on content abstracts requires a comprehensive approach that integrates natural language processing (NLP) techniques, image analysis, and content abstraction methods.

Develop a multi-model that combines features from both plot summaries and poster images to predict movie genres. Train the model using the prepared dataset, optimizing for genre predictions.

Evaluate the model's performance using metrices like accuracy, precision, recall, etc., on the test dataset. Analyse any misclassifications to understand areas for improvement.

Utilize the trained model to recommend movies based on user-specified genres. You can build a simple content-based recommendation system that suggests movies with similar plot summaries or genres.

Table of Contents

Chapter 1

1.1 Introduction

Movie genres are classification that explain about the films based on story or scenes. The film genre used to regulate the characters, story plot, movie structure. Example horror movies hold more action than dialogue, action movies hold fight scenes, high background music, high elevation and slow-motion camera shots.

Genre categorizes movies. This makes easier for viewer to discover movies based on genre he or she likes to watch. Genre is categorized based on 4 elements: story, plot, character, setting. Movies often have genres overlap, such as drama movies can also contain comedy or action movies can contains crime.

There is a specific style of filmmaking where a film possesses both the genre and subgenre. The main plot of the film is called genre and the side plot of the story is called sub-genre. Subgenres vary stylistically. For example, drama and comedy movies can also have some action part. But they both differ in story plot, dialogues, scenes.

1.2 Problem statement

To create a model that predicts the genre of a movie based on its plot summary or poster part and to propose movies based on stated genre and catalogue based on content.

1.3 Objectives

Model Interpretability: To understand how the model makes predictions by analysing feature importance or attention mechanisms.

To implement a recommendation system that suggests movies based on user-specified genres or preferences.

Model Training: Split your data set into training and testing sets to evaluate the performance of your model.

Content-Based Filtering: Incorporate other features such as movie ratings, release year, and cast.

Hybrid Approaches: combine content-based and collaborative filtering techniques for more personalized and accurate recommendations.

Chapter 2

S. SNO	Paper Title	Journal/conference published	Methods proposed	Datasets used	Limitations	links
1 1.	Poster-Based Multiple Movie Genre Classification Using Inter-Channel Features	IEEE Access, 2020	CNN model, convolution layers, Support vector system,	MOVIELENS, Kaggle movie data sets		https://ieeexplore.ieee.org/abstract/document/9057706
2.	Movie Popularity and Target Audience Prediction Using the Content-Based Recommender System	Journal IEEE ACCESS	1-D CNN deep learning model,	IMDb beta set, TMDb database		https://ieeexplore.ieee.org/abstract/document/9758691

3.	Evaluating folksonomy information sources for genre prediction	2014 IEEE International Advanced Computing Conference (IACC)	Decision support system, Multi-label, Threshold Method, Unsupervised Learning Techniques			https://ieeexplore.ieee.org/document/6779440
4.	Using Generative Adversarial Networks for Conditional Creation of Anime Posters	IEEE ACCESS	Advances in deep learning, Generative Adversarial Networks, Thematic Content, Image Segmentation	NETFLIX DATASET		https://ieeexplore.ieee.org/abstract/document/9887491

5.	Predicting Film Genres with Implicit Ideals	Institute for Intelligent Systems, University of Memphis, TN, USA	CNN, NLP	9,249 matched films and associated IMDB		https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2012.00565/full

Chapter 3

3.1 Description of data used:

The data set we have used is accessed from Kaggle which is used for our problem statement. As per our problem statement, we have 2 major datasets of plot summaries and another one of movie posters.

Data sets that we have used for movie plot based genre prediction are Kaggle movie train.csv and Kaggle movie test.csv . For training we have used train.csv data set and for testing we have test.csv . The data inside the trained dataset consists of id, text and genre whereas the data inside the tested data set consists of only id and text. The trained dataset contains more than 22000 entries and where as for testing dataset contains more than 5500 data entries. We have also got the good accuracy for the model we trained.

Data set that we have used for movie recommendation system is movies.csv. It consists of different data which is divided into various columns. The data inside dataset consists of index, budget, genres, homepage, id, keywords, original language, original title, overview, popularity, production companies, production country, release date, revenue, run time, spoken languages, status, tagline, title, vote count, cast, crew, director. This data set contains more than 4500 entries.

Data set that we have used for movie poster based genre prediction is movie poster.csv .It consists of different labels consists of various columns. The dataset consists of various poster images of different images of various movies more than 2000 entries.

3.2 Data preprocessing and Data cleaning:

Plot based Genre prediction

We have used nltk (Natural language Toolkit) algorithm for cleaning and preprocessing the data.

- Cleaning special character from the dialog/script.
- Converting the entire dialog/script into lower case.
- Tokenizing the dialog/script by words.
- Removing the stop words.
- Stemming the words.
- Joining the stemmed words.
- Creating a corpus.

Movie recommendation system

We have used matrix factorization techniques like Singular value decomposition (SVD), and can use CNN enhance text processing using lemmatization.

- Filled missing values in selected features (‘genre’, ‘keywords’, ‘fasline’, ‘cast’, ‘director’) with empty strings.
- Combine selected features into a single feature vector (‘combined features’) for each movie.
- Used Tfidfvectorizer to convert text data (‘combined-features’) into numerical feature vectors (‘features-vectors’).

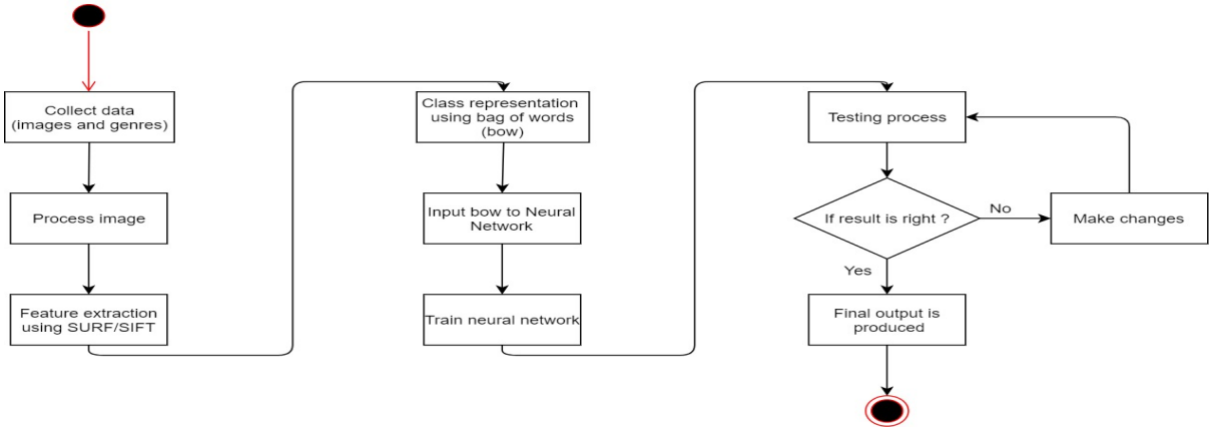
Chapter 4

4.1 Existing system:

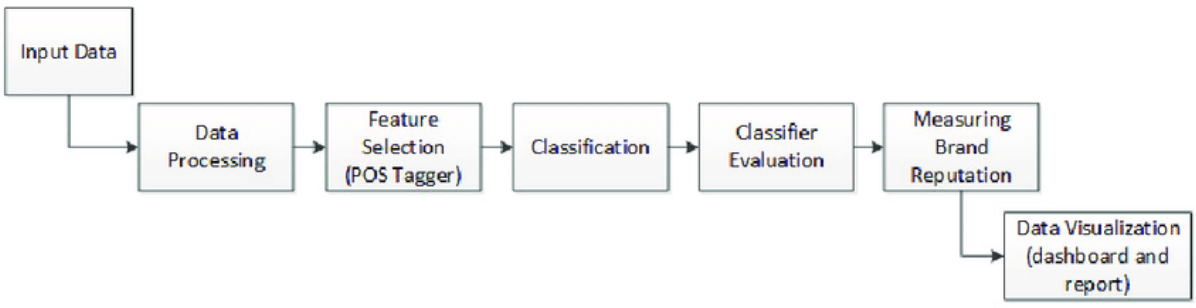
One existing system for project movie genre classification from poster images is to use deep learning models, such as convolutional neural networks (CNNs). A CNN architecture suitable for image classification tasks. Split the dataset into training, validation, and test sets. Train the CNN using the training set, optimizing the model's weights to minimize a chosen loss function (e.g., categorical cross-entropy) based on predictions compared to ground truth labels.

4.2.1 Architecture:

For genre classification from poster image:



For genre classification from plot summary:



4.2.2 Machine learning algorithms used:

Plot based genre recommendation:

We have NLTK (Natural Language Toolkit) algorithms along with train-test split for the model building. Multinomial Naïve bayes classifier for model training. NLTK is used for data cleaning to remove stemming words from the given plot in data base and to make the data case free we used this Natural Language Toolkit.

Traian-Test split is used to classify the dataset into training data and testing data.

Movie genre recommendation:

We have used embedded techniques like Wordtovec, Glove or fast test for converting data into numerical values. Beside cosine similarity we can use Jaccard similarity, Eucledian distance or Pearson correlation coefficient depending on nature of your data.

Wordtovec is used to create a distributed representation of words to numerical vectors. True context words can be distinguished from false context words.

Poster based genre recommendation:

We have used CNN and also Keras and along with Terasorflow.

CNN is used to analyse the poster, Keras is used to turn array of class integers into an array of one-hot vectors and Tensor flow is an open sourced end-to-end platform, a library for multiple machine learning tasks.

4.2.3 Model Selection and Evaluation metrics:

Create a word cloud visualization for drama, action and comedy to vizualise the most common used in each genre. To split the data set in test data set and train data set. Then model training is done using Multinomial Naïve Bayes Classifier.

Model Evaluation:

Prediction: Made predictions on the test set using trained classifier.

Accuracy calculation: calculated the accuracy of the model using ‘accuracy_score’.

Create a confusion Matrix using ‘confusion matrix’ and vizualize it using ‘sns.head map’.

Hyper perameter Tuning:

Tuned the alpha Hyper parameter of the Naïve Base Classifier to improve the performance of the model.

Chapter 5

5.1 Hardware and Software used:

We have used Dell 11th Gen Intel® Core™ i5-1135G7@ 2.40GHz 2.42GHz With installed RAM of 16 GB and used Google Colaboratory as our software of python version 3.6.9 for executing our code.

5.2 Parameter tuning process:

Hyper parameter is used for Multinomial naive bayes classifier for the better performance of the model.

Chapter6

Results and Discussion:

To find the best classification algorithms that can be used for the movie name prediction, we have used Systematic Literature Review method. We have gathered some works from the study that could help us to identify best classification algorithms to get good accuracy in prediction. The research articles and the identifications from the literature study are presented.

The results of the experiment with preprocessed dataset and the identified machine learning algorithms from literature study are presented in this section. Multinomial Naïve bayes classifier, TfidfVectorizer, cosine_similarity, CNN, Tensorflow and keras are used to train the model individually for the same dataset of movie scripts and the predictions of some random paraphrased sentences from the script are done for each model and recorded.

Research papers from the literature study which are obtained by using the search of keywords such as Natural Language Processing, classification algorithms, movie prediction in repositories like Google Scholar, IEEE, Science Direct. All the results related to our research are noted down which includes the use of classification in different prediction models, movie-related models that use machine learning techniques, and NLP-based articles that are used in the preprocessing of data. We have observed that there are several machine algorithms such as Naive Bayes, Decision Tree, Support Vector Machine, K- Nearest Neighbor classifier, Neural Network, Logistic Regression, Random Forest, Ada Boost, Gradient Boost, and some more algorithms are mostly used machine learning algorithms in the classification models. From the study of all the conclusions of research papers, we have opted to use 4 algorithms Multinomial Naïve bayes classifier, TfidfVectorizer, cosine_similarity, CNN in the prediction of movie genre using movie plot, poster image, classification of movie genre data to obtain good prediction accuracy.

Chapter 7

Conclusion and Future Scope:

In this thesis, we have implemented a model that predicts the movie genre when the random scenario from the movie script is given as input, predicts the movie genre from the poster image, and also classifies the movies. We have used a Systematic Literature Review to identify the suitable classification algorithms for the prediction of movie genre using movie plot, movie poster image and to classify the genres. As a result, we choose algorithms such as Multinomial Naïve bayes classifier, TfidfVectorizer, cosine_similarity, CNN, Tensorflow and keras . To obtain the accuracy of prediction, we have trained the model with 22,000 All the models are tested with random paraphrased sentences from the script, 7,500 posters and the results are noted. The accuracy of each model is calculated to compare with the remaining models. We can conclude that the model trained using RELU, Batch Normalisation has shown good performance with an accuracy of 91% in the prediction of movie genre, when compared to remaining models.

For future work, we can use ensemble methods to obtain more accurate and use a large dataset of movie plots, movie posters and multi-classification algorithms can be used to predict all the movie genre that have similar plots in their movie, when a random plot is given as input. In addition to that, we can use more advanced NLP techniques for text preprocessing that could help in efficient classification.

Chapter 8

References:

1. W.-T. Chu and H.-J. Guo, "Movie genre classification based on poster images with deep neural networks", *Proc. Workshop Multimodal Understand. Social Affect. Subjective Attributes*, pp. 39-45, 2017.

2. J.-M. Perez-Rua, V. Vielzeuf, S. Pateux, M. Baccouche and F. Jurie, "MFAS: Multimodal fusion architecture search", *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 6966-6975, Jun. 2019.

3. Ali, K., Van Stam, W.: Tivo: making show recommendations using a distributed collaborative filtering architecture. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 394–401. ACM, New York (2004)

4. S. R. S. Reddy, S. Nalluri, S. Kunisetti, S. Ashok and B. Venkatesh, "Content-based movie recommendation system using genre correlation" in Smart Intelligent Computing and Applications, Singapore:Springer, pp. 391-397, 2019.

5. Antol S, Agrawal A, Lu J, Mitchell M, Batra D, Lawrence Zitnick C, Parikh D (2015) Vqa: visual question answering. In: Proceedings of the IEEE international conference on computer vision, pp 2425–2433

6. Y. Hati, G. Jouet, F. Rousseaux and C. Duhart, "PaintsTorch: a User-Guided Anime Line Art Colorization Tool with Double Generator Conditional Adversarial Network", *European Conference on Visual Media Production*, pp. 1-10, Dec. 2019.

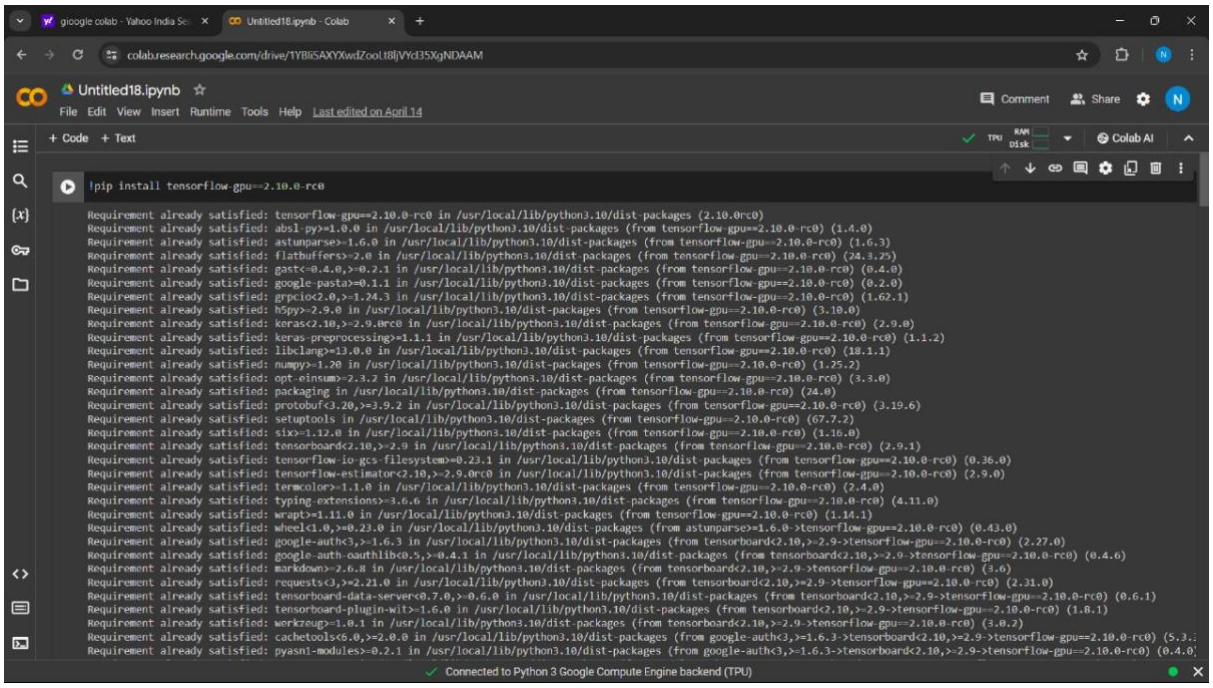
7. Griffiths, T. L., and Steyvers, M. (2002). “A probabilistic approach to semantic representation,” in *Proceedings of the 24th Annual Conference of the Cognitive Science Society*, eds W. D. Gray and C. D. Schunn (Hillsdale: Lawrence Erlbaum Associates), 381–386.

