Movie Recommendations by Creating A Hybrid System Using Deep Learning Approach

First: Pratibha Jyoti Negi. Author, Second: Dr. Brojo Kishore Mishra. Author

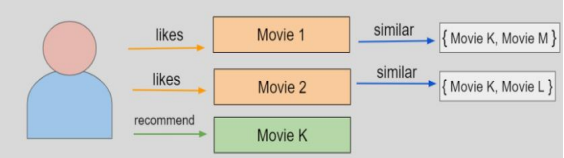
***Abstract*—** Recommendation systems are the world of artificial intelligence becomes extremely popular and large advantage to streaming services like YouTube, Netflix, Amazon Prime, and plenty of and many. The main purpose is to focus on their content to a particular audience. These recommendation systems were terribly robust in their predictions within which they'll dynamically improve the state of what the user sees on the page that's primarily based on the user's interface with the app. during this paper; we tend to have proposed a hybrid recommender system by using Deep learning approach. This approach deals with the content based information and collaborative data separately at first, then it combines the applications to come up with a system with the simplest of each data. By using the MovieLens of 20M Dataset, we have developed a movie-to-movie recommender system that recommends the movies like that movie was given as input . To develop the hybrid based model, we tend to aggregate the results associated with an auto encoder that determines the content-based movie impacts from the tag data, and an entity of neural network that determines data embeddings of collaborative-based knowledge from ratings data.

***Keywords*-**Content Based Filtering, Collaborative Filtering, Movie Recommendation System, NLTK, Deep Learning, NLTK

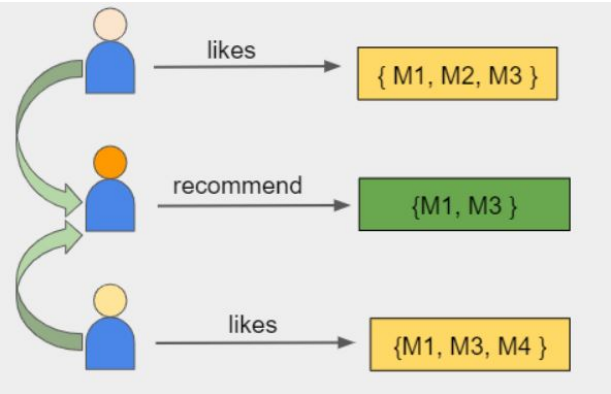
I. INTRODUCTION

Recommendation systems are very important in today's active world. People are always having lack of time with the infinite tasks that they need to bring off in the limited 24 hours. Hence the recommendation systems are very important as they help them to make the right choices and also help them without having to make use of their cognitive resources. If we will see nowadays Netflix, Amazon Prime and those who are movie streaming service providers, their customers are increasing rapidly for enjoying video content. In 2017 Netflix subscribers had watched more than 140 million hours per day, as a result Netflix earned $11bilion in yield. So roughly 80% of hours had streamed at Netflix which were influenced by their proper recommendation system. So no doubt, currently movie gliding services are the integral aspects that how we use up the video contents today. By taking this context, we have designed a hybrid filter system that will work on movie recommendations through the deep learning approach.
For movie gliding services like Netflix, recommendation systems were very important for helping the users to fin out new content for entertainment. The feature of this system is the associations of various recommendation systems like Content-based movie recommendation system, Collaborative Filtering movie recommendation system.

Content based filtering is one of the oldest methods for filtering trickle of data. Content-based methods are determining the similarities of movie features. By using this recommendation system, if one user watches a movie, then similar types of movies will recommended to them. For example, if a user "A" watches a comedy-drama movie starring Shahrukh Khan, then the system will automatically recommend them those movies with the same genres or starring the same actor or both. Hence, the movie attributes are used as the input for building a content-based recommendation system



In **Collaborative filtering**, It is related to past interactions between users and movies. The collaborative filtering system is made up of the past data of user interactions as the input for the movies what they have watched.
For example, if user "A" watches Movie1, Movie2, and Movie3, and user "B" watches sMovie1, Movie3, Movie4, we recommend Movie1 and Movie3 to a similar user "C". The following figure has shown the collaborative recommendation system for better clarity.



