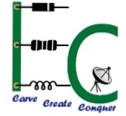
****

**THE NATIONAL INSTITUTE OF**

**ENGINEERING**

**MYSURU-570008**

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

**MINI PROJECT [EC0201] –VI Semester**

**Report On**

**BOOK RECOMMENDER USING MACHINE LEARNING**

**Under the guidance of:**

**Mrs. PRATHEEKSHA C S**

Assistant Professor, Dept. of ECE.

**Submitted By:**

CHETAN SK : 4NI17EC021

DHANUSH SHETTY : 4NI17EC031

**2019-20**

**ABSTRACT**

Recommendation systems are tools in e-commerce websites which helps user to find the appropriate products. With the rapid development of internet technologies the number of online book selling websites has increased which enhanced the competition among them.

The main motive of this project is to develop the technique which recommends most suitable books to the user or costumer according to their likes and recent favorites and even provide filters like book ratings, price and authors.

**Table of Contents**

1. Introduction
2. Literature Survey
3. Tool survey
4. Implementation
5. Results/Conclusion/Observation
6. Applications
7. Limitations
8. Future scope
9. References

**INTRODUCTION**

Now-a-days the amount of information especially in Internet growth very rapidly. Finding the necessary information becomes more difficult. Recommendation systems aim to solve this kind of problems. With the help of them one can quickly access relevant information without searching the web manually. As such many web sites today benefit from the recommendation systems to promote and sell their products. There is a wide range of products like music, movies, articles and etc. that can be recommended to the customer based on their profiles in internet shops or even social networks, browsing history such as visited links, browsing activity like number and time of visits and other online behavior.

In this project, we are using recommendation systems for recommending books. We are to develop a system, which learns user preferences by asking random questions about themselves as to understand the user emotionally, so that we can recommend books according to the user’s emotions and likes and to rate books and choosing favorite categories and then generate the list of books user most probably would like to read

**LITERATURE SURVEY**

…We have referred the following paper for our project:

* Stuti Goel, Arun Rana, Manpreet Kaur “A Review of Comparison Techniques of Image Steganography”, IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE) e-ISSN: 2278-1676, p-ISSN: 2320-3331, Volume 6, Issue 1 (May. - Jun. 2013), PP 41-48

**TOOL SURVEY**

* **Google Colab** is a free cloud service and now it supports free GPU. You

improve your Python programming language coding skills. develop deep learning applications using popular libraries such as Keras, TensorFlow, PyTorch

* The **Jupyter Notebook** is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text ,data cleaning and transformation, numerical simulation, statistical modeling, machine learning and much more
* **Kaggle** is a platform for predictive modelling and analytics competitions in which companies and researchers post data and statisticians and data miners compete to produce the best models for predicting and describing the data

**IMPLEMENTATION /WORKING**

1 Loading data

2 Creating dot model product

3 Creating neural network

4 Visualization

5 Making recommendations

Loading data:

dataset = pd.read\_csv('../input/ratings.csv')

dataset.head()

dataset.shape

from sklearn.model\_selection import train\_test\_split

train, test = train\_test\_split(dataset, test\_size=0.2, random\_state=42)

train.head()

test.head()

n\_users = len(dataset.user\_id.unique())

n\_users

n\_books = len(dataset.book\_id.unique())

n\_books

# creating book embedding path

book\_input = Input(shape=[1], name="Book-Input")

book\_embedding=Embedding(n\_books+1,5,name="BookEmbedding")(book\_input)

book\_vec = Flatten(name="Flatten-Books")(book\_embedding)

# creating user embedding path

user\_input = Input(shape=[1], name="User-Input")

user\_embedding = Embedding(n\_users+1, 5, name="User-Embedding")(user\_input)

user\_vec = Flatten(name="Flatten-Users")(user\_embedding)

# performing dot product and creating model

prod = Dot(name="Dot-Product", axes=1)([book\_vec, user\_vec])

model = Model([user\_input, book\_input], prod)

model.compile('adam', 'mean\_squared\_error')

from keras.models import load\_model

if os.path.exists('regression\_model.h5'):

model = load\_model('regression\_model.h5')

else:

history = model.fit([train.user\_id, train.book\_id], train.rating, epochs=5, verbose=1)

model.save('regression\_model.h5')

plt.plot(history.history['loss'])

plt.xlabel("Epochs")

plt.ylabel ("Training Error")

model.evaluate([test.user\_id, test.book\_id], test.rating)

predictions = model.predict([test.user\_id.head(10), test.book\_id.head(10)])

[print(predictions[i], test.rating.iloc[i]) for i in range(0,10)]

Visualization

book\_em = model.get\_layer('Book-Embedding')

book\_em\_weights = book\_em.get\_weights ()[0]

book\_em\_weights[:5]

from sklearn.decomposition import PCA

import seaborn as sns

pca = PCA(n\_components=2)

pca\_result = pca.fit\_transform(book\_em\_weights)

sns.scatterplot(x=pca\_result[:,0], y=pca\_result[:,1])

book\_em\_weights = book\_em\_weights / np.linalg.norm(book\_em\_weights, axis = 1).reshape((-1, 1))

book\_em\_weights[0][:10]

np.sum(np.square(book\_em\_weights[0]))

pca = PCA(n\_components=2)

pca\_result = pca.fit\_transform(book\_em\_weights)

sns.scatterplot(x=pca\_result[:,0], y=pca\_result[:,1])

from sklearn.manifold import TSNE

tsne = TSNE(n\_components=2, verbose=1, perplexity=40, n\_iter=300)

tnse\_results = tsne.fit\_transform(book\_em\_weights)

sns.scatterplot(x=tnse\_results[:,0], y=tnse\_results[:,1])

Making Recommendations

*# Creating dataset for making recommendations for the first user*

book\_data = np.array(list(set(dataset.book\_id)))

book\_data[:5]

user = np.array([1 for i **in** range(len(book\_data))])

user[:5]

predictions = model.predict([user, book\_data])

predictions = np.array([a[0] for a **in** predictions])

recommended\_book\_ids = (-predictions).argsort()[:5]

recommended\_book\_ids

*# print predicted scores*

predictions[recommended\_book\_ids]

books = pd.read\_csv('/content/books (3).xls')

books.head()

books[books['id'].isin(recommended\_book\_ids)]



**APPLICATIONS**

Used to recommend suitable books to users.

* The filtering algorithm used here can be applied to other recommenders like movie recommender, software recommender.

**LIMITATIONS**

* + The output is accurate up to 95%, even after a lot of iterations
  + Sometimes the working speed can decrease, because the system has to process a lot of database.
  + The input is not always perfect as humans are not good enough to give appropriate ratings. Eg : Some people are unable to understand the main context of the book due to which they give bad ratings.

**FUTURE SCOPE**

* + Can be modified further to recommend movies, songs, softwares
  + Online shopping brands can use this model to improve their products based on the customers reviews.

**REFERNCES**

<https://www.kaggle.com/tannergi/book-recommendation-system/data>

<https://ieeexplore.ieee.org/document/7416895>

**Project Coordinators Project Guide**

Smt. K.V.NAGALAKSHMI Mrs. PRATHEEKSHA C S

Associate Professor, Dept. of ECE Assistant Professor, Dept. of ECE

NIE, Mysore NIE, Mysore

Mr. LOKESHA H R

Assistant Professor, Dept. of ECE

NIE, Mysore