빅데이터 청년 인재 일자리 연계사업 프로젝트 산출물

IMDb 영화 데이터를 활용한 영화평점예측 및 추천 시스템 구축

팀 명 : 개미지옥

프로젝트 명 : imbd영화 데이터를 활용한 영화평점예측 및 추천 시스템 구축

조원 : 한병림, 변정, 이성준, 조은솔

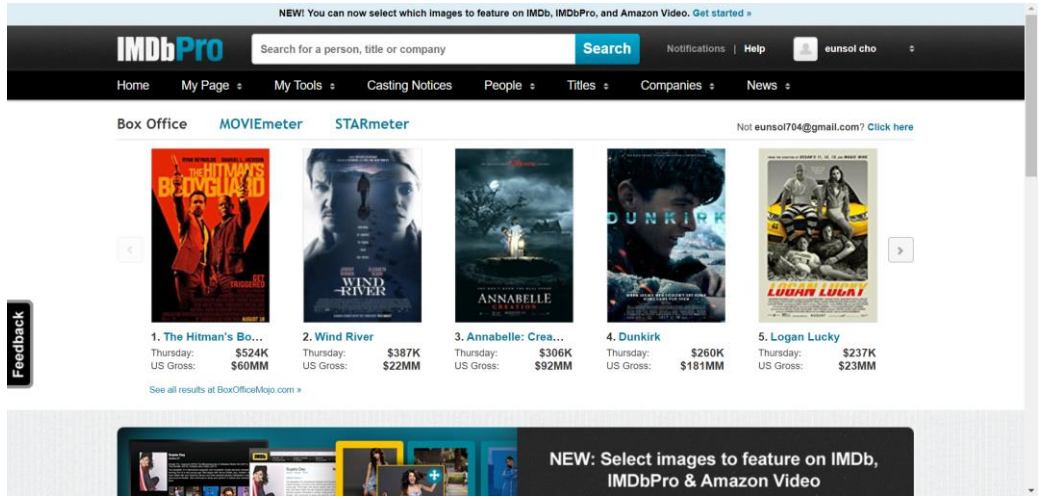
content

1 개 요

introduction

1. 개요 : IMDb 데이터 활용 _ IMDb 소개 및 선정 배경

IMDb : 인터넷 영화 데이터베이스(Internet Movie Database)



https://pro-labs.imdb.com/

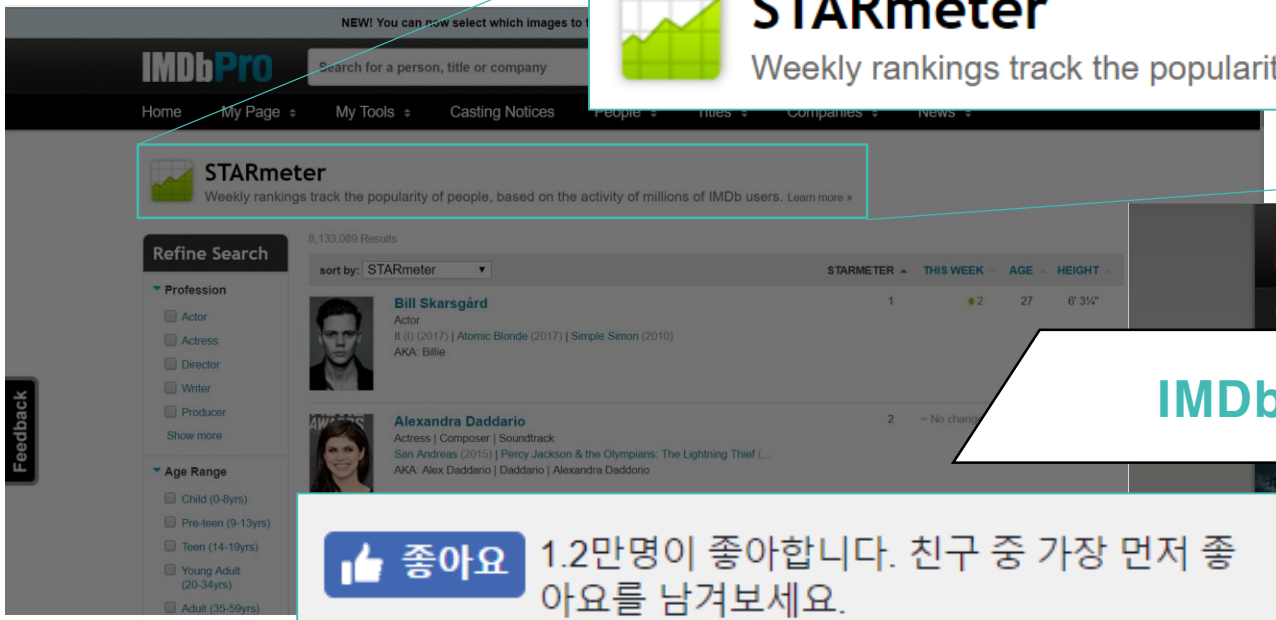
영화, 배우, 텔레비전 드라마, 비디오 게임 등에 관한 정보를 제공하는 온라인 데이터베이스

영화, 에피소드 정보 2,950,317건, 인물 정보 6,029,621건을 소유, 약 5,400만 명이 가입해 있다. (2014년 8월 1일을 기준)

