Day 43 — Jan 11. Jupyter NB

```
cosi_sim = cosine_similarity(count_matrix)
movie_user_likes = "Avatar"

def get_index_from_title(title):
    return df[df.title == title]["index"].values[0]

def get_title_from_index(index):
    return df[df.index == index]["title"].values[0]

movie_index = get_index_from_title(movie_user_likes)

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

generates a tuple where first element is "index" and second is "score".

enumerate gives index and value

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

largest to highest

x[1]: sim score

x[0]: index of movies

Print 10 similar movies

```
i = 0
for element in sorted_similar_movies:
    print (get_title_from_index (element[0]))
    i = i + 1
    if i > 10:
        break.
```

```
for i in range (10):
    index = sorted_similar_movies[i][0]
    title = get_title_from_index(index)
    print(title)
```

Generate full code on paper.

1. Import libraries.

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

Helper fxn

```
def get_index_from_title(title):
    return df[df.title == title]["index"].values[0]

def get_title_from_index(index):
    return df[df.index == index]["title"].values[0]
```

2. Read movie_dataset

```
df = pd.read_csv("movie_dataset.csv")
```

3. Select features of interest

```
features = ["keyword", "genre", "cast", "director"]
```

4. Create a new column in the dataset that combines all selected features

```
for feature in features:
    df[feature]: df[feature].fillna('')
```

```
def combine_features(row):
    try:
        return row["keyword"] + " " + row["genre"]
        + " " + row("cast") + " " +
        row("director")
    except:
        print("Error:", row)

df["combine features"]: df.apply(combine_features,
                                 axis=1)
```

5. Create count matrix

```
cv= CountVectorizer()
count_matrix = cv.fit_transform(df["combine
    features"])
array : count_matrix.toarray()
```

6. Compute the cosine similarity b/n the movies based on count matrix

cos_sim = cosine_similarity(count_matrix)

7. Retrieve movie user like = "Avatar" user likes

movie_index = get_index_from_title(movie_user_like)

similar_movies = list(enumerate(cos_sim[movie_index]))

sort_similar_movies = sorted(similar_movies, key=lambda x: x[0], reverse=True)

7.

8. List of similar movies

```
for i in range(10):
    index = sorted_similar_movies[i][0]
    title = get_title_from_index(index)
    print(title)
```