<빅데이터 발표회 최종보고서>

2015150040 통계학과 배현주 2015150283 통계학과 김응채

2015150020 통계학과 김순영 2015150058 통계학과 김어진

1. 서론

1) 연구 배경

영화 산업이 발달하면서 치열한 경쟁에 돌입하게 되었습니다. 영화제작자의 입장에서 흥행 요소를 미리 예측할 수 있다면 더욱 효과적인 접근을 통해 큰 성공을 기대할 수 있을 것입니다.

따라서 영화 흥행요소에 영향을 미치는 변수를 찾는 것을 목표로 정했습니다. 이를 위해 수집한 데이터는 2016년까지의 영화들의 특성에 관한 정보가 담긴 데이터입니다. 이 자료의 특징은 세계적으로 가장 유명한 영화 데이터베이스인 iMDB (Internet Movie Database)를 기준으로, 수많은 실제 사용자의 평가를 기반으로 하여 신뢰도가 높다는 점입니다

수집된 데이터를 토대로 전공 시간에 배웠던 다양한 분석 기법들을 수집된 데이터에 적용 해보았습니다. 구체적으로 logistic regression, KNN, LDA, QDA, decision tree 모델들의 옵션을 설정하고 다양한 평가 메트릭으로 평가하는 과정을 거쳤습니다.

2) 변수 설명 및 데이터 전처리

데이터의 변수들은 구체적으로 가장 오래된 영화인 <Intolerance: Love's Struggle Throughout the Ages> (D.W. Griffith, 1916)부터 가장 최신인 <Captain America: Civil War> (Anthony Russo, 2016)에 이르기까지 66개국 5043개의 영화, 그리고 총 27개의 변수로 이루어져 있습니다. 분석을 위해 target variable 설정단계에서 gross를 duration으로 나누어 상영기간을 고려한 매출로 설정하였습니다. 영화 수익 자체는 영화 상영기간과 상관성이 높다고 생각해 영화 수익률, 즉 영화 상영기간의 효과를 control할 수 있는 파생변수를 생성했습니다. Explanatory variable로는 "num_critic_for_reviews(평론가의 비평 수)", "director_facebook_likes(감독의 페이스북 좋아요 수)", "cast_total_facebook_likes(출연배우들의 총 페이스북 좋아요 수)", "facenumber_in_poster(포스터에 있는 얼굴 개수)", "num_user_for_reviews(영화 리뷰 개수)", "budget(예산)", "imdb_score", "movie_facebook_likes(영화 페이스북 좋아요 수)" 들을 사용하였습니다.

데이터 전처리 단계에서는 결측치 처리에 집중하였습니다. 주요 변수인 gross 와 budget에 결측치가 많이 존재해 이를 단순 삭제하는 것은 데이터의 손실이 크다고 생각하여 "mice" 패키지를 사용하여 결측값을 대체했습니다. "mice"패키지를 사용한 결측치 처리는 multiple imputation의 일종으로 시뮬레이션을 반복하여 누락된 데이터를 채워 넣는 방법으로 이루어집니다. 이러한 과정을 적용한 데이터를 바탕으로 logistic regression, KNN, LDA, QDA, decision tree를 적용했습니다.

2. 본론

1) Logistic regression

로지스틱 회귀를 적용하기 이전에, 모델에 필요한 변수들을 선택하기 위해서 stepwise function을 이용했습니다. 그 결과, 평론가 수, 출연배우들의 총 페이스북 좋아요 수, 영화 리뷰 개수, Imdb 점수가 유의미한 변수였습니다. 모델 해석 결과, 다른 변수들을 고정시킨 채, 평론가의 비평 수를 한 단위 증가하거나/ 영화 리뷰 작성 수를 한 단위 증가하거나/ iMDB 점수가 한 단위 증가할 때마다 영화가 흥행할 odds ratio는 각각 1.005/ 1.001/ 0.81배 증가합니다. 나아가, 이를 바탕을 로지스틱 회귀를 적합한 결과, 예측력은 0.83로 "good fit"을 보여줍니다.

