KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY (KIIT)

Deemed to be University U/S 3 of UGC Act, 1956

MOVIE RECOMMENDATION SYSTEM

-RAHUL KR. BISWAS

INTRODUCTION

- A recommendation system is a filtering system that anticipates the users' choices and then suggests the most accurate results based on the users' past preferences.
- We have a variety of applications for our recommendation system that we have used throughout the years and are actively applying on various internet platforms.
- The purpose of a recommendation system is to search for content that would be interesting to an individual.
- We compare the different things with the user's interest profile in the content-based filtering technique. So, fundamentally, the user profile contains material that is much more relevant to using the form of the features.
- Previous actions or feedback is often considered, as is the description of the information that users of various selections have modified.
- The following are the stages required in obtaining a movie recommendation:
  With the title, you may locate the index of the movie.
  Computing the cosine similarity scores for each film
  Arrange the scores in ascending order, starting with the greatest priority.
  The group is then pruned based on the similarity scores.
- he following are the benefits of content-based filtering:
  We could suggest the unrated items.
  Can suggest movies depending on the user's ratings
  It is not possible to make the user like with un-likes.

METHODOLOGY:

CONTENT BASED FILTERING

- We compare the different things with the user's interest profile in the content-based filtering technique. So, fundamentally, the user profile contains material that is much more relevant to using the form of the features.
- Previous actions or feedback is often considered, as is the description of the information that users of various selections have modified.
- The following are the stages required in obtaining a movie recommendation:
  With the title, you may locate the index of the movie.
  Computing the cosine similarity scores for each film
  Arrange the scores in ascending order, starting with the greatest priority.
  The group is then pruned based on the similarity scores.
- The following are the benefits of content-based filtering:
  We could suggest the unrated items.
  Can suggest movies depending on the user's ratings
  It is not possible to make the user like with un-likes

SYSTEM DESIGN & DEVELOPMENT:

Dataset:-

For content-based filtration:

The dataset has been taken from Kaggle. It is used as the standard dataset by the movie
recommendation system.

1) Movie dataset is used in “content-based movie recommendation system” in this project.
2) The ratings and movies are taken into account.
3) Total number of movies.
4) Total number of ratings in a dataset.
5) An uniquid is assigned to every movie and the user.

This chapter involves both the hardware and software requirements needed for the project and detailed explanation of the specifications:-

Hardware Requirements:

- A PC with windows/linux OS
- Processor with 1.7-2.4ghz speed
- Minimum of 8gb RAM
- 2gb Graphic card

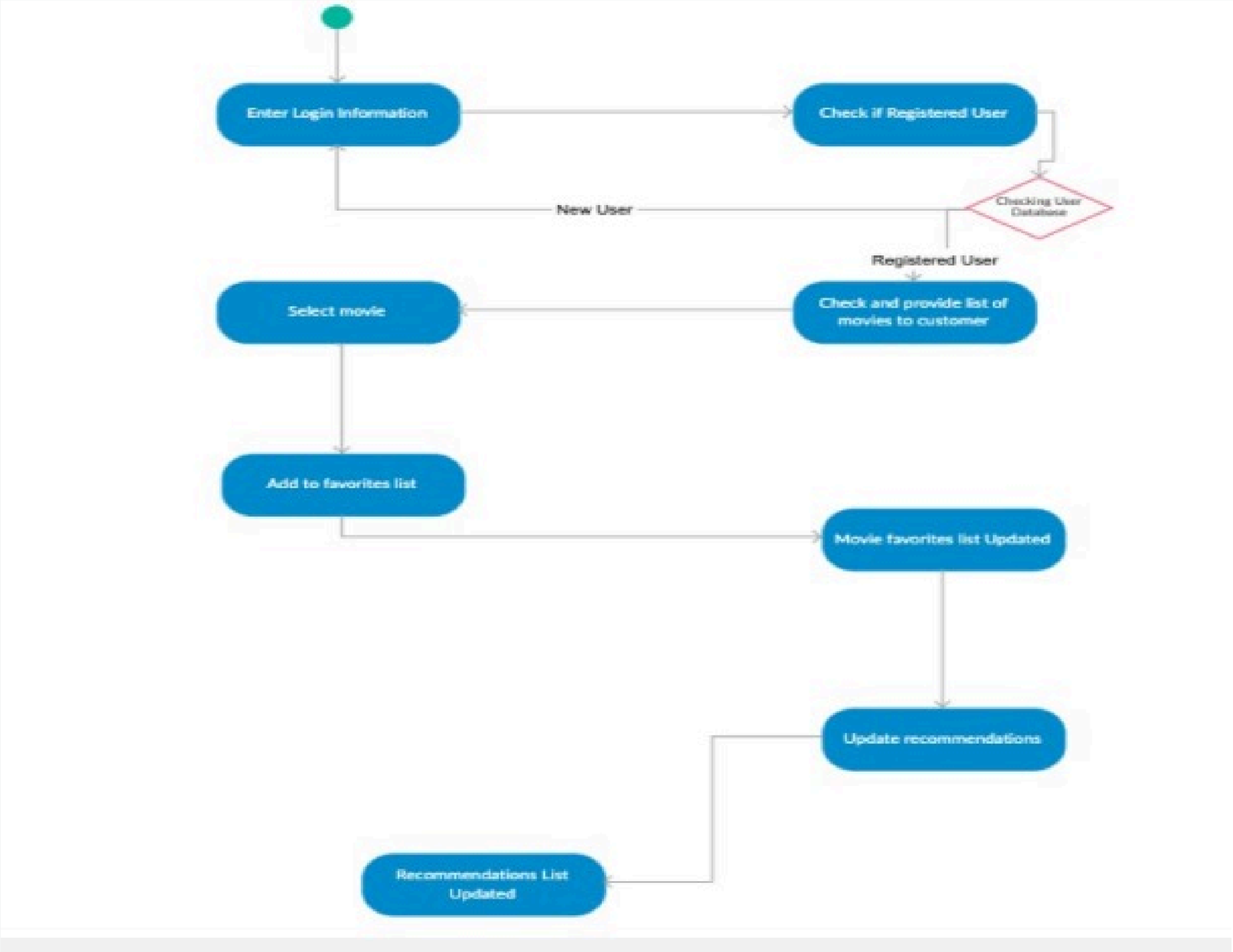
Software Specification:

