APPENDIX 1

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

ONLINE MOVIE REVIEW PREDICTION SYSTEM

A Project Report of Capstone Project - 2

Submitted by

SURBHI SINGH

(1613101766 / 16SCSE101098)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the Supervision of

Mr. JANARTHANAN S

(ASSISTANT PROFESSOR)

APRIL / MAY- 2020

APPENDIX 2

SCHOOL OF COMPUTING AND SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report **“ONLINE MOVIE REVIEW PREDICTION SYSTEM”** is the bona fide work of **“SURBHI SINGH (1613101766)”** who carried out the project work under my supervision.

SIGNATURE OF HEAD
DR. MUNISH SHABARWAL,
PhD (Management), PhD (CS)
Professor & Dean
School of Computing Science & Engineering

SIGNATURE OF SUPERVISOR
Mr. JANARTHANAN S
Assistant professor
School of Computing Science & Engineering

APPENDIX 3

TABLE OF CONTENTS

1. ABSTRACT

The data available on the web is of highly dynamic nature especially in the case if e-commerce systems and needs to be handled efficiently in order to provide competitive and efficient applications such as recommendation systems that would be able to predict the everchanging tastes of users accurately. In recent years, movie industry is getting more and more prosperous. There are hundreds of movies released every year. However, it is difficult to notice the releasing of every movie, not to mention actually seeing it. Therefore, movie recommender system has become more and more popular as a research topic. To date, a number of recommendation algorithms have been proposed, where collaborative filtering and content-based filtering are the two most famous and adopted recommendation techniques. As the world is staying at home to quarantine and because of the uninvited Virus, COVID-19. Earlier because of the tight schedule we never get time to watch movies, and also because of the quarantine we are getting bored. To overcome from the boredom, we are engaging ourselves in some extra-curricular activities, like- Exercise, Cooking, Watching Movies, etc. This Android app will help the people to know about Semantics of the movies. In this project, we propose a Movie Recommendation System by combining Naive Bayes Algorithm with Collaborative filtering. The recommendation system is able to predict the everchanging area of interest or tastes of users about the movies.

2.INTRODUCTION

Recommendation System is a subclass of information filtering system that seeks to predict the ‘rating’ or ‘preference’ that user would give to an item. In this project, we have combined Naive Bayes Algorithm with Collaborative filtering for predicting which movie the user will like the best.

Collaborative filtering algorithm usually works by searching a large group of people and finds a smaller set with tastes similar to the user. It looks at other things; they like and combine them to create a ranked list of suggestions. Finally, it shows the suggestion to the user. Sentiment analysis is a field dedicated to extracting subjective emotions and feelings from text. One common use of sentiment analysis is to figure out if a text expresses negative or positive feelings. Written reviews are great datasets for doing sentiment analysis, because they often come with a score that can be used to train an algorithm.

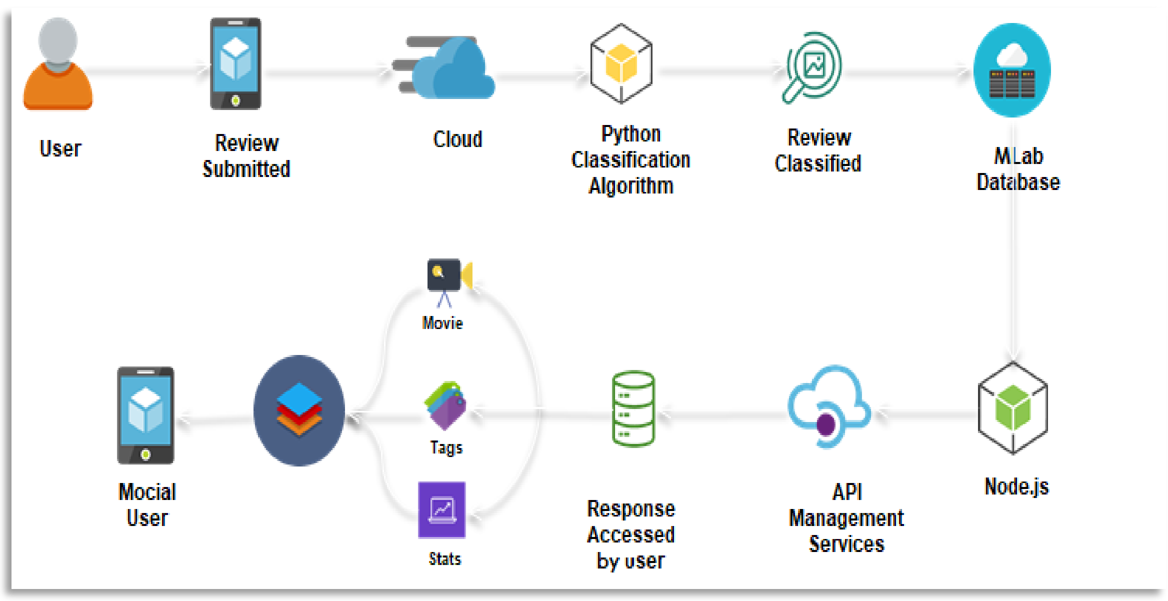
The recommender system is represented as an intelligent system, which identifies the user category based on user information analysis and user interest analysis. Once such information is obtained, in second stage, the analysis is performed to obtain the similarity group respective to available products and services. For sentiment analysis, written reviews playing a very important role, as the datasets. These ratings are provided as input to the system. By using the Naive Bayes Algorithm, the system will analyze this dataset to check for the user sentiments associated with each comment. A new collaborative filtering algorithm is being designed to provide better performance of the algorithms to provide users with more accurate recommendations and consider the basics of naïve Bayesian algorithm for similarity. While providing a review or rating to a movie tells us about the success or failure of a movie. A content movie review informs us about the movie and deeper analysis of a movie review can inform us if the movie, in general, meets the expectations of the reviewer. By this we also get to know the attitude of the reviewer with respect to various topics or the overall polarity of review. With the help of collaborative filtering and sentimental analysis we can find the state of mind of the reviewer while providing the review and understand if the person is liking or disliking the particular Genres by feeding the positive or negative reviews. New collaborative filtering algorithm are being designed to provide better performance of the algorithm to provide users with more accurate recommendations and consider the basics of naïve Bayesian algorithm for similarity.

