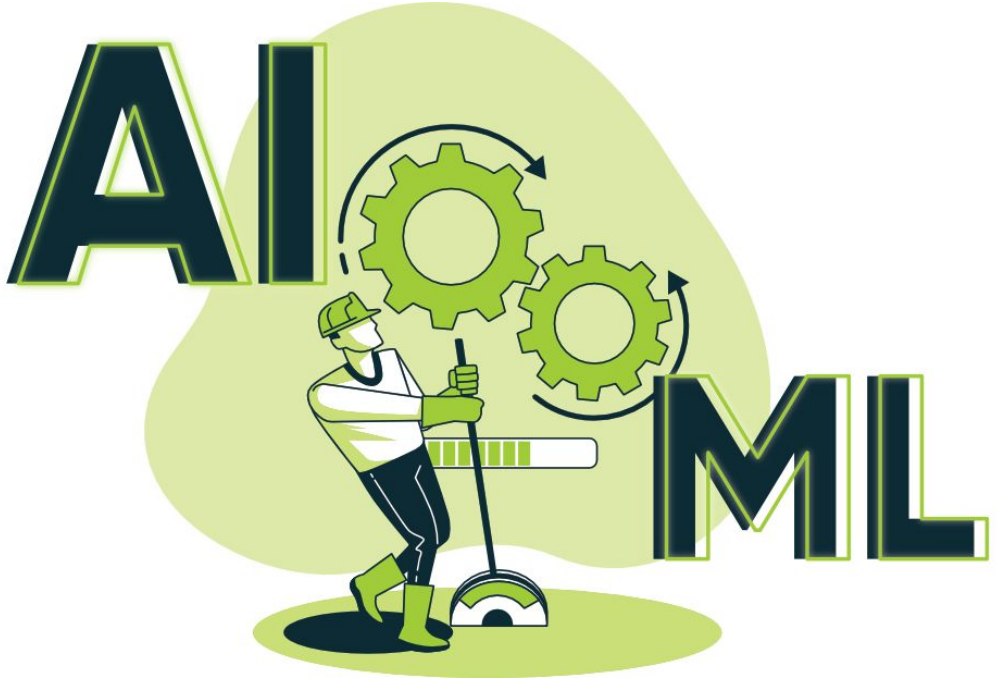
Movie Recommendation System(Pearson Correlation based)

Project By:

- Survi Pandey 21301221046
- Mohit Naskar 21301221108
- Siddh Kumar 21301221100
- Anshu Kumar 21301221009