Deep Learning has signifies the effectiveness in coping with recommendation tasks. In comparison with traditional recommendation models, deep learning provides high quality recommendations and better understanding of user's demand. In our context, we have applied the deep learning approach for movie recommendation .The rest of the paper is organized as follows. There are many approaches for content and collaborative filtering are discussed in Section II. In Section III, we propose a deep learning approach for content and collaborative filtering based on Naive Bayes and NLTK model. We demonstrate in the Section IV

our experimental analysis and results. Section V is giving a concluding remark of my work.

II. RELATED WORK

Naïve-Bayes formula is one among the popular classification machine learning algorithms that helps to classify the info primarily based upon the probability values computation.[1] It implements the Bayes theorem for the computation and used class levels delineated as feature values or vectors of predictors for classification. Naive Bayes algorithmic rule may be a quick formula for classification issues. This algorithm may be a smart appropriate real-time prediction, multi-class prediction, recommendation system, text classification, and sentiment analysis use cases. Naive Thomas Bayes formulas are often designed using Gaussian, Multinomial and bernoulli distribution. This formula is scalable and simple to implement for the massive data set.

It helps to calculate the posterior chance P(c|x) using the previous chance of class P(c), the previous probability of predictor P(x) and therefore the probability of predictor given category, conjointly known as as probability P(x|c). The formula or equation to calculate posterior chance is: P(c|x) = (P(x|c) * P(c)) / P(x) we use Multinomial distribution of Naïve_Bayes formula from Scikit-learn library.

Deep learning is that the deep artificial neural network is ready to find out advanced decision boundaries for classification or for advanced non-linear regression. One dataset contains Reviews and Comments of many users we want to use sentiment analysis for classify these reviews to positive, negative or average for that we tend to use nlp (Natural Language processing) conception.[17] used libraries square measure NLTK (Natural Language Toolkit) and from NLTK we tend to import Stopwords from corpus.

Stopwords are an English word that doesn't add a lot of meaning to a sentence. They will safely be neglected without sacrificing the meaning of the sentence. As an example, the words just like the, he, have etc. This types of words were already captured this in anatomy named corpus. We tend to 1st transfer it to our python setting. The word count from text documents is incredibly basic at the start line. But easy word count isn't enough for text process due to the words like "the", "an", "your", etc. are extremely occurred in text documents. Their massive word count is vacuous towards the analysis of the text. Tf-idf will be with success used for stop-words filtering from the text document. The other way to solve this drawback is word frequency. This methodology is named the TF-IDF that is stands for "Term Frequency – Inverse Document Frequency". TF-IDF may be a numerical statistic that measures the importance of the word in a document.[4] Term Frequency: range of your time a word seems during a text document. Inverse Document Frequency: measure the word may be a rare word or common word in a document.

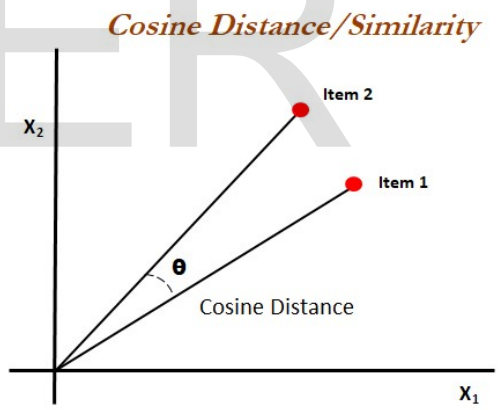
Example:

Consider a document that contains a complete of 100 words and so the word "book" has occurred 5 times in a very document. Term frequency (tf) will be 5 / 100 = 0.05. Let's assume we've 10,000 documents and so the word "book" has occurred in a very thousand of these. Inverse Document Frequency (IDF) will be log[10000/1000] + 1 = 2 TF-IDF = 0.05 * 2 = 0.1. A Scikit-Learn has provided the implementation of the Tfidf Vectorizer

Similarity Score:

How can it decide that item is most an equivalent because the item user likes? Here we tend to use the similarity scores. It's a numerical worth ranges between zeros to one that helps to figure out what amount two things are an equivalent as one another on a scale of zero to one. This similarity score is obtained activity the similarity between the text details of every of the items. So, similarity score is that the measure of similarity between given text details of two things. This might be done by cosine-similarity. Circular function Similarity: cos similarity can be a metric accustomed measure however similar the documents are regardless of their size.[6]

Mathematically, it measures the cosine of the angle between a pair of vectors projected in an extremely multi-dimensional space. The cosine similarity is advantageous as a result of although the two similar documents are most apart by the euclidean distance (due to the scale of the document), chances are high that they will still be headed nearer on. Once the angle is smaller, then the cosine similarity is higher.



