

# THE NETFLIX PRIZE

Recommender System

- Lim Jungmin

NETFLIX

## Netflix Prize

[Home](#)

[Rules](#)

[Leaderboard](#)

[Update](#)

### Leaderboard

Showing Test Score. [Click here to show quiz score](#)

Display top  leaders.

Rank	Team Name	Best Test Score	% Im
------	-----------	-----------------	------

Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic








1	<a href="#">BellKor's Pragmatic Chaos</a>	0.8567	
2	<a href="#">The Ensemble</a>	0.8567	
3	<a href="#">Grand Prize Team</a>	0.8582	
4	<a href="#">Opera Solutions and Vandelay United</a>	0.8588	
5	<a href="#">Vandelay Industries !</a>	0.8591	
6	<a href="#">PragmaticTheory</a>	0.8594	
7	<a href="#">BellKor in BiqChaos</a>	0.8601	
8	<a href="#">Dace</a>	0.8612	
9	<a href="#">Feeds2</a>	0.8622	
10	<a href="#">BiqChaos</a>	0.8623	
11	<a href="#">Opera Solutions</a>	0.8623	
12	<a href="#">BellKor</a>	0.8624	

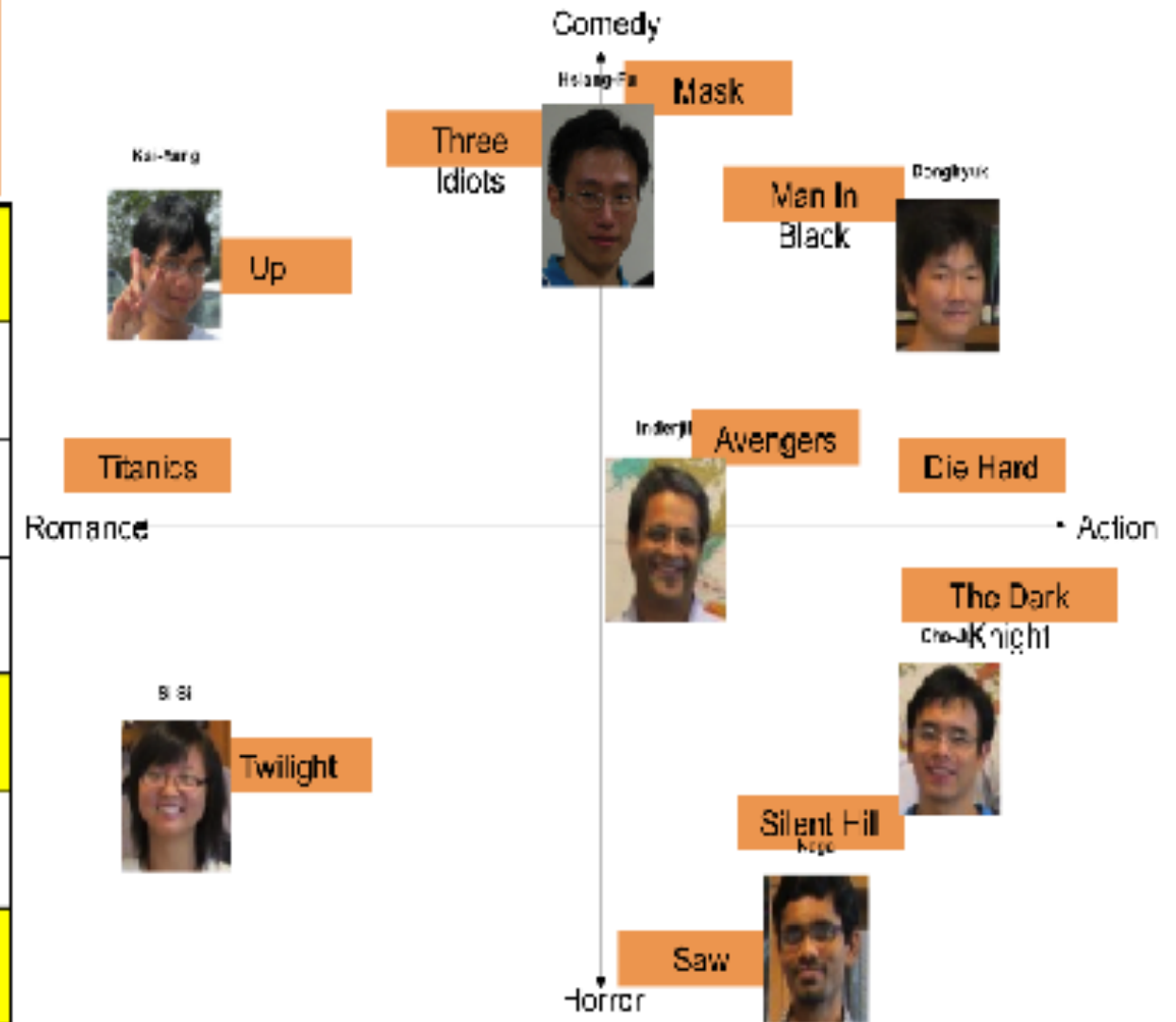
Progress Prize 2008 - RMSE = 0.8627 - Winning Team: BellKor in