- Project Guide
  Prof. Madhurima Banerjee

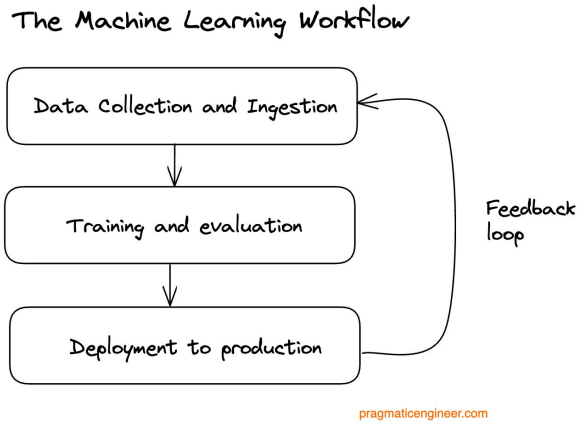
Index

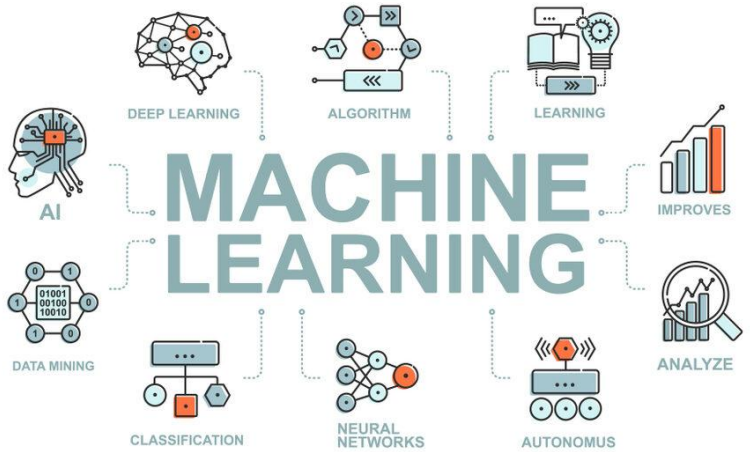


What is Machine Learning?

Machine learning (ML) is a type of artificial intelligence (AI) focused on building computer systems that learn from data.

This project is based on Machine Learning Modules and its development.

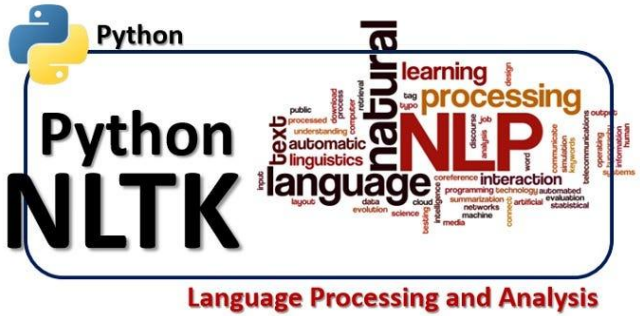




What is NLTK(Natural Language Toolkit)?

The **Natural Language Toolkit**, or more commonly **NLTK**, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language. It supports classification, tokenization, stemming, tagging, parsing, and semantic reasoning functionalities.

Natural language processing (**NLP**) is an interdisciplinary subfield of computer science and linguistics. It is primarily concerned with giving computers the ability to support and manipulate human language.



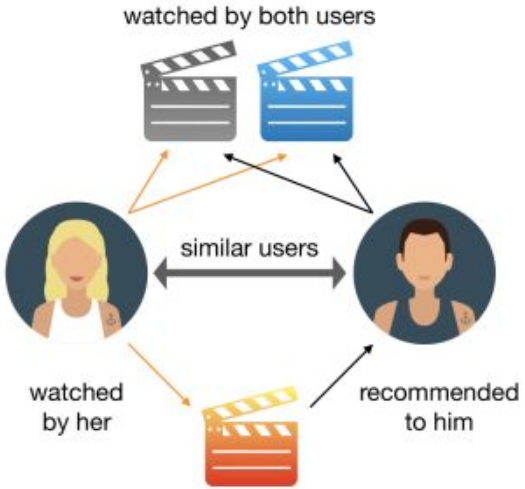
Introduction about the project topic!

What is a Recommendation System?

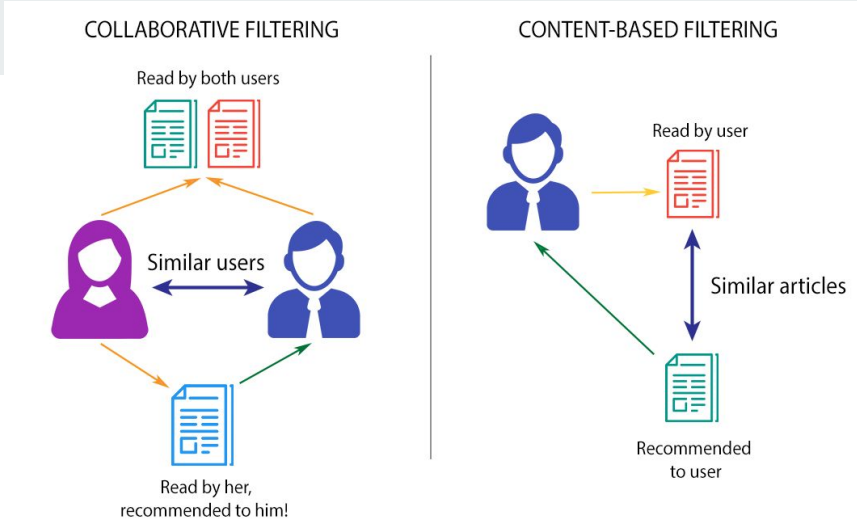
A recommendation system is a type of information filtering system which attempts to predicts the preferences of a user, and make suggestions based on these preferences.