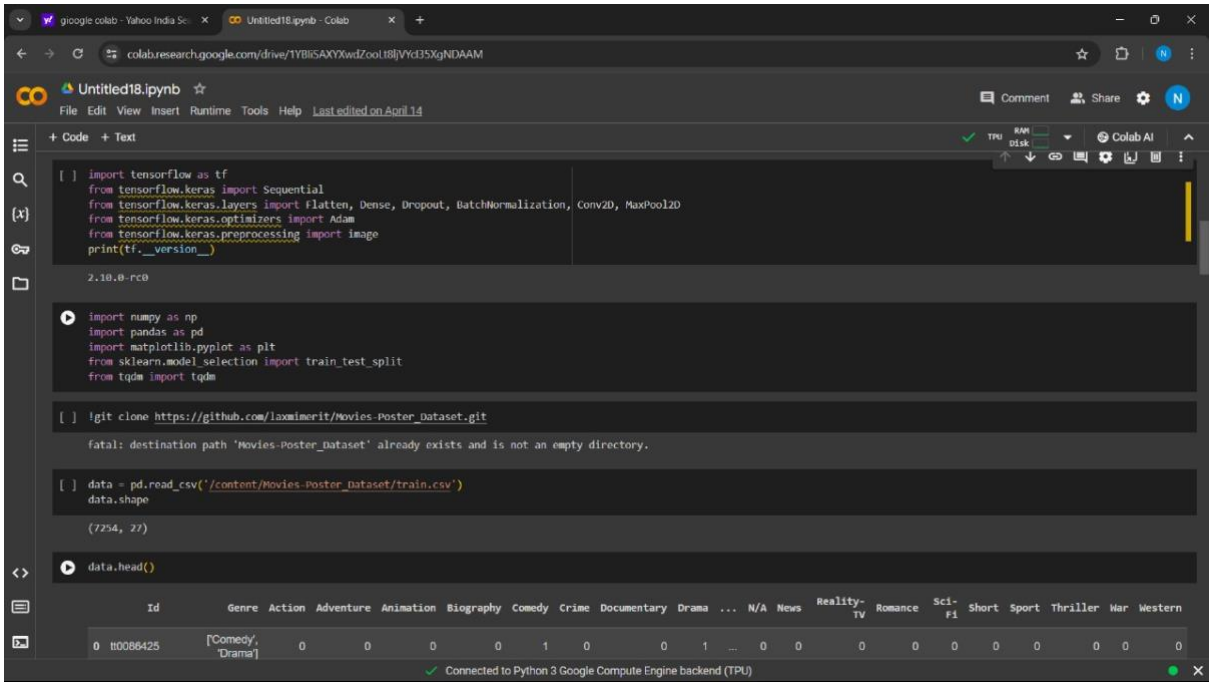
8. RiyahiM. *et al.*
Providing effective recommendations in discussion groups using a new hybrid recommender system based on implicit rating and semantic similarity
Electron. Commer. Res. Appl.
(2020)

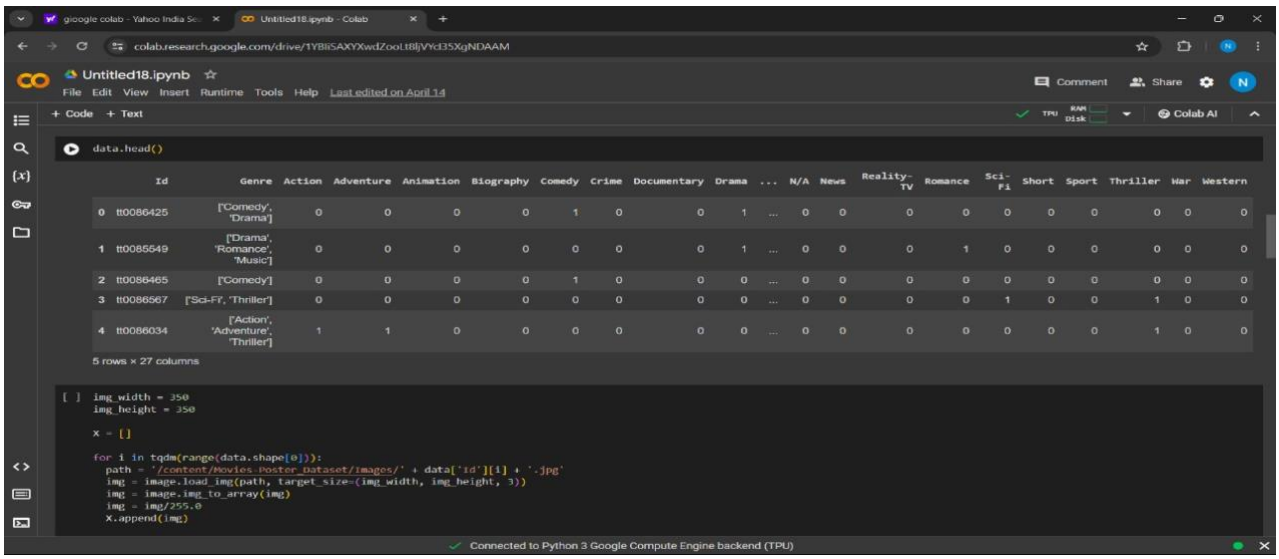
Chapter 9

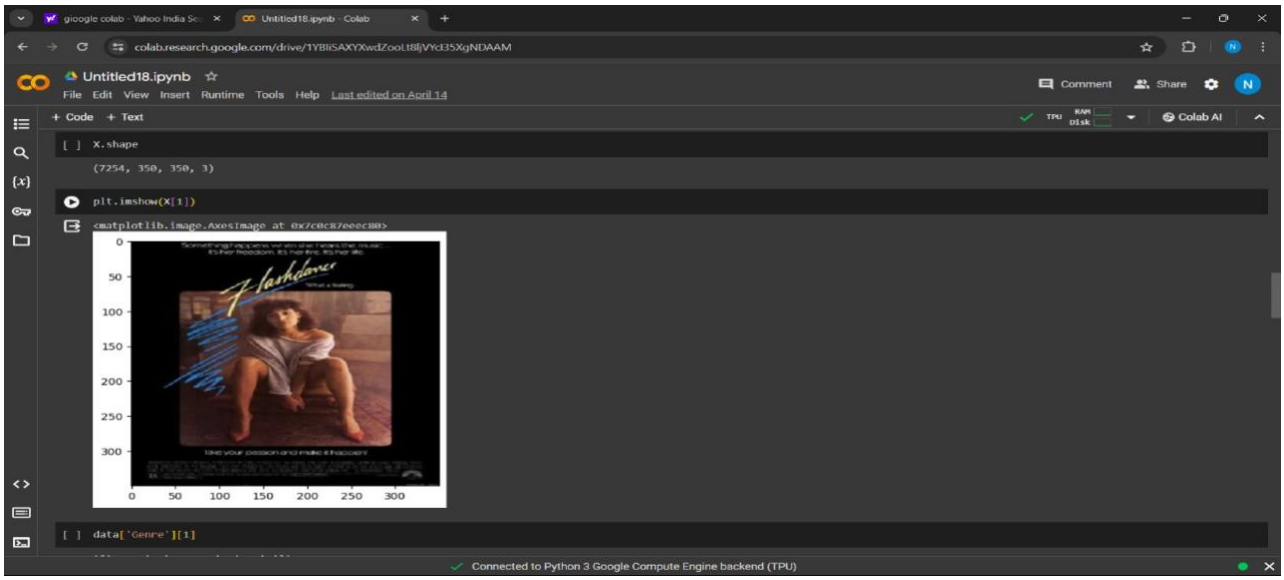
Appendix:

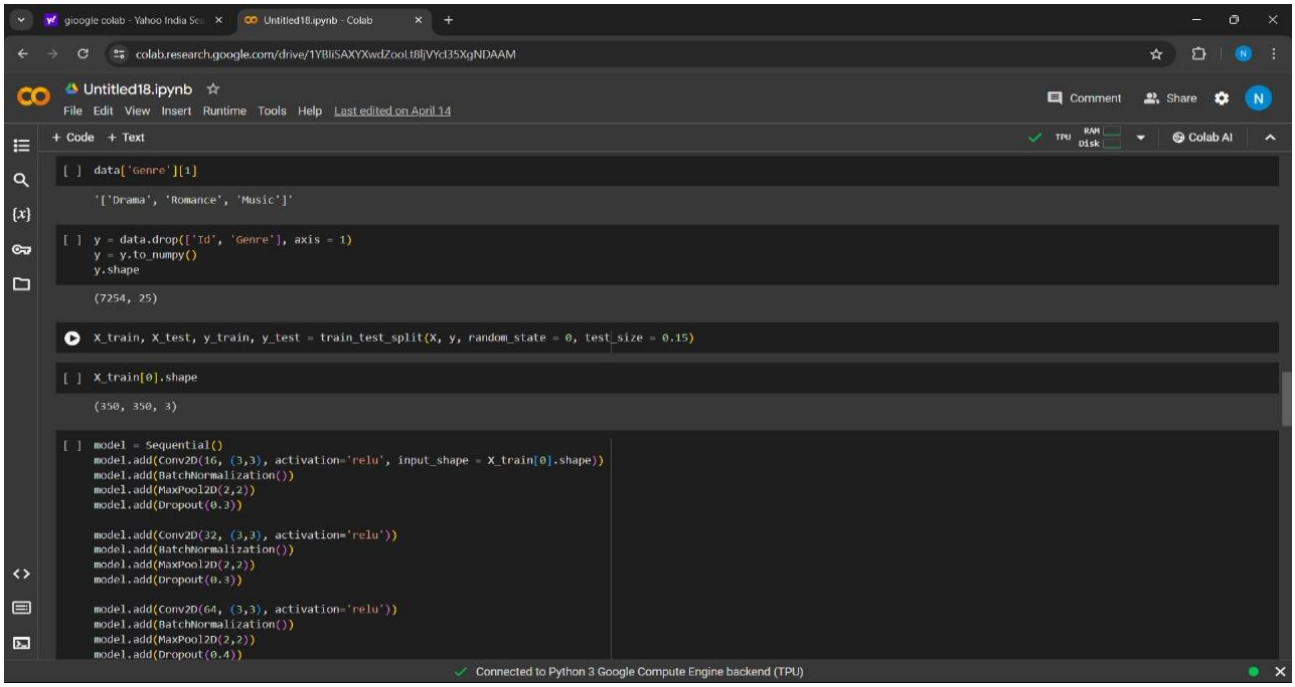
poster classification

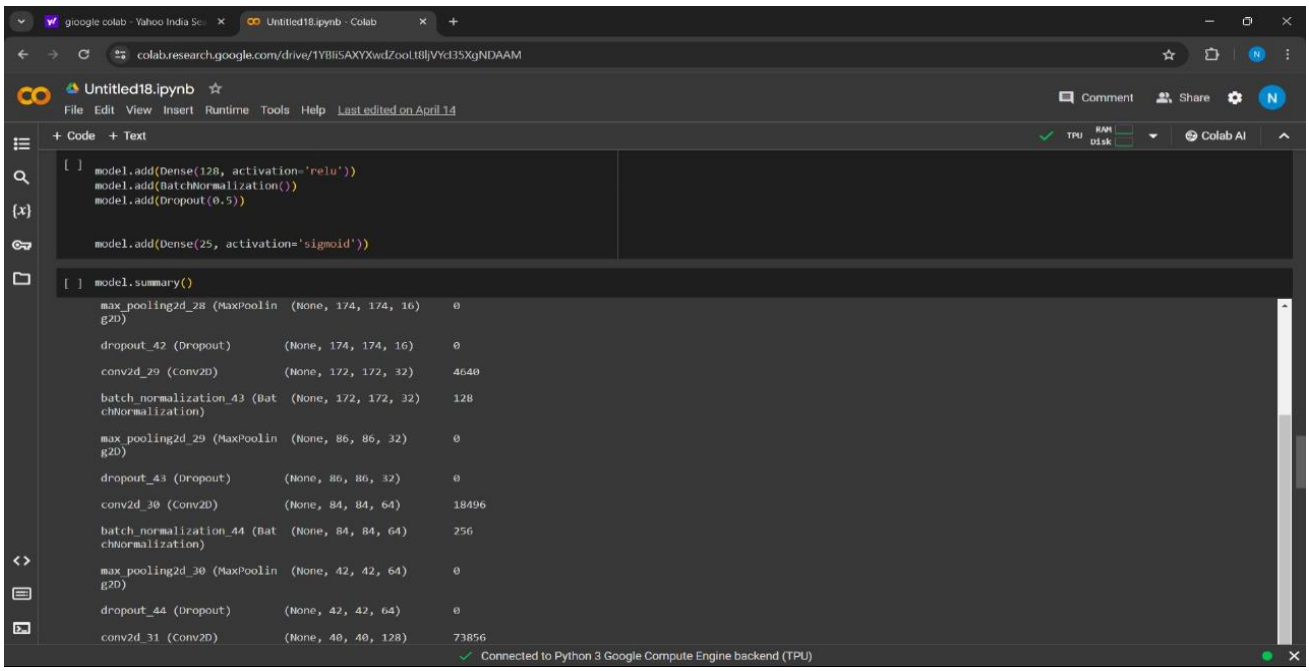


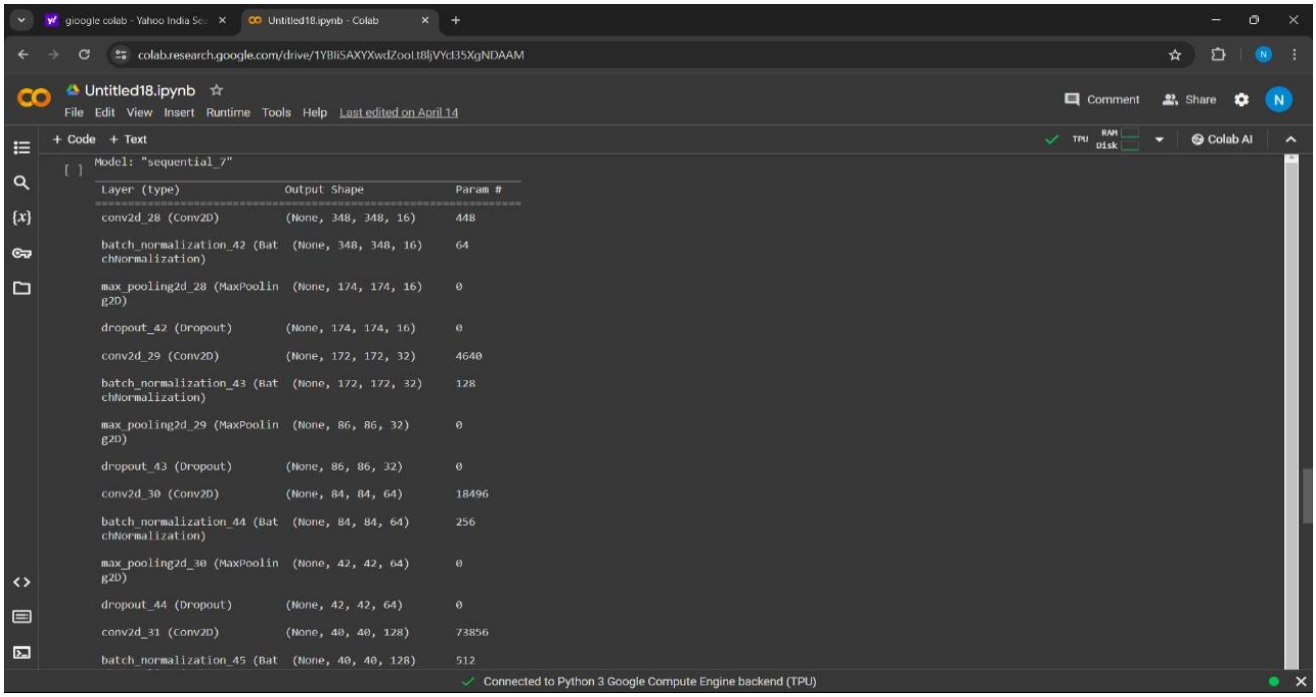


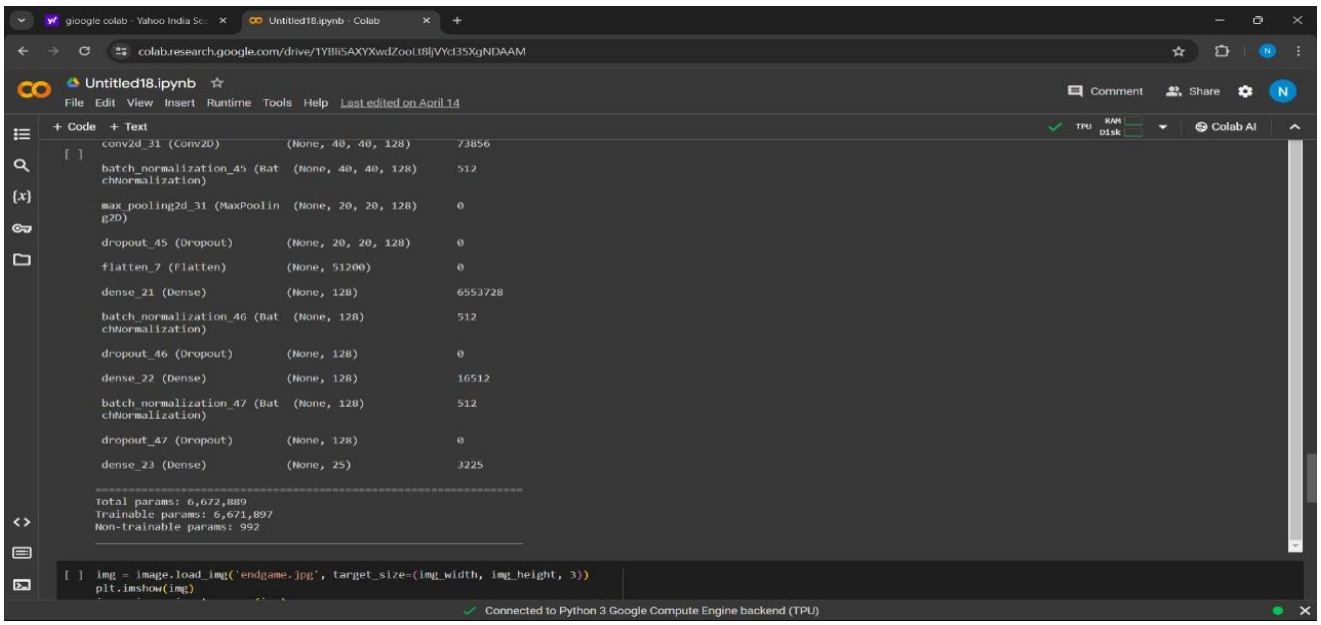

Untitled18.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 14
+ Code + Text
data.head()
Id Genre Action Adventure Animation Biography Comedy Crime Documentary Drama ... N/A News Reality-TV Romance Sci-Fi Short Sport Thriller War Western
0 tt0086425 ['Comedy', 'Drama'] 0 0 0 0 1 0 0 1 ... 0 0 0 0 0 0 0 0 0 0
1 tt0085549 ['Drama', 'Romance', 'Music'] 0 0 0 0 0 0 0 1 ... 0 0 0 1 0 0 0 0 0 0
2 tt0086465 ['Comedy'] 0 0 0 0 1 0 0 0 ... 0 0 0 0 0 0 0 0 0 0
3 tt0086567 ['Sci-Fi', 'Thriller'] 0 0 0 0 0 0 0 0 ... 0 0 0 0 1 0 0 1 0 0
4 tt0086034 ['Action', 'Adventure', 'Thriller'] 1 1 0 0 0 0 0 0 ... 0 0 0 0 0 0 0 1 0 0
5 rows × 27 columns
img_width = 350
img_height = 350
x = []
for i in tqdm(range(data.shape[0])):
path = '/content/movies_poster_dataset/images/' + data['Id'][i] + '.jpg'
img = image.load_img(path, target_size=(img_width, img_height, 3))
img = image.img_to_array(img)
img = img/255.0
x.append(img)
Connected to Python 3 Google Compute Engine backend (TPU)