Collaborative filtering algorithmic program is a technique used for recommendation systems. It's additionally a preferred and notable algorithmic program within the industries. Collaborative filtering is a methodology used for creating automatic predictions or filtering concerning the interests of a user by collection priority or style information from several users.

Filters out the content in keeping with user similar interest with different users, it essentially recommends the things to users that have similar style. Supported past analysis and increasing the requirement for system recommendations, we want new technology to dramatically improve system recommendations performance. This study focuses on the user-based collaborative filtering algorithm (CFA) that predicts film rating to be suggested to users by utilizing film ratings from different users who have similar behaviors "the belief of the collaborative filtering approach is that if

someone A has an equivalent opinion as someone B on a problem, A is more probably to have B's opinion on a special issue than that of a haphazardly chosen person".

Collaborative filtering depends on the past preference\rating correlation with different users. Supported this correlation, individuals with similar likes are taken under consideration for recommendation. Hybrid strategies are the mixture of both. Content based mostly filtering is one in all the oldest ways for percolating information. it is the method of filtering for information.Filtering using several techniques involving collaboration among multiple "agents, viewpoints, information sources" etc. features of collaborative filtering involves terribly massive information sets. For adding collaborative filtering in movie Recommendation System we tend to use **scikit-learn-surprise** library.

Surprise is a Python scikit library for building and examining recommendation systems that deal with precise rating information.

Surprise was designed for the subsequent purposes:

• It offers user an ideal management over their experiments. a robust prominence is takes place on documentation, that we've tried to form limpid and specific as possible by determine each details of the algorithms.

• Ease the pain of Dataset handling. Users will use each inherent datasets like movie lens, Jester, and along with their own custom datasets.

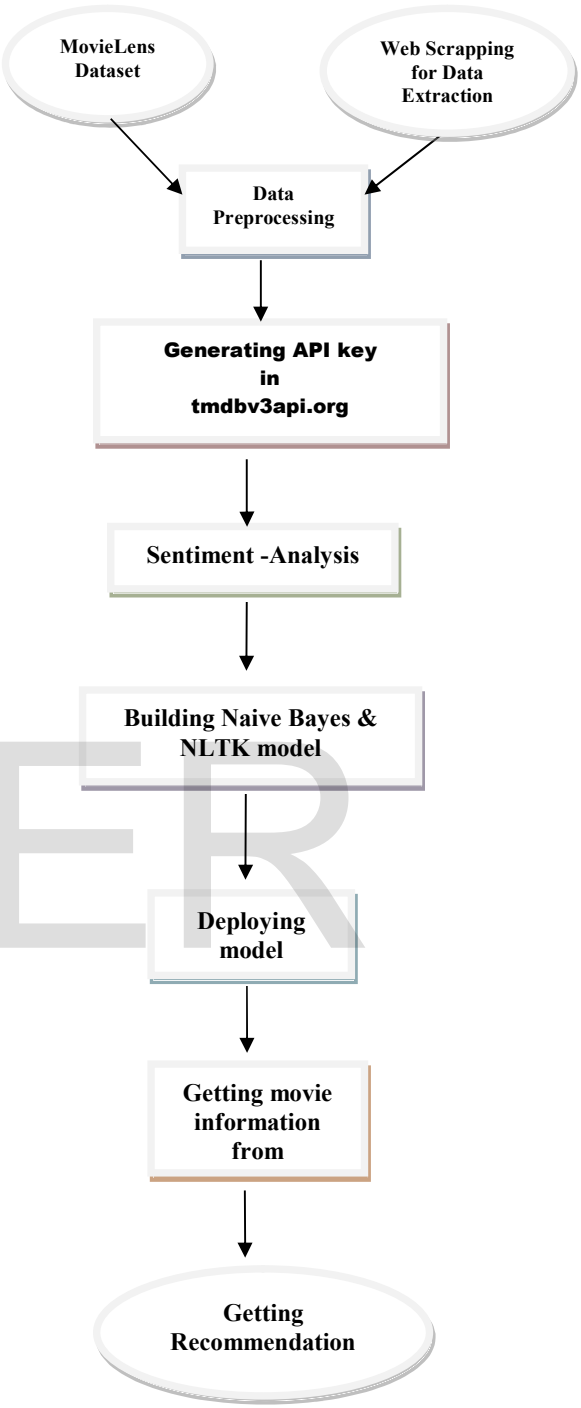
• Make it simple to implement new algorithmic program ideas.