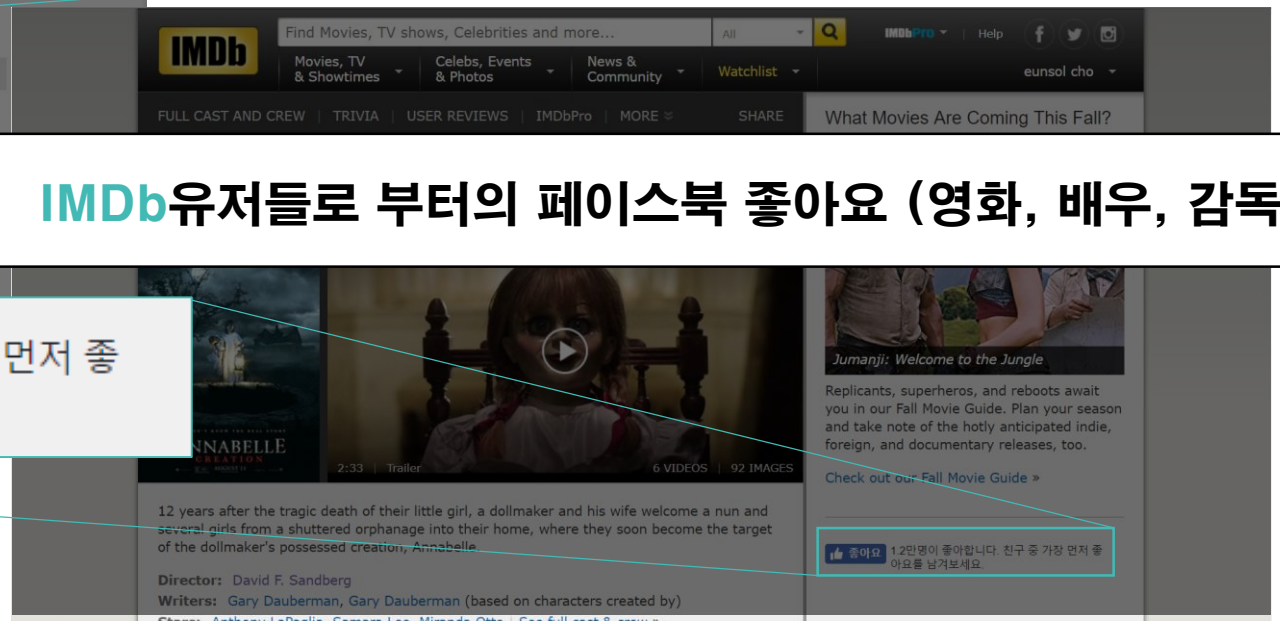
이용자가 1~10점 사이로 별점을 줄 수 있다.

다른 특징으로는 자체적으로 배우에 대한 랭킹인 STARmeter와 페이스북 좋아요 가 있다.

IMDb유저들 활동에 따라 매긴 배우들의 랭킹

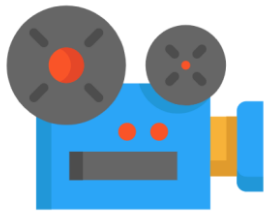


IMDb유저들로 부터의 페이스북 좋아요 (영화, 배우, 감독)



