

## CODE

```
import pandas as pd

import numpy as np

from sklearn.feature_extraction.text import CountVectorizer

from sklearn.metrics.pairwise import cosine_similarity


def get_title_from_index(index):

    return df[df.index == index]["title"].values[0]


def get_index_from_title(title):

    return df[df.title == title]["index"].values[0]


df = pd.read_csv("F://New folder (2)//movie_dataset1.csv")


#Printing dataframe created


features = ['keywords','cast','genres','director']

##Step 3: Create a column in DF which combines all selected features

for feature in features:

    df[feature] = df[feature].fillna("")


def combine_features(row):

    try:
```

```
        return row['keywords'] + " " + row['cast'] + " " + row['genres'] + "  
        " + row['director']
```

```
    except:
```

```
        print("Error:" + row)
```

```
df["combined_features"] = df.apply(combine_features,axis=1)
```

```
#print "Combined Features:", df["combined_features"].head()
```

```
#Step 4: Create count matrix from this new combined column
```

```
cv = CountVectorizer()
```

```
count_matrix = cv.fit_transform(df["combined_features"])
```

```
##Step 5: Compute the Cosine Similarity based on the count_matrix
```

```
cosine_sim = cosine_similarity(count_matrix)
```

```
#Now taking input from the user
```

```
movie_index = get_index_from_title(movie_user_likes)
```

```
similar_movies = list(enumerate(cosine_sim[movie_index]))
```

```
## Step 7: Get a list of similar movies in descending order of similarity score
```

```
sorted_similar_movies = sorted(similar_movies,key=lambda x:x[1],reverse=True)
```

```
## Step 8: Print titles of first 5 movies
```

```
i=0
```

```
for element in sorted_similar_movies:
```

```
    print(get_title_from_index(element[0]))
```

```
    i=i+1
```

```
    if i>5:
```

```
        break
```