• Provide tools to judge analyse and compare the algorithms' performance. Cross-validation procedures is run terribly simply using powerful CV iterators (inspired by scikit-learn wonderful tools), additionally as thorough search over a group of parameters.

• SurPRISE (roughly :)) stands for easy Python RecommendatIon System Engine.[8]

• Surprise doesn't support inexplicit ratings or content-based information.

OUR PROPOSED RECOMMENDATION MODEL



III. OUR PROPOSED RECOMMNDATION MODEL

In machine learning we are ready to determine the advanced decision boundaries for classifications and advanced nonlinear regressions. We are ready to find the hidden patterns and learning to extract each low-level and high-level element from the information and additionally composing them in classification. Data analysis steps demonstrates the utilization of various knowledge manipulation and correlation of various connected related are underlined during this project.

In a formal definition, recommendation systems address the subsequent problem: Given a group of items A, a subset B of A, with |B| << |A| and also the values f(b) ∀b∈B of a function f(x), A-> {0,1}, wherever the closed form of f(x) is unknown, find an estimate of f(x), specifically g(x), A-> {0,1}, and a subset C of A, with |C| << |A| and B∩C = ∅ , in order that the 1 P(g(c) = f(c) = one), ∀c∈C, is maximized.

The common target set of f(x) and g(x) are often additionally numerical, using a sort of threshold completely differentiate between different categories. The problems are often divided into three separate sub-problems that address totally different areas of the problem. Before we are able to apply the mathematical models and the algorithms to form an estimate for the user's interest, we should have to create a illustration of the real world items, in order that they will be processed by an algorithm.

Another words, we have to determine the outline that may be used for classifying the items. Then, we should decide that what model or algorithmic rule we are going to use to make an estimate for the user's interests and execute it into a software. Finally, so as to set that we tend to use this estimate if we should always recommend an item or not. the items that appear to match the user's interests, that much are chosen and presented the user. we are going to analyse each sub problem within the remainder of section and presents the range of solutions to them, and also the limitations, merits and demerits of those solutions.

We show that however these subproblems are interconnected to solve the initial problem and the way selections for one subproblems would possibly have an effect on for the selections of the another ones. Once it had been out there, we tend to provide the experimental data through the accuracy of that results that is made by that technique. At first the information has got to be pre-processed using natural language processing and to get only one column that contains all the columns of every movie. Then, these informations are converted into numbers by vectorizations, wherever scores are assigned to every word.

Subsequently cosine similarities will be calculated.Once i get the matrix that containing the count for all words, I function apply the cosine similarity perform to create a comparision between movies. The final step is that to make a function that takes title as input in a movie, and returns the highest ten alike movies as the output. This function can match the input movie title with the Similarity Matrix,then take away the similarity values within the dropping order. the highest ten alike movies are going to be found by getting the highest eleven values and later throw out the first index that is that the input movie .

IV. EXPERIMENTAL ANALYSIS & RESULTS

In proposed model we tend to use a pre filter before applying Naive Bayes algorithm. The attributes used to calculate distance of every purpose from centre of mass are one. Genre 2, Actor 3,Director 4,Year 5. Rating completely different attributes have different weights. In our analysis we've found that the foremost acceptable recommendations which will be generated should be supported the ratings given to the films by previous users, thus we've given additional importance to the rating attribute than different attributes. These ratings are taken from www.imdb.com as a result of maybe it's the biggest collection of movies along with the rating given to those movies by a large number of users from different parts of the universe. Another necessary parameter in our proposed model is total number of votes received by a specific movie. We've divided number of votes in to three classes that is but or less than or equal to 1000, more than 1000 however less than or equal to 10,000 and bigger than 10000.