3.LIITERATURE SURVEY

Recommender system has been so extensively used these days that it has become a preferable choice for researchers. First paper on recommender system was published in year 1998. Since then a significant number of papers had been published. Different factors have been explained to increase the reliability of recommender system. In year 2005 John O'Donovan, Barry Smyth [1] have taken trust as the percentage of correct predictions that a profile has made in general (profile-level trust) or with respect to a particular item (item- level trust). Authors have described a number of ways in which these different types of trust values might be incorporated into a standard collaborative filtering algorithm and evaluated each against a tried-and-test benchmark approach and on a standard data-set. This decreases the prediction error by 22%.

In another work by [2], an approach is presented that is not limited to any specific recommendation algorithm. The intuition behind this approach comes from the assumption that multi criteria ratings represent user preferences for different components of an item, such as story, acting, direction, and visuals in the case of movies. So, an item's overall rating is not just another rating that is independent of others; rather, it serves as some aggregation function f of the item’s multi criteria ratings. In other words, this approach assumes that the overall rating has a certain relationship with the multi criteria ratings. For instance, in a movie recommendation application, the story criterion rating might have a very high priority—that is, movies with high story ratings are well liked overall by some users, regardless of other criteria ratings. So, if a system predicts that a movie’s story rating will be high, it must also predict that the overall rating will be high in order to be accurate.

4.METHODOLOGY ADOPTED



4.1Naive Bayes

The Naive Bayes Classifier [7] contributes a simple method, representing and learning probabilistic knowledge with clear semantics. It is termed naïve because it relies on two important simplifying assumes that predicting attributes are conditionally self-reliant given the class, and considers that the prediction method has not influenced by any of the hidden attributes. They are amid the uncomplicated Bayesian network template. In machine learning, Naive Bayes classifiers are ancestry of simple "probabilistic classifiers". In easy terms, a Naïve Bayes classifier presume that the existence of a specific attribute in a class is unassociated to the appearance of any other attribute. Text classification/ Spam Filtering/ Sentiment

Analysis: Naive Bayes classifiers are used in analysis due to their better results in multi-class problems and independence rule and also have a higher success rate as compared to other algorithms.

The theorem is P(A|B) = {P(B/A) * P(A)}/P(B) OR Posterior = {Prior*likelihood}/evidence. This basically states "the probability of A given that B is true equals the probability of B given that A is true times the probability of A being true, divided by the probability of B being true.

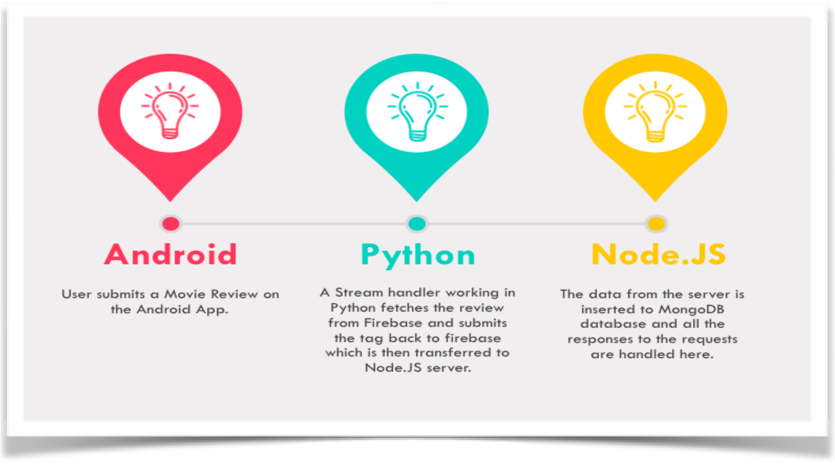
The Naïve Bayes Algorithm is the one of best algorithms for basic classification approaches with numerous applications in email spam detection, document categorization, sentiment detection, language detection, in personal email sorting [8].

4.1 Collaborative Filtering

“We are leaving the age of information and entering the age of recommendation”. The aim of Recommendation system is to show items of interest to a user. Collaborative Filtering is the most familiar approach used when it comes to establishment of sharp recommender systems that can be acquired to give superior recommendations as more statistics about users is collected. For the advanced Recommendation system, the collaborative filtering is used by many of the websites such as Netflix, YouTube, Amazon, and Torrents.

5.DATA FLOW OF THE APPLICATION

The data flow in this application is between 3 objects which are client, backend, and python algorithm. Once, the user submits the review for a particular movie, it gets uploaded on Firebase which is a real- time cloud service, from there, a stream handler which is working continuously fetches the data and passes the review to the model through a destined function under it. From the model, there comes a response that whether the review is positive or negative. Now, that response is sent back to the Nodejs server through which it is sent back to the client. Simultaneously, the response along with other user details gets uploaded to the NoSQL database MongoDB. All the data is collected in the MongoDB Database to provide users with other functionalities such as grouping users according to their movie taste, arranging a favorites section for the users, and creating a group chat for every particular group of users.


Android
User submits a Movie Review on the Android App.
Python
A Stream handler working in Python fetches the review from Firebase and submits the tag back to firebase which is then transferred to Node.JS server.
Node.JS
The data from the server is inserted to MongoDB database and all the responses to the requests are handled here.

6.EXPERIMENTAL TOOLS

XML:

X ML stands for Extensible Markup Language. XML is a markup language much like HTML. XML was designed to store and transport data. XML was self-descriptive, it has-

1. It has sender information
2. It has receiver information
3. It has heading
4. It has a message body

It has some goals like simplicity, generality, and usability across the application.

Node.js:

Node.JS is a cross-platform and open-source runtime environment. It is popular tool for almost any kind of project.

In this app, data is received by the POST routes form Android through Retrofit. After that through database calls, data is saved on MongoDB, and also the review is saved on Firebase from which the python script takes it and gives the rating which is taken by the POST route and saved in MongoDB Database.