1. 개요 : 도전 과제

introduction

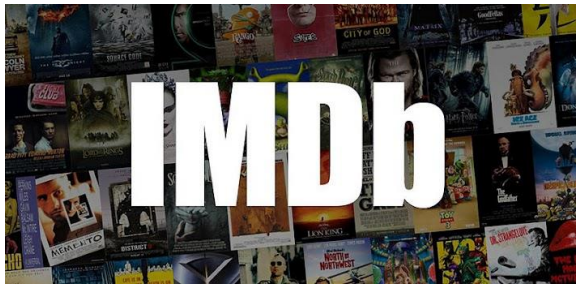
IMDb영화 데이터

영화평점예측 및 추천 시스템 구축

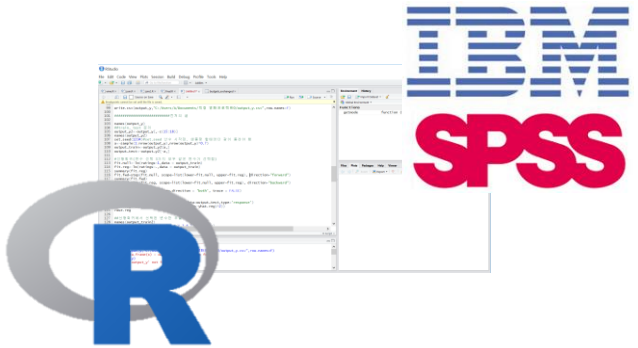
2 시스템 아키텍처
System Architecture

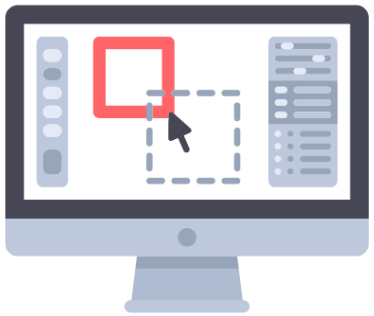
2. 시스템 아키텍처 System Architecture

1. Python을 이용 Crawling을 통해 IMDb 의 데이터 수집 및 저장



3. Oracle DB에 data 저장





4. WEB Application 구축을 통한 시각화

2. R / SPSS을 이용한 data 전처리 및 기계학습기술을 적용한 데이터 분석 처리

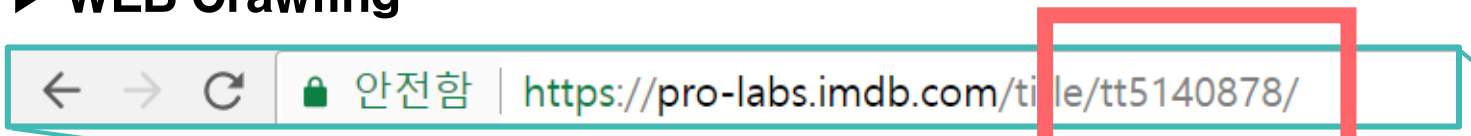
3 데이터 수집 및 저장

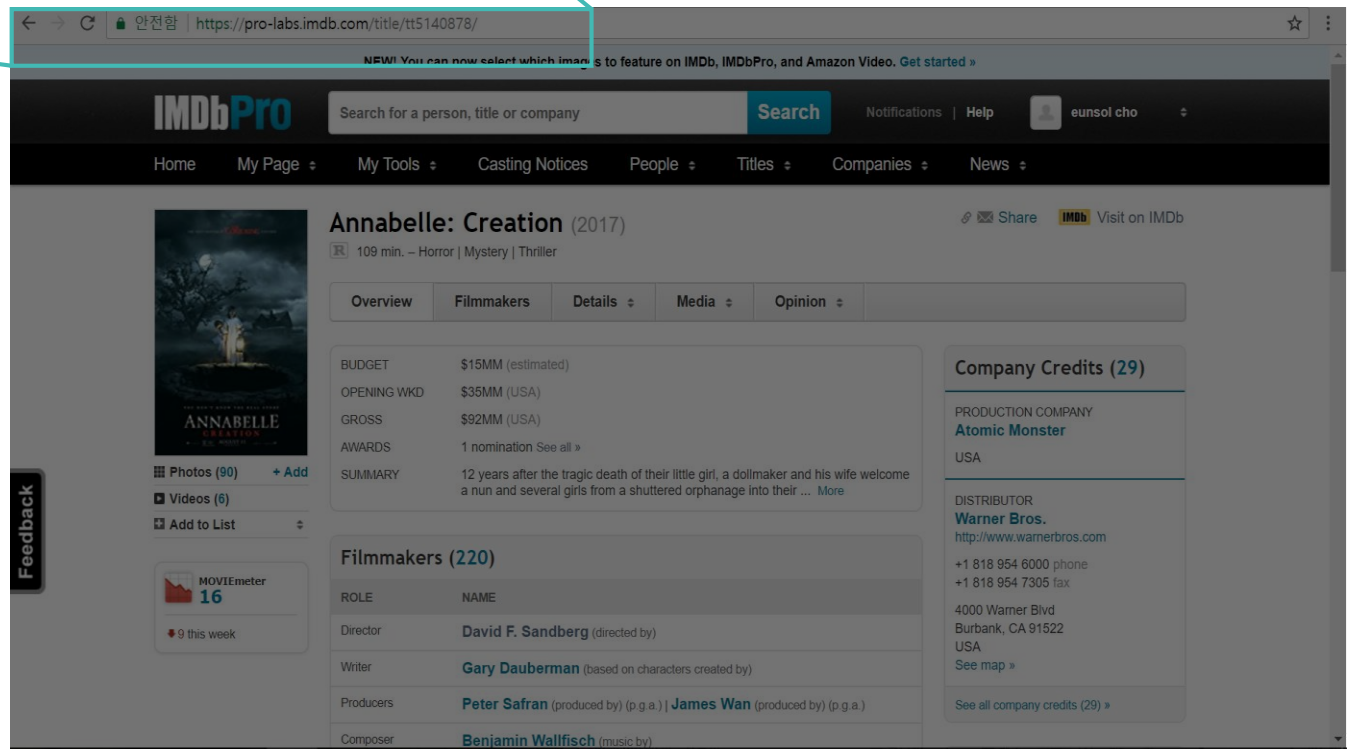
collection&storage

3. 데이터 수집 및 저장

collection&storage

▶ WEB Crawling





Python을 이용 WEB Crawling을 통해 IMDb 의 데이터 수집 및 저장 하였다.

IMDb의 컨텐츠(영화, TV시리즈 등)는 위의 URL주소 와 같이 각 고유한 시리얼 넘버를 갖는다.

시리얼 넘버 : tt + 랜덤한 6자리의 숫자

▶ WEB Crawling

다양한 IMDb컨텐츠 중 영화 시리얼 넘버 만을 추출하기 위해

시리얼 넘버 생성 규칙을 IMDb측 문의 하였으나, 정보를 얻지 못하였다.

Re: Other-Data

by - IMDb Help Desk (28 Aug 2017 02:37:12 PM)

Hi,

First of all, the IMDb content on Kaggle was stolen from us without authorization. If you use it, you're using stolen content. We would strongly urge you to not do that.

Second, the only source of IMDb content for purely academic non-commercial usage is available here with these terms and conditions: http://www.imdb.com/interfaces

---
Regards,

The IMDb Help Desk

Please let us know if this response resolved your question: Yes No What's this? reply

▶ WEB Crawling

1. 영화 컨텐츠의 시리얼 넘버 추출

```
15 for n in range(5140878, 1000000, -1):
```

최신 영화의 시리얼 넘버를 기준으로 for문을 돌린다.

```
22     if soup.h1 is None:
```

영화 컨텐츠 일 경우를 if 문을 통해 분류

2. output.csv 파일로 영화 컨텐츠 시리얼 넘버 저장

```
12 f = open('output.csv', 'w', encoding='utf-8', newline='')
13 wr = csv.writer(f)
```

```
34             wr.writerow(["tt"+str(n), mname])
35
36 f.close()
```

3. output.csv 파일을 이용 영화 컨텐츠로 부터 원하는 데이터를 추출한다.

영화 컨텐츠의 시리얼 넘버 추출 코드

```
7
8  import requests
9  from bs4 import BeautifulSoup
10 import csv
11
12 f = open('output.csv', 'w', encoding='utf-8', newline='')
13 wr = csv.writer(f)
14
15 for n in range(5140878, 1000000, -1):
16     url = 'https://www.imdb.com/title/tt'+str(n)+'/'
17
18     source_code = requests.get(url)
19     plain_text = source_code.text
20     soup = BeautifulSoup(plain_text, 'lxml')
21
22     if soup.h1 is None:
23         print("tt"+str(n))
24         print("null")
25
26     else:
27         print('tt'+str(n))
28         tp =soup.find(id='type')
29
30         if tp is None:
31             #제목
32             mname=soup.h1.span.get_text()
33             print(mname)
34             wr.writerow(["tt"+str(n), mname])
35
36 f.close()
37
```