Data sources:

A. IMDB 5000 Movie Dataset | Kaggle
B. The Movies Dataset | Kaggle
C. List of American films of 2018 - Wikipedia
D. List of American films of 2019 - Wikipedia
E. List of American films of 2020 - Wikipedia

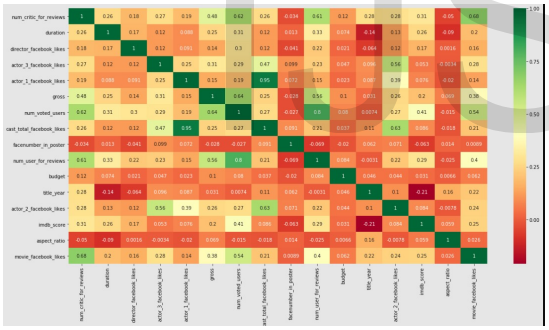
	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	gross
0	Color	James Cameron	723.0	178.0	0.0	855.0	Joel David Moore	1000.0	760505847.0
1	Color	Gore Verbinski	302.0	169.0	563.0	1000.0	Orlando Bloom	40000.0	309404152.0
2	Color	Sam Mendes	602.0	148.0	0.0	161.0	Rory Kinnear	11000.0	200074175.0
3	Color	Christopher Nolan	813.0	164.0	22000.0	23000.0	Christian Bale	27000.0	448130642.0
4	NaN	Doug Walker	NaN	NaN	131.0	NaN	Rob Walker	131.0	NaN

	genres	actor_1_name	movie_title	num_voted_users	cast_total_facebook_likes	actor_3_name	facenumber_in_poster	
	Action|Adventure|Fantasy|Sci-Fi	CCH Pounder	Avatar	886204	4834	Wes Studi	0.0	avatar|future|marine
	Action|Adventure|Fantasy	Johnny Depp	Pirates of the Caribbean: At World's End	471220	48350	Jack Davenport	0.0	goddess|marriage c
	Action|Adventure|Thriller	Christoph Waltz	Spectre	275868	11700	Stephanie Sigman	1.0	bomb|espionage|s
	Action|Thriller	Tom Hardy	The Dark Knight Rises	1144337	106759	Joseph Gordon-Levitt	0.0	deception|imprisonment|s
	Documentary	Doug Walker	Star Wars: Episode VII - The Force Awakens	8	143	NaN	0.0	

plot_keywords	movie_imdb_link	num_user_for_reviews	language	country	content_rating	budget	title_year	actor
avatar|future|marine|native|paraplegic	http://www.imdb.com/title/tt0499549/?ref_=fn_t...	3054.0	English	USA	PG-13	237000000.0	2009.0	
goddess|marriage ceremony|marriage proposal|pi...	http://www.imdb.com/title/tt0449088/?ref_=fn_t...	1238.0	English	USA	PG-13	300000000.0	2007.0	
bomb|espionage|sequel|spy|terrorist	http://www.imdb.com/title/tt2379713/?ref_=fn_t...	994.0	English	UK	PG-13	245000000.0	2015.0	
deception|imprisonment|lawlessness|police offi...	http://www.imdb.com/title/tt1345836/?ref_=fn_t...	2701.0	English	USA	PG-13	250000000.0	2012.0	
NaN	http://www.imdb.com/title/tt5289954/?ref_=fn_t...	NaN	NaN	NaN	NaN	NaN	NaN	

	director_name	actor_1_name	actor_2_name	actor_3_name	genres	movie_title	comb
0	James Cameron	CCH Pounder	Joel David Moore	Wes Studi	Action Adventure Fantasy Sci-Fi	avatar	CCH Pounder Joel David Moore Wes Studi James C...
1	Gore Verbinski	Johnny Depp	Orlando Bloom	Jack Davenport	Action Adventure Fantasy	pirates of the caribbean: at world's end	Johnny Depp Orlando Bloom Jack Davenport Gore ...
2	Sam Mendes	Christoph Waltz	Rory Kinnear	Stephanie Sigman	Action Adventure Thriller	spectre	Christoph Waltz Rory Kinnear Stephanie Sigman ...
3	Christopher Nolan	Tom Hardy	Christian Bale	Joseph Gordon-Levitt	Action Thriller	the dark knight rises	Tom Hardy Christian Bale Joseph Gordon-Levitt ...
4	Doug Walker	Doug Walker	Rob Walker	unknown	Documentary	star wars: episode vii - the force awakens ...	Doug Walker Rob Walker unknown Doug Walker Doc...

