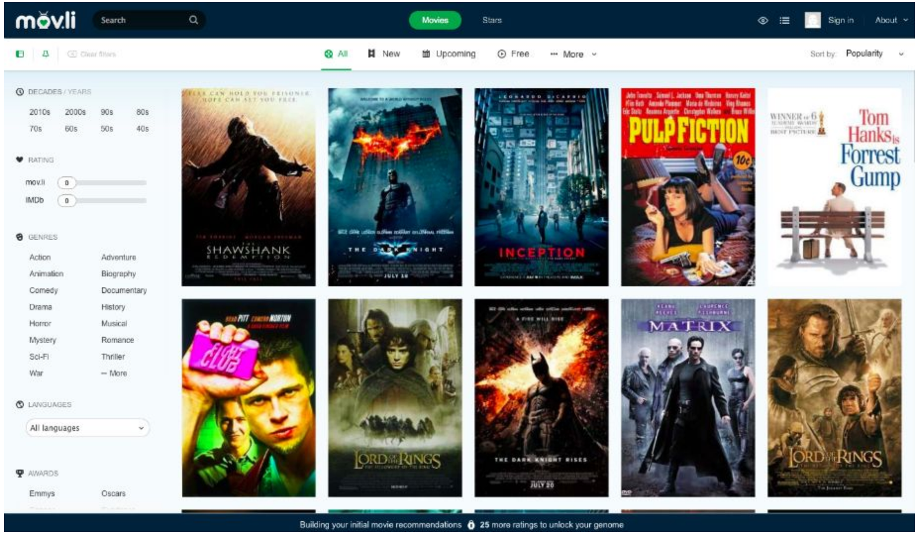
‘Bad Movies’ Recommendation

Xinyu Zang & Lin Jiang

Tired of these recommended movies?

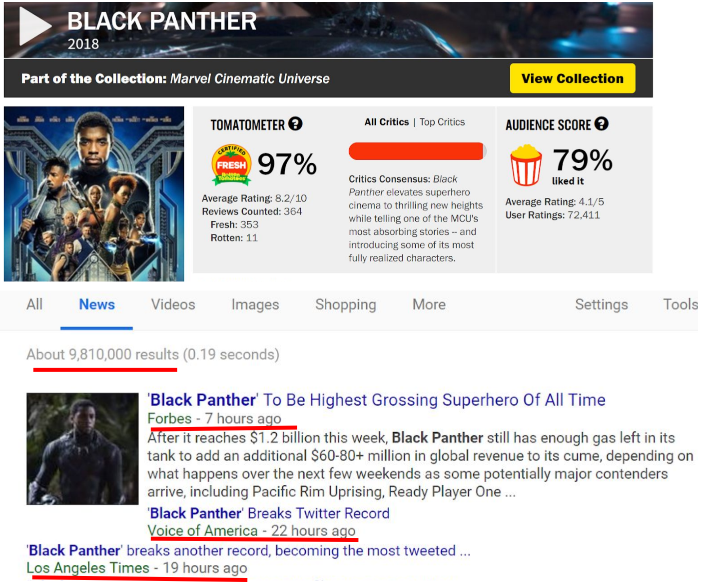


Famous & Highly rate & Seen before

We will recommend some “bad” ones you might haven’t seen!

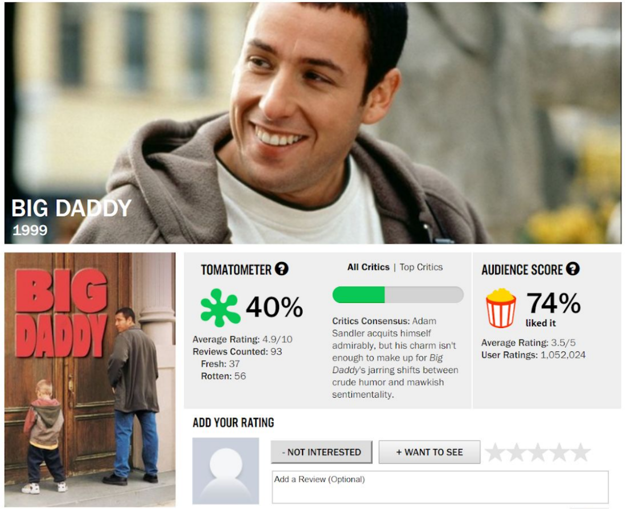
Intuition --- Only Bad for Movie Critics

- Some movies were surprisingly panned by critics, but had been very popular by general audiences
- These movies might not get the headlines in magazine or newspaper written by professional critics and therefore less exposure are very likely to be ignored by us.