3. 데이터 수집 및 저장

collection&storage

▶ **DATA**

no	Column name	Describe	Data type
1	serial	영화 별 고유 시리얼 번호	String
2	title	영화 제목	String
3	year	개봉 연도	Int
4	duration	상영 시간	Int
5	genres	장르	String
6	budget	예산	String
7	director_name	감독 이름	String
8	actor_n_name	출연 배우 이름(n=3, 3명)	String
9	actor_n_starmeter	출연 배우 스타미터 (n=3, 3명)	Int
10	actor_n_likes	출연 배우 페이스북 좋아요 수 (n=3, 3명)	Int
11	director_likes	감독 페이스북 좋아요 수	Int
12	movie_likes	영화 페이스북 좋아요 수	Int
13	ratingCounts	평점 남긴 유저의 수	Int
14	ratings	평점(범위 0~10)	Double

4 데 이 터 전 처 리

p r e p r o c e s s i n g

: 분석에 앞서 R을 이용 WEB Crawling을 통해 수집한 데이터의 전처리를 진행.

Rating(평점) 여부로 데이터 분리(약 15000개 -> 5000개)

결측치 : 10054 건

```
> apply(out,2,function(x) sum(is.na(x)))
          serial            title             year         duration
               0                0             2802             6038
          genres           budget    director_name     actor_1_name
             907             8414              347             4260
    actor_2_name     actor_3_name actor_1_starmeter actor_2_starmeter
            5408             6178             4261             5408
actor_3_starmeter     actor_1_likes     actor_2_likes     actor_3_likes
            6180             11869            11932            11090
   director_likes           ratings       ratingCounts       movie_likes
           10480             10054            10054             9901
     writer_likes
           11203
```

1970년대 이후 개봉 영화 : Title과 Year의 결측치를 최소화한 지점

4. 데이터 전처리

preprocessing

Budget(예산)은 모두 달러로 환산

budget
€1.2MM (estimated)
€1.3MM (estimated)
€1.4MM (estimated)
€1.5MM (estimated)
€1.6MM (estimated)
€1.6MM (estimated)
€1.7MM (estimated)
€1.8MM (estimated)

country	describe	rate
NGN	나이지리아 나이라	0.003000
ZAR	남아공 랜드	0.077280
NPR	네팔 루피	0.010000
NOK	노르웨이 크로네	0.128200
NZD	뉴질랜드 달러	0.715700
TWD	대만 신타이비	0.033200
KRW	대한민국 원	0.000892
DKK	덴마크 크로네	0.159500

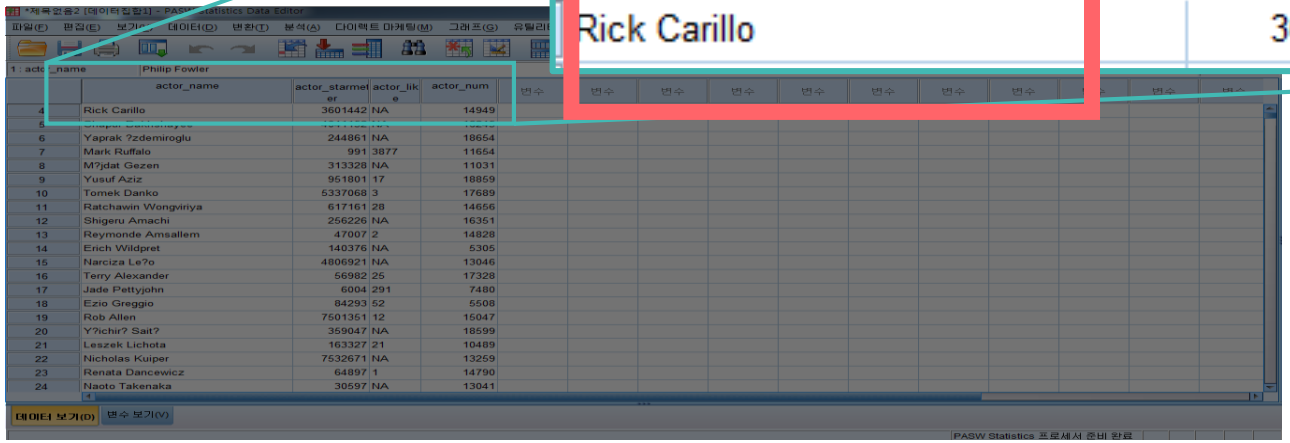
budget
150000.0
150000.0
6700000.0
150000.0
150000.0
150000.0
559400.0
150000.0

4. 데이터 전처리

preprocessing

Factor변수(감독, 장르, 배우)에 고유 번호 붙이기

actor_name	actor_starmeter	actor_like	actor_num
Rick Carillo	3601442	NA	14949



결측치 -> 중앙값으로 대체

```
##결측치 대체->중앙값
apply(output_y,2,function(x) sum(is.na(x)))
output_y$duration[is.na(output_y$duration)]<-median(output_y$duration, na.rm = T)
output_y$budget[is.na(output_y$budget)]<-median(output_y$budget, na.rm = T)
output_y$actor_1_starmeter[is.na(output_y$actor_1_starmeter)]<-median(output_y$actor_1_starmeter, na.rm = T)
output_y$actor_2_starmeter[is.na(output_y$actor_2_starmeter)]<-median(output_y$actor_2_starmeter, na.rm = T)
output_y$actor_3_starmeter[is.na(output_y$actor_3_starmeter)]<-median(output_y$actor_3_starmeter, na.rm = T)
output_y$actor_1_likes[is.na(output_y$actor_1_likes)]<-median(output_y$actor_1_likes, na.rm = T)
output_y$actor_2_likes[is.na(output_y$actor_2_likes)]<-median(output_y$actor_2_likes, na.rm = T)
output_y$actor_3_likes[is.na(output_y$actor_3_likes)]<-median(output_y$actor_3_likes, na.rm = T)
output_y$director_likes[is.na(output_y$director_likes)]<-median(output_y$director_likes, na.rm = T)
```

5 데이터 분석 및 시각화

alaysis&visualization

평점예측

: 선형회귀
회귀나무
신경망

예측 모델으로 오분류율을 사용하지 못함.
-> **RMSE(평균제곱오차)**를 사용.
: 절대적인 기준이 없어 가장 낮은 값을 가지는 모델을 최종 선택.

