PROJECT REPORT ON

IMDB SENTIMENTAL ANALYSIS & RECOMMENDATION SYSTEMS

SUBMITTED IN PARTIAL REQUIREMENT FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE

SUBMITTED BY

P CHARISHMA	G VENNELA	S SRAVANTHI
17011A0509	18015A0526	18015A0526

UNDER THE SUPERVISION OF

Dr K P SUPREETHI

PROFESSOR OF CSE, JNTUHCEH

DEPARTMENT OF COMPUTER SCIENCE

JNTUH COLLEGE OF ENGINEERING

KUKATPALLY, HYDERABAD TELANGANA

2020 - 2021

J.N.T.U.H. COLLEGE OF ENGINEERING
KUKATPALLY, HYDERABAD – 500 085

CERTIFICATE

This is to certify that the thesis entitled "***IMDB SENTIMENTAL ANALYSIS & RECOMMENDATION SYSTEM***" submitted to JNTUH college of engineering, Hyderabad by *P. CHARISHMA, G. VENNELA, S. SRAVANTHI* for the award of *BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE* is a record of bonafide research work carried by them. The results embodied in this thesis have been not submitted to any other university or institution for the award of any degree

Supervisor	**Head of the Department**
Dr K P SUPREETHI	**Dr M CHANDRA MOHAN**
PROFESSOR OF CSE	**PROFESSOR OF CSE & HOD**
JNTUHCEH, HYDERABAD	**JNTUHCEH, HYDERABAD**

DECLARATION

We hereby declare that the project work entitled "***IMDB SENTIMENTAL ANALYSIS & RECOMMENDATION SYSTEM***" submitted to the submitted to JNTUH college of engineering, Hyderabad is a record of an original work carried out by *P. CHARISHMA, G. VENNELA, S. SRAVANTHI* under the guidance of *Dr K P SUPREETHI, PROFESSOR OF CSE, JNTUHCEH.*This report is submitted in partial fulfilment of requirement for the award of BACHELOR OF TECHNOLOGY.I declare that the work reported in this report has not been submitted and will not be submitted to any university or institution for the award of degree.

SIGNATURE

P. CHARISHMA [17011A0509]

G. VENNELA [18015A0511]

S. SRAVANTHI [18015A0526]

ACKNOWLEDGEMENT

We would like to express my gratitude to our project supervisor *Dr K P SUPREETHI, PROFESSOR OF CSE, JNTUHCEH.* For her guidance and constant encouragement throughout the course of the desertion work and I also thank *Dr M. CHANDRA MOHAN, Professor of CSE & Head of the Department, JNTUHCEH* for his valuable support and motivation regarding project work. I also thank all the faculty of the DEPARTMENT OF COMPUTER SCIENCE for their valuable suggestion at all our approaches.

DATE

SIGNATURE

PLACE
KUKATPALLY, HYDERABAD

P. CHARISHMA [17011A0509]
G. VENNELA [18015A0526]
S. SRAVANTHI [18015A0526]

INDEX

ABSTRACT

Any opinion of an individual through which the feelings, attitudes and thoughts can be expressed is known as sentiment. The kinds of data analysis which is attained from the news reports, user reviews, social media updates or microblogging sites is called sentiment analysis which is also known as opinion mining. The responses of general public are collected and improvised by researchers to perform evaluations. In essence, it is the process of determining the emotional tone behind a series of words, used to gain an understanding of the attitudes, opinions and emotions express on social media and other online websites. Break each text document down into its component parts.

IMDb (an acronym for Internet Movie Database) is an online database of information related to films, television programs, home Movies, video games, and streaming content online – including cast, production crew and personal biographies, plot summaries, trivia, ratings, and fan and critical reviews. The online users express their opinions, views and sentiments on the Movies. An additional fan feature, message boards, was abandoned.

The main objective of this project is to assign a user likability of the movie. It is done by finding the emotions for comments by training a model on data collected from various sources, and also recommend movies. This project presents a brief survey of techniques to analyse opinions/views posted by users about a particular movie to assign a rating.

LIST OF FIGURES

INTRODUCTION

A large amount of data that is generated today is unstructured, which requires processing to generate insights. Some examples of unstructured data are news articles, posts on social media, and search history. The process of analysing natural language and making sense out of it falls under the field of **Natural Language Processing** (NLP). Sentiment analysis is a common NLP task, which involves classifying texts or parts of texts into a predefined sentiment. You will use the **Natural Language Toolkit** (NLTK), a commonly used NLP library in Python, to analyse textual data.

We will prepare a dataset of sample tweets from the NLTK package for NLP with different data cleaning methods. Once the dataset is ready for processing, you will train a model on pre-classified tweets and use the model to classify the sample tweets into negative and positive sentiments.

The Dataset's will be IMDB datasets, where **IMDb** (an acronym for **Internet Movie Database)** is an online database of information related to films, television programs, home videos, video games, and streaming content online – including cast, production crew and personal biographies, plot summaries, trivia, ratings, and fan and critical reviews. An additional fan feature, message boards, was abandoned in February 2017. Originally a fan-operated website, the database is now owned and operated by IMDb.com, Inc., a subsidiary of Amazon.

SENTIMENTAL ANALYSIS:

FIG 1.1 MODEL FOR SENTIMENTAL ANALYSIS WORK FLOW

Lemmatization: It entails reducing the various inflected forms of a word into a single form for easy analysis.