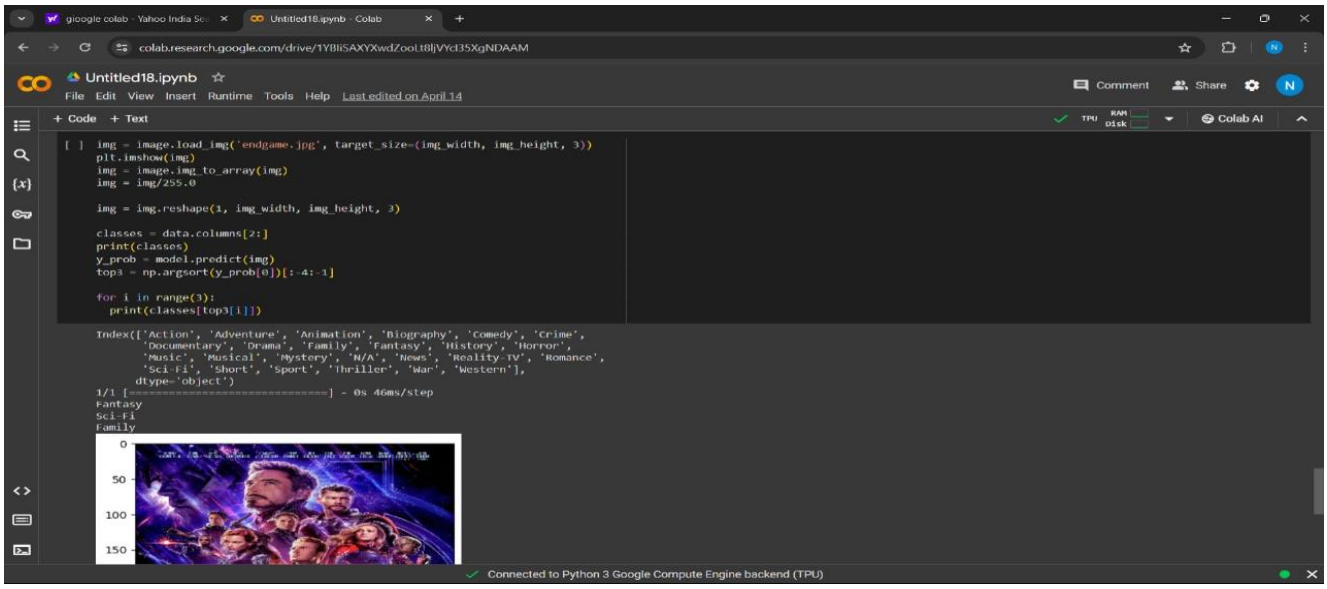

Untitled18.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 14
+ Code + Text
X.shape
(7254, 350, 350, 3)
plt.imshow(X[1])
<matplotlib.image.AxesImage at 0x7c0c87eeec80>
data['Genre'][1]
Connected to Python 3 Google Compute Engine backend (TPU)

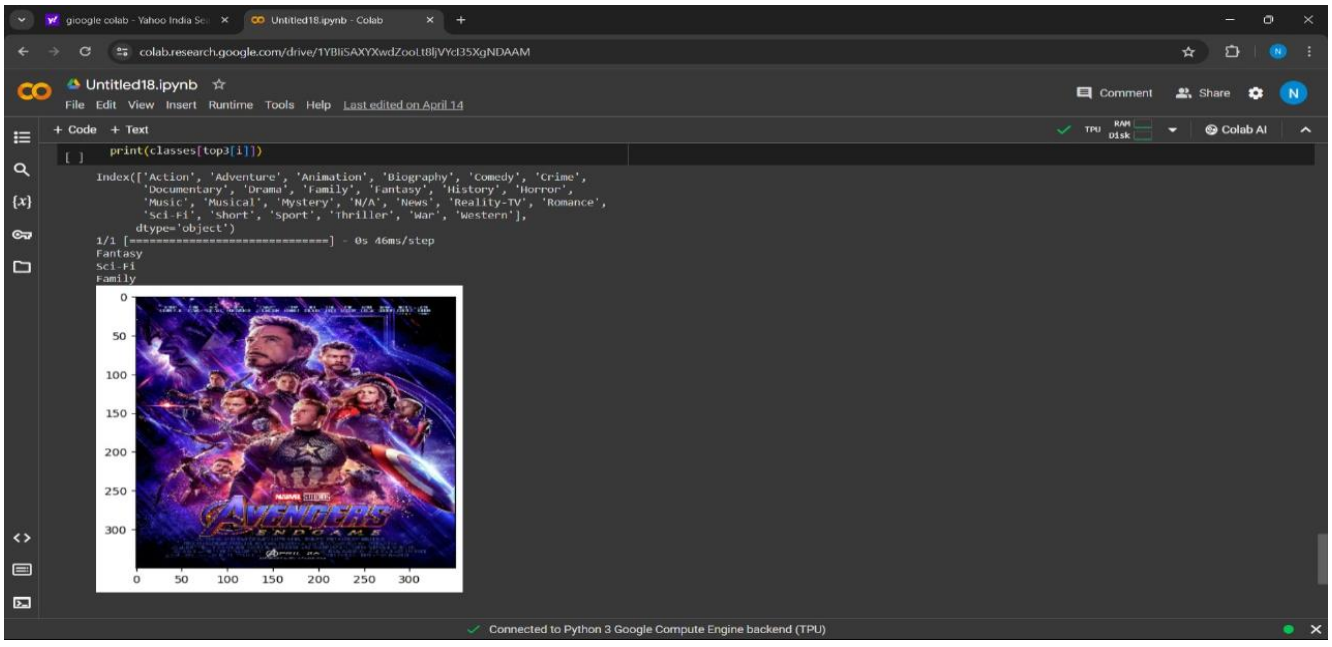

Untitled18.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 14
+ Code + Text
data['Genre'][1]
'['Drama', 'Romance', 'Music']'
y = data.drop(['Id', 'Genre'], axis = 1)
y = y.to_numpy()
y.shape
(7254, 25)
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state = 0, test_size = 0.15)
X_train[0].shape
(350, 350, 3)
model = Sequential()
model.add(Conv2D(16, (3,3), activation='relu', input_shape = X_train[0].shape))
model.add(BatchNormalization())
model.add(MaxPool2D(2,2))
model.add(Dropout(0.3))
model.add(Conv2D(32, (3,3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPool2D(2,2))
model.add(Dropout(0.3))
model.add(Conv2D(64, (3,3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPool2D(2,2))
model.add(Dropout(0.4))
Connected to Python 3 Google Compute Engine backend (TPU)


Untitled18.ipynb - Colab
colab.research.google.com/drive/1YBliSAXYXwdZooLt8ljVYcl35XgNDAAM
Untitled18.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 14
Comment
Share
+ Code + Text
Colab AI
model.add(Dense(128, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Dense(25, activation='sigmoid'))
model.summary()
max_pooling2d_28 (MaxPoolin g2D) (None, 174, 174, 16) 0
dropout_42 (Dropout) (None, 174, 174, 16) 0
conv2d_29 (Conv2D) (None, 172, 172, 32) 4640
batch_normalization_43 (Bat chNormalization) (None, 172, 172, 32) 128
max_pooling2d_29 (MaxPoolin g2D) (None, 86, 86, 32) 0
dropout_43 (Dropout) (None, 86, 86, 32) 0
conv2d_30 (Conv2D) (None, 84, 84, 64) 18496
batch_normalization_44 (Bat chNormalization) (None, 84, 84, 64) 256
max_pooling2d_30 (MaxPoolin g2D) (None, 42, 42, 64) 0
dropout_44 (Dropout) (None, 42, 42, 64) 0
conv2d_31 (Conv2D) (None, 40, 40, 128) 73856
Connected to Python 3 Google Compute Engine backend (TPU)

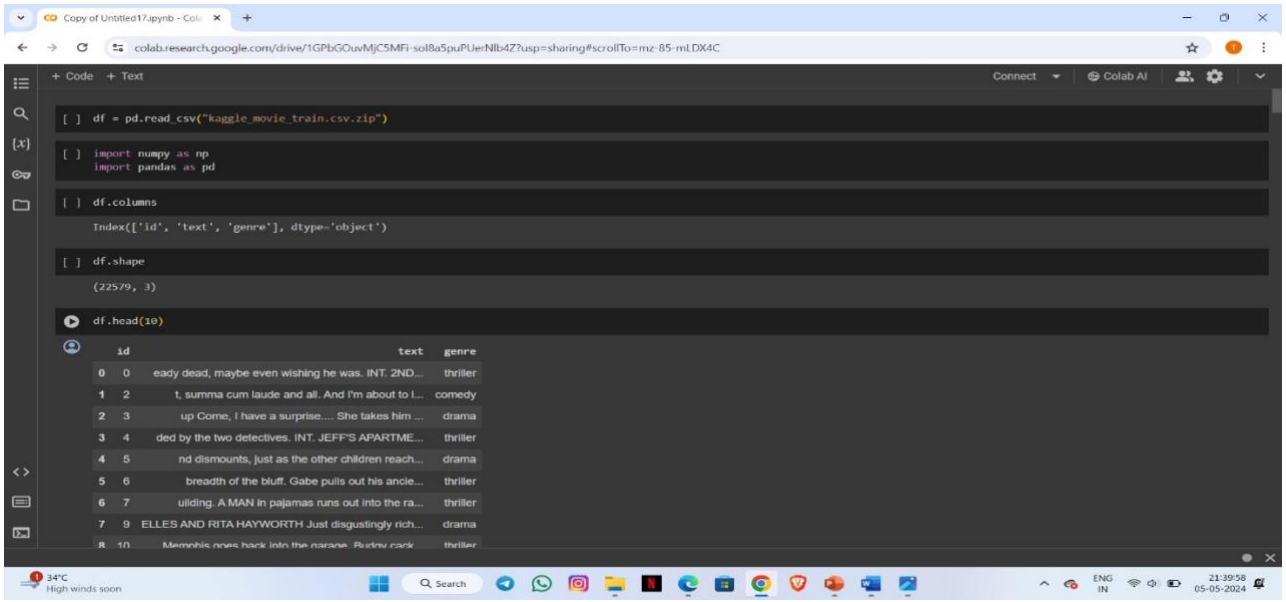

Untitled18.ipynb - Colab
colab.research.google.com/drive/1YBliSAXYXwdZooLt8ljVYcl35XgNDAAM
Untitled18.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 14
Comment
Share
+ Code + Text
Colab AI
Model: "sequential_7"
Layer (type) Output Shape Param #
conv2d_28 (Conv2D) (None, 348, 348, 16) 448
batch_normalization_42 (Bat chNormalization) (None, 348, 348, 16) 64
max_pooling2d_28 (MaxPoolin g2D) (None, 174, 174, 16) 0
dropout_42 (Dropout) (None, 174, 174, 16) 0
conv2d_29 (Conv2D) (None, 172, 172, 32) 4640
batch_normalization_43 (Bat chNormalization) (None, 172, 172, 32) 128
max_pooling2d_29 (MaxPoolin g2D) (None, 86, 86, 32) 0
dropout_43 (Dropout) (None, 86, 86, 32) 0
conv2d_30 (Conv2D) (None, 84, 84, 64) 18496
batch_normalization_44 (Bat chNormalization) (None, 84, 84, 64) 256
max_pooling2d_30 (MaxPoolin g2D) (None, 42, 42, 64) 0
dropout_44 (Dropout) (None, 42, 42, 64) 0
conv2d_31 (Conv2D) (None, 40, 40, 128) 73856
batch_normalization_45 (Bat (None, 40, 40, 128) 512
Connected to Python 3 Google Compute Engine backend (TPU)

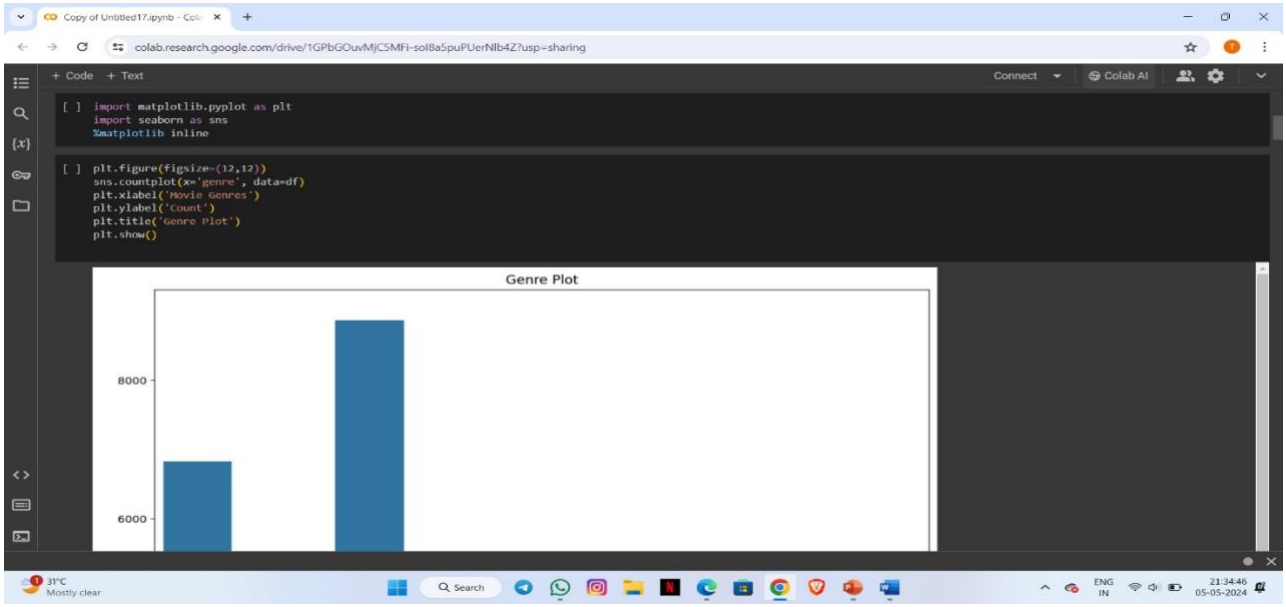

Untitled18.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 14
+ Code + Text
conv2d_31 (Conv2D) (None, 40, 40, 128) 73856
batch_normalization_45 (Bat chNormalization) (None, 40, 40, 128) 512
max_pooling2d_31 (MaxPoolin g2D) (None, 20, 20, 128) 0
dropout_45 (Dropout) (None, 20, 20, 128) 0
flatten_7 (Flatten) (None, 51200) 0
dense_21 (Dense) (None, 128) 6553728
batch_normalization_46 (Bat chNormalization) (None, 128) 512
dropout_46 (Dropout) (None, 128) 0
dense_22 (Dense) (None, 128) 16512
batch_normalization_47 (Bat chNormalization) (None, 128) 512
dropout_47 (Dropout) (None, 128) 0
dense_23 (Dense) (None, 25) 3225
Total params: 6,672,889
Trainable params: 6,671,897
Non-trainable params: 992
img = image.load_img('endgame.jpg', target_size=(img_width, img_height, 3))
plt.imshow(img)
Connected to Python 3 Google Compute Engine backend (TPU)

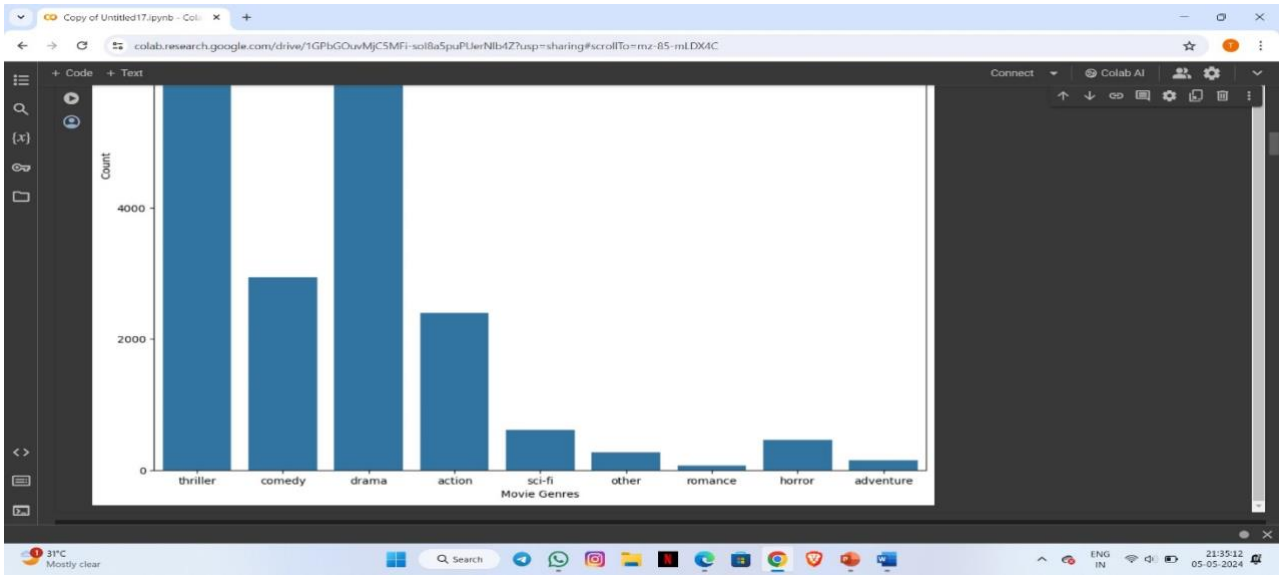

Untitled18.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 14
+ Code + Text
img = image.load_img('endgame.jpg', target_size=(img_width, img_height, 3))
plt.imshow(img)
img = image.img_to_array(img)
img = img/255.0
img = img.reshape(1, img_width, img_height, 3)
classes = data.columns[2:]
print(classes)
y_prob = model.predict(img)
top3 = np.argsort(y_prob[0])[:-4:-1]
for i in range(3):
print(classes[top3[i]])
Index(['Action', 'Adventure', 'Animation', 'Biography', 'Comedy', 'Crime',
'Documentary', 'Drama', 'Family', 'Fantasy', 'History', 'Horror',
'Music', 'Musical', 'Mystery', 'N/A', 'News', 'Reality-TV', 'Romance',
'Sci-Fi', 'Short', 'Sport', 'Thriller', 'War', 'Western'],
dtype='object')
1/1 [==============================] - 0s 46ms/step
Fantasy
Sci-Fi
Family
Connected to Python 3 Google Compute Engine backend (TPU)