추천

: 코사인 유사도(Cosine Similarity)

▶ 선형회귀

CODE

```
112  #선형회귀(변수 선택 3가지 모두 같은 변수가 선택됨)
113  fit.null<-lm(ratings~1,data = output_train)
114  fit.reg<-lm(ratings~.,data = output_train)
115  summary(fit.reg)
116  fit.fwd=step(fit.null, scope=list(lower=fit.null, upper=fit.reg), direction="forward")
117  summary(fit.fwd)
118  fit.bwd=step(fit.reg, scope=list(lower=fit.null, upper=fit.reg), direction="backward")
119  summary(fit.bwd)
120  fit.step.reg = step(fit.reg,direction = 'both', trace = FALSE)
121  summary(fit.step.reg)
```

SUMMARY

선형 회귀로 부터 선택된 변수 값

```
> summary(fit.step.reg)

Call:
lm(formula = ratings ~ year + duration + actor_3_starmeter +
    actor_1_likes + ratingCounts + genre_num, data = output_train)

Residuals:
    Min      1Q  Median      3Q     Max
-5.4131 -0.7560  0.0579  0.8550  4.2324

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    -3.399e+01  7.673e+00  -4.429 9.71e-06 ***
year            1.961e-02  3.815e-03   5.140 2.89e-07 ***
duration        4.438e-03  9.448e-04   4.698 2.72e-06 ***
genre_num24    -2.949e-01  6.612e-01  -0.446 0.655563
genre_num25    -2.763e+00  1.315e+00  -2.101 0.035671 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.313 on 3912 degrees of freedom
Multiple R-squared:  0.2315,    Adjusted R-squared:  0.2264
F-statistic: 45.32 on 26 and 3912 DF,  p-value: < 2.2e-16
```

RMSE

```
> rmse.reg
[1] 1.304811
```

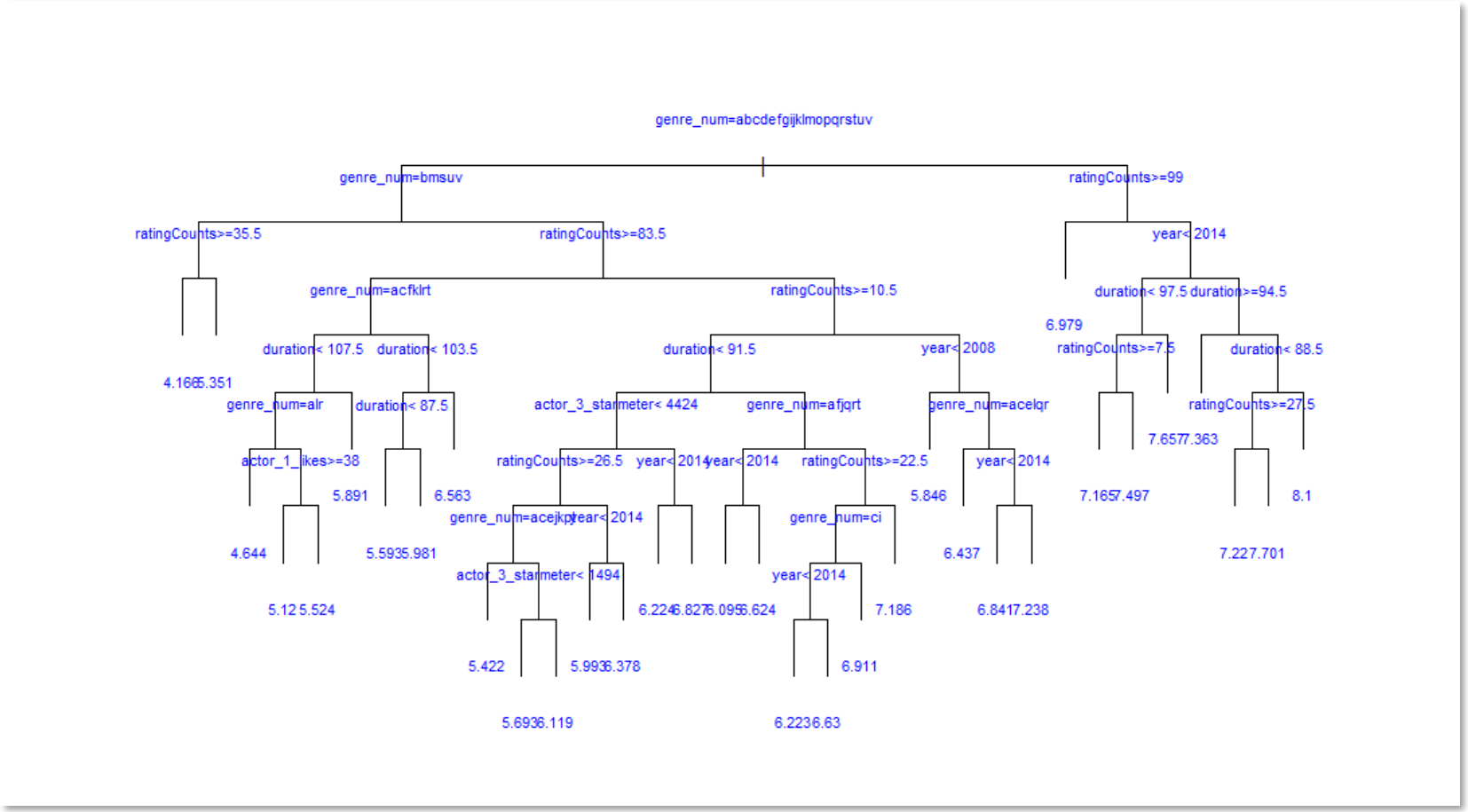
▶ 회귀나무

CODE

```
132 ##회귀나무
133 library(rpart)
134 my.control = rpart.control(xval=10, cp=0, minsplit=nrow(output_train2)*0.05)
135 fit.tree = rpart(ratings~year+duration+actor_3_starmeter+
136                   actor_1_likes+ratingCounts+genre_num,
137                 data=output_train2, method='anova',control=my.control)
138 fit.tree
139 which.min(fit.tree$cp[,4])#에러 최소값 찾
140 ii = which.min(fit.tree$cp[,4])
141
142 fit.prune.tree = prune(fit.tree,cp=fit.tree$cp[ii,1])
```