Morphological segmentation: It involves dividing words into individual units called morphemes. **Word segmentation:** It involves dividing a large piece of continuous text into distinct units.

Part-of-speech tagging: It involves identifying the part of speech for every word

Parsing: It involves undertaking grammatical analysis for the provided sentence.

Sentence breaking: It involves placing sentence boundaries on a large piece of text.

Stemming: It involves cutting the inflected words to their root form.

PROBLEM STATEMENT

Based on the massive movie information, analyzing the factors that make a movie successful. In other words, gaining a high IMDB score. We also want to show the results of this analysis in an intuitive way. In this project, we take IMDB scores as response variables and focus on operating predictions by analyzing the rest of variables in the IMDB. The results can help film companies to understand the secret of generating a commercial success movie.

To make effective search movies are recommended to users based on likability

OBJECTIVE:

By the review's of the user, the rating of particular movie is predicted.
To develop a user interface where the content-like of the user are provided
Based on the preference the recommendation of a particular movie is predicted
The related movie's based and the likability is projected.

SCOPE:

Prediction of the rating of the movie
Recommendation of that movie based on rating and user likeable content.

SOFTWARE REQUIREMENTS:

Programming Language : Python
Inbuilt Python Libraries:

Keras.

Tkinter

Keras.layers

Keras.modelspickle

Environment : Anaconda

DataSet : imdb.csv,user.csv,movie.csv

Library :Algorithm : Sequential model for categorical regression

OPERATING SYSTEM:WINDOWS,MAC

HARDWARE REQUIREMENTS:

RAM:4GB

HARDDISK:1TB

CONCEPT:

1. Rating prediction if done using the keras sequential model activation layer's relu and softmax and the utiis, pickle for predicting and storing the model computed.
2. predected rating will be assigned to the particular movie
3. The user will login to the system
4. The login username is taken for the validating the user authority to login in to the system
5. Based on the content-like of the user using content-based filtering the movie is recommended
6. If the movie recommendation is valid then the related-movie's of the same genre will be displayed.

EXISTING SYSTEM

In the existing system,

Rating of the movie:

IMDB provides the rating based on the review given by the users using the IMDBpy (LINK) to load relational data into AgensGraph in relational form.

Recommendation of the movie:

Recommendation's are not provided by the website but search bar will be provided for searching and retrieving the related movies.

Related movies are provided by the system based on the rating-based filtering, that make the recommended movies of the high rating will be displayed.

Rating-based collaborative filtering recommender systems do this by finding patterns that are consistent across the ratings of other users.

Disadvantages of rating-based filtering-

Rating-based filtering the highest rating of the movie is taken in to the account and which make's user to think for movie selection

The website doesn't use the content-based filtering, which makes the user cannot predict what movie he should select from the entire bundle of movie's recommended.

PROPOSED STATEMENT

This investigates and outlines a portable framework for movie rating and audit rundown in which semantic introduction of remarks. For all intents and purposes, when we are not acquainted with a particular item, we request that our trusted sources suggest one. Today, the prominence of the Internet drives individuals to look for conclusions from the Internet before acquiring an item or seeing a film. For instance, the client audit area in Amazon.com records the quantity of surveys, the rate for various appraisals, and remarks from commentators. At the point when individuals need to buy books, CDs, or DVDs, these remarks, and evaluations, as a rule, impact their buying practices. User is interested in items of similar category thus providing recommendation based on user interest.

SENTIMENT ANALYSIS IN THE SYSTEM:

Sentiment analysis is a common NLP task, which involves classifying texts or parts of texts into a pre-defined sentiment. Essentially, sentiment analysis or sentiment classification fall into the broad category of text classification tasks where you are supplied with a phrase, or a list of phrases and your classifier is supposed to tell if the sentiment behind that is positive, negative or neutral. Sometimes, the third attribute is not taken to keep it a binary classification problem. by providing user reviews(unlabeled) it assigns the labels of their respective sentiments to the machine learning model, and analyzes the sentiment based rating to the movie

A recommendation system provides suggestions to the users through a filtering process that is based on user preferences and browsing history. The information about the user is taken as an input. The information is taken from the input that is in the form of browsing data. This information reflects the prior usage of the product as well as the assigned ratings. A recommendation system is a platform that provides its users with various contents based on their preferences and likings. A recommendation system takes the information about the user as an input. The recommendation system is an implementation of the machine learning algorithms.

There are two types of recommendation systems

- Content-Based Recommendation System
- Collaborative Filtering Recommendation.

Benefit's of Content-Based Recommendation System:

- The model doesn't need any data about other users, since the **recommendations** are specific to this user. This makes it easier to scale to a large number of users. The model can capture the specific interests of a user, and can recommend niche items that very few other users are interested in.

- We don't need domain knowledge because the embeddings are automatically learned. The model can help users discover new interests. In isolation, the ML **system** may not know the user is interested in a given item, but the model might still recommend it because similar users are interested in that item.

- generate recommendations for each unique user **based** on how similar users liked the item. In other words, this method creates a matching system of educated guesses.