2) KNN (K-Nearest Neighbors)

```
> for (i in 1:10){
+   fit.knn = knn(train=movie.train[,-9],test=movie.test[,-9],cl=movie.train[,9], k=i, prob=T)
+   yhat = fit.knn
+   ctable = table(movie.test[,9], yhat, dnn=c("Actual", "Predicted"))
+   miss.err = 1-sum(diag(ctable))/sum(ctable)
+   pred.acc = 1 - miss.err; round(pred.acc, 3)
+   print(pred.acc)
+ }
[1] 0.7190161
[1] 0.7114475
[1] 0.7341533
[1] 0.730369
[1] 0.7473983
[1] 0.7398297
[1] 0.7492904
[1] 0.7511826
[1] 0.7606433
[1] 0.7587512
```

[그림 2.1] loop 구문을 이용해 최적의 k값 구한 결과

KNN을 적합시키기 이전, numberic 변수에 대해 scaling을 진행해 각 변수들간의 단위를 통일시키고 overfitting을 방지하고자 했습니다. 또한 KNN에서 가장 적합한 k의 값을 찾기 위해 for loop 구문을 이용해 prediction accuracy를 계산했습니다. 그 결과, k=9일 때 prediction accuracy가 0.7604로 예측력이 가장 높았습니다.

3) Linear Discriminant Analysis & Quadratic Discriminant Analysis

-LDA와 QDA란??

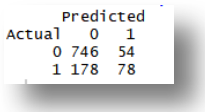
LDA는 Linear Discriminant Analysis로 집단에 대한 정보로부터 구별할수 있는 선형 함수를 만들고, 새로운 개체에 대해 어느 집단에 속하는지 분류하는 방법 입니다. LDA는 집단이 정규분포를 따르

며 등분산이라고 가정합니다. 판별함수를 만들 때 적은 수의 모수를 예측할수록 , 즉 함수의 모형이 간단할수록 정확한 예측이라고 말할 수 있습니다. 간단히 얘기하면, 하나의 공간의 선을 그어서 공간 속에서 개체들을 분류하는 방법입니다.

QDA는 Quadratic Discriminant Analysis로 LDA와 마찬가지로 판별함수를 만들어 새로운 개체가 어느 집단에 속하는 지를 예측합니다. LDA와 마찬가지로 집단이 정규분포를 가정하지만, LDA와 달리 판별함수가 비선형 함수이고 LDA는 등분산 가정이 필요하지 않다는 장점이 있습니다. 따라서 따로 등분산 test를 거치지 않아도 되지만, 비선형 함수이기 때문에 많은 모수가 필요한 예측 방법입니다.

저희는 다른 분석 기법들과 마찬가지로 LDA와 QDA를 적용하기 위해 전체 데이터를 7:3으로 분할하여 Train-Test data를 만들고, Train Data로 각 기법의 판별함수를 만들어 Test Data에 적용하였습니다.

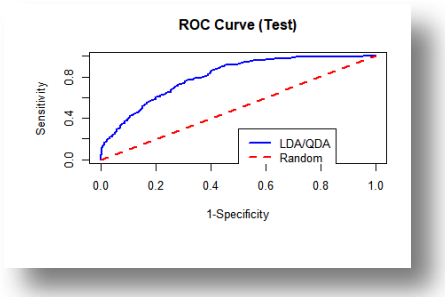
-LDA 적용



Actual	Predicted 0	1
0	746	54
1	178	78

[그림 2.2] Train Data로 LDA분석 후 Test Data에 적용한 분류표

[그림 2.1]을 기준으로 정확도, Seneitivity,Specificity를 계산한 결과, 정확도는 0.78, Sensitivity 0.3, Specificity 0.93로 나타났습니다. 이를 바탕으로 ROC 커브를 그려 얻은 AUC 값은 0.8로 [그림 2.2]를 통해 볼 수 있었습니다.



[그림 2.3] Train Data로 LDA분석 후 그린 ROC커브

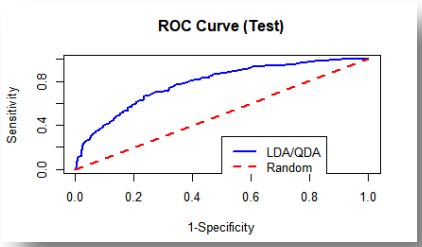
마지막으로 Test error를 계산한 결과 K=10일 때 0.20687, K=5일 때 0.20689로 미세한 차이로 K=10일 때의 test error가 가장 낮았습니다.

-QDA 적용

```
          Predicted
Actual    0   1
     0  704  96
     1  146 110
```

[그림 2.4] Train Data로 QDA분석 후 Test Data에 적용한 분류표

[그림 2.3]을 기준으로 정확도, Seneitivity,Specificity를 계산한 결과, 정확도는 0.77, Sensitivity 0.43, Specificity 0.88로 나타났습니다. 이를 바탕으로 ROC 커브를 그려 얻은 AUC 값은 0.779로 [그림 2.4]를 통해 볼 수 있었습니다.