▶ 회귀나무

SUMMARY



RMSE

```
> text(fit.pru
> rmse.tree
[1] 1.269992
>
```

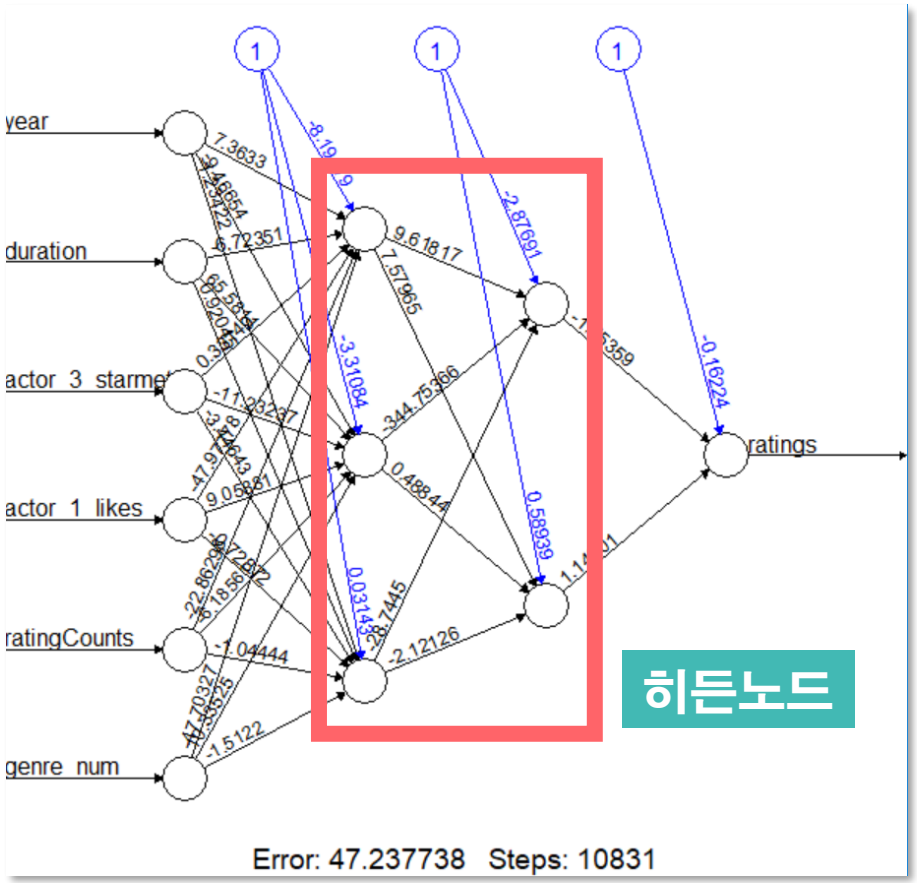
▶ 신경망

CODE

```
151 ##신 경 망
152 output_y3 <- output_y2[,c(1,2,6,7,11,12,15)]
153 for(i in 1:ncol(output_y3))if(!is.numeric(output_y3[,i]))
154   output_y3[,i]=as.numeric(output_y2[,i])
155 str(output_y3)
156 max1 = apply(output_y3,2,max)
157 min1 = apply(output_y3,2,min)
158
159 output_y3_scale = scale(output_y3,center=min1,scale=max1-min1)
160 output_y3_scale=as.data.frame(output_y3_scale)
161
162 set.seed(1234)
163 i=sample(1:nrow(output_y3_scale),round(nrow(output_y3_scale)*0.7))
164 output_y3.train = output_y3_scale[i,]
165 output_y3.test = output_y3_scale[-i,]
166
167 vname = names(output_y3.train)
168 form = as.formula(paste('ratings~',paste(vname[!vname %in% "ratings"],collapse='+')))
169 form
170 install.packages("neuralnet")
171 library(neuralnet)
172 fit.nn = neuralnet(form,data=output_y3.train,hidden=c(3,2),linear.output=T)
173 plot(fit.nn)
174
175 pred=compute(fit.nn,output_y3.test[,-5])
176 yhat.nn = pred$net.result*(max(output_y3
177                                $ratings)-min(output_y3$ratings))+min(output_y3$ratings)
178 output_y3.test$ratings=output_y3.test$
179   ratings*(max(output_y3$ratings)-min(output_y3$ratings))+min(output_y3$ratings)
```