IMPLEMENTATION

DATA LOADING:

- LOAD THE DATASET OF IMDB, USER, MOVIES DATASET’S:

```
1.read CSV file of IMDB dataset for
the for the rating computation
2.read the user dataset for the user
validation for login
3.read the movie dataset for the
related movie’s display
```

- USER INTERFACE:

```
1.Open python interface from using
tkinter package
2.Input the Username by this command:
      L1 = Label(top, text="UserName")
      E1 = Entry(top, bd =5)
3.Input the Password by this command:
      L2 = Label(top, text=“Passcode")
      E2 = Entry(top, bd =5,show=“*”)
4.Append the submit button for the
login.
```

DATA PREPROCESSING:

Data pre-processing is done using the tokenizer and label encoder:

TOKENISING:

The tokenize module provides a lexical scanner for Python source code, implemented in Python. The scanner in this module returns comments as tokens

1. Tokenize:

- Tokenizer is used to create token's for the sentence
- below command is used to tokenzie-

tokenize=Tokenizer(num_words=max_words,char_level=False)

2.Indexing:

- Tokenizer is used to give indexes for each and every unique word
- tokenize.index_word is the command used

3.Term frequency:(TF)

Term frequencies computed using tokenize.word_docs command

It is a measure of how frequently a term, t, appears in a document, d:

$$tf_{t,d} = \frac{n_{t,d}}{\text{Number of terms in the document}}$$

Here, in the numerator, n is the number of times the term "t" appears in the document "d". Thus, each document and term would have its own TF value.

We will again use the same vocabulary we had built in the Bag-of-Words model to show how to calculate the TF for Review #2:

Review 2: This movie is not scary and is slow

Here,

- Vocabulary**:** 'This', 'movie', 'is', 'very', 'scary', 'and', 'long', 'not', 'slow', 'spooky', 'good'
- Number of words in Review 2 = 8
- TF for the word 'this' = (number of times 'this' appears in review 2)/(number of terms in review 2) = 1/8

Similarly,

- TF('movie') = 1/8
- TF('is') = 2/8 = 1/4
- TF('very') = 0/8 = 0
- TF('scary') = 1/8
- TF('and') = 1/8
- TF('long') = 0/8 = 0
- TF('not') = 1/8
- TF('slow') = 1/8
- TF('spooky') = 0/8 = 0
- TF('good') = 0/8 = 0

it can calculate the term frequencies for all the terms and all the reviews in this manner:

Term	Review 1	Review 2	Review 3	TF (Review 1)	TF (Review 2)	TF (Review 3)
This	1	1	1	1/7	1/8	1/6
movie	1	1	1	1/7	1/8	1/6
is	1	2	1	1/7	1/4	1/6
very	1	0	0	1/7	0	0
scary	1	1	0	1/7	1/8	0
and	1	1	1	1/7	1/8	1/6
long	1	0	0	1/7	0	0
not	0	1	0	0	1/8	0
slow	0	1	0	0	1/8	0
spooky	0	0	1	0	0	1/6
good	0	0	1	0	0	1/6

TABLE 5.1: CALCULATION OF THE TERM FREQUENCIES OF REVIEW 1,2,3

We plan and build up a film rating and survey rundown framework in a versatile domain. The motion picture rating data depends on the assumption characterization result. The consolidated portrayals of film audits are produced from the component-based synopsis. We propose a novel approach in view of inert semantic investigation (LSA) to distinguish item includes. Besides, we discover a way to diminish the measure of rundown in light of the item includes acquired from LSA. We consider both notion grouping precision and framework reaction time to outline the framework. The rating and survey rundown framework can be reached out to other item audit areas effectively.

LABEL ENCODER:

- **LabelEncoder** can be used to normalize labels. It can also be used to transform non-numerical labels (as long as they are hashable and comparable) to numerical labels.

- **label encoding in Python**, we replace the categorical value with a numeric value between 0 and the number of classes minus 1.
- Below command are used for lable encoding:

```
encoder=LabelEncoder()
encoder.fit(train_sent)
```

DATA TRAINING

Data training is done by the Sequential model of keras which have the layer activation functions and keras utils, pickle are used.

RELU FUNCTION:

The ReLU function is another non-linear activation function that has gained popularity in the deep learning domain. ReLU stands for Rectified Linear Unit. The main advantage of using the ReLU function over other activation functions is that it does not activate all the neurons at the same time.

This means that the neurons will only be deactivated if the output of the linear transformation is less than 0. The plot below will help you understand this better-
$f(x)=max(0,x)$

For the negative input values, the result is zero, that means the neuron does not get activated. Since only a certain number of neurons are activated, the ReLU function is far more computationally efficient when compared to the sigmoid and tanh function.