Android Studio:

Android Studio provides the fastest tools for building apps on every type of Android device. It is Google's officially supported IDE for developing Android Apps. This IDE is based on IntelliJ IDEA, which offers a powerful code editor and developer tools. It has lint tools to help you catch performance, usability, version compatibility, and other problems. Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and Google App Engine

MongoDB:

MongoDB is cross-platform document-oriented database program. It is an open-source document database and a leading NoSQL database. It uses JSON-like documents with schema and classified as NoSQL database program.

The unique MongoDB architecture combines the best of both relational and non-relational databases, addressing the needs of organizations for performance, scalability, flexibility, and reliability while maintaining the strengths of legacy databases. JSON-based, document data model with dynamic schemas while still preserving the functionality of relational databases, like secondary indexes, a full query language and aggregations.

JAVA:

Java is at the heart of our digital lifestyle. It's the platform for launching careers, exploring human-to-digital interfaces, architecting the world's best applications, and unlocking innovation everywhere – from garages to global organizations. Java is an object-oriented language that gives a clear structure to programs and allows code to be reused, lowering developmental costs. It is one of the most popular programming languages easy to use, open-source, fast, free, and secure. Java code can run on all platforms that support Java without need for recompilation.

Python:

Python is a general-purpose interpreted, interactive object-oriented, and high-level programming language. It's great as a first language because it is concise and easy to read, and programmer's stack as it can be used for everything from web development to software development and scientific applications.it has a syntax that allows developers to write programs with fewer lines than some other programming languages, it runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick. Python can be treated in a procedural way, an object-orientated way, or a functional way.

Firebase:

Firebase is built on Google infrastructure and scales automatically, for even the largest apps. It gives functionality like analytics, databases, messaging, and crash reporting so you can move quickly and focus on your Users. Firebase products work great individually but share data and insights, so they work even better together. Deploy functionality to your app quickly using pre-packaged solutions. Firebase Extensions are configurable and work with Firebase and other Google Cloud Platforms products.

7.EXPERIMENTAL SETUP

The Setup is pretty simple the Android app acts as the interface to interact with the users so that they can feed the review in so that it can be fetched back in the database and be furthered passed on to the python algorithm which later classifies the review as positive or negative and sends the tag back to the database and it’s shown on the app.

First, the users have to sign up on the app giving his/her required details. After that he/she have to login. Now, the users have many choices like they can rate and review a movie of their choice or they can see how other people are rating movies of a particular genre or they can form movie plans with different people having similar movie interests.

Choice A: User wants to rate/review a movie.

The user should select a particular genre from the spinner.

As soon as a genre is selected, a route is called which have the genre as input and in response it receives a list of movies of that genre in the spinner given below.

Now, the user can select a movie of their choice and then press the next button. A new activity appears in which a text box is given to enter the review and give star rating to the movie. Now, a route is called having genre, movie, review and star rating as input. As soon as the review is submitted, the python script picks up the review from firebase and give it positive/negative tag to classify users by doing sentimental analysis. After that the review is submitted and user receives a list of the people who have submitted positive reviews in that genre with whom the user can form plans if they want.

Choice B: User wants to see what rating people have given to different movies.

For this, the user has to press the button given at bottom right and select a genre. After the genre is selected, a route is called with the genre as input and an array of different people is returned who have given positive reviews in that genre.

8.SOURCE CODE

```
#!/usr/bin/env python3 #

-*- coding: utf-8 -*-

import pickle

from nltk.corpus import stopwords

from nltk.tokenize import word_tokenize

from firebase import firebase

import pyrebase

def create_word_features(words):

    useful_words = [word for word in words if word not in stopwords.words('english')] my_dict =

    dict([(word, True) for word in useful_words])

    return my_dict

model = pickle.load(open("model.pkl","rb"))

config = {

  "apiKey" : "key",

  "authDomain": "mocial-91b4a.firebaseapp.com",
   "databaseURL": "https://mocial-91b4a.firebaseio.com/",
   "storageBucket": "mocial-91b4a.appspot.com"
 }
 firebase1 = pyrebase.initialize_app(config) db
 = firebase1.database()
 app = firebase.FirebaseApplication('https://mocial-91b4a.firebaseio.com/',None)
 #Streaming the Input.(Forever Stream) def
 stream_handler(message):
  print('event={m[event]}; path={m[path]}; data={m[data]}'.format(m=message)) data2 =
message["path"]
    x = "Review"
    if x in data2:
data1 = message["data"]
print(data1)
       print("\n\nThe updated Review is:", data1['review'])
       strng = "/Users"
       data2 = data2[:-6]
       data3 = strng + data2 + 'Rating'
       print(data3)
       #rating = "Positive" def
       reviewmovie(dat):
          words=word_tokenize(dat)
          words=create_word_features(words)
          return(model.classify(words))
       send = reviewmovie(data1['review'])
       app.put(data3,'rating',send)
    else:
        print("\n\nThe Review is not updated yet.")
my_stream =db.child('Users').stream(stream_handler) # Run
Stream Handler forever
while True:
    data = input("[{}] Type exit to disconnect: ".format('?')) if
    data.strip().lower() == 'exit':
        print('Stop Stream Handler')
        if my_stream: my_stream.close()
        break
response = db.child('Users').get()
print(response.val())
```