Data corelation heatmap(movie-metadata.csv)



V. CONCLUSION:

We have proposed a simple Naive-Bayes and NLTK Model for content based filtering and designed a collaborative filtering approach using Scikit-learn-Surprise which performs a good movie recommendation System. This work makes refinement in terms of forecasting ratings and suggesting movies for the users. Our recommendation system use an API of **TMDBV3API** for getting additional movie information and movie start casts information from IMDB. According to users comment it classifies which comment is good or bad by sentiment analysis to provide best recommendation to the users.

References:

[1] McCallum, Andrew. "Graphical Models, Lecture2: theorem Network Represention" (PDF). Retrieved twenty two Gregorian calendar months 2019.

[2] Piryonesi,S.Madeh; El-Diraby, doi:10.1061/JPEODX.0000175, Tamer E. (2020-06-01).Journal of Transportation Engineering, half B: Pavements. 146 (2): 04020022.

[3] Hastie, Trevor. (2001). data processing, inference, and prediction: with two hundred full-color illustrations. Tibshirani, Robert. Friedman, J. H. (Jerome H.). New York: Springer. ISBN 0-387-95284-5. OCLC 46809224.

[4] Russell, Stuart; Norvig, Peter (2003) [1995]. Artificial Intelligence: a contemporary Approach (2nd impotence.). Tyro Hall. ISBN 978-0137903955.

[5] Hand, D. J.; Yu, K. (2001). "Idiot's mathematician — not thus stupid when all?". International applied mathematics Review. Sixty nine (3): 385–399. Doi: 10.2307/1403452. ISSN 0306-7734. JSTOR 1403452. [6] W. Croft, D. Metzler, and T. Strohman. Search Engines: data Retrieval in apply. Addison Wesley, 2010.

[7] Doruk Kilitcioglu, Ravi Raju avatar, Victor Wang, microphone Lee Williams, Jay Wong, Chenchen Xu, YaoZh1918. Hengji Liu, Maher Malaeb, Manoj K, James McNeilis, Naturale0, nju-luke, Jay Qi, Nicolas Hug, Janniks, jyesawtellrickson, screenwriter Rebscher, Skywhat, David Stevens, TrWestdoor,

[8] T. Chai and R. Draxler. Root Mean sq. Error (RMSE) or Mean Absolute Error (MAE)? Geoscientific Model Development Discussions, 7(1):1525–1534, 2014.

[9] C. Gomez-Uribe and N. Hunt. Algorithms, Business worth, and Innovation. ACM TMIS, 6(4): Article thirteen, 2016.

[10] F. instrumentalist and J. Konstan. The Movielens Datasets: History and Context. ACM TiiS, 5(4): Article nineteen, 2016.

[11] Y. Koren, R. Bell, and C. Volinsky. Matrix resolution Techniques for Recommender Systems. Computer, 42(8):30– thirty seven, 2009.

[12] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. Alsaadi. A Survey of Deep Learning Neural Network Architectures related to Their Applications. Neurocomputing, 234:11–26, 2017.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In IEEE CVPR, pages 770–778, 2016.

[14] "Modern data Retrieval: Singhal, Amit (2001). a short Overview". Bulletin of the IEEE laptop Society Technical Committee on knowledge Engineering twenty four (4): 35–43.

[15] ^ P.-N. Tan, M. Steinbach & V. Kumar, Introduction to data processing, Addison-Wesley (2005), ISBN 0-321-32136-7, chapter 8; page five hundred.[18] ^ "COSINE DISTANCE, COS SIMILARITY, www.itl.nist.gov. Retrieved 2020-07-11.

1st Author Name: Pratibha Jyoti Negi
Branch: MTech
Email: 19mtcs022.pratibhajyotinegi@giet.edu
Department of computer science engineering
GIET University, Gunupur, Odisha, 765022

2nd Author Name: Dr. Brojo Kishore Mishra
Email: bkmishra@giet.edu
Professor, Department of computer science engineering
GIET University, Gunupur, Odisha, 765022