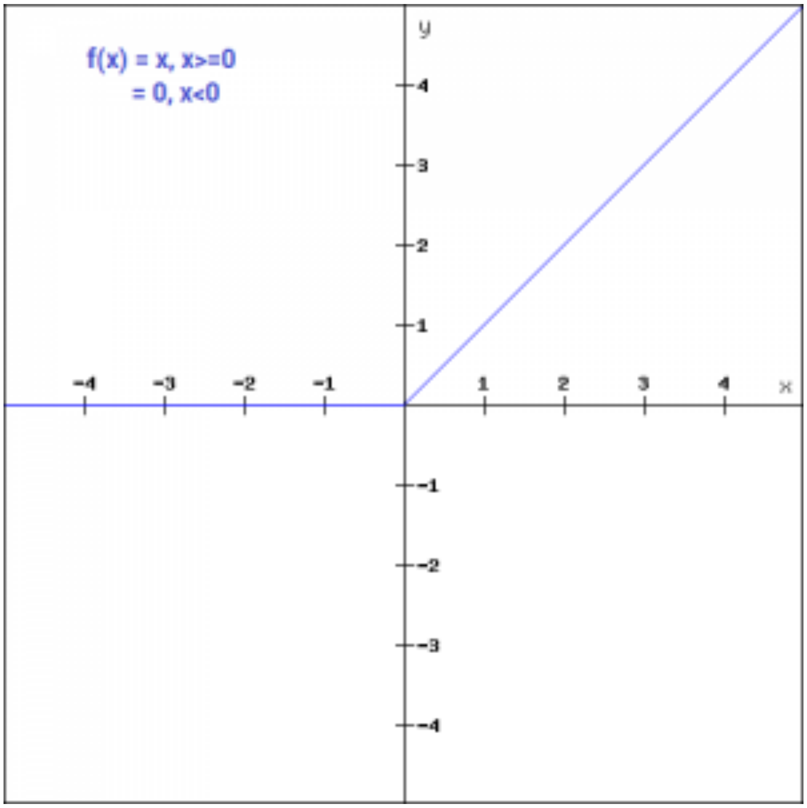

FIG 5.1: GRAPH FOR f(x)=max(0,x) RELU FUNCTION

SOFTMAX:

Softmax function is often described as a combination of multiple sigmoids. We know that sigmoid returns values between 0 and 1, which can be treated as probabilities of a data point belonging to a particular class. Thus sigmoid is widely used for binary classification problems.

The softmax function can be used for multiclass classification problems. This function returns the probability for a datapoint belonging to each individual class. Here is the mathematical expression of the same-

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

While building a network for a multiclass problem, the output layer would have as many neurons as the number of classes in the target. For instance if you have three classes, there would be three neurons in the output layer. Suppose you got the output from the neurons as [1.2 , 0.9 , 0.75].

Applying the softmax function over these values, you will get the following result – [0.42 , 0.31, 0.27]. These represent the probability for the data point belonging to each class. Note that the sum of all the values is 1.

KERAS LOSES:

CategoricalCrossentropy class is used for the keras loses for the computing of crossentropy loss between the labels and predictions.

Use this crossentropy loss function when there are two or more label classes. We expect labels to be provided in a one_hot representation. If you want to provide labels as integers, please use SparseCategoricalCrossentropy loss. There should be # classes floating point values per feature.

In the snippet below, there is # classes floating pointing values per example. The shape of both y_pred and y_true are [batch_size, num_classes].

Following command is used for this module:

model.compile(loss="categorical_crossentropy",optimizer="adam",metrics=['accuracy'])

KERAS UTILS:

- The **utils** module contains classes and funtions of general utility **used** in multiple places throughout astropysics . Some of these are astropyhysics-specific algorithms while others are more python tricks. The **utils** module is composed of three submodules to make organization clearer
- utils.to_categorical is used for the binary matrix representation of the input. The classes axis is placed last.
- In order to find out the accuracy, we use fit function.

history=model.fit(x_train,y_train_new,batch_size=32,epochs=2,verbose=1,validation_split=0.1)

An **epoch** is a term used in **machine learning** and indicates the number of passes of the entire training dataset the **machine learning** algorithm has completed.

If the batch size is the whole training dataset then the number of **epochs** is the number of iterations.

Units of epoch:us/step

Where us=units of second

- evaluation

Model evaluation aims to estimate the generalization accuracy of a model on future (unseen/out-of-sample) data.

Methods for evaluating a model's performance are divided into 2 categories: namely, holdout and Cross-validation

KERAS PICKLE:

Pickle is the standard way of serializing objects in Python. You can use the pickle operation to serialize your machine learning algorithms and save the serialized format to a file. Later you can load this file to deserialize your model and use it to make new predictions.

```
import pickle

f=open('tok.pkl','wb')

pickle.dump(tokenize,f)
```

DATA TESTING:

Data testing is done by the 2 modules:

RATING PREDICTION:

PSEUDO CODE:

```
1.IMPORT TKINTER PACKAGE FOR THE GUI AND KERAS.MODELS,
KERAS.PREPROCESSING FOR TESTING THE MODULE

2.OPEN PICKLE FILE AND EXTRACT THE TOKENIZE ALGORITHMS

3.CREATE A PYTHON GUI TKINTER WIDGET WITH FUNCTION TK()

4.INITIALIZE MOVIE AND GENRE NAME VARIABLE'S

5.CREATE LABEL AND ENTRY FOR REVIEWER INPUT AND GRID
THEM

L2 = Label(top, text="Reviewer1")

E1 = Entry(top , bd =5)

L2.grid(row=1,column=0)
```

E1.grid(row=1,column=1)

6.USE KERAS LOSSES MODEL.COMPILE FUNCTION FOR ACCURACY

7.DEFINE A FUNCTION PREDICT()

a.CREATE GLOBAL VARIABLE PRED FOR PREDICTION

b.INITIALISE RATING=0

c.TRAIN_DATA RATING PREDICTION:

i.FOR PREDICTING THE NEG OR POS,USE PREDICT_CLASSES() FUNCTION AND STORE IN Y_PRED

ii.IF Y_PRED[0]==1 THEN INCREMENT RATING WITH Y_PRED[0]

d.INITIALIZE ENTERD_INPUT=[] AND APPEND E1.GET(),E2.GET(),E3.GET(),E4.GET(),E5.GET()

e.FOR EVERY ELEMENT IN ENTERED_INPUT:

i.USE TOKENIZE.TEXTS_TO_MATRIX([ENTERED_INPUT[I]]) FOR TOKENIZING

ii.FOR PREDICTING THE NEG OR POS,USE PREDICT_CLASSES() FUNCTION AND STORE IN Y_PRED

iii.IF Y_PRED[0]==0 THEN INITIALISE text1="The input is a NEGATIVE sentiment”