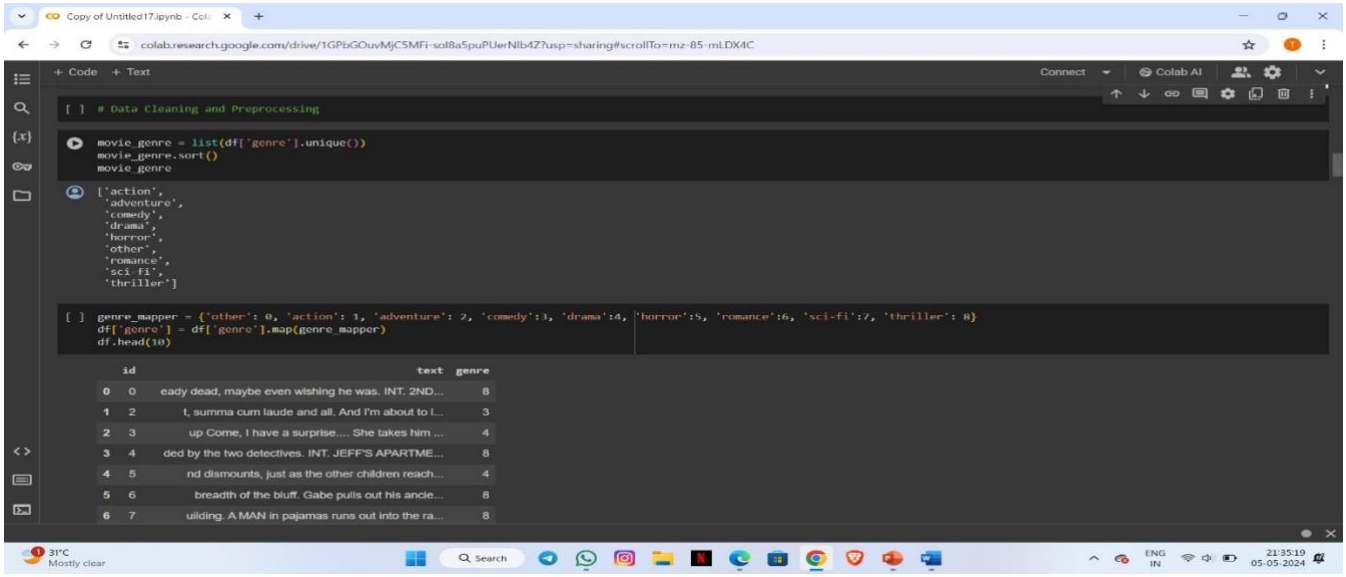

Untitled18.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 14
+ Code + Text
print(classes[top3[i]])
Index(['Action', 'Adventure', 'Animation', 'Biography', 'Comedy', 'Crime',
'Documentary', 'Drama', 'Family', 'Fantasy', 'History', 'Horror',
'Music', 'Musical', 'Mystery', 'N/A', 'News', 'Reality-TV', 'Romance',
'Sci-Fi', 'Short', 'Sport', 'Thriller', 'War', 'Western'],
dtype='object')
1/1 [==============================] - 0s 46ms/step
Fantasy
Sci-Fi
Family
Connected to Python 3 Google Compute Engine backend (TPU)

PLOT BASED


Copy of Untitled17.ipynb - Colab
+ Code + Text
Connect
Colab AI
[ ] df = pd.read_csv("kaggle_movie_train.csv.zip")
[ ] import numpy as np
import pandas as pd
[ ] df.columns
Index(['id', 'text', 'genre'], dtype='object')
[ ] df.shape
(22579, 3)
df.head(10)
id text genre
0 0 eady dead, maybe even wishing he was. INT. 2ND... thriller
1 2 t, summa cum laude and all. And I'm about to l... comedy
2 3 up Come, I have a surprise.... She takes him ... drama
3 4 ded by the two detectives. INT. JEFF'S APARTME... thriller
4 5 nd dismounts, just as the other children reach... drama
5 6 breadth of the bluff. Gabe pulls out his ancie... thriller
6 7 ulding. A MAN in pajamas runs out into the ra... thriller
7 9 ELLES AND RITA HAYWORTH Just disgustingly rich... drama
34°C
High winds soon
Search
ENG
IN
21:39:58
05-05-2024

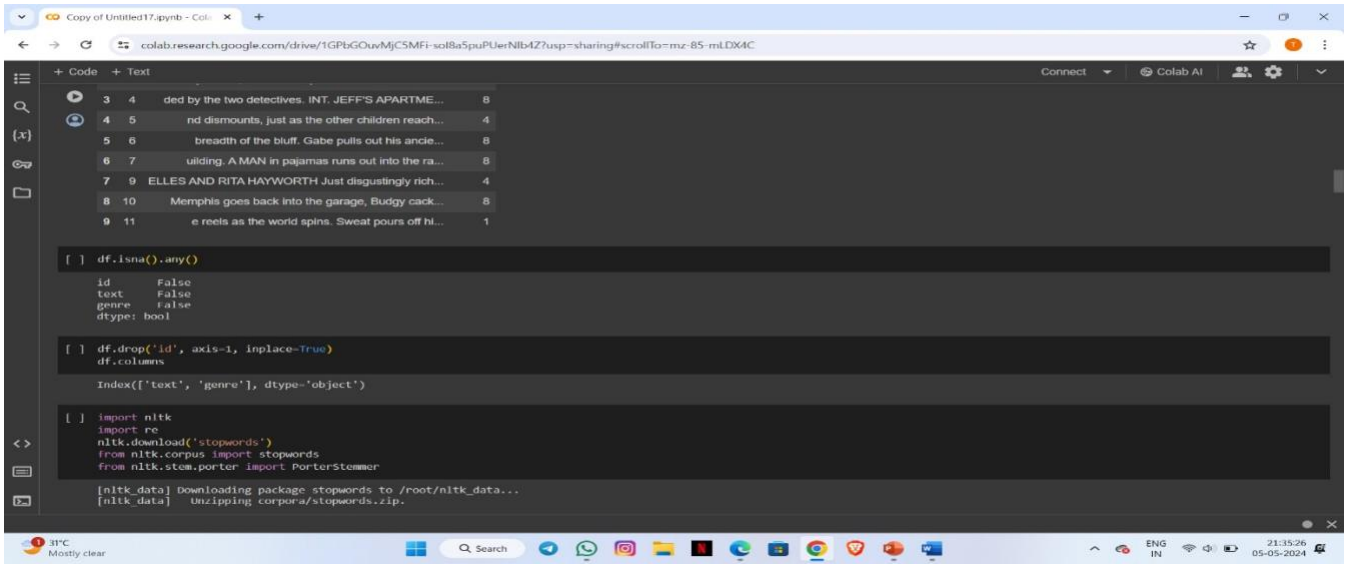

Copy of Untitled17.ipynb - Colab
colab.research.google.com/drive/1GPbGOuvMjC5MFi-sol8a5puPUerNlb4Z?usp=sharing
+ Code + Text
Connect
Colab AI
[ ] import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
[ ] plt.figure(figsize=(12,12))
sns.countplot(x='genre', data=df)
plt.xlabel('Movie Genres')
plt.ylabel('Count')
plt.title('Genre Plot')
plt.show()
Genre Plot
8000
6000
31°C
Mostly clear
Search
ENG
IN
21:34:46
05-05-2024


Copy of Untitled17.ipynb - Colab
colab.research.google.com/drive/1GPbGOuvMjC5MFi-sol8a5puPUerNlb4Z?usp=sharing#scrollTo=mz-85-mLDX4C
+ Code + Text
Connect
Colab AI
Count
4000
2000
0
thriller
comedy
drama
action
sci-fi
other
romance
horror
adventure
Movie Genres
31°C
Mostly clear
Search
ENG
IN
21:35:12
05-05-2024



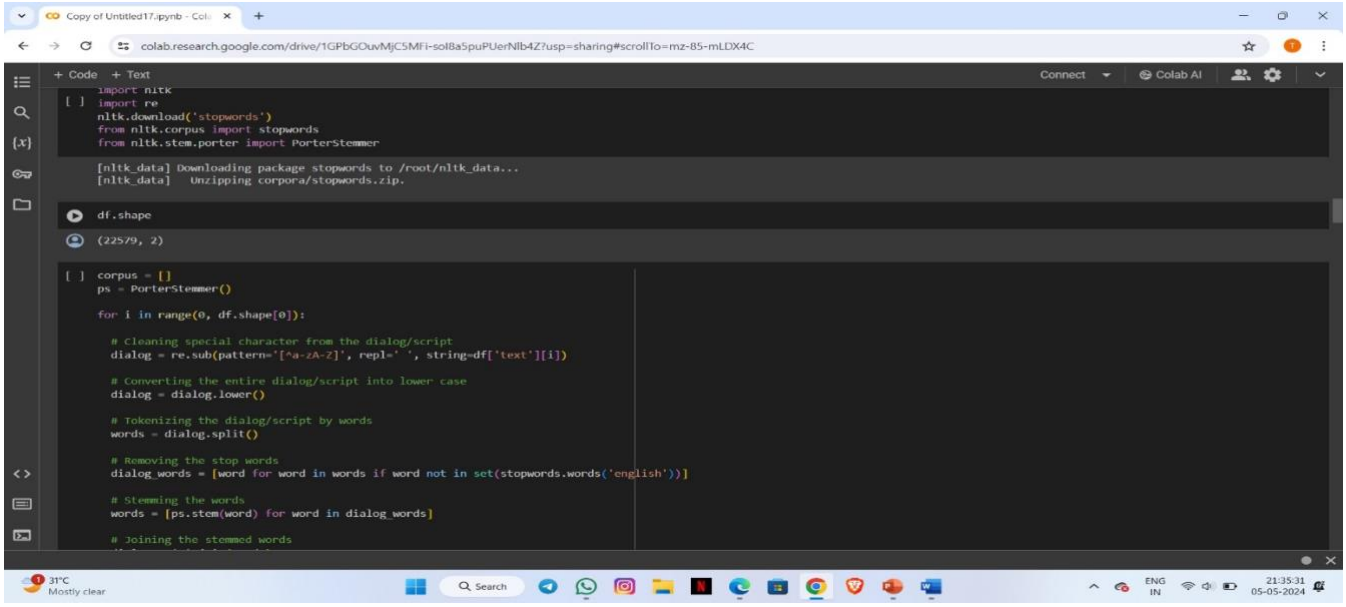
```
# Data Cleaning and Preprocessing

movie_genre = list(df['genre'].unique())
movie_genre.sort()
movie_genre

['action',
 'adventure',
 'comedy',
 'drama',
 'horror',
 'other',
 'romance',
 'sci-fi',
 'thriller']

genre_mapper = {'other': 0, 'action': 1, 'adventure': 2, 'comedy':3, 'drama':4, 'horror':5, 'romance':6, 'sci-fi':7, 'thriller': 8}
df['genre'] = df['genre'].map(genre_mapper)
df.head(10)
```

	id	text	genre
0	0	eady dead, maybe even wishing he was. INT. 2ND...	8
1	2	t, summa cum laude and all. And I'm about to l...	3
2	3	up Come, I have a surprise.... She takes him ...	4
3	4	ded by the two detectives. INT. JEFF'S APARTME...	8
4	5	nd dismounts, just as the other children reach...	4
5	6	breadth of the bluff. Gabe pulls out his ancie...	8
6	7	uilding. A MAN in pajamas runs out into the ra...	8

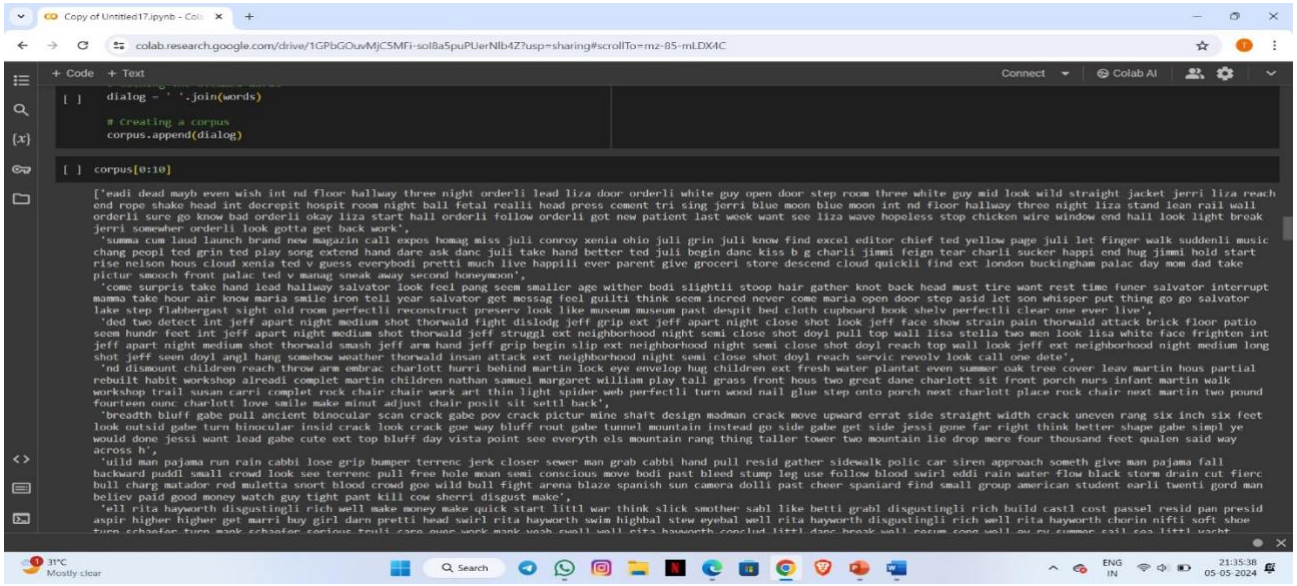


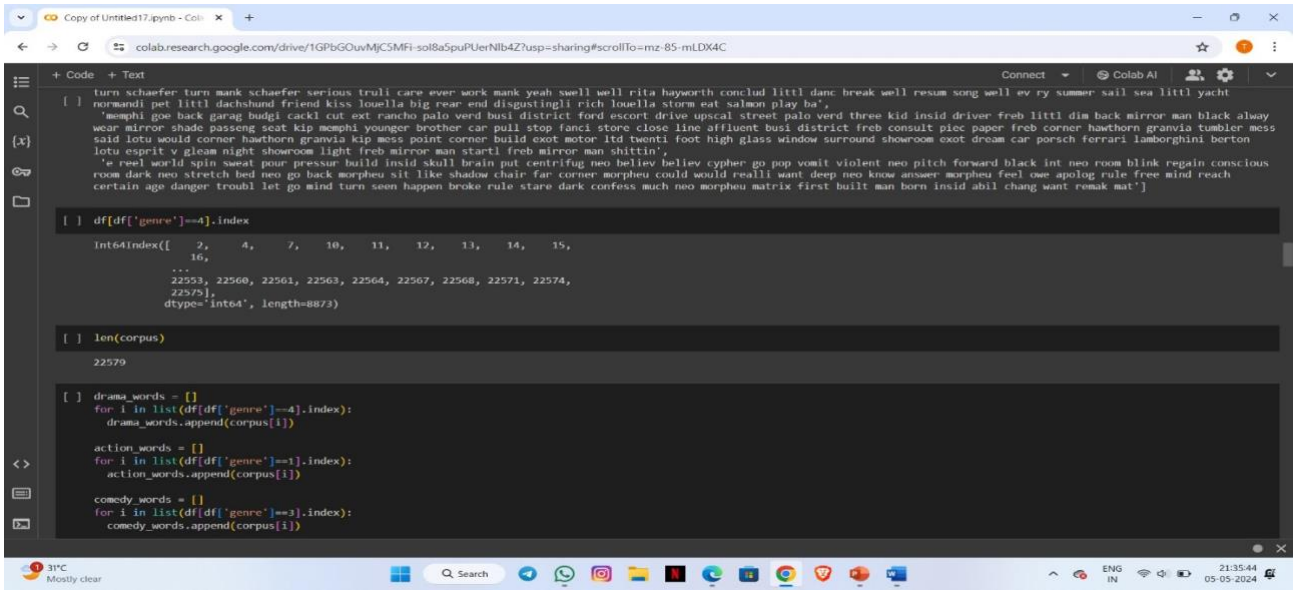
3	4	ded by the two detectives. INT. JEFF'S APARTME...	8
4	5	nd dismounts, just as the other children reach...	4
5	6	breadth of the bluff. Gabe pulls out his ancie...	8
6	7	uilding. A MAN in pajamas runs out into the ra...	8
7	9	ELLES AND RITA HAYWORTH Just disgustingly rich...	4
8	10	Memphis goes back into the garage, Budgy cack...	8
9	11	e reels as the world spins. Sweat pours off hi...	1