▶ 신경망

SUMMARY



RMSE

```
> rmse.nn
[1] 1.392791898
```

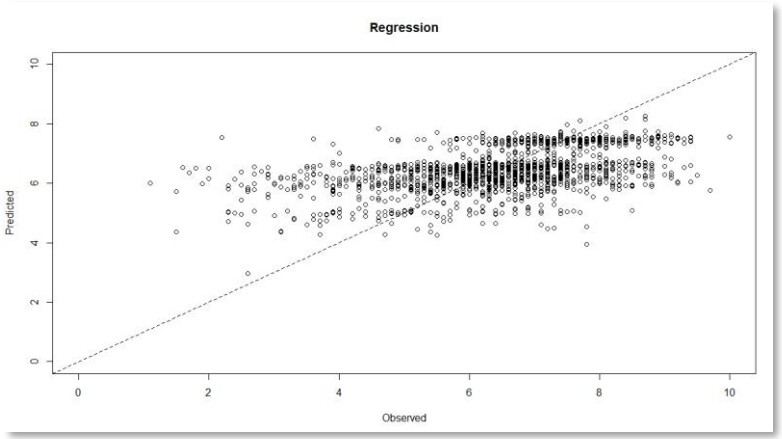
평점예측

▶선형회귀

 # RMSE

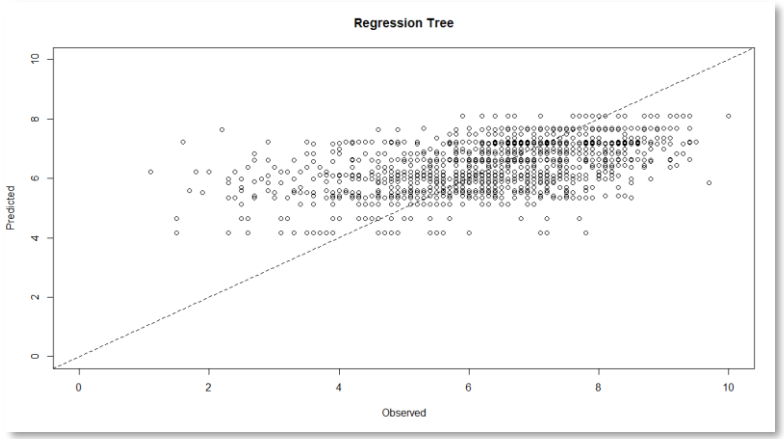
```
> rmse.reg
[1] 1.304811
```

 # 산점도



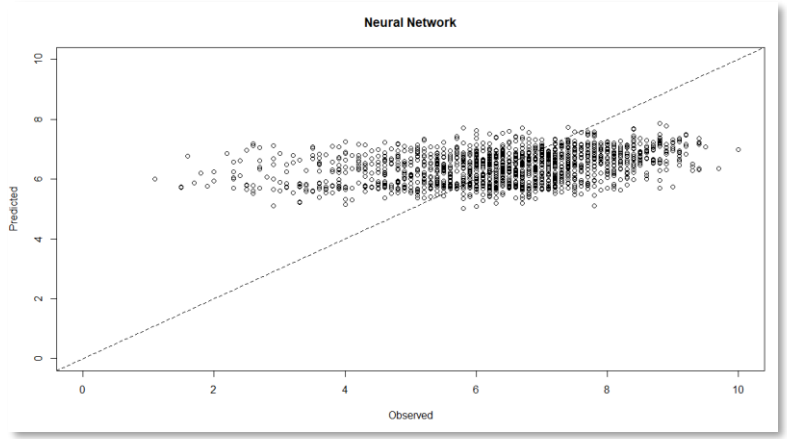
▶회귀나무

```
> text(rit.pr
> rmse.tree
[1] 1.269992
>
```