Mission Statement

Based on user's taste such as favorite movie or favorite genres, recommend similar movie that users are very likely to ignore because of low ratings from critics



Data

MovieLens Latest Datasets

Collected by GroupLens, a research group at University of Minnesota.

Datasets include 100,000 ratings and 1,300 tag applications applied to 9,000 movies.

The Open Movie Database

Dataset contains movie ratings from both Rotten Tomatoes critics and the general audience. It also contains urls of movie posters.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
id	title	imdbID	spanishTit	imdbPictu	year	rtID	rtAllCriticsRating	rtAllCriticsNumReviews	rtAllCriticsNumFresh	rtAllCriticsNumRo	rtAllCriticsScore	rtTopCriticsRating	rtTopCriti	rtTopCriti	rtTopCriti	rtTopCriti	rtAudienc	rtAudienc	rtAudienceScore	rtPictureURL					
1	Toy story	114709	Toy story (	http://ia.r	1995	toy_story	9	73	73	0	100	8.5	17	17	0	100	3.7	102338	81	http://content7.flixster.com/movie/10/93/63/10936393_det.jpg					
2	Jumanji	113497	Jumanji	http://ia.r	1995	1068044-ju	5.6	28	13	15	46	5.8	5	2	3	40	3.2	44587	61	http://content8.flixster.com/movie/56/79/73/5679734_det.jpg					
3	Grumpy O	107050	Dos viejos	http://ia.r	1993	grumpy_o	5.9	36	24	12	66	7	6	5	1	83	3.2	10489	66	http://content6.flixster.com/movie/25/60/256020_det.jpg					
4	Waiting to	114885	Esperando	http://ia.r	1995	waiting_to	5.6	25	14	11	56	5.5	11	5	6	45	3.3	5666	79	http://content9.flixster.com/movie/10/94/17/10941715_det.jpg					
5	Father of t	113041	Vuelve el	http://ia.r	1995	father_of_	5.3	19	9	10	47	5.4	5	1	4	20	3	13761	64	http://content8.flixster.com/movie/25/54/255426_det.jpg					

Data Preprocessing

MovieLens Latest Datasets

- Clean movie title to build dictionary

	A	B
1	movieId	title
2	1	Toy Story (1995)
3	2	Jumanji (1995)
4	3	Grumpier Old Men (1995)
5	4	Waiting to Exhale (1995)
6	5	Father of the Bride Part II (1995)
7	6	Heat (1995)
8	7	Sabrina (1995)
9	8	Tom and Huck (1995)
10	9	Sudden Death (1995)
11	10	GoldenEye (1995)
12	11	American President, The (1995)

	A	B
1	movieId	title
2	1	Toy Story
3	2	Jumanji
4	3	Grumpier Old Men
5	4	Waiting to Exhale
6	5	Father of the Bride Part II
7	6	Heat
8	7	Sabrina
9	8	Tom and Huck
10	9	Sudden Death
11	10	GoldenEye
12	11	The American President

The Open Movie Database

- Check missing values
- Select movies: Critics Score<50 AND Audience Score>50 AND Number of Audience Ratings>200 as potential recommendations