9.EXPERIMENTAL RESULT

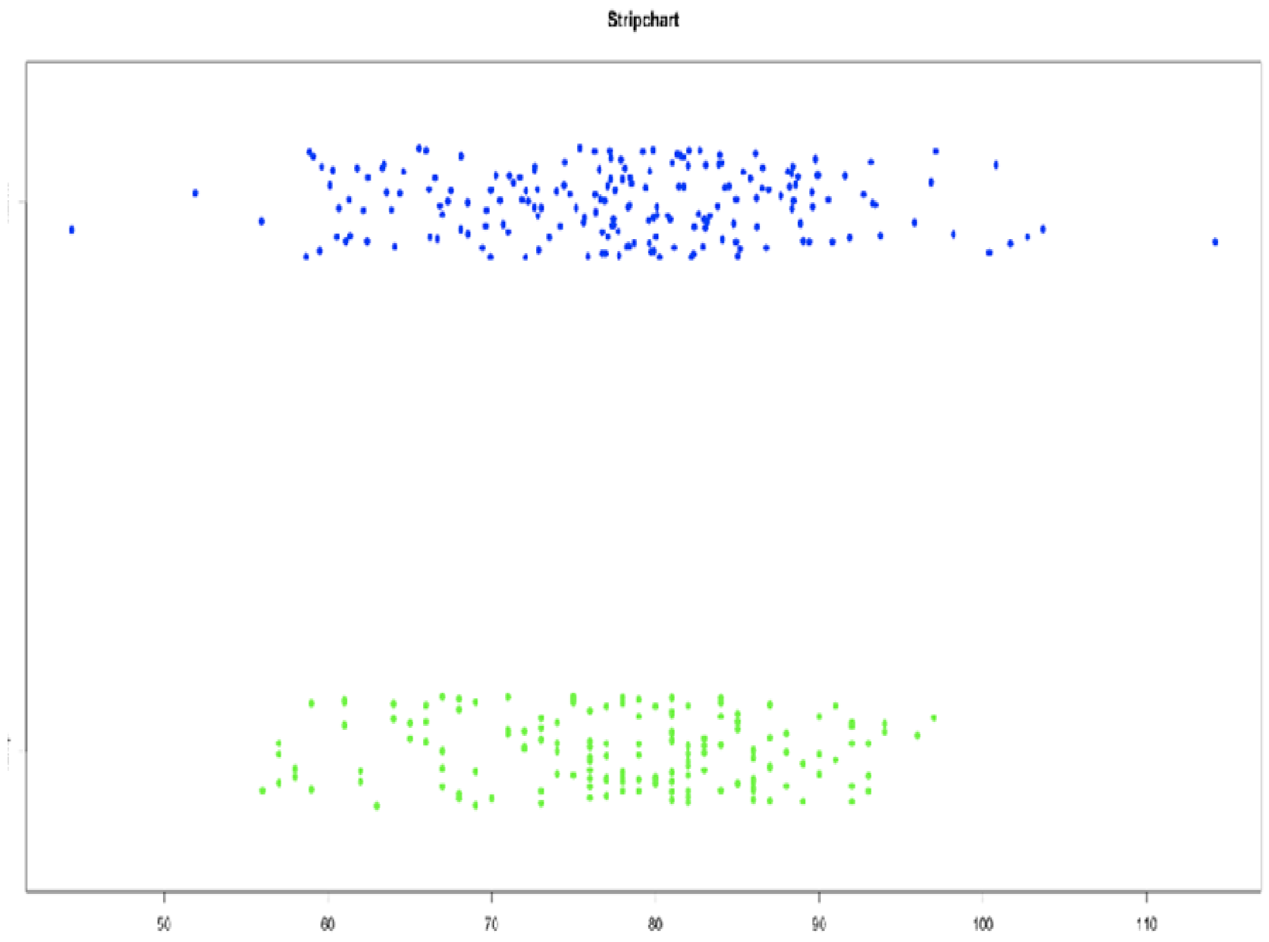
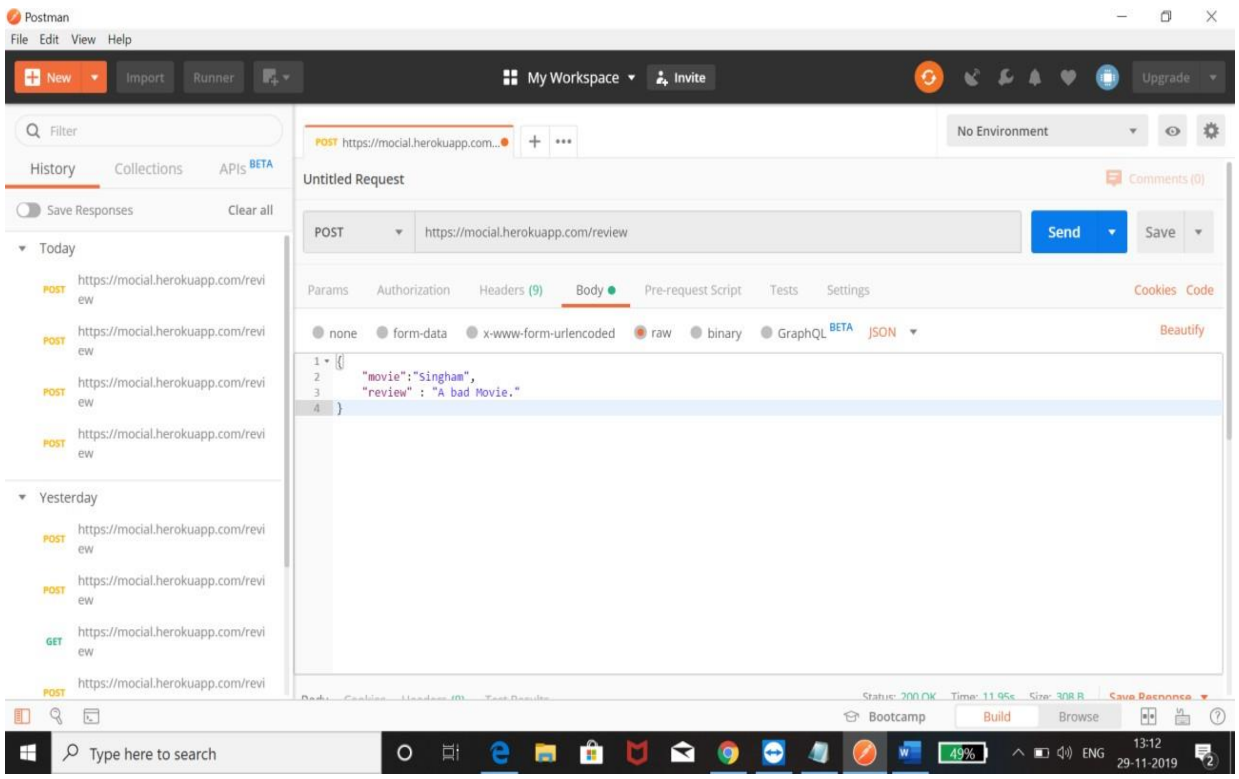


Figure 1. A strip chart of negative & positive reviews.