OR

IF Y_PRED[0]==1 THEN INCREMENT RATING WITH Y_PRED[0] AND INITIALISE text1="The input is a POSITIVE sentiment"

iv.CREATE LABEL FOR DISPLAYING THE TEXT1 AND GRID THEM

f.CREATE LABEL FOR DISPLAYING THE RATING AND GRID THE LABEL

```
    g.IF RATING GREATER AND EQUAL 7.5:
        i.INITIALIZE PRED=1
        ii.DECLARE TEXT2="GOOD MOVIE"
      OR
        i.INITIALIZE PRED=1
        ii.DECLARE TEXT2="GOOD MOVIE"
    h.CREATE LABEL FOR DIAPLAYING TEXT2 AND GRID THE
LABEL
8.CREATE BUTTON PREDICT WITH COMMAND PREDICT1 AND GRID
THE BUTTON
9.TOP.MAINLOOP()
```

<u>RECOMMENDATION:</u>

RECOMMENDATION MODULE:

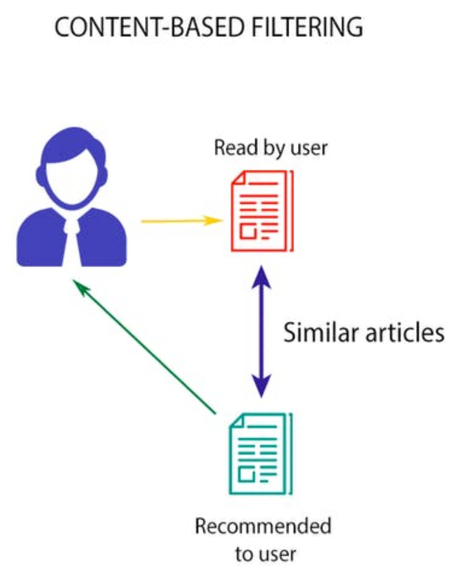

FIG 5.2 - MODULE FOR CONTENT-BASED FILTERING FOR RECOMMENDATION

Recommendation model uses the content-based filtering, where this filtering is for thegenerate recommendations for each unique user **based** on how similar users liked the item. In other words, this method creates a matching system of educated guesses

PSEUDO CODE:

```
1.IMPORT TKINTER PACKAGE AND PANDAS PACKAGE
2.CREATE A PYTHON GUI TKINTER WIDGET WITH FUNCTION TK()
3.CREATE LABEL AND ENTRY FOR USERNAME AND PACK IT
4.CREATE LABEL AND ENTRY FOR PASSWORD, WHERE SHOW="*"AND
PACK IT
5.DEFINE FUNCTION MOVIE_PRED():
    a.READ THE CSV FILE USER.CSV AND STORE IN DATA1
b.READ CSV FILE MOVIES.CSV AND STORE IN MOVIES
c.CONVERT DATA1.USERNAME SERIES TO LIST AND STORE IN
DATA1_LIST
d.IF ENTRY1 IN DATA1.LIST ARE EQUAL AND ENTRY2 IS
"12345":
    i.STORE INDEX OF ENTRY1 IN DATA1_LIST IN I
    ii..CREATE A PYTHON GUI TKINTER WIDGET WITH FUNCTION
TK()
    iii.DECLARATION OF LISTBOX
    iv.IF PRED EUQAL TO 1 AND GENRE EQUAL TO
DATA1.GENRE[I]:
     A. CREATE LABEL FOR DISPLAYING GENRE AND PACK IT
```

B. CREATE LABEL FOR DISPLAYING THE MOVIE AND PACK IT

C. CREATE LABEL FOR DISPLAYING RELATED MOVIES AND PACK IT

FOR EVERY ELEMENT IN MOVIES:

IF GENRE EQUAL TO MOVIES.GENRE ELEMENT:

INSERT RESPECTIVE FILMS TO THE LISTBOX AND DISPLAY IT

OR

A. NO RECOMMENDATION’S FOR PARTICULAR USER

OR

i.POP UP INVALID LOGIN WIDGET USING LABEL’S

6.CREATE BUTTON PREDICT WITH COMMAND MOVIE_PRED AND GRID THE BUTTON

9.TOP.MAINLOOP()

DESIGN

USE CASE DIAGRAM:

- The first step in the process is user will login to the system.
- The login username is taken for the validating the user authority to login in to the system
- Based on the content-like of the user using content-based filtering the movie is recommended
- If the movie recommendation is valid then the related-movie's of the same genre will be displayed.