Model Selection

- Memory-Based (User-User & Item-Item) v.s. Model-Based (SVD)
- Use Cross Validation and Root Mean Squared Error (RMSE) to evaluate the model
  - In each iteration of CV, use ratings from the training folds to estimate the the rating in the test fold and compute RMSE
  - Average RMSE in each iteration to get the CV-RMSE
- Model-Based Collaborative Filtering has relatively low RMSE.

```
#Print out the RMSE
print('User-based CF cross-valication RMSE: ' + str(np.mean(user_rmse)))
print('Item-based CF cross-valication RMSE: ' + str(np.mean(item_rmse)))
```

```
User-based CF cross-valication RMSE: 3.09088051696
Item-based CF cross-valication RMSE: 3.43873881381
```

```
user_ratings_mean = np.mean(train_data_matrix, axis = 1)
normal_train_data_matrix = train_data_matrix- user_ratings_mean.reshape(-1, 1)

u, s, vt = svds(normal_train_data_matrix, k = 20)
s_diag_matrix=np.diag(s)
X_pred = np.dot(np.dot(u, s_diag_matrix), vt) + user_ratings_mean.reshape(-1, 1)
print('User-based CF MSE with K=20: ' + str(rmse(X_pred, test_data_matrix)))
```

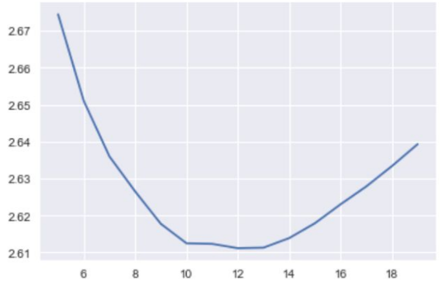
```
User-based CF MSE with K=20: 2.6461986099138306
```

Model-Based Collaborative Filtering

- Apply SVD to get a low- dimensional hidden feature matrix from the sparse rating matrix
- Use 5-folds Cross Validation to choose the optimal K-value which achieves minimal RMSE.

```
plt.plot(k_range, mean_rmse_value)
k_optima=k_range[np.argmin(mean_rmse_value)]
print('The optimal K value is ' + str(k_optim) + ' with RMSE=' + str(np.min(mean_rmse_value)))
```

```
The optimal K value is 12 with RMSE=2.61112085349
```



Model-Based Collaborative Filtering

- Measure cosine similarity of the hidden feature matrix. For input movie, find 10 similar underrated movies Ids

```
def top_cosine_similarity(low_dim_data, movie_name, top_n):
    movie_name = movie_name
    low_case_movie_name = movie_name.lower()
    if low_case_movie_name in lower_case_movie_dict:
        index = Rating_df.columns.get_loc(lower_case_movie_dict[low_case_movie_name])
        movie_row = low_dim_data[:, index]
        magnitude = np.sqrt(np.einsum('ij, ij -> i', low_dim_data.T, low_dim_data.T))
        similarity = np.dot(movie_row, low_dim_data) / (magnitude[index] * magnitude)
        pre_sort_indexes = np.argsort(-similarity)
        #Delete the index of input movie itself
        sort_indexes = np.delete(pre_sort_indexes, 0)
        top_movie_id = Rating_df.columns[sort_indexes]
        frozen_selected_id = frozenset(selected_id)
        selected_top_movie_id = [x for x in top_movie_id if x in frozen_selected_id][:top_n]
        return selected_top_movie_id
    else:
        return("Sorry, no matching found in the dataset")
```

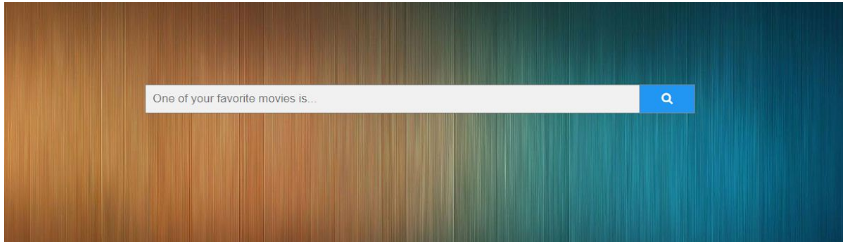
- Retrieve the relative information from the recommended Ids

```
def get_movie_infor(movie_df, movie_id):
    indexes = []
    for id in movie_id:
        index = movie_df[movie_df['id']==id].index.astype(int)[0]
        indexes.append(index)
    get_content = movie_df.iloc[indexes]
    return get_content[['title','rtAllCriticsScore', 'rtAudienceScore', 'rtPictureURL']]
```