In this project we tried to build a Recommendation System by using Collaborative Filtering and Naive Bayes Algorithm. I tried to purpose a Simple Recommendation System using movie reviews dataset from the users. The users according to their interest and taste firstly choose the genre of the movies and depends upon the likes and dislikes of them to feed the reviews as Positive and Negative accordingly.

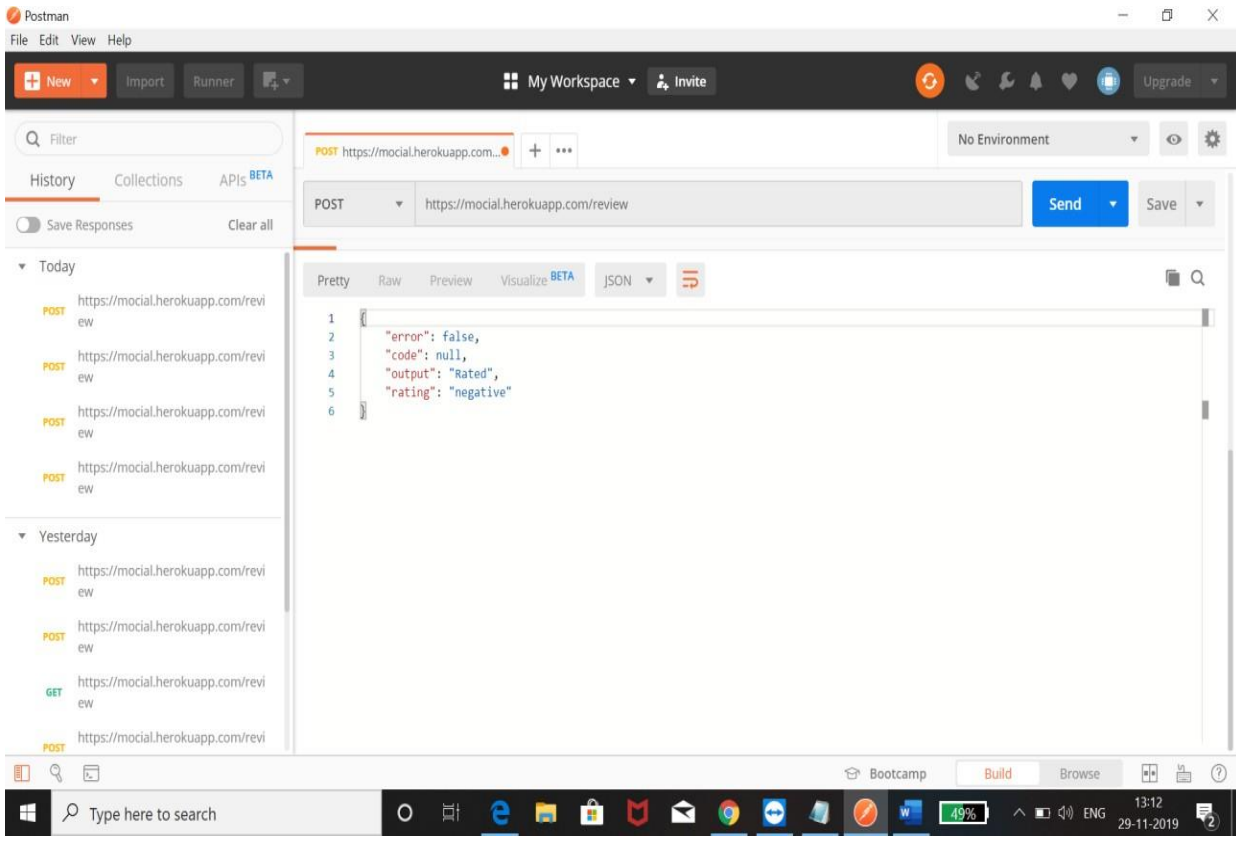
The simple Recommender offers generalized recommendations to every user based on the movie popularity and genre. The movies which are more in demand, will have a soaring probability of being liked by the moderate crowd. We can expand it by adding some feature, earlier it only gives the review on the personalized way, depending on the genre they choose, we will try to add a feature in which that will have the crew member details, together with the genre. For Example, if a person Loves the film DDLJ, MNIK, and K3G from this one inference we can obtain is that the person also loves the actor Shahrukh Khan and the director Karan Johar. So next time when that person search for the movies of Romantic genre, he will get this type of Recommendation on the topmost of the list. As we know, because of the COVID-19, and the world stays home to quarantine using social media for interacting with each other and asking on Instagram for movie recommendations on their stories from their friends and the followers, and others are helping their friends by giving some ideas and names of movies depends on their likes and dislikes. So will also try to add a chat box so that people can easily become social with the others who has the same interest in the movies and able to form a groups and plan movie accordingly.