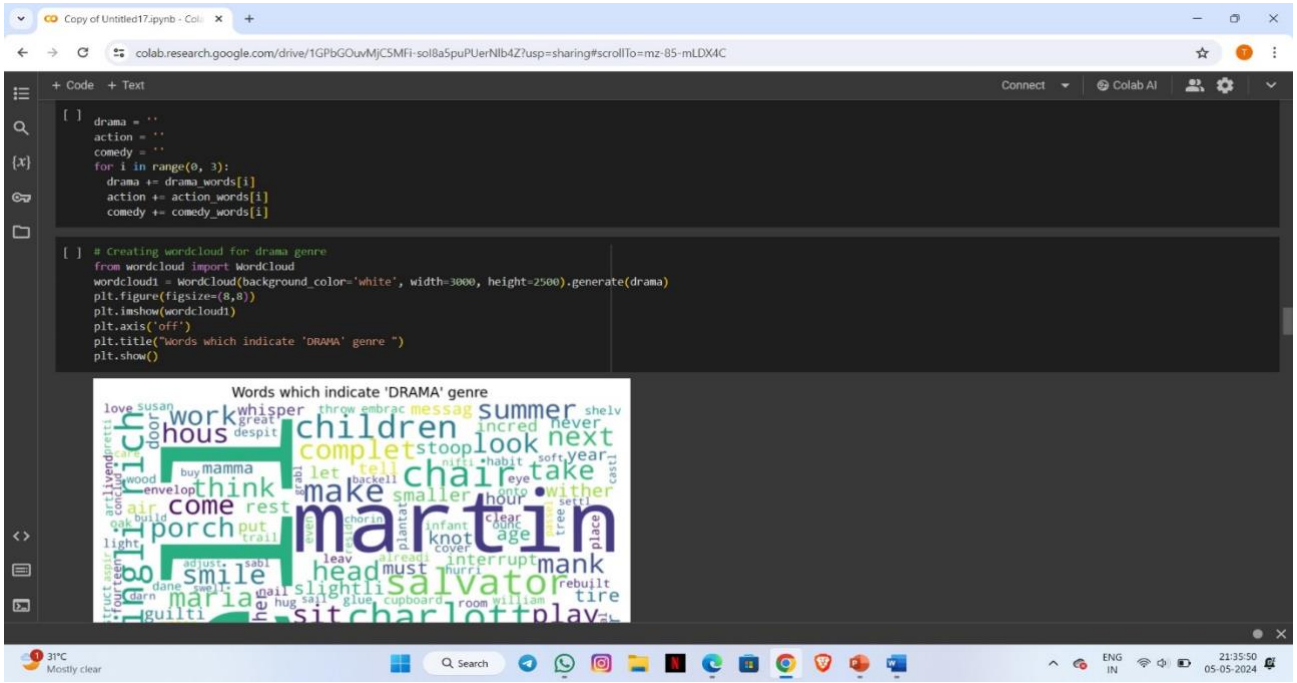
```
df.isna().any()

id      False
text    False
genre   False
dtype: bool

df.drop('id', axis=1, inplace=True)
df.columns

Index(['text', 'genre'], dtype='object')

import nltk
import re
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```



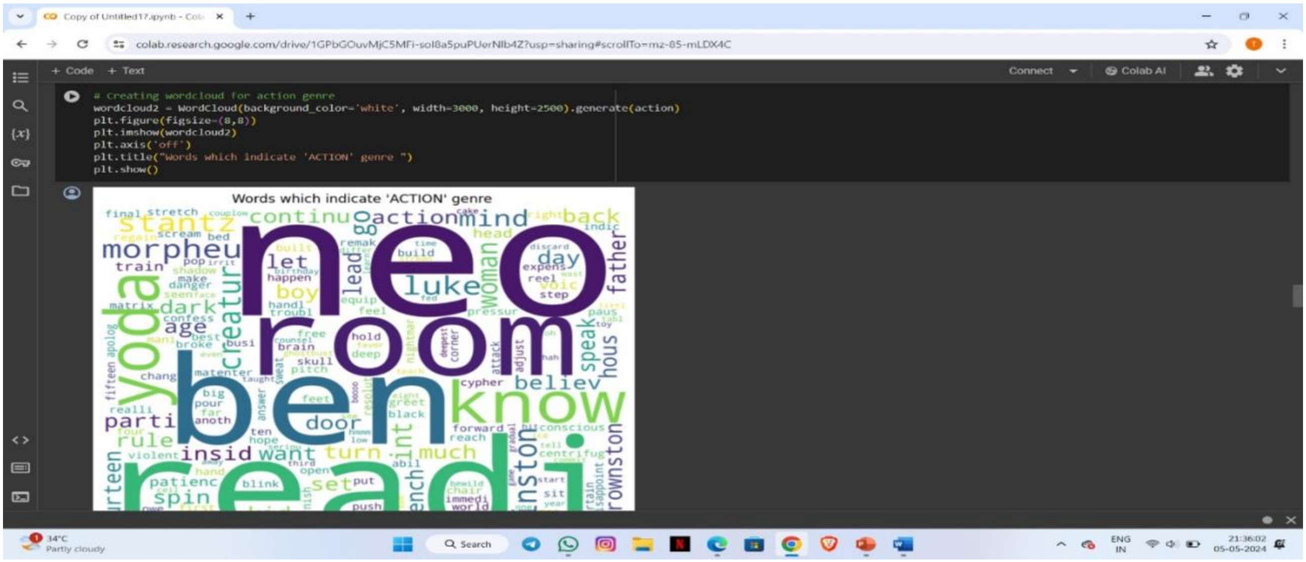
```
import nltk
import re
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.

df.shape

(22579, 2)

corpus = []
ps = PorterStemmer()

for i in range(0, df.shape[0]):

  # Cleaning special character from the dialog/script
  dialog = re.sub(pattern='[^a-zA-Z]', repl=' ', string=df['text'][i])