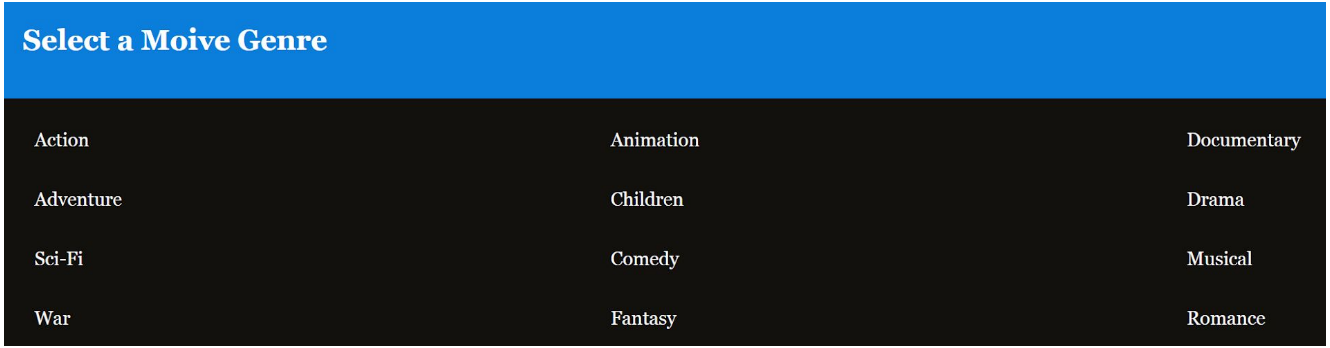
- For recommendation based on genres, get top 10 highest audience scores movies from each genre.