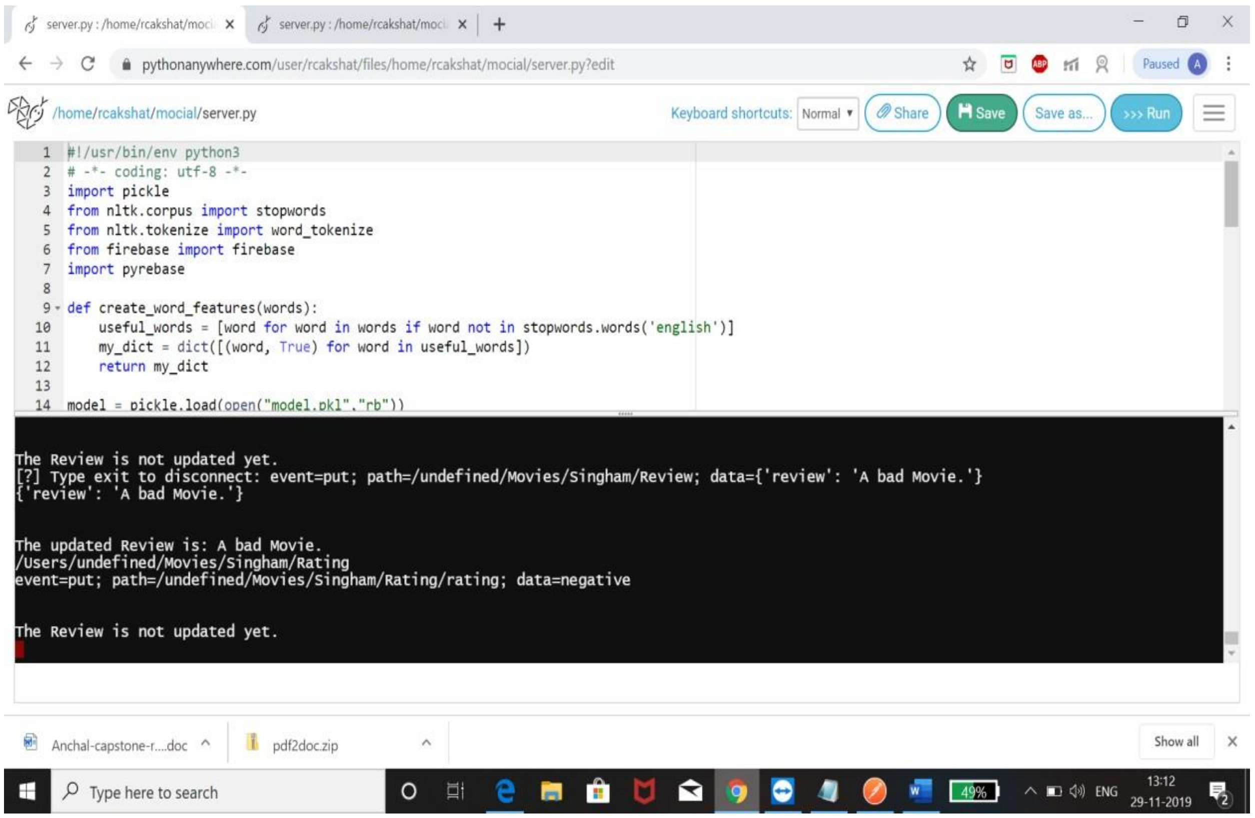
POST https://mocial.herokuapp.com/review

```
{
    "movie":"Singham",
    "review" : "A bad Movie."
}
```

Status: 200 OK Time: 11.95s Size: 308 B

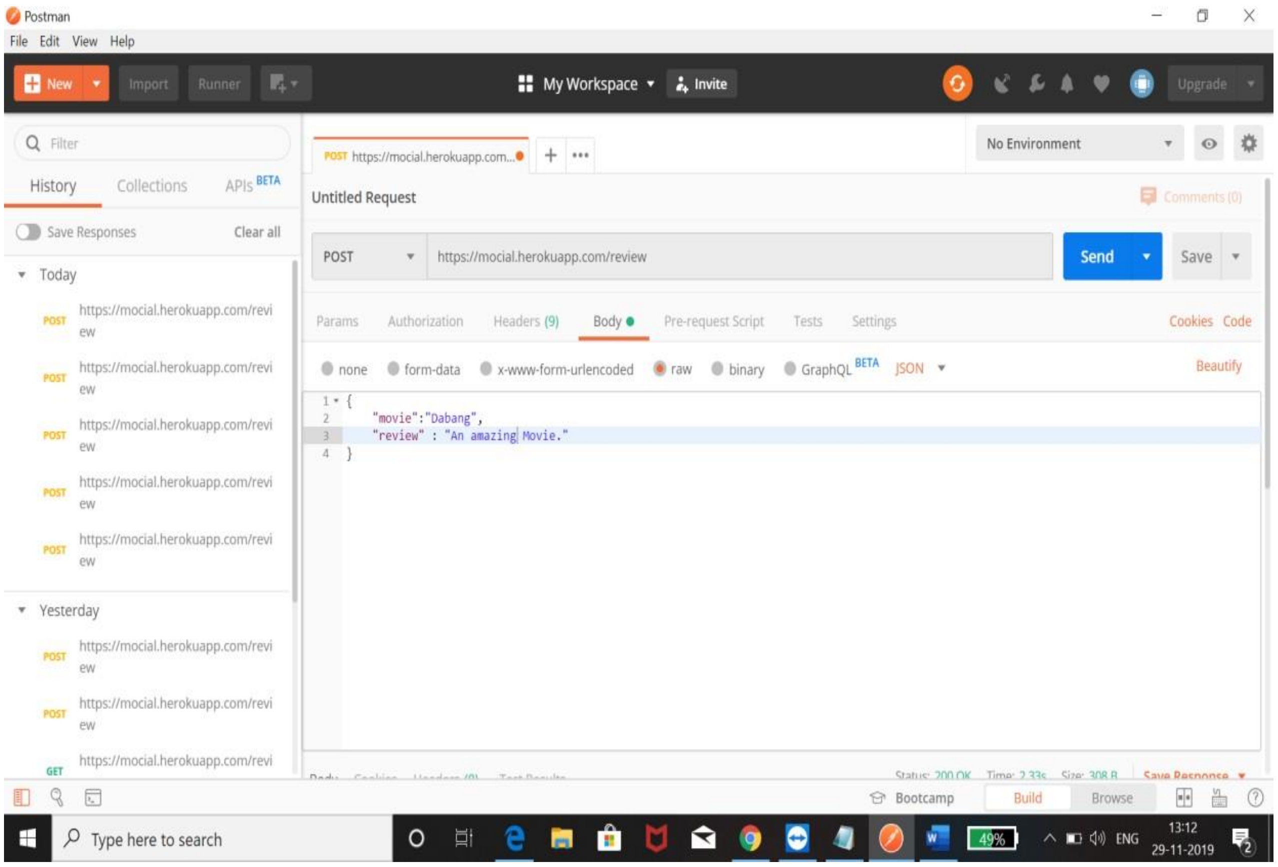


```
{
    "error": false,
    "code": null,
    "output": "Rated",
    "rating": "negative"
}
```


pythonanywhere.com/user/rcakshat/files/home/rcakshat/mocial/server.py?edit
/home/rcakshat/mocial/server.py
Keyboard shortcuts: Normal
Share
Save
Save as...
>>> Run