We used Pearson Correlation Threshold value for the

recommendation.



Methodology



- Content-Based Filtering: This system suggests similar items based on a particular item. It uses item metadata, such as genre, director, description, actors, etc. for movies, to make these recommendations.
- Collaborative Filtering: These systems try to predict the rating or preference that a user would give an item based on past ratings and preferences of other users.
-
- Hybrid Engine: This is a combination of both content-based and collaborative filtering systems.

About the dataset.

```
[369]: df.head()
```

[369]:

	Movie Name	Movie ID	User ID	Reviews	Ratings
0	The Shawshank Redemption	1	100	The Shawshank Redemption holds the Number 1 sp...	10
1	The Shawshank Redemption	1	101	However delightful as it is The Shawshank Rede...	10
2	The Shawshank Redemption	1	102	However delightful as it is The Shawshank Rede...	10
3	The Shawshank Redemption	1	104	"The Shawshank Redemption" is a cinematic gem ...	10
4	The Shawshank Redemption	1	105	"Hope is a good thing probably best of all & g...	10

```
[370]: df.shape
```

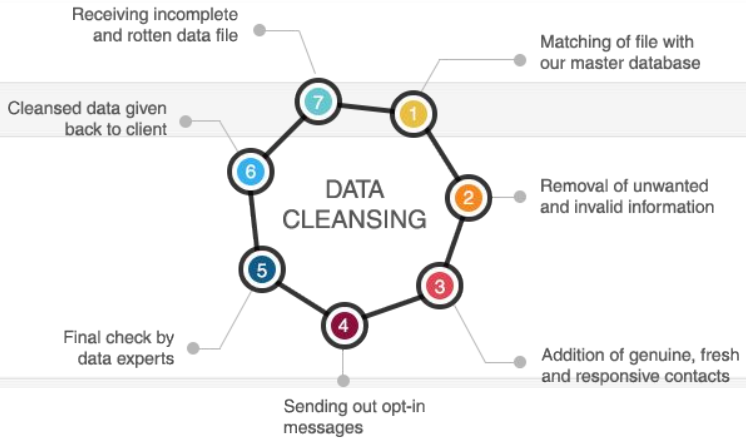
```
[370]: (101, 5)
```

This dataset has Columns named Movie Name,Movie ID,User ID, Reviews of the Movies and Ratings

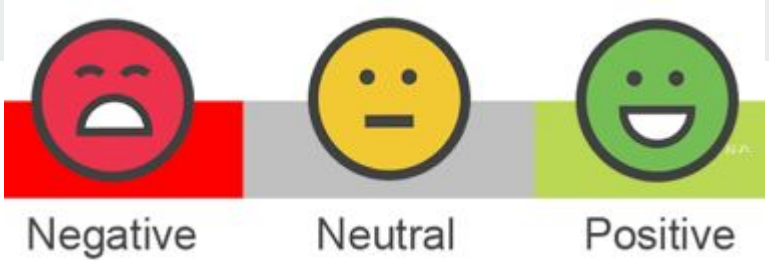
DATASET

Data Cleaning.

```
[371]: df.duplicated().sum()

[371]: 0

[372]: df.isna().sum()

[372]: Movie Name    0
       Movie ID      0
       User ID       0
       Reviews       0
       Ratings       0
       dtype: int64
```