▶신경망

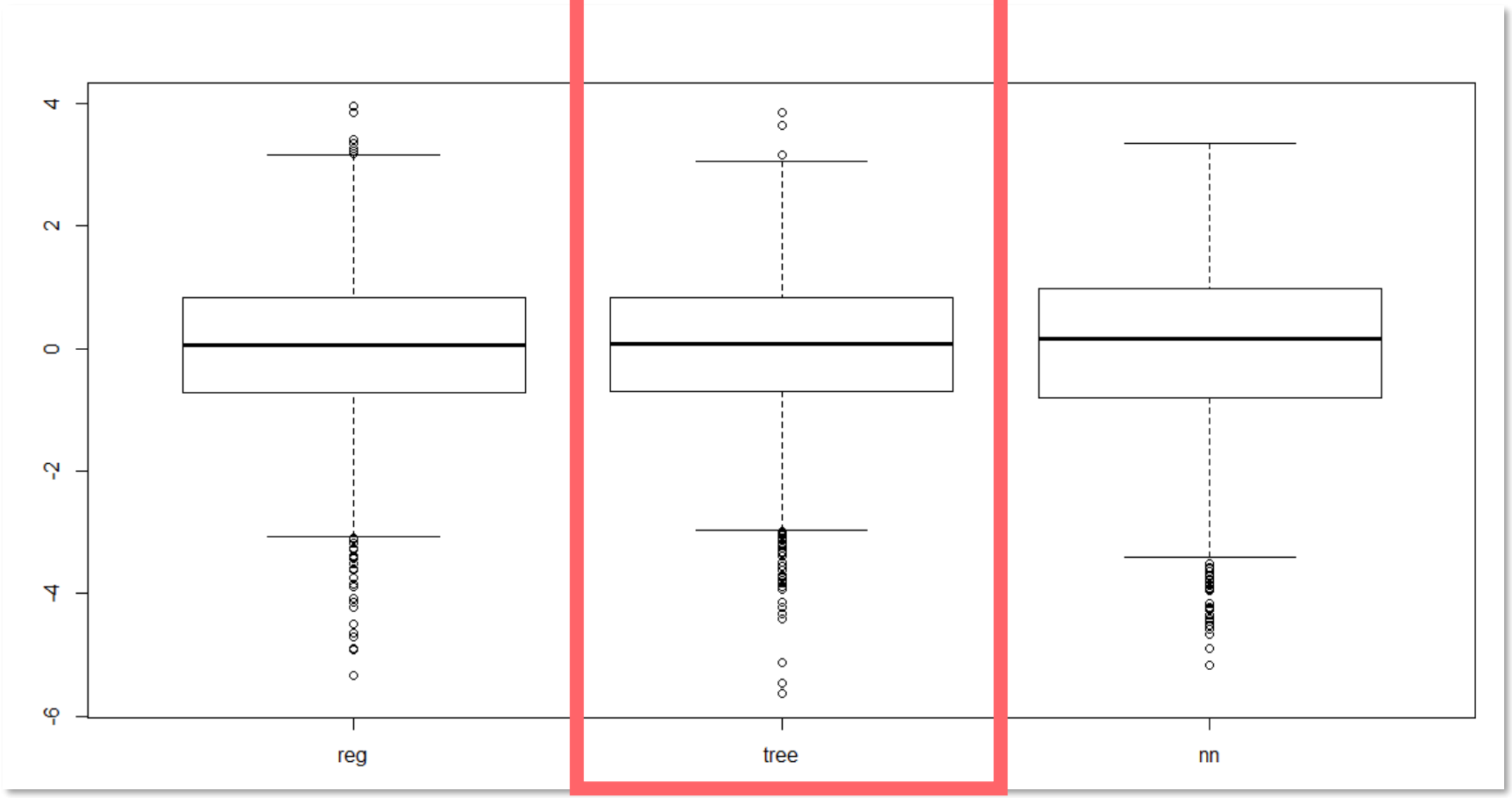
```
> rmse.nn
[1] 1.392791898
```



평점예측

▶선형회귀 ▶회귀나무 ▶신경망

오차의 분포



추천

▶코사인 유사도(Cosine Similarity)

CODE

```
12 Doc_1 <- movie[1,c(2:19)]
13 Doc_2 <- movie[2,c(2:19)]
14 Doc_3 <- movie[3,c(2:19)]
15
16 Doc_corpus <- rbind(Doc_1, Doc_2, Doc_3) # matrix
17
18 colnames(Doc_corpus) <- <- c("year",  "duration", "budget", "actor_1_starmeter",  "actor_2_starmeter", "actor_3_starmeter", "actor_1_likes",
19 ")
20
21 Doc_corpus
```

SUMMARY

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.847659	0.917802	0.804671	0.763931	0.776821	0.898557	0.840143	0.899894	0.725794	0.812231	0.918327	0.782332	0.818902	0.805165	0.817403	0.551166
4	0.410266	0.44861	0.364605	0.338626	0.346608	0.43832	0.385402	0.437746	0.397997	0.3627	0.448921	0.349918	0.393581	0.350177	0.377218	0.420452
1	0.189536	0.22567	0.180276	0.172823	0.196616	0.215739	0.21164	0.214567	0.152905	0.186469	0.225981	0.179741	0.181375	0.216205	0.191498	0
4	0.109681	0.153159	0.089059	0.079799	0.113494	0.144717	0.134047	0.140815	0.088449	0.098661	0.153668	0.090852	0.098077	0.124859	0.11141	0.12422
2	0.085268	0.14804	0.084615	0.070387	0.100317	0.133947	0.125561	0.132697	0.021754	0.095521	0.14868	0.083465	0.073267	0.113419	0.094484	0.163239
1	0.047858	0.082474	0.06208	0.049517	0.04913	0.071672	0.070193	0.073023	0.015734	0.061785	0.082854	0.052147	0.039876	0.061309	0.048041	0.207878
	0.040541	0.08518	0.04785	0.034641	0.048473	0.073623	0.070745	0.073478	0	0.051527	0.085648	0.041495	0.031841	0.061176	0.044071	0.152905
2	0.01181	6.49E-06	0.01358	0.022632	0.014029	0.000558	0.004224	0.000561	0.086169	0.009721	4.18E-06	0.015557	0.015182	0.010531	0.009955	0.227012
4	0.011787	0.000729	0.012497	0.020965	0.012275	0.000824	0.003979	0.000858	0.081177	0.008462	0.000739	0.01402	0.013645	0.010349	0.00904	0.209239
1	0.011625	8.77E-07	0.013428	0.022451	0.013996	0.000521	0.004236	0.000516	0.085648	0.009629	0	0.015424	0.014986	0.010565	0.009855	0.225981
	0.011452	0	0.013249	0.02222	0.013855	0.000489	0.004184	0.000478	0.08518	0.009497	8.77E-07	0.015238	0.014773	0.010473	0.009716	0.22567
3	0.008748	0.010033	0.00327	0.006869	0.005494	0.008012	0.004717	0.007403	0.061484	0.001199	0.01016	0.00397	0.007011	0.004505	0.004651	0.203185
2	0.008374	0.00035	0.009945	0.017487	0.010428	5.46E-05	0.002811	5.17E-05	0.075595	0.006582	0.000378	0.011372	0.010876	0.00802	0.006579	0.214261
1	0.008347	0.010473	0.006674	0.007919	0.001628	0.007728	0.002167	0.00809	0.061176	0.003446	0.010565	0.004484	0.008494	0	0.002611	0.216205
4	0.008347	0.010473	0.006674	0.007919	0.001628	0.007728	0.002167	0.00809	0.061176	0.003446	0.010565	0.004484	0.008494	4.89E-09	0.002611	0.216204
3	0.008156	0.000352	0.0102	0.017862	0.010697	5.18E-05	0.002964	6.80E-05	0.075432	0.007112	0.000384	0.011733	0.010769	0.008177	0.006801	0.219854

컬럼별 유사도

6 한계점 및 활용방안

Limitation

6. 한계점 및 활용방안

Limitation

한계점 : 예측도가 떨어진 원인

결측치가 많았다.

: 시리얼 번호의 규칙을 몰라, 크롤링 시 정보가 거의 없는 데이터를 다수 가져옴.

Factor변수(감독, 배우)의 요소가 너무 다양했다.

: 5000여개 데이터에서 감독 3700여명 배우 9600여명 등장.

일반적인 변수만으로는 예측이 어렵다.

: 작품성이라는 변수가 필요하나, 작품성은 추상적인 지표라 변수화 하기 어려움.

Ex. 리얼 vs 워낭소리

활용방안

1. 개봉 전 평점 예측을 통한 흥행 예측 가능

2. 영화 추천을 통한 영화 산업 부흥

Q & A