- Text Editor(VS-code)
- Anaconda distribution package(PyCharm Editor)
- Python libraries

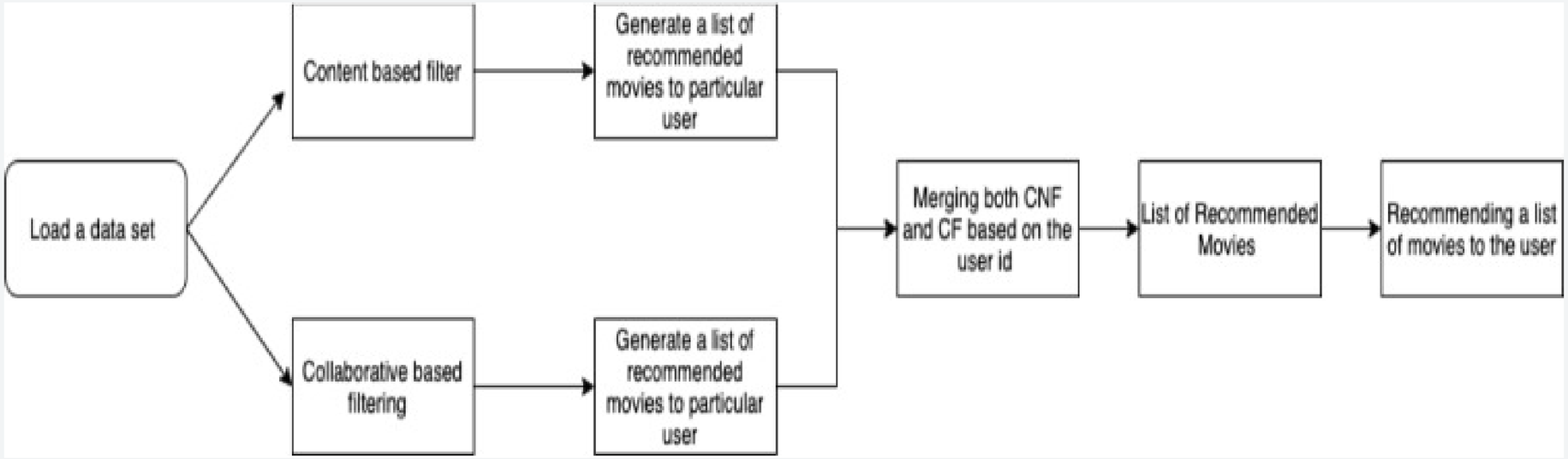
Software Requirements:

- Annaconda Distribution
- Python Libraries

ACTIVITY DIAGRAM:



DATA FLOW DIAGRAM:



INITIALLY LOAD THE DATA SETS THAT ARE REQUIRED TO BUILD A MODEL THE DATA SET THAT ARE REQUIRED IN THIS PROJECT ARE MOVIES.CSV, RATINFG.CSV, USERS.CSV ALL THE DATA SETS ARE AVAILABLE IN THE KAGGLE.COM. COMBINING BOTH BASED ON THE USEID A SINGLE FINAL LIST OF MOVIES ARE RECOMMENDED TO THE PARTICULAR USER

PERFORMANCE ANALYSIS:

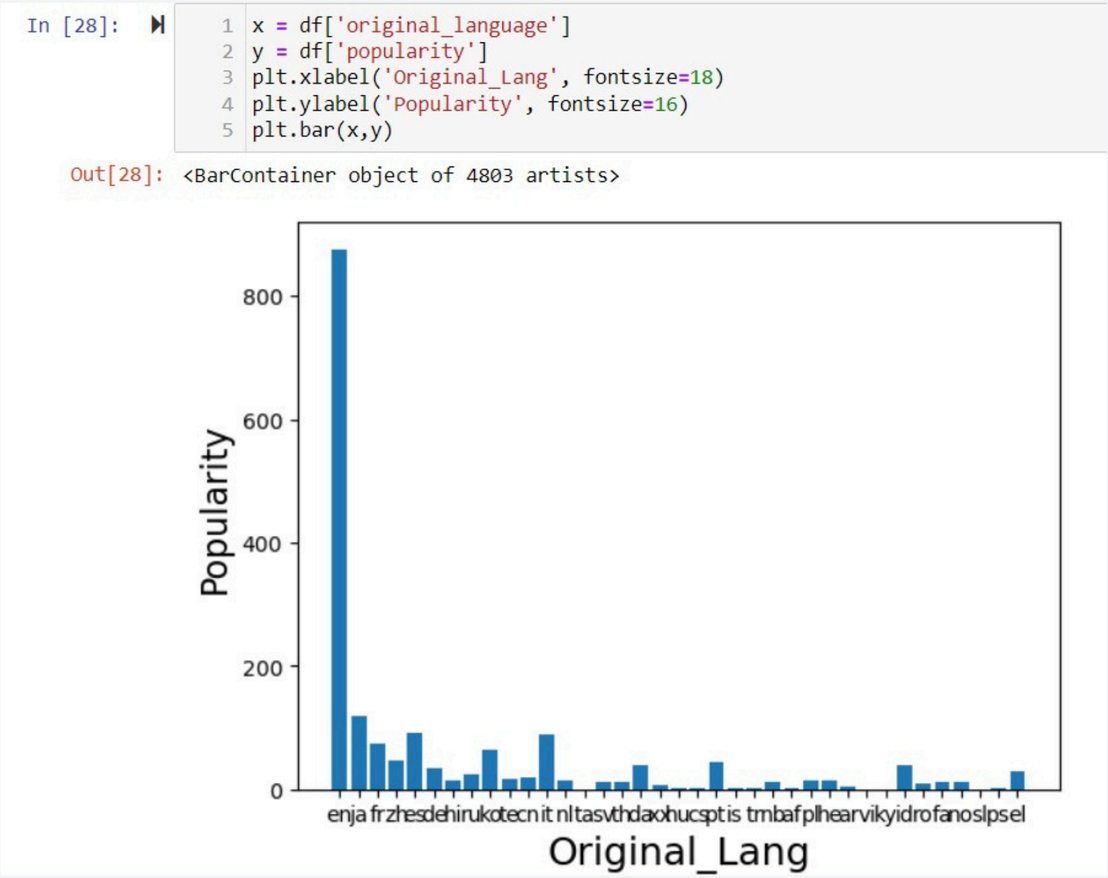
Content Based Filtering User profile holds the content that is much more matching to use the form of the features. The previous actions or for the feedback is taken into account a generally takes into account the description of the content that has been edited by the users of different choices.

Bar graph upon which language is being preferred most while watching a movie

In [28]:

```
x = df['original_language']
y = df['popularity']
plt.xlabel('Original_Lang', fontsize=18)
plt.ylabel('Popularity', fontsize=16)
plt.bar(x,y)
```