  # Converting the entire dialog/script into lower case
  dialog = dialog.lower()

  # Tokenizing the dialog/script by words
  words = dialog.split()

  # Removing the stop words
  dialog_words = [word for word in words if word not in set(stopwords.words('english'))]

  # Stemming the words
  words = [ps.stem(word) for word in dialog_words]

  # Joining the stemmed words
```

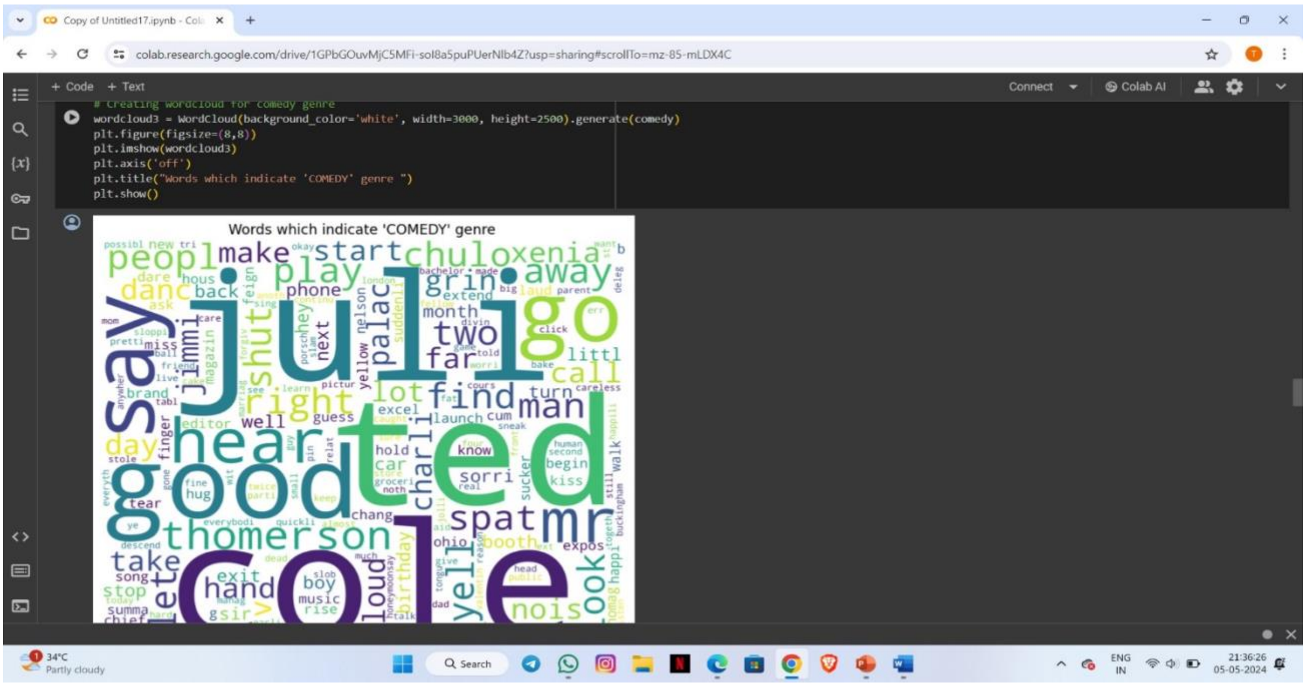

Copy of Untitled17.ipynb - Colab
colab.research.google.com/drive/1GPbGOuvMjC5MFi-sol8a5puPUerNlb4Z?usp=sharing#scrollTo=mz-85-mLDX4C
+ Code + Text
Connect
Colab AI
dialog = ' '.join(words)
# Creating a corpus
corpus.append(dialog)
corpus[0:10]
['eadi dead mayb even wish int nd floor hallway three night orderli lead liza door orderli white guy open door step room three white guy mid look wild straight jacket jerri liza reach end rope shake head int decrepit hospit room night ball fetal realli head press cement tri sing jerri blue moon blue moon int nd floor hallway three night liza stand lean rail wall orderli sure go know bad orderli okay liza start hall orderli follow orderli got new patient last week want see liza wave hopeless stop chicken wire window end hall look light break jerri somewher orderli look gotta get back work',
'summa cum laud launch brand new magazin call expos homag miss juli conroy xenia ohio juli grin juli know find excel editor chief ted yellow page juli let finger walk suddenli music chang peopl ted grin ted play song extend hand dare ask danc juli take hand better ted juli begin danc kiss b g charli jimmi feign tear charli sucker happi end hug jimmi hold start rise nelson hous cloud xenia ted v guess everybodi pretti much live happili ever parent give groceri store descend cloud quickli find ext london buckingham palac day mom dad take pictur smooch front palac ted v manag sneak away second honeymoon',
'come surpris take hand lead hallway salvator look feel pang seem smaller age wither bodi slightli stoop hair gather knot back head must tire want rest time funer salvator interrupt mamma take hour air know maria smile iron tell year salvator got messag feel guilti think seem incred never come maria open door step asid let son whisper put thing go go salvator lake step flabbergast sight old room perfectli reconstruct preserv look like museum museum past despit bed cloth cupboard book shelv perfectli clear one ever live',
'ded two detect int jeff apart night medium shot thorwald fight dislodg jeff grip ext jeff apart night close shot look jeff face show strain pain thorwald attack brick floor patio seem hundr feet int jeff apart night medium shot thorwald jeff struggl ext neighborhood night semi close shot doyl pull top wall lisa stella two men look lisa white face frighten int jeff apart night medium shot thorwald smash jeff arm hand jeff grip begin slip ext neighborhood night semi close shot doyl reach top wall look jeff ext neighborhood night medium long shot jeff seen doyl angl hang somehow weather thorwald insan attack ext neighborhood night semi close shot doyl reach servic revolv look call one dete',
'nd discount children reach throw arm embrac charlott hurri behind martin lock eye envelop hug children ext fresh water plantat even summer oak tree cover leav martin hous partial rebuilt habit workshop alreadi complet martin children nathan samuel margaret william play tall grass front hous two great dane charlott sit front porch nurs infant martin walk workshop trail susan carri complet rock chair chair work art thin light spider web perfectli turn wood nail glue step onto porch next charlott place rock chair next martin two pound fourteen ounc charlott love smile make minut adjust chair posit sit settl back',
'breadth bluff gabe pull ancient binocular scan crack gabe pov crack pictur mine shaft design madman crack move upward errat side straight width crack uneven rang six inch six feet look outsid gabe turn binocular insid crack look crack goe way bluff rout gabe tunnel mountain instead go side gabe get side jessi gone far right think better shape gabe simpl ye would done jessi want lead gabe cute ext top bluff day vista point see everyth els mountain rang thing taller tower two mountain lie drop mere four thousand feet qualen said way across h',
'uild man pajama run rain cabbi lose grip bumper terrenc jerk closer sewer man grab cabbi hand pull resid gather sidewalk polic car siren approach someth give man pajama fall backward puddl small crowd look see terrenc pull free hole moan semi conscious move bodi past bleed stump leg use follow blood swirl eddi rain water flow black storm drain cut fierc bull charg matador red muletta snort blood crowd goe wild bull fight arena blaze spanish sun camera dolli past cheer spaniard find small group american student earli twenti gord man believ paid good money watch guy tight pant kill cow sherri disgust make',
'ell rita hayworth disgustingli rich well make money make quick start littl war think slick smother sabl like betti grabl disgustingli rich build castl cost passel resid pan presid aspir higher higher get marri buy girl darn pretti head swirl rita hayworth swim highbal stew eyebal well rita hayworth disgustingli rich well rita hayworth chorin nifti soft shoe turn schaefer turn mank schaefer serious truli care ever work mank yeah swell well rita hayworth conclud littl danc break well resum song well ev ry summer sail sea littl yacht

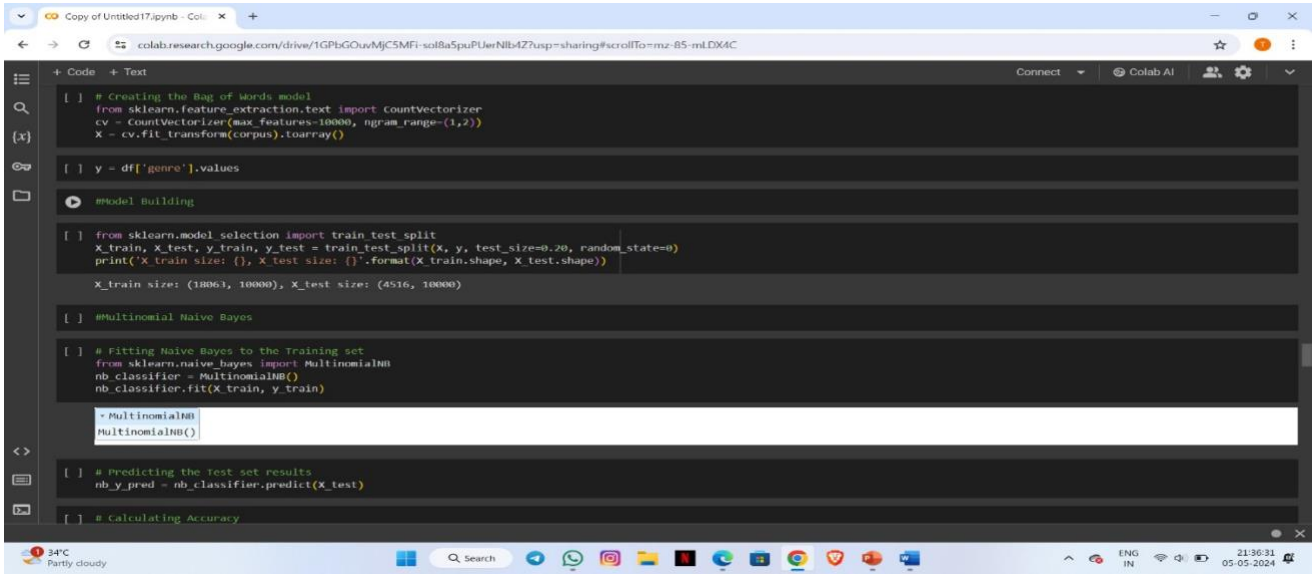

Copy of Untitled17.ipynb - Colab
colab.research.google.com/drive/1GPbGOuvMjC5MFi-sol8a5puPUerNlb4Z?usp=sharing#scrollTo=mz-85-mLDX4C
+ Code + Text
Connect
Colab AI
turn schaefer turn mank schaefer serious truli care ever work mank yeah swell well rita hayworth conclud littl danc break well resum song well ev ry summer sail sea littl yacht
normandi pet littl dachshund friend kiss louella big rear end disgustingli rich louella storm eat salmon play ba',
'memphi goe back garag budgi cackl cut ext rancho palo verd busi district ford escort drive upscal street palo verd three kid insid driver freb littl dim back mirror man black alway wear mirror shade passeng seat kip memphi younger brother car pull stop fanci store close line affluent busi district freb consult piec paper freb corner hawthorn granvia tumbler mess said lotu would corner hawthorn granvia kip mess point corner build exot motor ltd twenti foot high glass window surround showroom exot dream car porsch ferrari lamborghini berton lotu esprit v gleam night showroom light freb mirror man startl freb mirror man shittin',
'e reel world spin sweat pour pressur build insid skull brain put centrifug neo believ believ cypher go pop vomit violent neo pitch forward black int neo room blink regain conscious room dark neo stretch bed neo go back morpheu sit like shadow chair far corner morpheu could would realli want deep neo know answer morpheu feel owe apolog rule free mind reach certain age danger troubl let go mind turn seen happen broke rule stare dark confess much neo morpheu matrix first built man born insid abil chang want remak mat']
df[df['genre']==4].index
Int64Index([2, 4, 7, 10, 11, 12, 13, 14, 15, 16, ... 22553, 22560, 22561, 22563, 22564, 22567, 22568, 22571, 22574, 22575], dtype='int64', length=8873)
len(corpus)
22579
drama_words = []
for i in list(df[df['genre']==4].index):
drama_words.append(corpus[i])
action_words = []
for i in list(df[df['genre']==1].index):
action_words.append(corpus[i])
comedy_words = []
for i in list(df[df['genre']==3].index):
comedy_words.append(corpus[i])

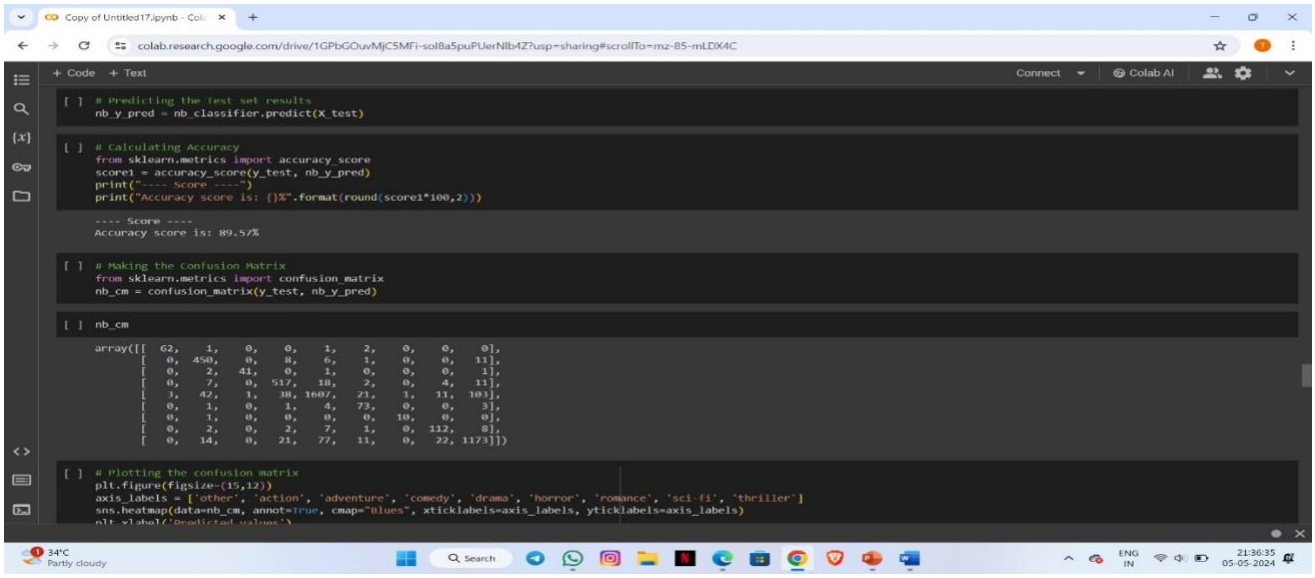

Copy of Untitled17.ipynb - Colab