[그림 2.5] Train Data로 LDA분석 후 그린 ROC커브

마지막으로 Test error를 계산한 결과 K=10일 때 0.21645, K=5일 때 0.21883로 미세한 차이지만 LDA보다는 다소 큰 차이로 K=10일 때의 test error가 가장 낮았습니다.

-LDA QDA 비교 및 정리

[표 2.1] LDA/QDA 비교 표

LDA		QDA
0.78	ACCURACY	0.77
0.80	AUC	0.78
0.207	5-fold-cv test error	0.212
0.207	10-fold-cv test error	0.206

[표 2.1]결과 LDA의 예측 정확도, AUC가 더 높고 test error 또한 더 낮다는 것 을 볼 수 있었습니

다. 따라서 LDA 분석기법이 새로운 영화 데이터를 대입 했을 경우 영화 흥행을 전반적으로 더 잘 예측한다는 것을 알 수 있습니다.

4) Decision Tree

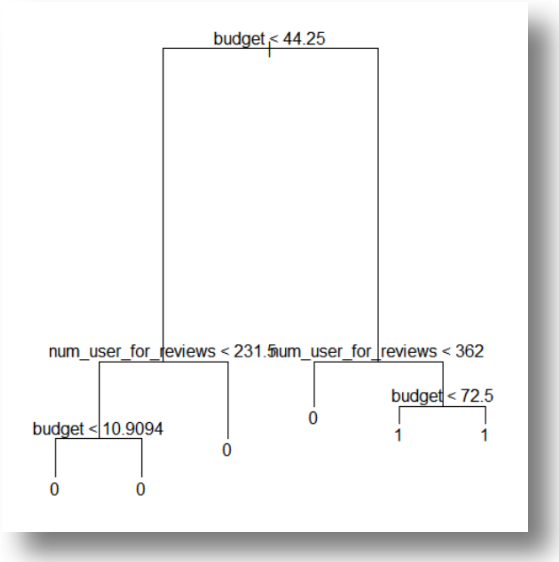
-Decision Tree 설명

Decision Tree란 의사결정 나무로 데이터의 특징에 대한 질문을 하면서 응답에 따라 데이터를 분류해 나가는 알고리즘입니다. 의사 결정 나무는 다른 분석기법들과 달리 특별한 가정이 필요하지 않고 분석과정이 직관적이라 이해하기가 쉽다는 장점이 있습니다. 또한 수치형, 범주형 변수 모두 사용가능하며 계산비용이 낮아 대규모의 데이터 셋에서도 빠르게 연산이 가능합니다.

의사 결정 나무에는 CHAID,tree,rpart,party라는 대표적인 4가지 패키지가 있는데, 저희는 범주형 변수만을 다루는CAHID를 제외한 나머지 세가지 패키지를 저희 데이터에 적용하여 분석했습니다.

저희는 다른 분석 기법들과 마찬가지로 의사결정나무를 적용하기 위해 전체 데이터를 7:3으로 분할하여 Train-Test data를 만들고, Train Data로 각 기법의 의사결정나무를 만들어 Test Data에 적용하였습니다.