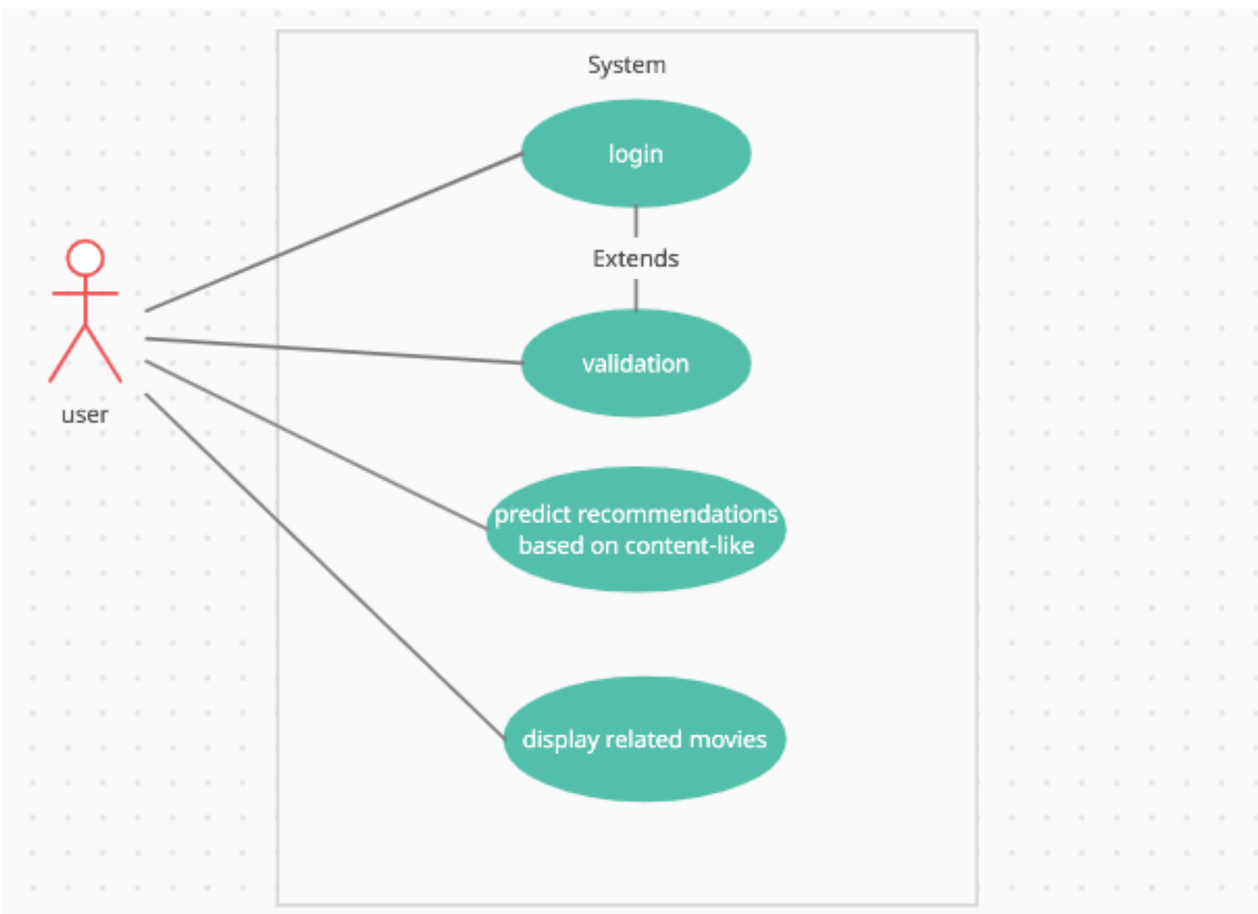

FIG 6.1 - USE CASE DIAGRAM OF USER LOGIN AND RECOMMENDATION

ACTIVITY DIAGRAM:

The work flow of this model will be as follows:

1.load the input data from the data set and divide it into training data and testing data

2.data cleaning of stop words are initiated

3.tokenization is used for the formation of tokens, indexing the tokens and computing the term frequency of each term

4.label encoder is used for the normalize labels, replace the categorical value with a numeric value between 0 and the number of classes minus 1

5.training of the data is begin by the activation fuction's relu and softmax of keras for high-level regression

6.keras utiliti is used to the prediction of the model, which can be negative or postive

7.keras pickle is used to store the model and retreive it to use

8.using testing data, using predictclasses function ,using user data testing it done and rating is computed

9.based of rating and user liked genre the movie recommendation is done, it the genre is equivalent

10.the movies related to the same genre will be displayed below.