Out[28]: <BarContainer object of 4803 artists>

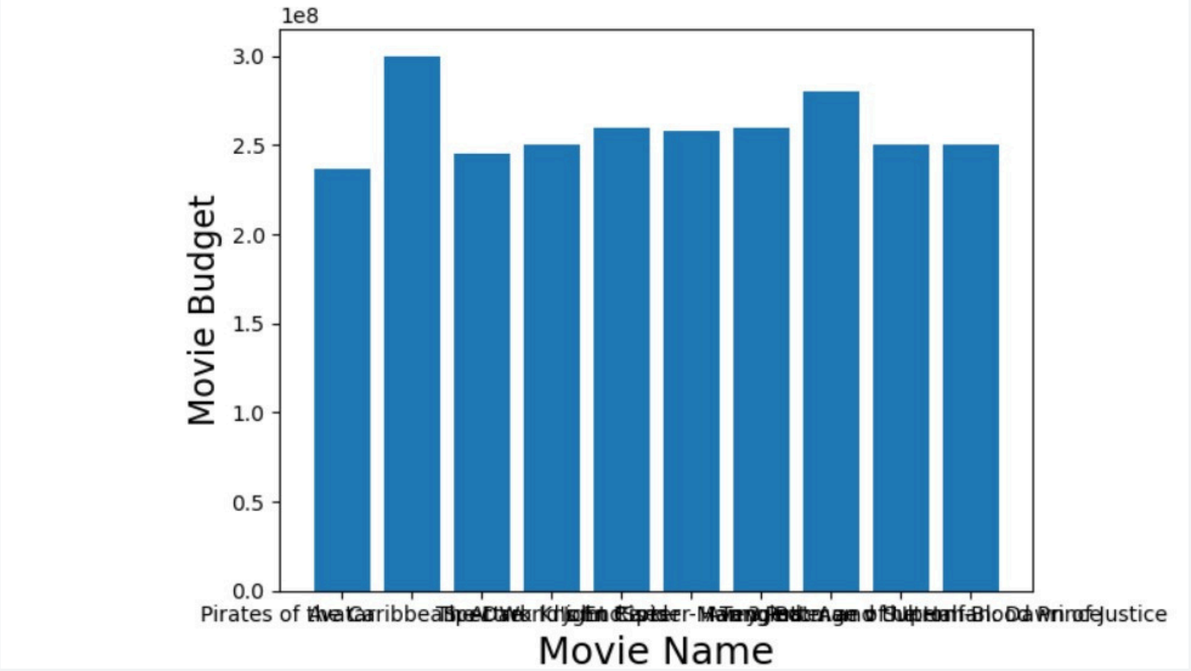


This graph shows first 10 films annual budget

In [71]:

```
x = df1['title']
y = df1['budget']
plt.xlabel('Movie Name', fontsize=18)
plt.ylabel('Movie Budget', fontsize=16)
plt.bar(x,y)
```

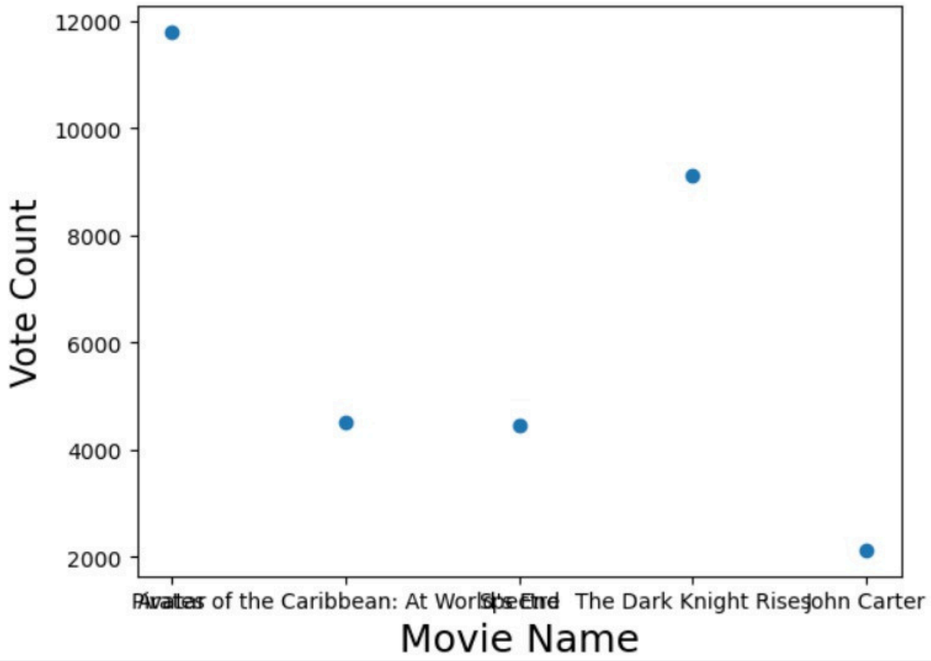
Out[71]: <BarContainer object of 10 artists>



Movie and their vote counts:

```
In [80]:
1 x = df2['title']
2 y = df2['vote_count']
3 plt.xlabel('Movie Name', fontsize=18)
4 plt.ylabel('Vote Count', fontsize=16)
5 plt.scatter(x,y)
```

```
Out[80]: <matplotlib.collections.PathCollection at 0x241df541630>
```



CONCLUSION:

Content-based movie recommendation systems continue to evolve, and they remain a vital component of the broader recommendation landscape. While they excel at providing personalized recommendations based on intrinsic movie features, overcoming challenges like the cold start problem and limited serendipity is crucial. Hybrid models and the integration of deep learning techniques are promising directions for improving the performance of content-based systems, ultimately enhancing the movie-watching experience for users. As the entertainment industry continues to grow, content-based filtering will undoubtedly play a significant role in helping users navigate an ever expanding catalog of movies.