13	<a href="#">xiangliang</a>	0.8642	
----	----------------------------	--------	--

## Rating Matrix

Users

	Movie 1	Movie 2	Items					Movie 10	Movie 11
	1		5		3		5		2
		2	3		5		2	5	
				3	?	5	3		
	2	5			3		4	2	
			5		5				1
		5		1				5	
	1		1				2		4



## Netflix Recommender System

- 추천 시스템(recommender system)이란 사용자(user)가 상품(item)에 대해 어떻게 평가하는지 예측하는 시스템의 일종이다. Netflix에서 제공한 netflix-prize-data의 경우 17770개의 영화에 대한 고객 2649429명 평점 정보가 저장되어있다. 이 정보를 가지고 사용자가 아직 평가하지 않은 영화에 대한 평점을 예측하는 것이 이번 프로젝트의 목표이다.
- 추후 공장데이터의 결측치를 수정하는 작업에도 활용될 수 있는 모듈 개발에 활용할 수 있다.

## 코드 리뷰

```
if __name__ == '__main__':

    # 1. 데이터 파일 읽어오기
    start_time = time.time()
    data_file = readFile("/Users/limjungmin/
Netflix_Recommender/netflix-prize-data/
combined_data_1.txt")
    run_time = time.time() - start_time
    print ( " Run time for readFile : %.4f (sec)" %
(run_time) )

    # 2. Surprise 패키지에 활용할 수 있도록 데이터 전처리
    start_time = time.time()
    custom_data_file =
makeCustomDataFile(data_file)
    run_time = time.time() - start_time
    print ( " Run time for makeCustomDataFile : %.
4f (sec)" % (run_time) )

    # 3. 사용할 알고리즘(SVD)를 통한 학습 진행
    start_time = time.time()
    predictions =
train_custom_data_file(custom_data_file, algo =
SVD())
    run_time = time.time() - start_time
    print ( " Run time for train_custom_data_file :
%.4f (sec)" % (run_time) )

    # 4. 예측 결과를 가지고 RMSE 측정값 구하기
    get_accuracy(predictions)
```

```

def readFile(path):

    data_file = open(path)
    return data_file

def makeCustomDataFile(data_file):

    # combined_data_1.txt에는 4499개의 movieID가 저장되어있음.
    custom_data_file = open("/Users/limjungmin/Netflix_Recommender/u.data", 'w')

    #cnt = 0 : 디버깅용 Count 계수
    for line in data_file:

        if ":" in line:
            movieID = line.split(":")[0]
            #print(movieID)
            #cnt+=1
        else :

            info = line.split(",")

            userID = info[0]
            rating = info[1]
            date = info[2].split('\n')[0]

            str = userID + ";" + movieID + ";" + rating + "\r\n"
            custom_data_file.write(str)

            #if cnt > 50 : break

    print("make Custom Data File Done")

    reader = surprise.Reader(line_format='user item rating', sep=';')
    data = surprise.Dataset.load_from_file('/Users/limjungmin/Netflix_Recommender/u.data', reader=reader)
    df = pd.DataFrame(data.raw_ratings, columns=["user", "item", "rate", "id"])
    del df["id"]

    print(df.head(10))

    return data

```

## 코드 리뷰

## 코드 리뷰

```
def train_custom_data_file(data, algo):  
    trainset, testset = train_test_split(data,  
test_size=.25)  
  
    algo.fit(trainset)  
  
    predictions = algo.test(testset)  
  
    return predictions  
  
def get_accuracy(predictions):  
    return accuracy.rmse(predictions)
```