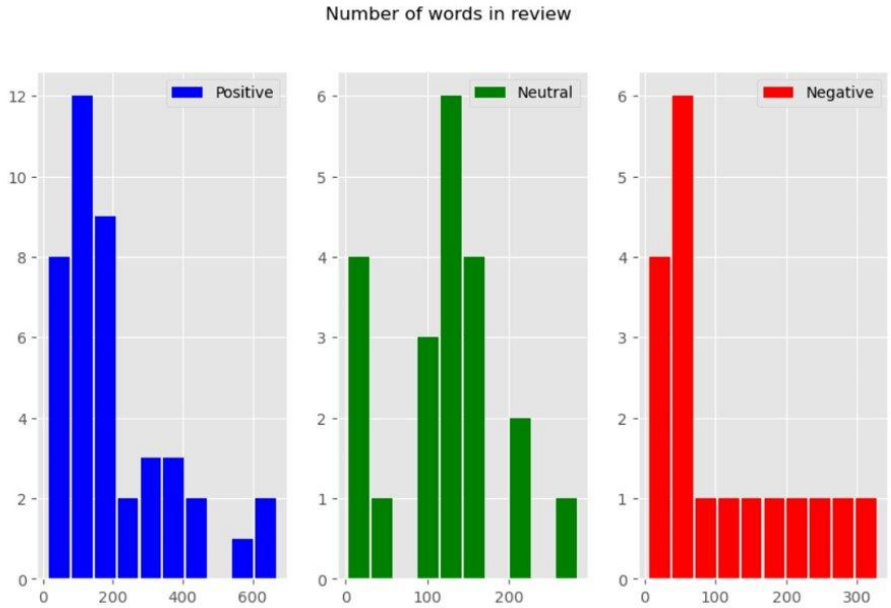
The above code shows that the dataset has no duplicacy as well as no NAN values



Calculating the Sentiment of the users on the basis the Reviews and Ratings provided

	Movie Name	Movie ID	User ID	Reviews	Ratings	word count	sentiment
96	There Will Be Blood	10	127	It's about expansion, it's about capitalism, a...	10	618	3
97	There Will Be Blood	10	107	Totally confused, but never fear, I got it.\r\...	5	32	2
98	There Will Be Blood	10	115	I was looking forward to seeing this when it w...	5	425	2
99	There Will Be Blood	10	128	I think the last significant movie I saw that ...	3	307	2
100	There Will Be Blood	10	102	Do yourself a favor and do not waste your time...	2	140	1

Data Visualization



Graph based on the words per Review

Tokenization

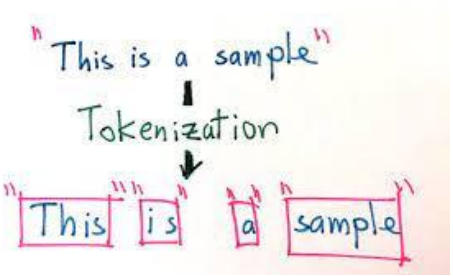
Tokenization refers to a process by which a piece of sensitive data, such as a credit card number, is replaced by a surrogate value known as a token

```
def tokenize_review(review):
    return nltk.word_tokenize(review)
```

```
df['tokenized_review'] = df['Reviews'].apply(lambda x: tokenize_review(x))
```

```
df[:5]
```

	Movie Name	Movie ID	User ID	Reviews	Ratings	word count	sentiment	tokenized_review
0	The Shawshank Redemption	1	100	shawshank redemption holds number 1 spot top25...	10	458	3	[shawshank, redemption, holds, number, 1, spot...
1	The Shawshank Redemption	1	101	however delightful shawshank redemption allego...	10	671	3	[however, delightful, shawshank, redemption, a...
3	The Shawshank Redemption	1	104	shawshank redemption cinematic gem transcends ...	10	401	3	[shawshank, redemption, cinematic, gem, transc...
4	The Shawshank Redemption	1	105	hope good thing probably best good thing never...	10	291	3	[hope, good, thing, probably, best, good, thin...
5	The Dark Night	2	106	confidently directed dark brooding packed impr...	10	45	3	[confidently, directed, dark, brooding, packed...