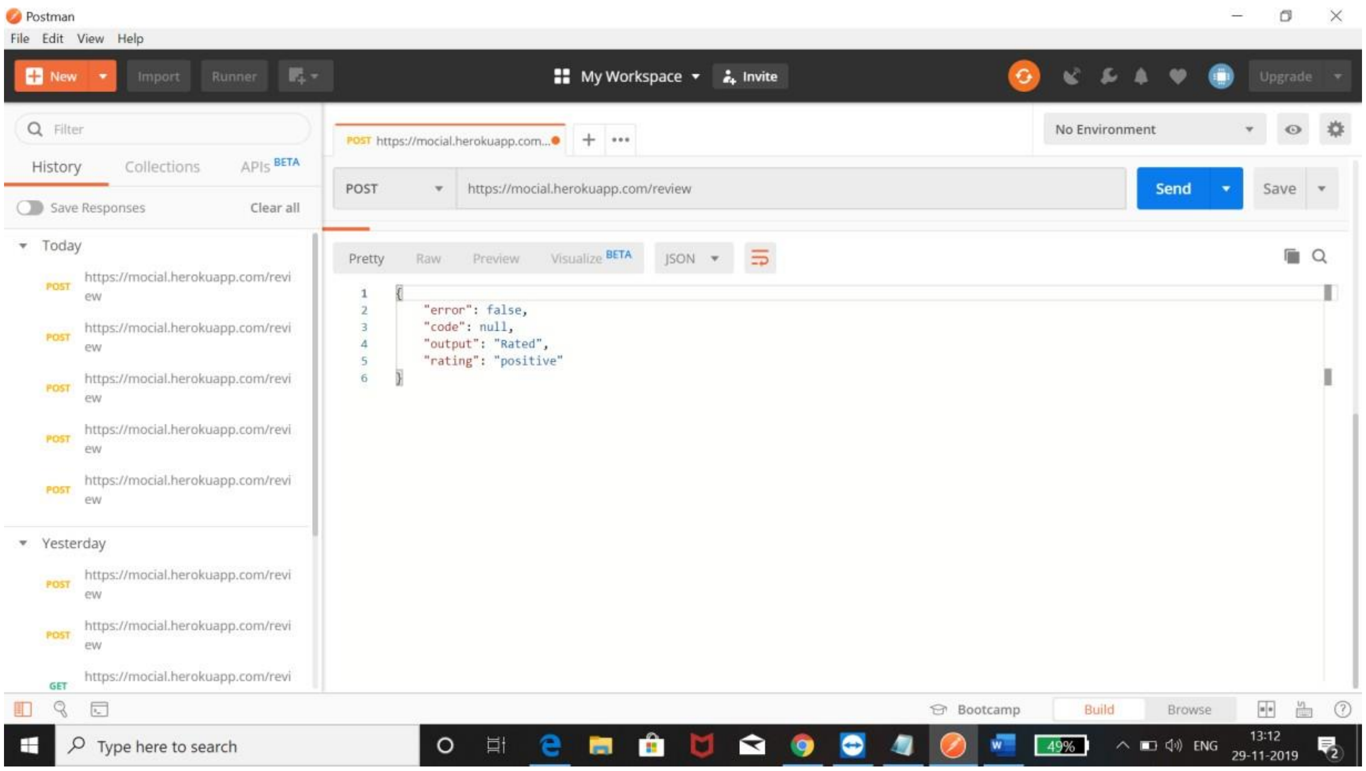
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import pickle
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from firebase import firebase
import pyrebase

def create_word_features(words):
    useful_words = [word for word in words if word not in stopwords.words('english')]
    my_dict = dict([(word, True) for word in useful_words])
    return my_dict

model = pickle.load(open("model.pkl","rb"))
```

```
The Review is not updated yet.
[?] Type exit to disconnect: event=put; path=/undefined/Movies/Singham/Review; data={'review': 'A bad Movie.'}
{'review': 'A bad Movie.'}

The updated Review is: A bad Movie.
/Users/undefined/Movies/Singham/Rating
event=put; path=/undefined/Movies/Singham/Rating/rating; data=negative