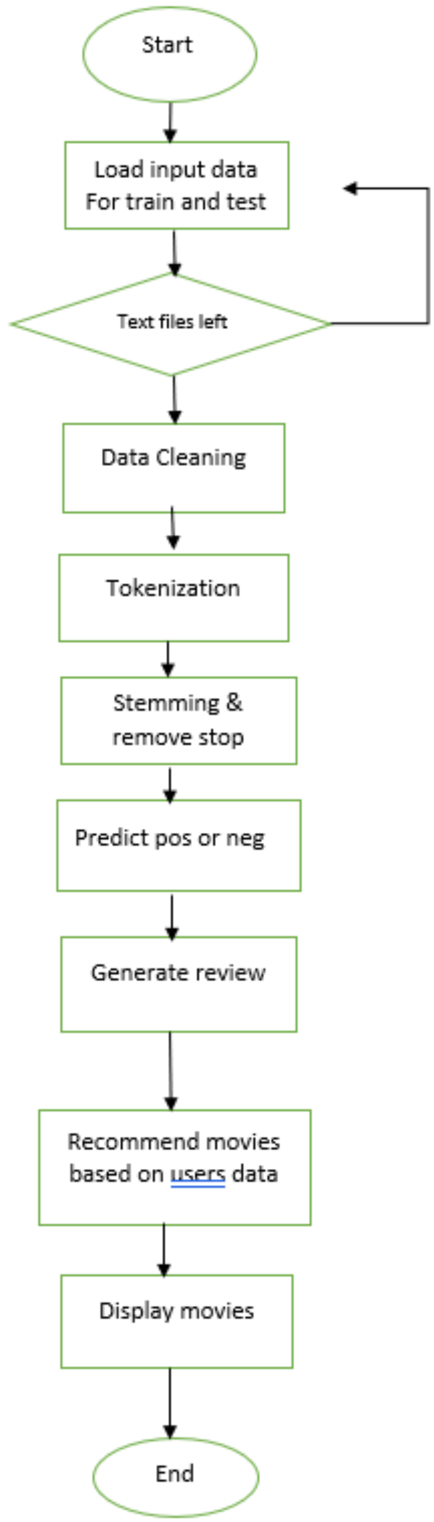

FIG 6.2 - ACTIVITY DIAGRAM OF THE RATING PREDICTION AND RECOMMENDATION

RESULTS

RATING PREDICTION:

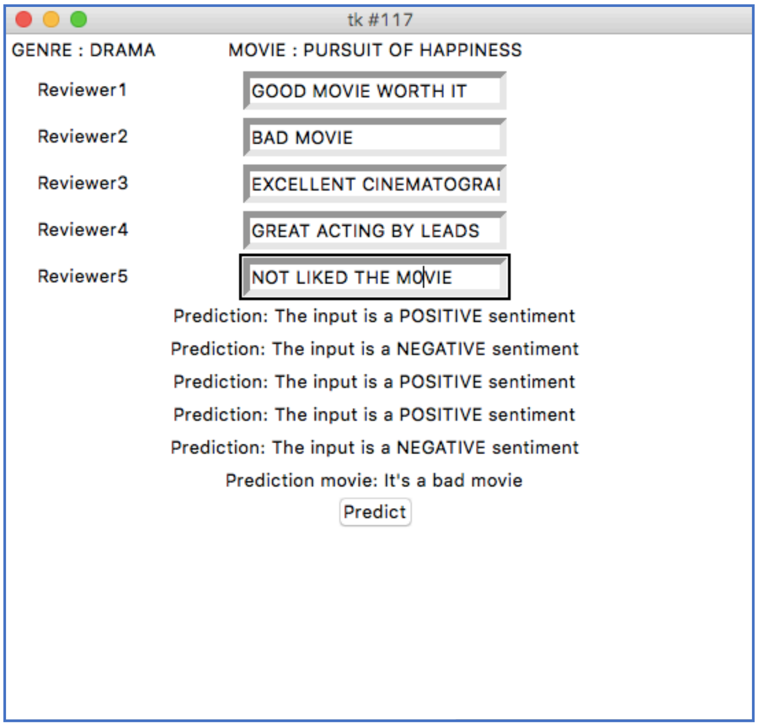

FIG 7.1 - WIDGET FOR RATING PREDICTION MODEL

USER INTERFACE:

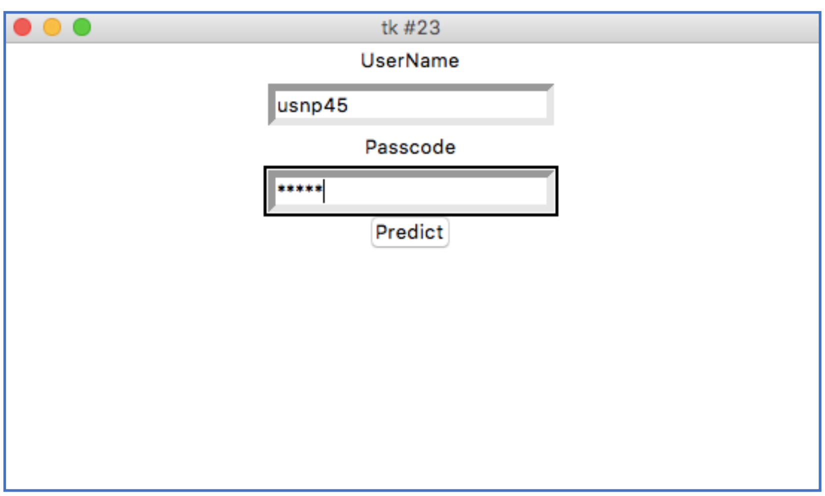

FIG 7.2 - WIDGET FOR USER LOGIN

NO RECOMMENDATION WIDGET:(if genre un-match or rating too low)