Stemming of Data

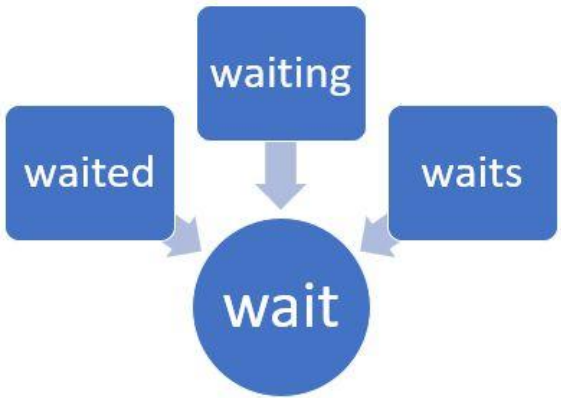
The goal of stemming is to reduce words to their base or root form, known as the "stem."

```
def stem_sentence(sentence):
   return [stemmer.stem(word) for word in sentence]
```

```
# Apply the function to the review column of the dataframe
df['stem_new'] = df['tokenized_review'].apply(lambda x: stem_sentence(x))
```

```
df[:5]
```

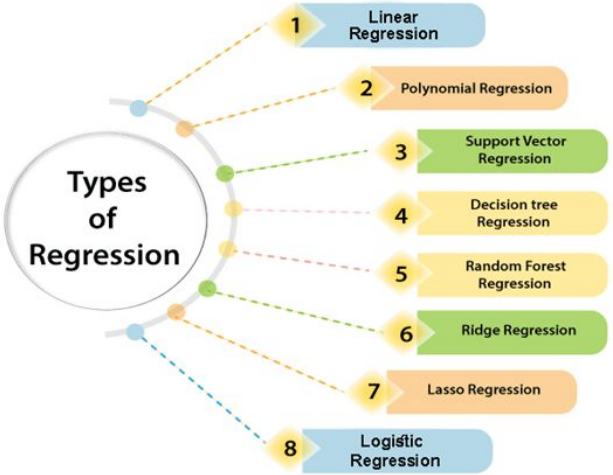
	Movie Name	Movie ID	User ID	Reviews	Ratings	word count	sentiment	tokenized_review	stem_new
0	The Shawshank Redemption	1	100	shawshank redemption holds number 1 spot top25...	10	458	3	[shawshank, redemption, holds, number, 1, spot...	[shawshank, redempt, hold, number, 1, spot, to...
1	The Shawshank Redemption	1	101	however delightful shawshank redemption allego...	10	671	3	[however, delightful, shawshank, redemption, a...	[howev, delight, shawshank, redempt, allegori,...
3	The Shawshank Redemption	1	104	shawshank redemption cinematic gem transcends ...	10	401	3	[shawshank, redemption, cinematic, gem, transc...	[shawshank, redempt, cinemat, gem, transcend, ...
4	The Shawshank Redemption	1	105	hope good thing probably best good thing never...	10	291	3	[hope, good, thing, probably, best, good, thin...	[hope, good, thing, probabl, best, good, thing...
5	The Dark Night	2	106	confidently directed dark brooding packed impr...	10	45	3	[confidently, directed, dark, brooding, packed...	[confid, direct, dark, brood, pack, impress, a...



