Movie Recommendation System

Group 3

Naman Parashar
201550087

Himanshu Sharma
201550062

Piyush Agrawal
201550098

Data Selection:

Dataset: TMDB 5000 Movie Dataset

Link: [TMDB 5000 Movie Dataset | Kaggle]

```
In [90]: 1 movies.head()
```

Out[90]:

	movie_id	title	overview	genres	keywords	cast	crew	tags
0	19995	Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...	[Action, Adventure, Fantasy, ScienceFiction]	[cultureclash, future, spacewar, spacecolony, ...	[SamWorthington, ZoeSaldana, SigourneyWeaver]	[JamesCameron]	[In, the, 22nd, century,, a, paraplegic, Marin...
1	285	Pirates of the Caribbean: At World's End	[Captain, Barbossa,, long, believed, to, be, d...	[Adventure, Fantasy, Action]	[ocean, drugabuse, exoticisland, eastindiatrad...	[JohnnyDepp, OrlandoBloom, KeiraKnightley]	[GoreVerbinski]	[Captain, Barbossa,, long, believed, to, be, d...
2	206647	Spectre	[A, cryptic, message, from, Bond's, past, send...	[Action, Adventure, Crime]	[spy, basedonnovel, secretagent, sequel, mi6, ...	[DanielCraig, ChristophWaltz, LéaSeydoux]	[SamMendes]	[A, cryptic, message, from, Bond's, past, send...
3	49026	The Dark Knight Rises	[Following, the, death, of, District, Attorney...	[Action, Crime, Drama, Thriller]	[dccomics, crimefighter, terrorist, secretiden...	[ChristianBale, MichaelCaine, GaryOldman]	[ChristopherNolan]	[Following, the, death, of, District, Attorney...
4	49529	John Carter	[John, Carter, is, a, war-weary,, former, mili...	[Action, Adventure, ScienceFiction]	[basedonnovel, mars, medallion, spacetravel, p...	[TaylorKitsch, LynnCollins, SamanthaMorton]	[AndrewStanton]	[John, Carter, is, a, war-weary,, former, mili...

Data Cleaning:

Data cleansing or data cleaning is the process of detecting and correcting corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect.

```
1 movies_new.head()
```

	title	popularity	vote_average	vote_count	tags
0	Avatar	150.437577	7.2	11800	Action Adventure Fantasy ScienceFiction cultur...
1	Pirates of the Caribbean: At World's End	139.082615	6.9	4500	Adventure Fantasy Action ocean drugabuse exoti...
2	Spectre	107.376788	6.3	4466	Action Adventure Crime spy basedonnovel secret...
3	The Dark Knight Rises	112.312950	7.6	9106	Action Crime Drama Thriller dccomics crimefigh...
4	John Carter	43.926995	6.1	2124	Action Adventure ScienceFiction basedonnovel m...

Splitting into Training and Testing:

Scikit-learn alias sklearn is the most useful and robust library for machine learning in Python. The scikit-learn library provides us with the model_selection module in which we have the splitter function train_test_split().

```
1 movies_new.head()
```

	title	popularity	vote_average	vote_count	tags
0	Avatar	150.437577	7.2	11800	Action Adventure Fantasy ScienceFiction cultur...
1	Pirates of the Caribbean: At World's End	139.082615	6.9	4500	Adventure Fantasy Action ocean drugabuse exoti...
2	Spectre	107.376788	6.3	4466	Action Adventure Crime spy basedonnovel secret...
3	The Dark Knight Rises	112.312950	7.6	9106	Action Crime Drama Thriller dccomics crimefigh...
4	John Carter	43.926995	6.1	2124	Action Adventure ScienceFiction basedonnovel m...

Decision Tree –

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal, but are also a popular tool in machine learning.

```
|: 1 print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.92      0.86      0.89       616
           1       0.86      0.92      0.89       584

    accuracy                           0.89      1200
   macro avg       0.89      0.89      0.89      1200
weighted avg       0.89      0.89      0.89      1200
```

Cosine Similarity –

In data analysis, cosine similarity is a measure of similarity between two sequences of numbers. For defining it, the sequences are viewed as vectors in an inner product space, and the cosine similarity is defined as the cosine of the angle between them, that is, the dot product of the vectors divided by the product of their lengths. It follows that the cosine similarity does not depend on the magnitudes of the vectors, but only on their angle. The cosine similarity always belongs to the interval {\displaystyle [-1,1].}{\displaystyle [-1,1].} For example, two proportional vectors have a cosine similarity of 1, two orthogonal vectors have a similarity of 0, and two opposite vectors have a similarity of -1. The cosine similarity is particularly used in positive space, where the outcome is neatly bounded in [0,1]

```
def recommend(movie):
    index = movies_new[movies_new['title'] == movie].index[0]
    distances = sorted(list(enumerate(cs[index])),reverse=True,key = lambda x: x[1])
    for i in distances[1:6]:
        print(movies_new.iloc[i[0]].title)
```

```
recommend('Avatar')
```

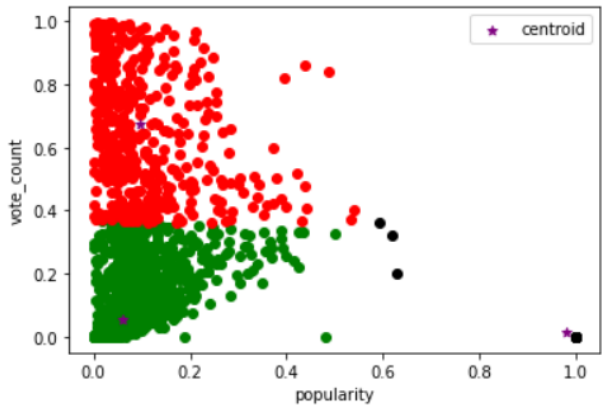
```
Star Trek Into Darkness
The Lovers
Jupiter Ascending
The Time Machine
The Mummy: Tomb of the Dragon Emperor
```

K-Means Clustering –

k-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. k-means clustering minimizes within-cluster variances (squared Euclidean distances), but not regular Euclidean distances, which would be the more difficult Weber problem: the mean optimizes squared errors, whereas only the geometric median minimizes Euclidean distances. For instance, better Euclidean solutions can be found using k-medians and k-medoids.

```
df1 = movies2[movies2.cluster==0]
df2 = movies2[movies2.cluster==1]
df3 = movies2[movies2.cluster==2]
plt.scatter(df1.popularity,df1['vote_count'],color='green')
plt.scatter(df2.popularity,df2['vote_count'],color='red')
plt.scatter(df3.popularity,df3['vote_count'],color='black')
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='purple',marker='*',label='centroid')
plt.xlabel('popularity')
plt.ylabel('vote_count')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x249787396d0>
```



Conclusion–

Cosine Similarity is performing best among the three (Decision Tree, Cosine Similarity, K-Means Clustering).

As cosine similarity is providing easy to find recommended movies on the basis of genre, tags, etc.

Where as,

Decision Tree is giving Boolean result on the basis of popularity, vote_count and vote_average.

And K-Means Clustering is just making group of huge number of movies which is making problematic to find the best by increasing the range.