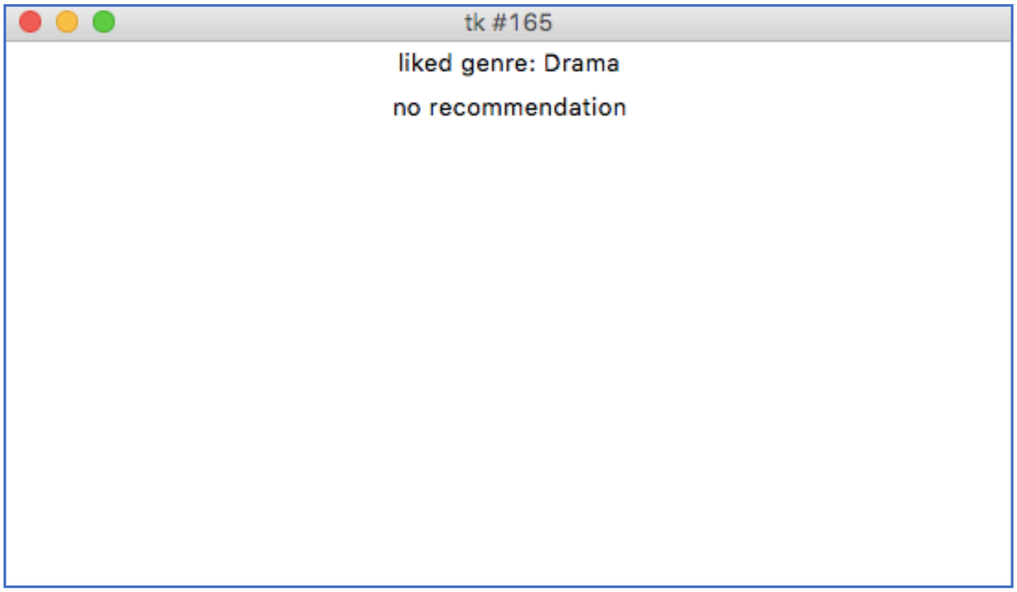

FIG 7.3 - NO MOVIE RECOMMENDATION FOR GENRE UNMATCH OR LOW RATING

RECOMMENDATION WIDGET:

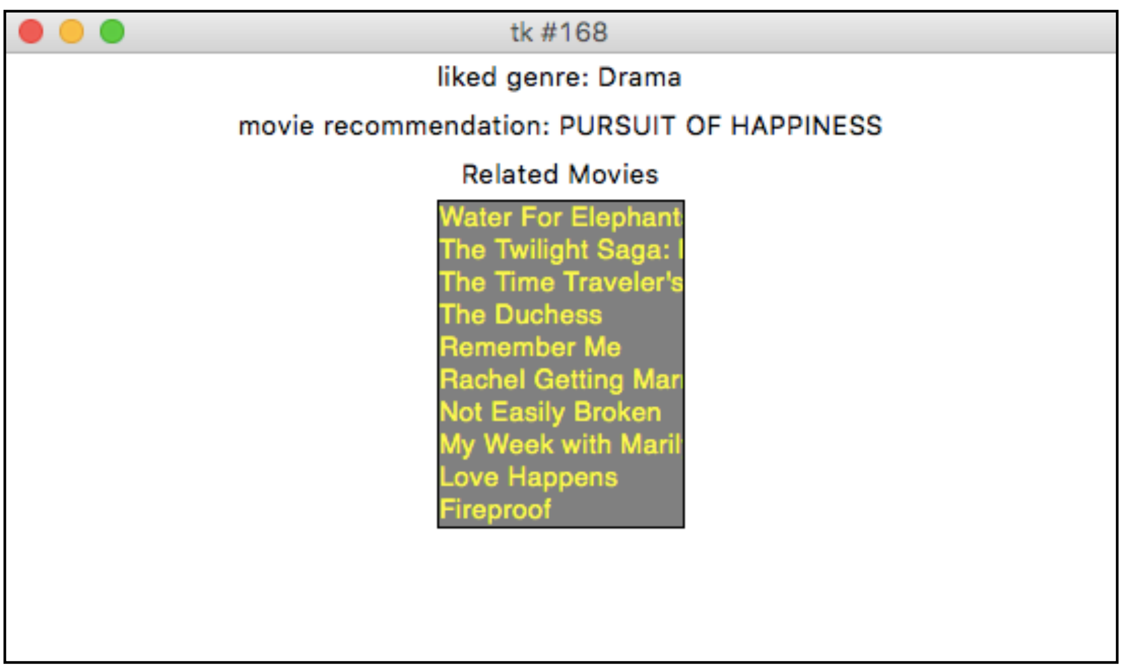

FIG 7.4 - RECOMMENDATION OF THE MOVIE AND RELATED MOVIE'S WIDGET

CONCLUSION

This project provides appropriate rating for a particular movie.Recommendations will be displayed to the user based on his interest.Related movie's are displayed on the interest of the user specified.A user interface is used for login and authentication.

REFERENCES

- https://www.analyticsvidhya.com/blog/2020/02/quick-introduction-bag-of-words-bow-tf-id/
- Java Projects on Movie Rating and Recommendation on Mobile - Free Projects For All
- https://builtin.com/data-science/how-build-neural-network-keras
- Machine Learning Project - Data Science Movie Recommendation System Project in R - DataFlair (data-flair.training)
- https://www.analyticsvidhya.com/blog/2020/01/fundamentals-deep-learning-activation-functions-when-to-use-them/
- https://keras.io/api/layers/activations/
- https://heartbeat.fritz.ai/introduction-to-machine-learning-model-evaluation-fa859e1b2d7f
- https://www.tutorialspoint.com/keras/keras_model_evaluation_and_prediction.htm\
- https://www.tensorflow.org/api_docs/python/tf/keras/utils/to_categorical
- https://keras.io/api/utils/
- https://keras.io/api/losses/
- https://www.tensorflow.org/api_docs/python/tf/keras/losses/categorical_crossentropy
- https://towardsdatascience.com/character-level-language-model-1439f5dd87fe
- https://www.tutorialspoint.com/python/python_gui_programming.htm
- https://en.wikipedia.org/wiki/IMDb