Building Models to find the accuracy of the train and test data

```
#linear Logistic Regression Model
logreg=linear_model.LogisticRegression()
logreg.fit(x_train,y_train)
logreg_pred=logreg.predict(x_test)
log_acc = accuracy_score(logreg_pred, y_test)
print("Test accuracy: {:.2f}%".format(log_acc * 100))
```

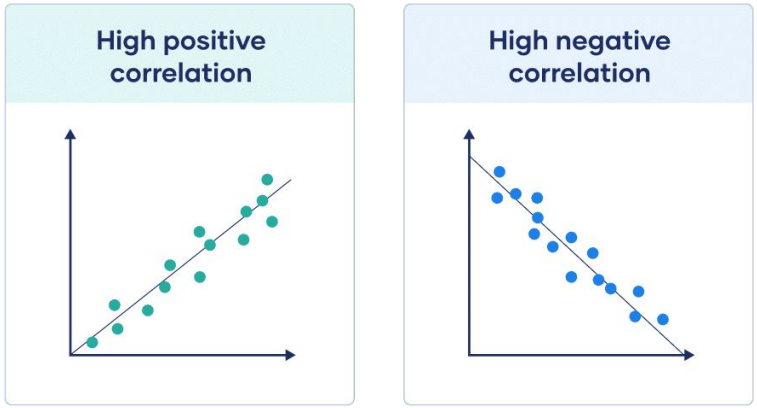
```
Test accuracy: 70.00%
```

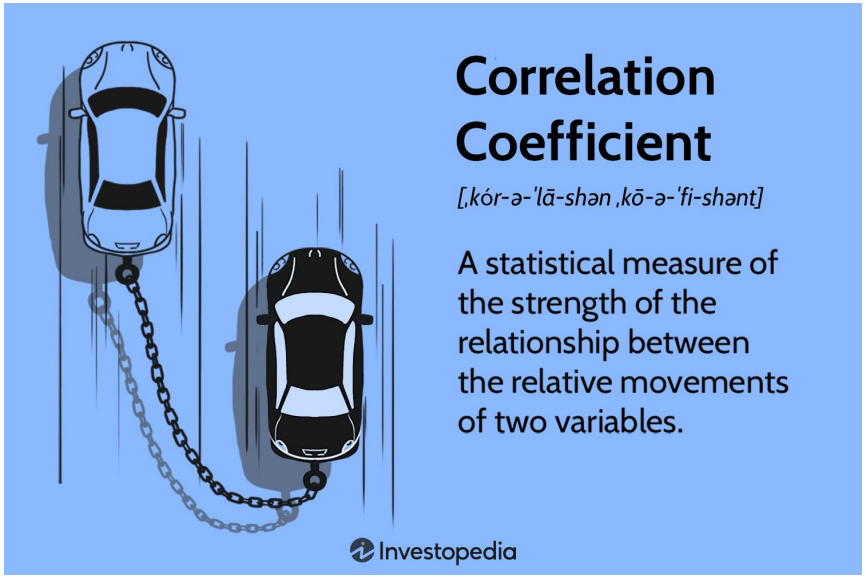


This is a image of one type of models developed in the project to check the accuracy of the data.

Correlation

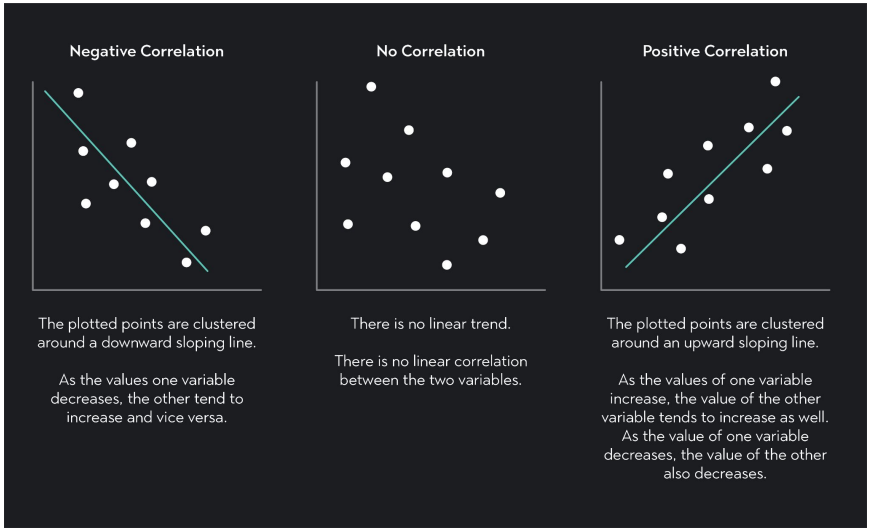
Correlation coefficients quantify the association between variables or features of a dataset.