Input

- One of your favourite movies

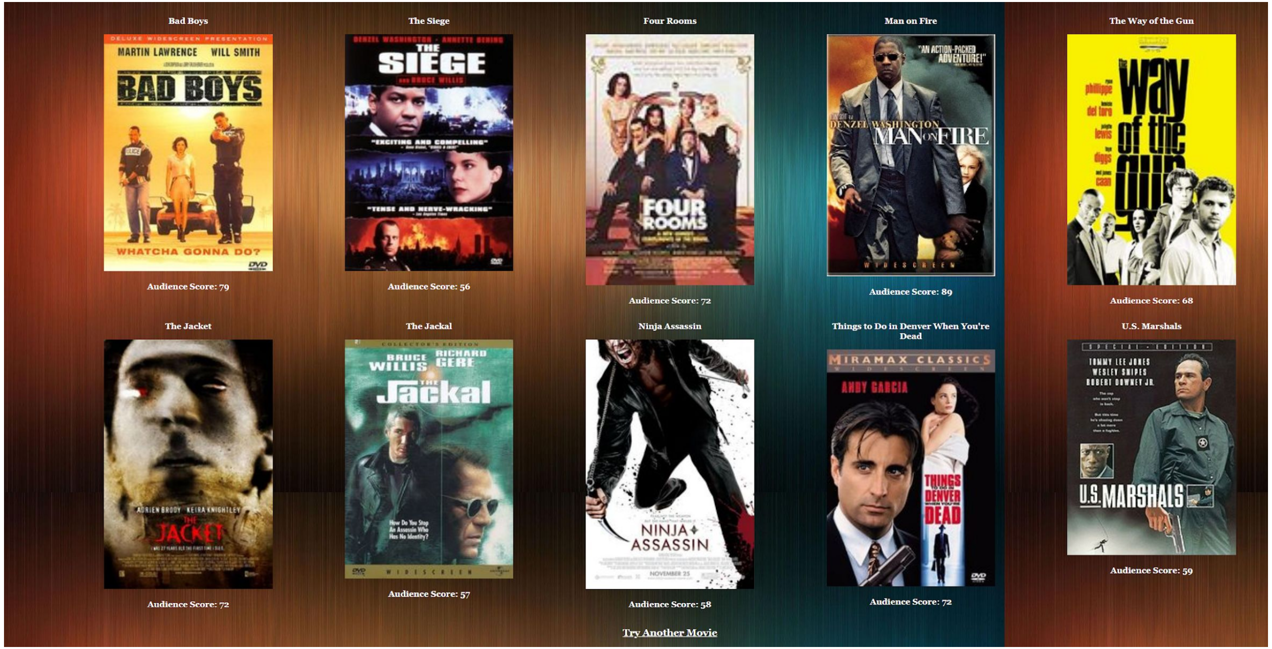


- One of you favorite film genres



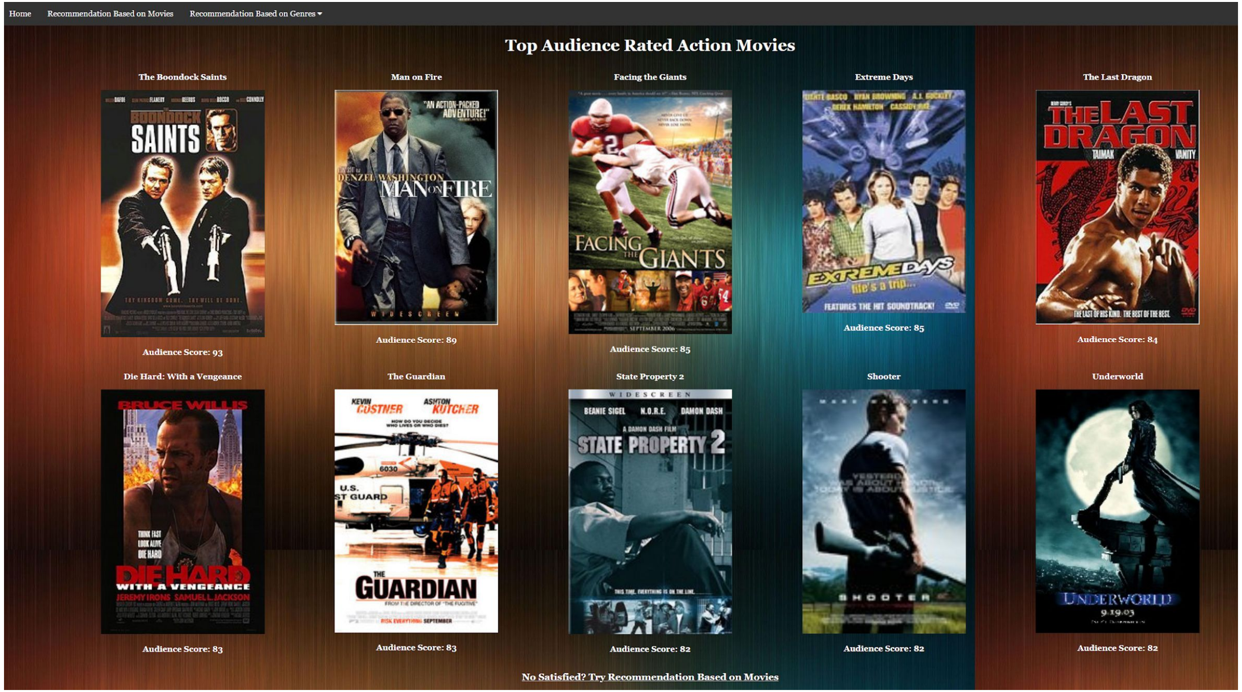
Output

- 10 similar ‘underrated’ movies you may like based on one of your favorite movies with audience scores



Output

- Top 10 ‘underrated’ movies by the critics from your favorite genre with audience scores



Demo

- Let’s try it with one of my favorite movies ‘Heat’, an American crime movie starring Robert De Niro, Al Pacino.

- Most recommendations are action/crime movies. Some with high iMDb scores.