colab.research.google.com/drive/1GPbGOuvMjC5MFi-sol8a5puPUerNlb4Z?usp=sharing#scrollTo=mz-85-mLDX4C
+ Code + Text
Connect
Colab AI
drama = ''
action = ''
comedy = ''
for i in range(0, 3):
drama += drama_words[i]
action += action_words[i]
comedy += comedy_words[i]
# Creating wordcloud for drama genre
from wordcloud import WordCloud
wordcloud1 = WordCloud(background_color='white', width=3000, height=2500).generate(drama)
plt.figure(figsize=(8,8))
plt.imshow(wordcloud1)
plt.axis('off')
plt.title("Words which indicate 'DRAMA' genre ")
plt.show()
Words which indicate 'DRAMA' genre
martin
children
chair
salvator
smile
maria
porch
house
think
make
look
next
take
work
summer
complet
head
mank
sit
charlott
play

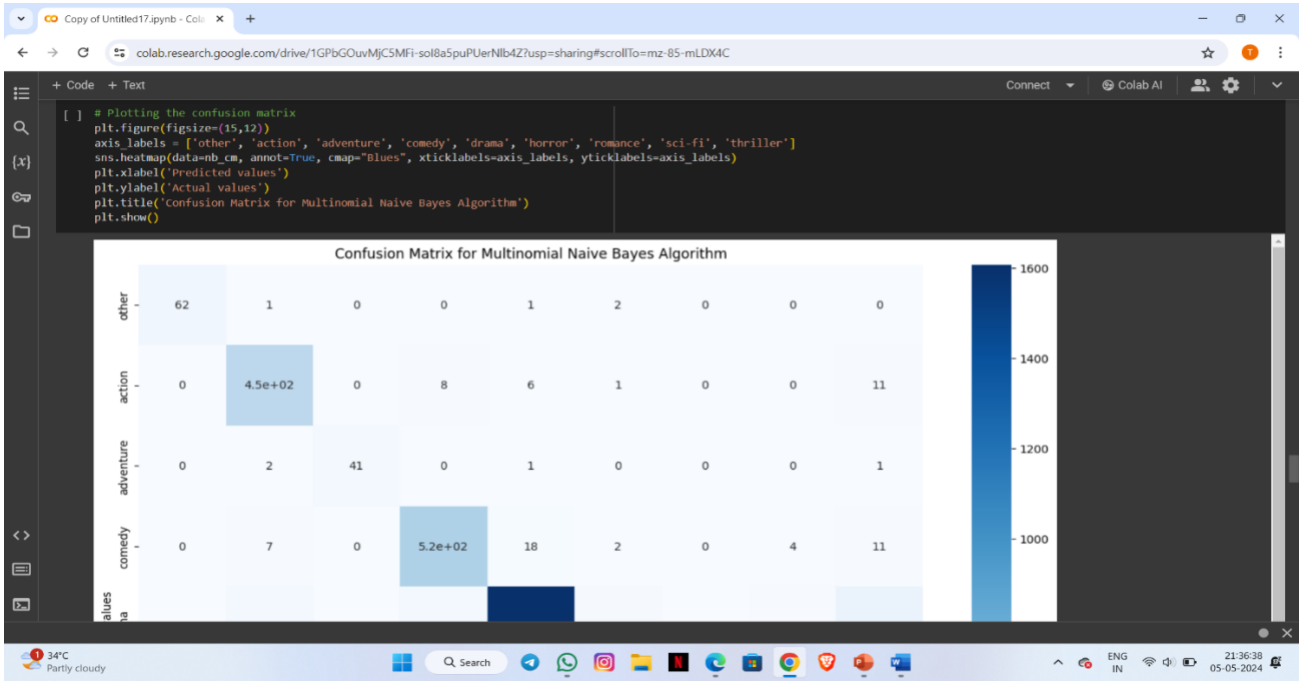

Words which indicate 'DRAMA' genre
# Creating wordcloud for action genre
wordcloud2 = WordCloud(background_color='white', width=3000, height=2500).generate(action)
plt.figure(figsize=(8,8))

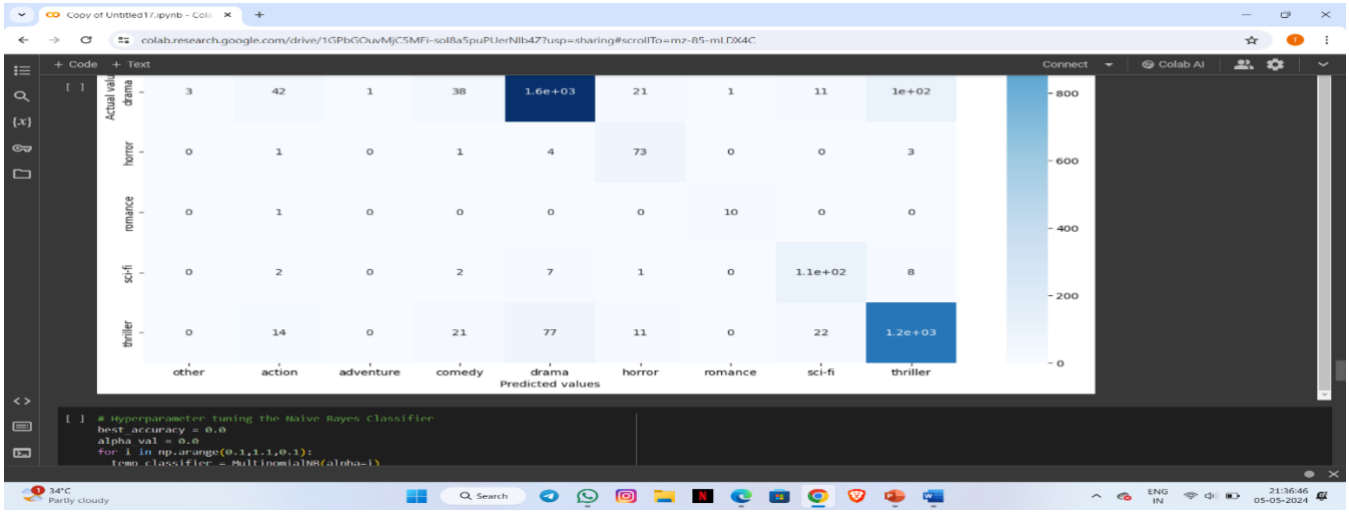

# Creating wordcloud for action genre
wordcloud2 = WordCloud(background_color='white', width=3000, height=2500).generate(action)
plt.figure(figsize=(8,8))
plt.imshow(wordcloud2)
plt.axis('off')
plt.title("Words which indicate 'ACTION' genre ")
plt.show()
Words which indicate 'ACTION' genre

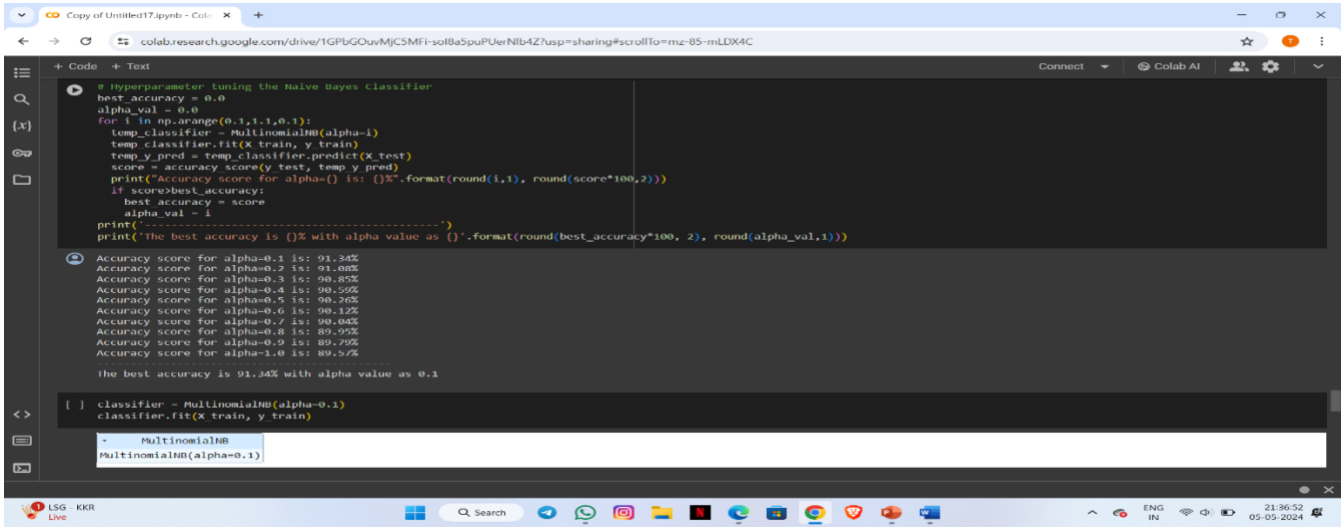

# Creating wordcloud for comedy genre
wordcloud3 = WordCloud(background_color='white', width=3000, height=2500).generate(comedy)
plt.figure(figsize=(8,8))
plt.imshow(wordcloud3)
plt.axis('off')
plt.title("Words which indicate 'COMEDY' genre ")
plt.show()
Words which indicate 'COMEDY' genre


Copy of Untitled17.ipynb - Colab
colab.research.google.com/drive/1GPbGOuvMjC5MFi-sol8a5puPUerNlb4Z?usp=sharing#scrollTo=mz-85-mLDX4C
+ Code + Text
Connect
Colab AI
# Creating the Bag of Words model
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=10000, ngram_range=(1,2))
X = cv.fit_transform(corpus).toarray()
y = df['genre'].values
#Model Building
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=0)
print('X_train size: {}, X_test size: {}'.format(X_train.shape, X_test.shape))
X_train size: (18063, 10000), X_test size: (4516, 10000)
#Multinomial Naive Bayes
# Fitting Naive Bayes to the Training set
from sklearn.naive_bayes import MultinomialNB
nb_classifier = MultinomialNB()
nb_classifier.fit(X_train, y_train)
MultinomialNB
MultinomialNB()
# Predicting the Test set results
nb_y_pred = nb_classifier.predict(X_test)
# Calculating Accuracy
34°C
Partly cloudy
Search
ENG
IN
21:36:31
05-05-2024

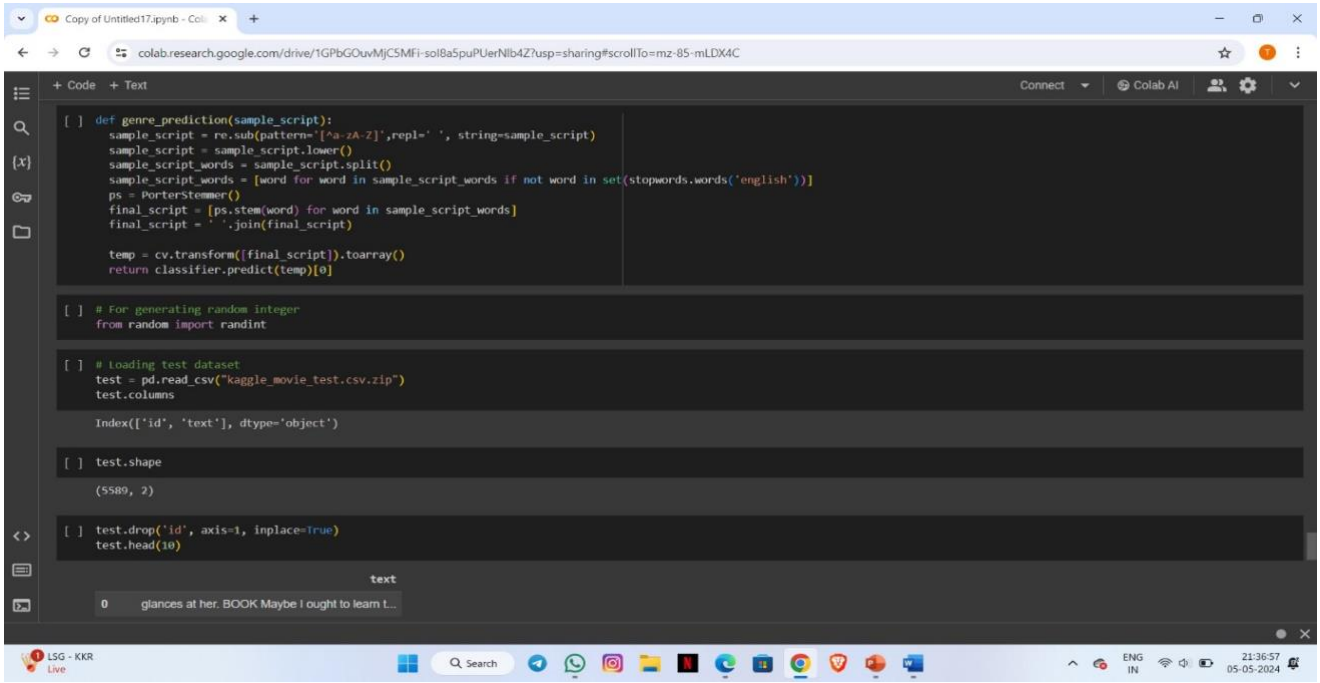

Copy of Untitled17.ipynb - Colab
colab.research.google.com/drive/1GPbGOuvMjC5MFi-sol8a5puPUerNlb4Z?usp=sharing#scrollTo=mz-85-mLDX4C
+ Code + Text
Connect
Colab AI
# Predicting the Test set results
nb_y_pred = nb_classifier.predict(X_test)
# Calculating Accuracy
from sklearn.metrics import accuracy_score
score1 = accuracy_score(y_test, nb_y_pred)
print("---- Score ----")
print("Accuracy score is: {}%".format(round(score1*100,2)))
---- Score ----
Accuracy score is: 89.57%
# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
nb_cm = confusion_matrix(y_test, nb_y_pred)
nb_cm
array([[62,   1,   0,   0,   1,   2,   0,   0,   0],
[0, 450,   0,   8,   6,   1,   0,   0, 11],
[0,   2, 41,   0,   1,   0,   0,   0,   1],
[0,   7,   0, 517, 18,   2,   0,   4, 11],
[3, 42,   1, 38, 1607, 21,   1, 11, 103],
[0,   1,   0,   1,   4, 73,   0,   0,   9],
[0,   1,   0,   0,   0,   0, 10,   0,   0],
[0,   2,   0,   2,   7,   1,   0, 112,   8],
[0, 14,   0, 21, 77, 11,   0, 22, 1173]])
# Plotting the confusion matrix
plt.figure(figsize=(15,12))
axis_labels = ['other', 'action', 'adventure', 'comedy', 'drama', 'horror', 'romance', 'sci-fi', 'thriller']
sns.heatmap(data=nb_cm, annot=True, cmap="Blues", xticklabels=axis_labels, yticklabels=axis_labels)
34°C
Partly cloudy
Search
ENG
IN
21:36:35
05-05-2024

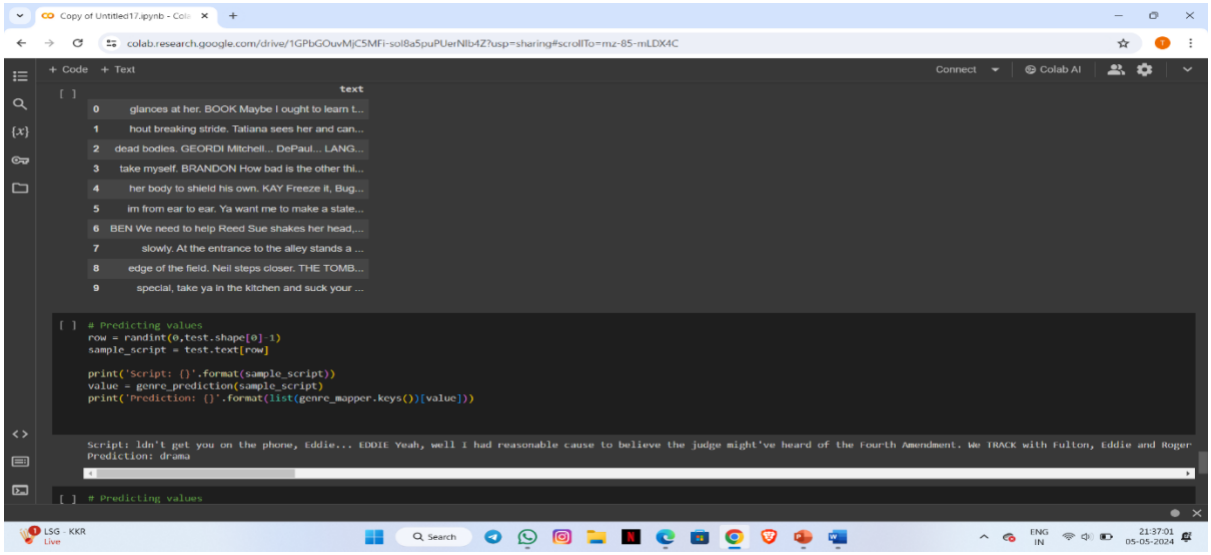

Copy of Untitled17.ipynb - Colab
colab.research.google.com/drive/1GPbGOuvMjC5MFi-sol8a5puPUerNlb4Z?usp=sharing#scrollTo=mz-85-mLDX4C
+ Code + Text
Connect
Colab AI
# Plotting the confusion matrix
plt.figure(figsize=(15,12))
axis_labels = ['other', 'action', 'adventure', 'comedy', 'drama', 'horror', 'romance', 'sci-fi', 'thriller']
sns.heatmap(data=nb_cm, annot=True, cmap="Blues", xticklabels=axis_labels, yticklabels=axis_labels)
plt.xlabel('Predicted values')
plt.ylabel('Actual values')
plt.title('Confusion Matrix for Multinomial Naive Bayes Algorithm')
plt.show()
Confusion Matrix for Multinomial Naive Bayes Algorithm
other
action
adventure
comedy
62
1
0
0
1
2
0
0
0
0
4.5e+02
0
8
6
1
0
0
11
0
2
41
0
1
0
0
0
1
0
7
0
5.2e+02
18
2
0
4
11
1600
1400
1200
1000
34°C
Partly cloudy
Search
ENG
IN
21:36:38
05-05-2024

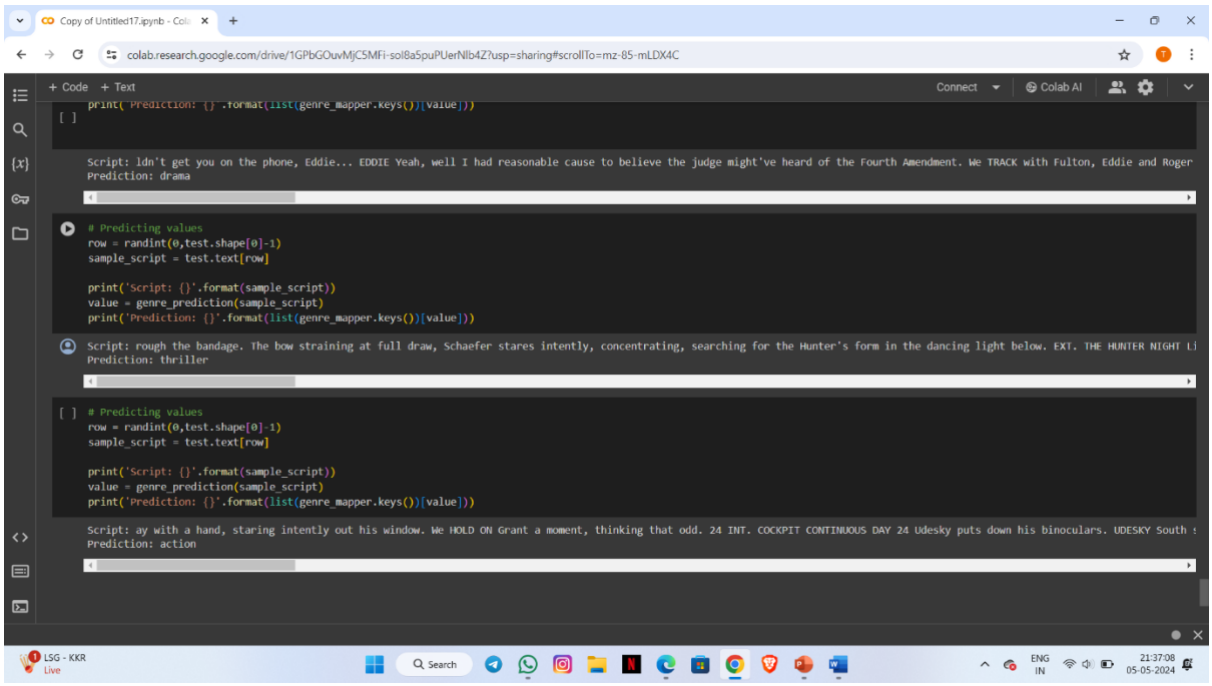

Actual values
drama
horror
romance
sci-fi
thriller
other
action
adventure
comedy
drama
horror
romance
sci-fi
thriller
Predicted values
3 42 1 38 1.6e+03 21 1 11 1e+02
0 1 0 1 4 73 0 0 3
0 1 0 0 0 0 10 0 0
0 2 0 2 7 1 0 1.1e+02 8
0 14 0 21 77 11 0 22 1.2e+03
800
600
400
200
0
# Hyperparameter tuning the Naive Bayes Classifier
best_accuracy = 0.0
alpha_val = 0.0
for i in np.arange(0.1,1.1,0.1):
temp_classifier = MultinomialNB(alpha=i)


colab.research.google.com/drive/1GPbGOuvMjC5MFi-sol8a5puPUerNlb4Z?usp=sharing#scrollTo=mz-85-mLDX4C
# Hyperparameter tuning the Naive Bayes Classifier
best_accuracy = 0.0
alpha_val = 0.0
for i in np.arange(0.1,1.1,0.1):
temp_classifier = MultinomialNB(alpha=i)
temp_classifier.fit(X_train, y_train)
temp_y_pred = temp_classifier.predict(X_test)
score = accuracy_score(y_test, temp_y_pred)
print("Accuracy score for alpha={} is: {}%".format(round(i,1), round(score*100,2)))
if score>best_accuracy:
best_accuracy = score
alpha_val = i
print('--')
print('The best accuracy is {}% with alpha value as {}'.format(round(best_accuracy*100, 2), round(alpha_val,1)))
Accuracy score for alpha=0.1 is: 91.34%
Accuracy score for alpha=0.2 is: 91.08%
Accuracy score for alpha=0.3 is: 90.85%
Accuracy score for alpha=0.4 is: 90.59%
Accuracy score for alpha=0.5 is: 90.26%
Accuracy score for alpha=0.6 is: 90.12%
Accuracy score for alpha=0.7 is: 90.04%
Accuracy score for alpha=0.8 is: 89.95%
Accuracy score for alpha=0.9 is: 89.79%
Accuracy score for alpha=1.0 is: 89.57%

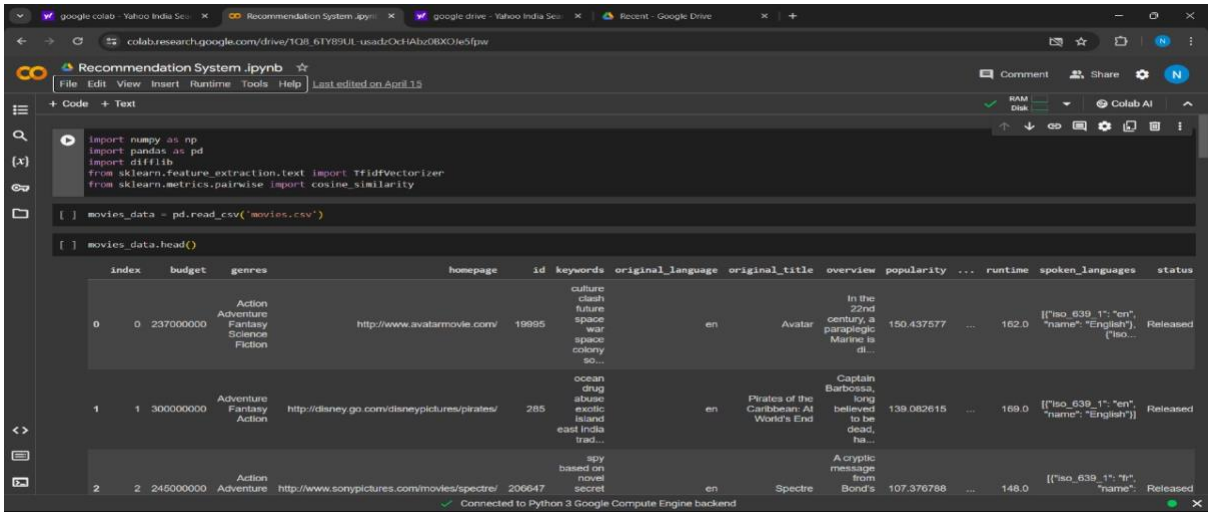
The best accuracy is 91.34% with alpha value as 0.1
classifier = MultinomialNB(alpha=0.1)
classifier.fit(X_train, y_train)
MultinomialNB
MultinomialNB(alpha=0.1)

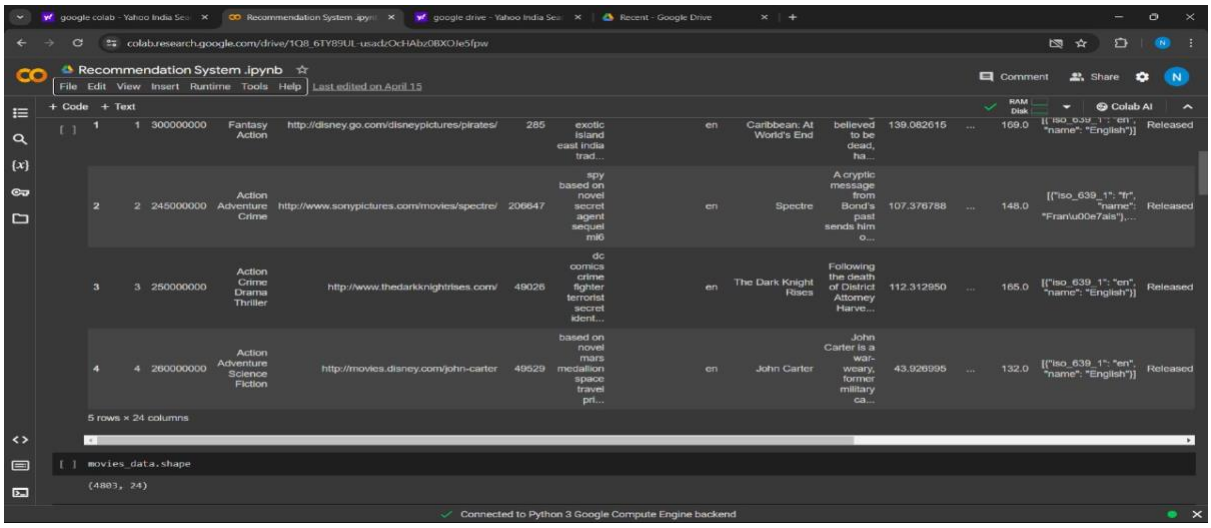

colab.research.google.com/drive/1GPbGOuvMjC5MFi-sol8a5puPUerNlb4Z?usp=sharing#scrollTo=mz-85-mLDX4C
def genre_prediction(sample_script):
sample_script = re.sub(pattern='[^a-zA-Z]',repl=' ', string=sample_script)
sample_script = sample_script.lower()
sample_script_words = sample_script.split()
sample_script_words = [word for word in sample_script_words if not word in set(stopwords.words('english'))]
ps = PorterStemmer()
final_script = [ps.stem(word) for word in sample_script_words]
final_script = ' '.join(final_script)
temp = cv.transform([final_script]).toarray()
return classifier.predict(temp)[0]
# For generating random integer
from random import randint
# Loading test dataset
test = pd.read_csv("kaggle_movie_test.csv.zip")
test.columns
Index(['id', 'text'], dtype='object')
test.shape
(5589, 2)
test.drop('id', axis=1, inplace=True)
test.head(10)
text
0 glances at her. BOOK Maybe I ought to learn t...

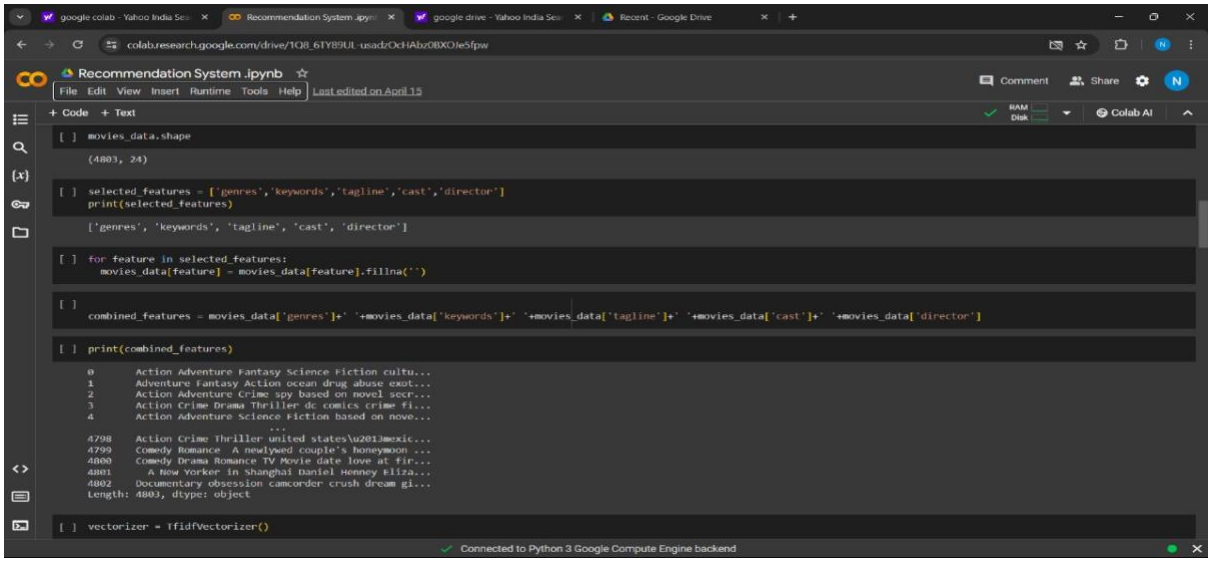

Copy of Untitled17.ipynb - Colab
colab.research.google.com/drive/1GPbGOuvMjC5MFi-sol8a5puPUerNlb4Z?usp=sharing#scrollTo=mz-85-mLDX4C