Pearson Correlation

The Pearson correlation measures the strength of the linear relationship between two variables.



It has a value between -1 to 1, with a value of -1 meaning a total negative linear correlation, 0 being no correlation, and + 1 meaning a total positive correlation.

```
def calculate_pearson_correlation():
        # I'm not checking the condition len(x),len(y) >1 (since no negative ratings are present)
        # Number of elements not checked margined to 3 for all

        # mean of ratings of x and y
        mean_x = df4['AverageRatings'].iloc[0]
        mean_y = df5['AverageRatings'].iloc[0]

        # substract mean of each element of both pair
        diff_x = []
        # for loop to iterate through the column and return the ratings for user 1
        for i in range(df4.shape[0]):
                result = df4['Ratings'].iloc[i]-df4['AverageRatings'].iloc[0]
                diff_x.append(result)
        diff_y =[]
        # for loop to iterate through the column and return the ratings of user 2
        for i in range(df5.shape[0]):
                result = df5['Ratings'].iloc[i]-df5['AverageRatings'].iloc[0]
                diff_y.append(result)


        # sum of the product of each pair of element from the two array
        sum_product_diff = sum([diff_x[i] * diff_y[i] for i in range(2)])


        # calculate the standard deviation of each array

        std_x =[]
        # for loop to iterate through the column and return the ratings for user 1
        for i in range(df4.shape[0]):
                result = df4['Ratings'].iloc[i]
                std_x.append(result)
        # print(std_x)
        standard_deviation_x = np.std(std_x)

        std_y =[]
        # for loop to iterate through the column and return the ratings for user 2
        for i in range(df5.shape[0]):
                result = df5['Ratings'].iloc[i]
                std_y.append(result)
        # print(std_y)
        standard_deviation_y = np.std(std_y)
        # print(standard_deviation_x ,standard_deviation_y)

        # calculate Pearson correlation
        pearson_correlation = sum_product_diff / (len(std_x) - 1) / standard_deviation_x / standard_deviation_y
        return pearson_correlation
```

Working of Pearson Correlation.

$$r = \frac{N\Sigma xy - (\Sigma x)(\Sigma y)}{\sqrt{[N\Sigma x^2 - (\Sigma x)^2][N\Sigma y^2 - (\Sigma y)^2]}}$$

Where:

- N = number of pairs of scores
- Σxy = sum of the products of paired scores
- Σx = sum of x scores
- Σy = sum of y scores
- Σx^2 = sum of squared x scores
- Σy^2 = sum of squared y scores

The Output of the System

```
Movie/Movies that can be suggested to user 1 : 112

   Movie ID
0         8
1         1
2         4
3         7

Movies/Movie that can be suggested to user 2 : 109

   Movie ID
0         9


Empty markdown cell, double-click or press enter to edit.
```

Based on the threshold value of the Pearson Correlation , the movies are suggested.

```
if (value >=0.65)
        suggest movies;
else
        printf("The users are not related");
```

Conclusion

This is a recommendation system project which will recommend movies to the users based on a threshold value from the pearson correlation. The data will be processed using different models and different data filtering techniques. Then the processed data will be used in different models to achieve the output of the project.

Acknowledgement

We would like to express my special thanks of gratitude to my project guide Prof. Madhurima Banerjee ma'am for her immense support and guidance in completing my project.
I would also like to extend my gratitude to the coordinator Prof. Atindra Nag and the Principal sir "Dr Gour Banerjee" for providing us the facility that was required.

Date:
19/11/2023