The Review is not updated yet.
```



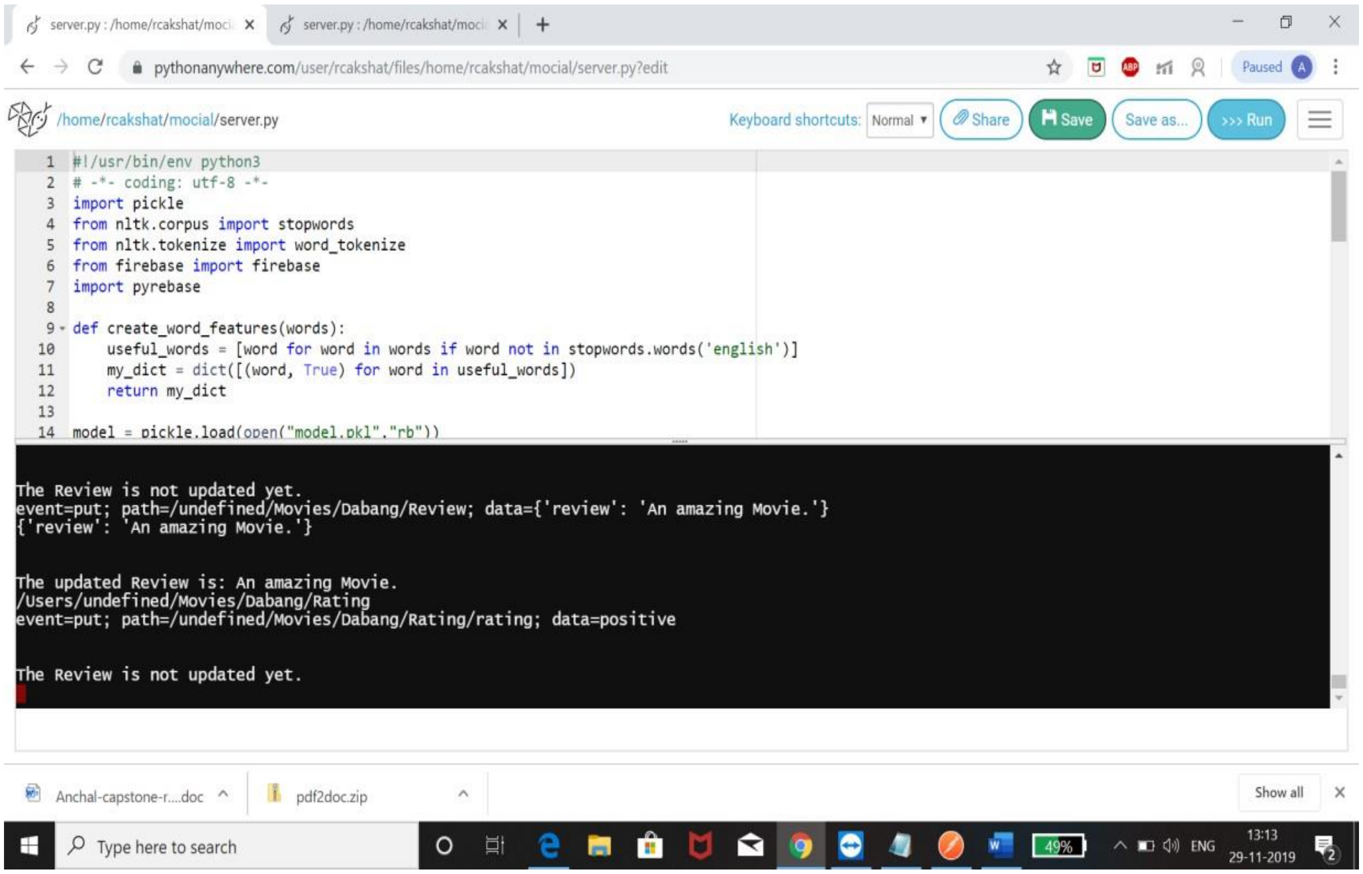
POST https://mocial.herokuapp.com/review

```
{
    "movie":"Dabang",
    "review" : "An amazing Movie."
}
```



POST https://mocial.herokuapp.com/review

```
{
    "error": false,
    "code": null,
    "output": "Rated",
    "rating": "positive"
}
```


server.py : /home/rcakshat/mocial
pythonanywhere.com/user/rcakshat/files/home/rcakshat/mocial/server.py?edit
/home/rcakshat/mocial/server.py
Keyboard shortcuts: Normal
Share
Save
Save as...
>>> Run
Anchal-capstone-r....doc
pdf2doc.zip
Show all
Type here to search

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import pickle
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from firebase import firebase
import pyrebase

def create_word_features(words):
    useful_words = [word for word in words if word not in stopwords.words('english')]
    my_dict = dict([(word, True) for word in useful_words])
    return my_dict

model = pickle.load(open("model.pkl","rb"))
```

```
The Review is not updated yet.
event=put; path=/undefined/Movies/Dabang/Review; data={'review': 'An amazing Movie.'}
{'review': 'An amazing Movie.'}

The updated Review is: An amazing Movie.
/Users/undefined/Movies/Dabang/Rating
event=put; path=/undefined/Movies/Dabang/Rating/rating; data=positive

The Review is not updated yet.
```

10.CONCLUSION

The overall task in this project is for the classification of reviews as favorable or unfavorable. This project is to make people get more social and get entertained in this quarantine. As the people are inside their home and some are also working from home and keeping themselves by doing a lot of stuff, to lightening up their mood they need some rest and a mode of entertainment. This app will not only help them in lightening up the mood but also help to see the movies of own choices with the help of the Reviews and they can also feed their review. The functionality of this app is for is not that tacky, it is a very simple stuff, firstly the users have to sign up and then login with the essential information/details. After doing so, they can choose Genres it maybe Romantic, Comic, Action, Thriller, and Horror. Now user can feed their review positive/negative according to his/her choice, and likes and dislikes. The users can also fetch the list of others who have to feed the review as Positive for any particular genre and the Movies.

11.REFERENCES

[1] John O'Donovan, Barry Smyth, Trust in recommender systems, Proceedings of the 10th international conference on Intelligent user interfaces, January 10-13, 2005, San Diego, California, USA

[2] Adomavicius G., Kwon Y. IEEE Intelligent Systems Volume 22 Issue 3, Pages 48-55 May 2007

[3] Movie Review Datasets, kaggle.com

[4] Movie Reviews, imdb.com

[5] Sentiment Analysis on Movie Reviews: A Comparative Study of Machine Learning Algorithms and Open Source Technologies, Mr. B. Narendra, Mr. K. Uday Sai, Mr. G. Rajesh, Mr. K. Hemanth, Mr. M. V. Chaitanya Teja, Mr. K. Deva Kumar

[6] Hasan Ogul, Emrah Ekmekciler, "Two-way collaborative filtering on semantically enhanced movie ratings", *Information Technology Interfaces (ITI) Proceedings of the ITI 2012 34th International Conference on*, pp. 361-366, 2012.

[7] Dave, K., Lawerence, S., Pennock, D.: Mining the Peanut Gallery-Opinion Extraction and Semantic Classification of Product Reviews. In: Proceedings of the 12th International WWW Conference, pp. 519–528 (2003) Google Scholar

[8] John, G. H., & Langley, P. Estimating continuous distributions in Bayesian classifier. In Proceedings of the Eleventh Conference on Uncertainty in artificial intelligence
(pp. 388-345). Morgan Kaufmann Publishers Inc.

[9] Nguyen, H. A., & Choi, D. Application of data mining to network intrusion detection: classifier selection model. In Challenges for Next Generation Network Operations and Service Management (pp. 399-408). Springer Berlin Heidelberg, 2008.

[10] Sentiment Analysis of movie review comments, uat essenov, Sas a isailovic , May 17, 2009

[11] Deep learning for sentiment analysis of movie reviews, Hadi Pouransari, Stanford University