**RECOMMENDER ENGINES**

Recommender engines are subclasses of record filtering structures that seeks to understand the preference and rating of a user to an item. In simple word, these are algorithms that suggests relevant items to users in a personalized way. Recommender engines lies among the most visible and successful applications of Artificial Intelligence and Machine learning. These system helps us through our daily online life e.g., in e-commerce websites, social media platforms and media network. These engines help us by signifying things that are of our interest and we our correspondingly likely to consume or purchase.

**IMPORTANCES**

Recommender engines are significant to both services providers and customers. These systems are used by companies to reduce transaction costs of searching and selecting items in online e-commerce purchasing environments. Theses engines are used to improve business decision making process and their values. In social media platforms recommender engines boost company revenues, for the fact, they are used as one of the most effective ways in terms of selling more products. In scientific libraries and researches websites these systems encourage users by allowing them to move beyond catalogue searches. Therefore, the need to use effective and accurate recommendation techniques in AI systems that have ability to predicts relevant and dependable recommendations for users cannot be over-emphasized.

**BUSINESS VALUE OF AI RECOMMENDER’S ENGINES**

**Recommender Engine and Videos**

There exist huge potential of personalization and recommendation in video (Netflix, YouTube, etc…) watching platforms. Netflix have about 75% watch history on some sort of recommendation. Whereas YouTube have 60% of home screen clicks due to recommendation algorithms. These recommendations have led to a measurable increase in user engagement. Such personalization and recommendation services has helped to decrease users churn by several percentage points over the years. The estimate business value of AI recommender engineer is more than 1 billion US dollars per day.

**Recommender Engine and E-commerce**

Recommender systems provides one-to-one marketing strategy to e-commerce users and customers. Amazon a pioneer platform uses collaborative recommender engines as part of their marketing strategy and offers “a personalized store to every customer”. Such personalization helps to benefits both consumers and services providers through understanding of purchase models between customer and companies. Serving these needs can result in greater success regarding cross-selling of related products, upselling, product affinities, one-to-one promotions, larger baskets, and customer retention.

**Challenges for recommendation engines**

Major challenge for recommender systems is its scalability and esurance of quality recommendations to the consumer. For example, for scalability recommender engines should be powerful enough to search through millions of potential neighbor users in real time. A website will have thousands of data points to search for customer preference if it is using browsing patterns. Therefore, ensuring recommendations that are of quality is very essential in order to gain customer trust. If customer follow a recommendation system but then do not end up liking the product, they are less likely to use the recommender system again.

**Working of Recommender Engine**

The following example make use of a dataset having 250 movies and utilizes Title, Genre, Director, Actor and Plot to recommend movie of user’s interest. The example uses cosine similarity matrix techniques for recommendation. The working of recommender engine is available at google colab.

CELL1:

!pip install rake-nltk

CELL2:

from rake\_nltk import Rake

import pandas as pd

import numpy as np

from sklearn.metrics.pairwise import cosine\_similarity

from sklearn.feature\_extraction.text import CountVectorizer

df  = pd.read\_csv('https://query.data.world/s/uikepcpffyo2nhig52xxeevdialfl7')

df.head()

CELL3:

df = df[['Title','Genre','Director','Actors','Plot']]

df.head()

CELL4:

df.shape

CELL5:

df\_ = df['Director'].value\_counts()[0:10]

df\_.plot(kind='bar', figsize=[8,5], fontsize=15, color='blue').invert\_xaxis()

CELL5:

import nltk

nltk.download('punkt')

r = Rake()

df['Key\_words'] = ""

for index, row in df.iterrows():

    r.extract\_keywords\_from\_text(row['Plot'])

    key\_words\_dict\_scores = r.get\_word\_degrees()

    row['Key\_words'] = list(key\_words\_dict\_scores.keys())

df.drop(columns = ['Plot'], inplace = True)

CELL6:

df['Genre'] = df['Genre'].map(lambda x: x.split(','))

df['Actors'] = df['Actors'].map(lambda x: x.split(',')[:3])

df1['Director'] = df['Director'].map(lambda x: x.split(','))

for index, row in df.iterrows():

      row['Genre'] = [x.lower().replace(' ','') for x in row['Genre']]

      row['Actors'] = [x.lower().replace(' ','') for x in row['Actors']]

      row['Director'] = [x.lower().replace(' ','') for x in row['Director']]

CELL7:

df['Bag\_of\_words'] = ''

columns = ['Genre', 'Director', 'Actors', 'Key\_words']

for index, row in df.iterrows():

    words = ''

    for col in columns:

        words += ' '.join(row[col]) + ' '

    row['Bag\_of\_words'] = words

df = df[['Title','Bag\_of\_words']]

df

CELL8:

count = CountVectorizer()

count\_matrix = count.fit\_transform(df['Bag\_of\_words'])

cosine\_sim = cosine\_similarity(count\_matrix, count\_matrix)

print(cosine\_sim)

CELL9:

indices = pd.Series(df['Title'])

CELL10:

def recommend(title, cosine\_sim = cosine\_sim):

    recommended\_movies = []

    idx = indices[indices == title].index[0]

    score\_series = pd.Series(cosine\_sim[idx]).sort\_values(ascending = False)

    top\_10\_indices = list(score\_series.iloc[1:11].index)

    for i in top\_10\_indices:

        recommended\_movies.append(list(df['Title'])[i])

    return recommended\_movies

CELL11:

recommend('The Godfather')