+ Code + Text
Connect
Colab AI
text
0 glances at her. BOOK Maybe I ought to learn t...
1 hout breaking stride. Tatiana sees her and can...
2 dead bodies. GEORDI Mitchell... DePaul... LANG...
3 take myself. BRANDON How bad is the other thi...
4 her body to shield his own. KAY Freeze it, Bug...
5 im from ear to ear. Ya want me to make a state...
6 BEN We need to help Reed Sue shakes her head,...
7 slowly. At the entrance to the alley stands a ...
8 edge of the field. Neil steps closer. THE TOMB...
9 special, take ya in the kitchen and suck your ...
# Predicting values
row = randint(0,test.shape[0]-1)
sample_script = test.text[row]
print('Script: {}'.format(sample_script))
value = genre_prediction(sample_script)
print('Prediction: {}'.format(list(genre_mapper.keys())[value]))
Script: ldn't get you on the phone, Eddie... EDDIE Yeah, well I had reasonable cause to believe the judge might've heard of the Fourth Amendment. We TRACK with Fulton, Eddie and Roger
Prediction: drama
Predicting values
LSG - KKR
Live
Search
ENG
IN
21:37:01
05-05-2024

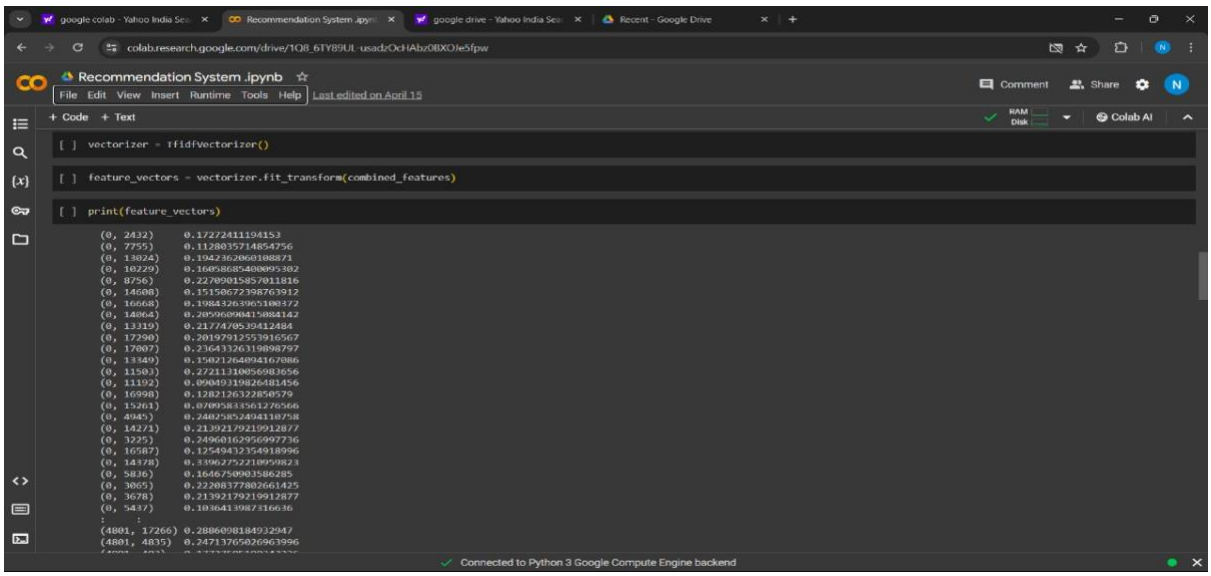

Copy of Untitled17.ipynb - Colab
colab.research.google.com/drive/1GPbGOuvMjC5MFi-sol8a5puPUerNlb4Z?usp=sharing#scrollTo=mz-85-mLDX4C
+ Code + Text
Connect
Colab AI
Script: ldn't get you on the phone, Eddie... EDDIE Yeah, well I had reasonable cause to believe the judge might've heard of the Fourth Amendment. We TRACK with Fulton, Eddie and Roger
Prediction: drama
# Predicting values
row = randint(0,test.shape[0]-1)
sample_script = test.text[row]
print('Script: {}'.format(sample_script))
value = genre_prediction(sample_script)
print('Prediction: {}'.format(list(genre_mapper.keys())[value]))
Script: rough the bandage. The bow straining at full draw, Schaefer stares intently, concentrating, searching for the Hunter's form in the dancing light below. EXT. THE HUNTER NIGHT LI
Prediction: thriller
# Predicting values
row = randint(0,test.shape[0]-1)
sample_script = test.text[row]
print('Script: {}'.format(sample_script))
value = genre_prediction(sample_script)
print('Prediction: {}'.format(list(genre_mapper.keys())[value]))
Script: ay with a hand, staring intently out his window. We HOLD ON Grant a moment, thinking that odd. 24 INT. COCKPIT CONTINUOUS DAY 24 Udesky puts down his binoculars. UDESKY South
Prediction: action
LSG - KKR
Live
Search
ENG
IN
21:37:08
05-05-2024

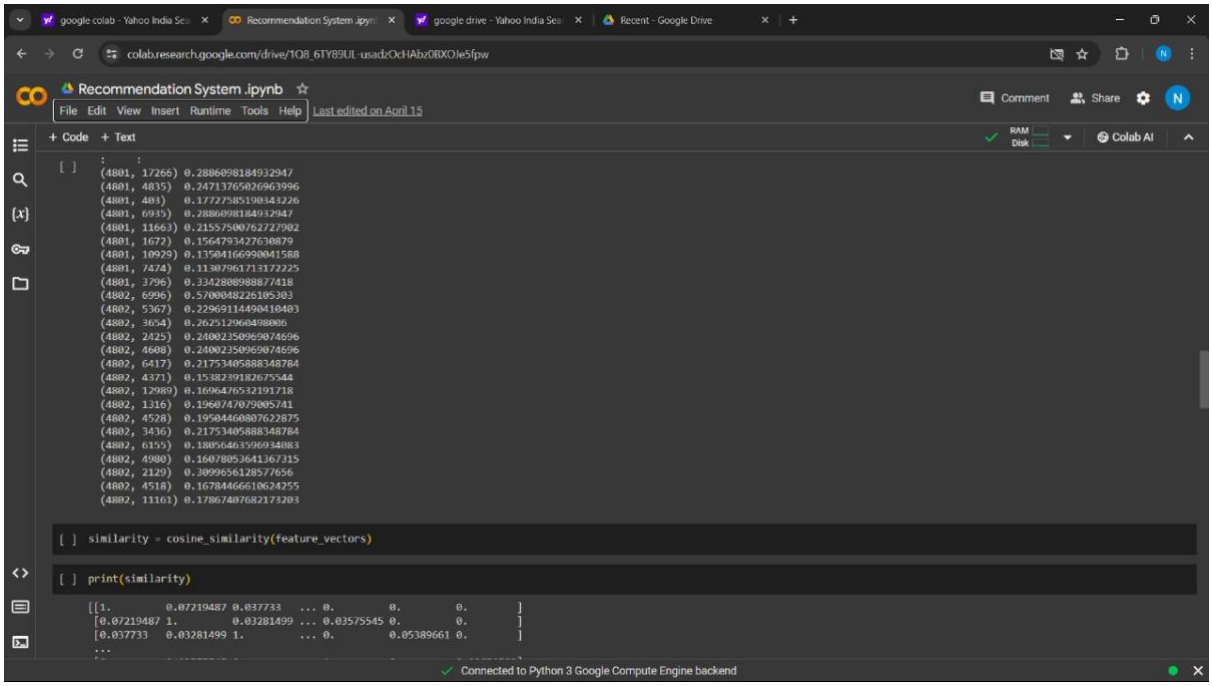
Movie recommendation system

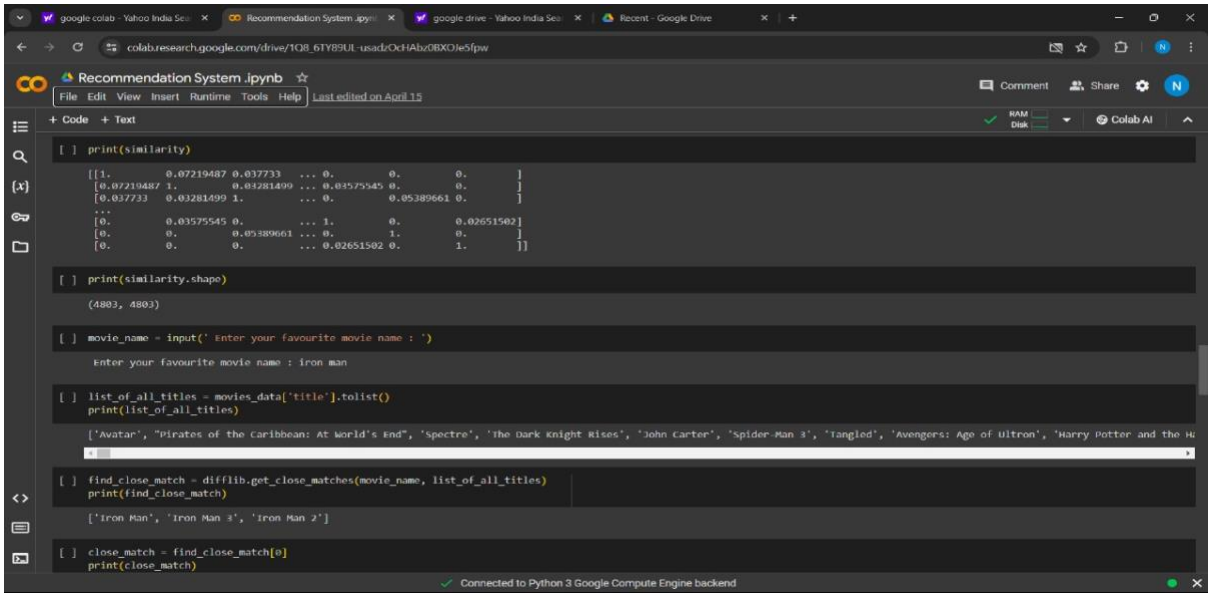


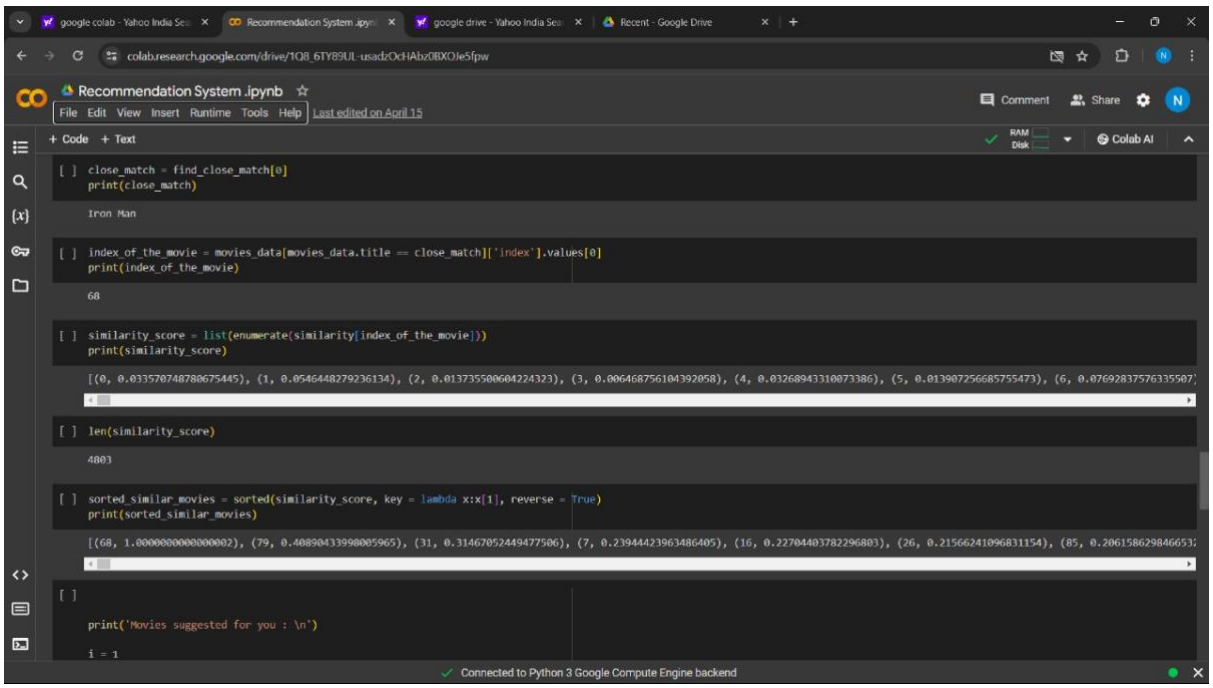


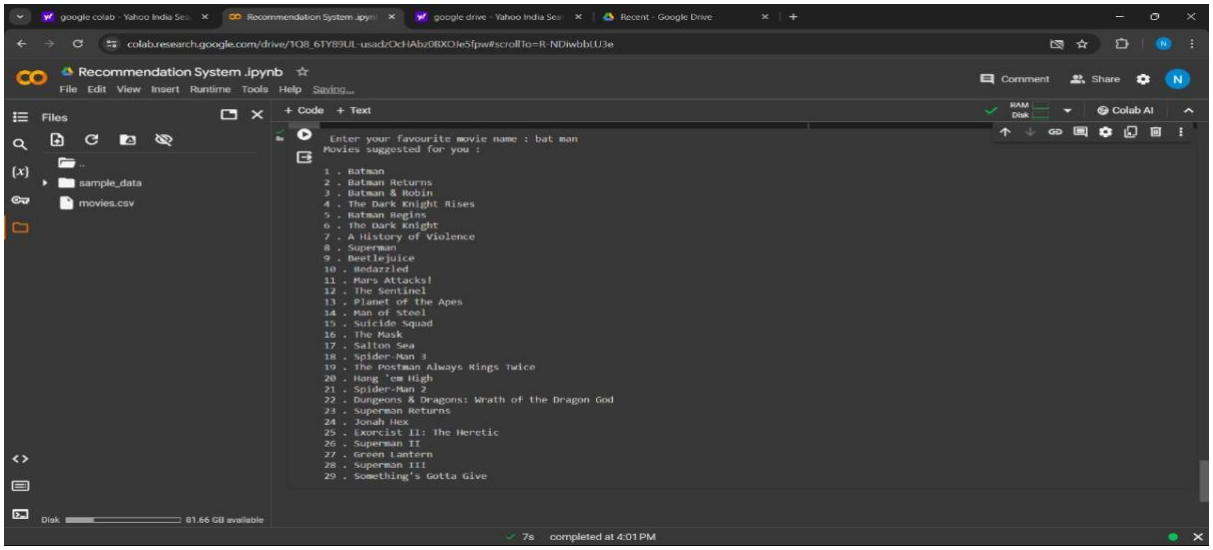


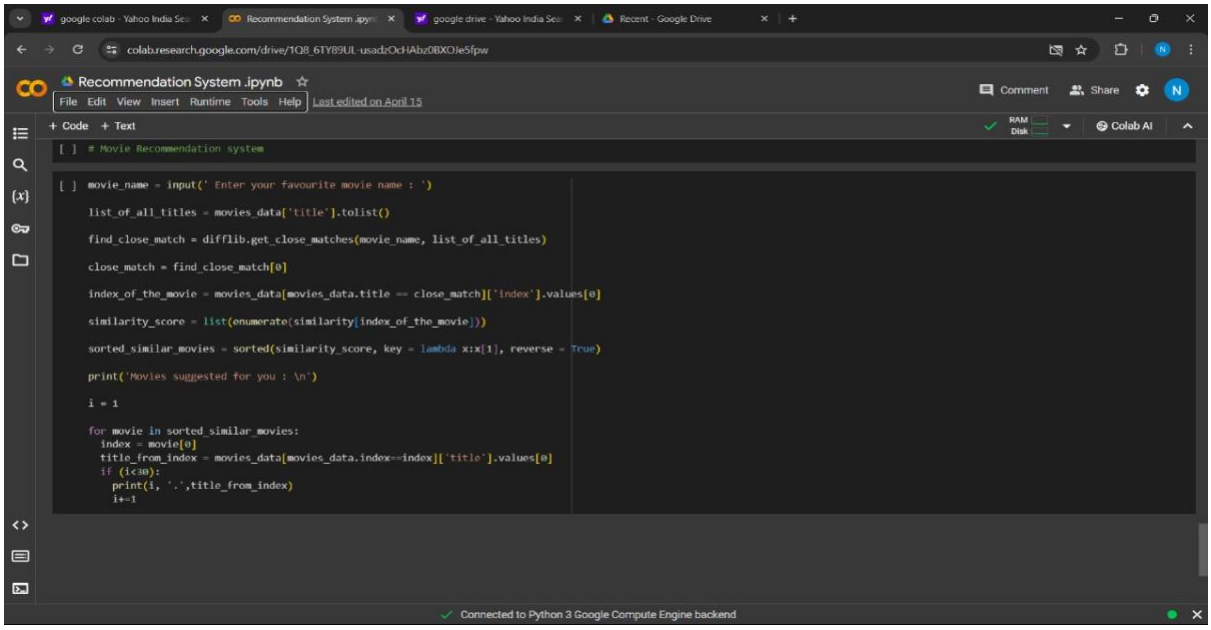

Recommendation System .ipynb
vectorizer = TfidfVectorizer()
feature_vectors = vectorizer.fit_transform(combined_features)
print(feature_vectors)
(0, 2432) 0.17272411194153
(0, 7755) 0.1128035714854756
(0, 13024) 0.1942362060308871
(0, 16229) 0.16058685480095302
(0, 8756) 0.2270901585701181
(0, 14608) 0.15150672398763912
(0, 16668) 0.19843263965108372
(0, 14864) 0.20596090415084142
(0, 13319) 0.2177470539412484
(0, 17290) 0.20197912553916567
(0, 17007) 0.23643326319898797
(0, 13349) 0.15021264894167086
(0, 11503) 0.2721131005693656
(0, 11192) 0.09049319826481456
(0, 16998) 0.128212632285079
(0, 15261) 0.07095833561276566
(0, 4945) 0.24025852494110758
(0, 14271) 0.21392179219912877
(0, 1225) 0.2496016295697736
(0, 16587) 0.12549432354918996
(0, 14378) 0.33962752210959823
(0, 5836) 0.16467509036286285
(0, 3065) 0.22208377802661425
(0, 3678) 0.21392179219912877
(0, 5437) 0.10364139873166636
(4801, 17266) 0.2886098184932947
(4801, 4835) 0.24713765026963996
Connected to Python 3 Google Compute Engine backend


Recommendation System .ipynb
(4801, 17266) 0.2886098184932947
(4801, 4835) 0.24713765026963996
(4801, 403) 0.17727585190343226
(4801, 6935) 0.2886098184932947
(4801, 11663) 0.21557500762727902
(4801, 1672) 0.15647934276308791
(4801, 10929) 0.13504166990041588
(4801, 7474) 0.11387961713172225
(4801, 3796) 0.33428089888774118
(4802, 6996) 0.5700048226105303
(4802, 5367) 0.22969114490410403
(4802, 3654) 0.262512960498806
(4802, 2425) 0.24002350969074696
(4802, 4608) 0.24002350969074696
(4802, 6417) 0.21753405888348784
(4802, 4371) 0.15382391826755444
(4802, 12989) 0.16964765321917184
(4802, 1316) 0.1960747079005741
(4802, 4528) 0.19504460807622875
(4802, 3436) 0.21753405888348784
(4802, 6155) 0.18056463596934083
(4802, 4900) 0.16078053641367315
(4802, 2129) 0.30996561285776566
(4802, 4518) 0.16784466610624255
(4802, 11161) 0.17867407682173203
similarity = cosine_similarity(feature_vectors)
print(similarity)
[[1. 0.07219487 0.037733 ... 0. 0. 0.]
[0.07219487 1. 0.03281499 ... 0.03575545 0. 0.]
[0.037733 0.03281499 1. ... 0. 0.05389661 0.]
Connected to Python 3 Google Compute Engine backend


Recommendation System .ipynb
print(similarity)
[[1. 0.07219487 0.037733 ... 0. 0. 0.]
[0.07219487 1. 0.03281499 ... 0.03575545 0. 0.]
[0.037733 0.03281499 1. ... 0. 0.05389661 0.]
[0. 0.03575545 0. ... 1. 0. 0.02651502]
[0. 0. 0.05389661 ... 0. 1. 0.]
[0. 0. 0. ... 0.02651502 0. 1.]]
print(similarity.shape)
(4803, 4803)
movie_name = input(' Enter your favourite movie name : ')
Enter your favourite movie name : iron man
list_of_all_titles = movies_data['title'].tolist()
print(list_of_all_titles)
['Avatar', "Pirates of the Caribbean: At World's End", 'Spectre', 'The Dark Knight Rises', 'John Carter', 'Spider-Man 3', 'Tangled', 'Avengers: Age of Ultron', 'Harry Potter and the H
find_close_match = difflib.get_close_matches(movie_name, list_of_all_titles)
print(find_close_match)
['Iron Man', 'Iron Man 3', 'Iron Man 2']
close_match = find_close_match[0]
print(close_match)
Connected to Python 3 Google Compute Engine backend


Recommendation System .ipynb
close_match = find_close_match[0]
print(close_match)
Iron Man
index_of_the_movie = movies_data[movies_data.title == close_match]['index'].values[0]
print(index_of_the_movie)
68
similarity_score = list(enumerate(similarity[index_of_the_movie]))
print(similarity_score)
len(similarity_score)
4803
sorted_similar_movies = sorted(similarity_score, key = lambda x:x[1], reverse = True)
print(sorted_similar_movies)
print('Movies suggested for you : \n')
i = 1
Connected to Python 3 Google Compute Engine backend


Recommendation System .ipynb
Files
sample_data
movies.csv
Enter your favourite movie name : bat man
Movies suggested for you :
1 . Batman
2 . Batman Returns
3 . Batman & Robin
4 . The Dark Knight Rises
5 . Batman Begins
6 . The Dark Knight
7 . A History of Violence
8 . Superman
9 . Beetlejuice
10 . Redazzled
11 . Mars Attacks!
12 . The Sentinel
13 . Planet of the Apes
14 . Man of Steel
15 . Suicide Squad
16 . The Mask
17 . Salton Sea
18 . Spider-Man 3
19 . The Postman Always Rings Twice
20 . Hang 'em High
21 . Spider-Man 2
22 . Dungeons & Dragons: Wrath of the Dragon God
23 . Superman Returns
24 . Jonah Hex
25 . Exorcist II: The Heretic
26 . Superman II
27 . Green Lantern
28 . Superman III
29 . Something's Gotta Give
completed at 4:01 PM


Recommendation System .ipynb
# Movie Recommendation system
movie_name = input(' Enter your favourite movie name : ')
list_of_all_titles = movies_data['title'].tolist()
find_close_match = difflib.get_close_matches(movie_name, list_of_all_titles)
close_match = find_close_match[0]
index_of_the_movie = movies_data[movies_data.title == close_match]['index'].values[0]
similarity_score = list(enumerate(similarity[index_of_the_movie]))
sorted_similar_movies = sorted(similarity_score, key = lambda x:x[1], reverse = True)
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
Connected to Python 3 Google Compute Engine backend