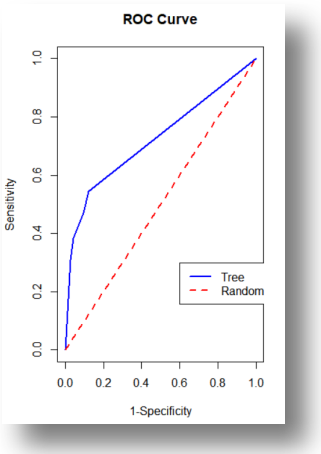
-Tree



[그림 2.6] tree를 적용한 의사결정나무

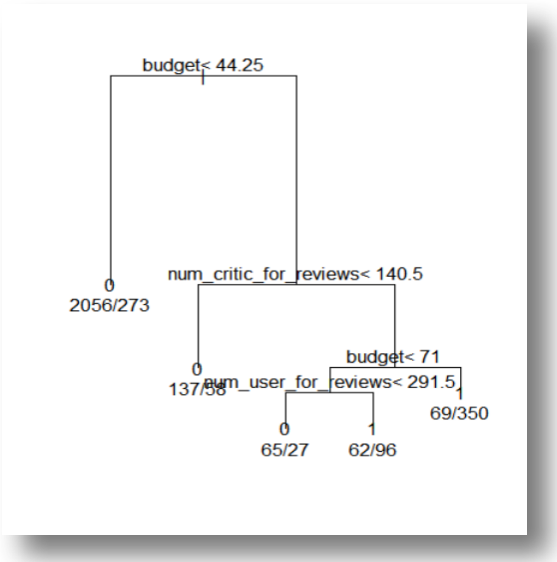
Tree를 적용하여 의사결정나무를 그린 결과 [그림 2.5]와 같이 나타났습니다. 최상위 노드에 있는 budget는 결국 영화흥행을 분류하는데 가장 중요한 변수라는 것을 알 수 있습니다. 좀더 아래 계층에 있는 노드를 살펴보면 num_user_for_reviews, 제작이후의 후기의 수라는 것을 알 수 있

었습니다. 만들어진 의사 결정 나무에 대한 예측정확도, AUC, k=5일때의 cv test eroor, k=10일일 때 cv test error는 각각 0.8111,0.869,0.1765,0.1777이었습니다.



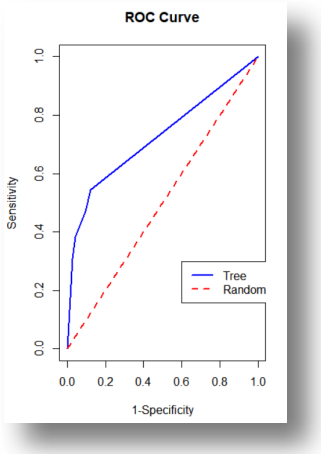
[그림 2.7] tree를 적용한 ROC커브

-rpart



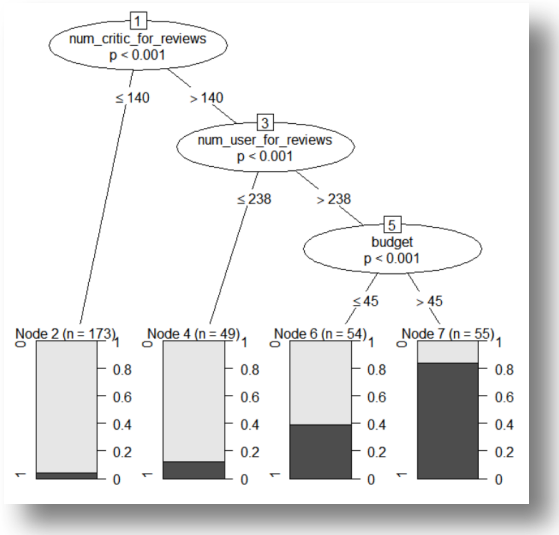
[그림 2.8] rpart를 적용한 의사결정나무

[그림 2.7]을 살펴보면, rpart를 적용한 결과 tree와 마찬가지로 예산과 평론가 후기의 개수가 상위 노드에 포함되어 있었습니다. tree와의 차이점은 예산을 하위노드에 다시 포함시켜서 예산이 71이 넘으면 가장 많이 분류된다는 것을 알 수 있었습니다. 만들어진 의사 결정 나무에 대한 예측정확도, AUC, k=5일때의 cv test eroor, k=10일일 때 cv test error는 각각 0.840,0.723,0.174,0.158이었습니다.



[그림 2.9] rpart를 적용한 ROC커브

-party



[그림 2.10] party를 적용한 의사결정나무

[그림 2.9]를 살펴보면, party를 적용한 결과 tree,rpart와 마찬가지로 예산과 평론가 후기의 개수가 상위 노드에 포함되어 있었습니다. 다른 의사 결정나무와의 차이점은 예산보다 평론가 후기 개수를 상위 노드에 포함시켜 분류되었음을 알 수 있었습니다. 만들어진 의사 결정 나무에 대한 예측정확도, AUC, k=5일때의 cv test eroor, k=10일일 때 cv test error는 각각 0.849,0.885,0.182,0.181이었습니다.

-Decision Tree 패키지 비교 및 정리

[표 2.2] Decision Tress 패키 비교 표

	Tree	Rpart	Party
ACCURACY	0.8111	0.840	0.849
AUC	0.869	0.723	0.885
5-fold-cv test error	0.1765	0.174	0.182
10-fold-cv test error	0.1777	0.158	0.181

[표 2.2]의 결과 party의 예측정확도와 AUC값은 가장 높은 값을 기록했고 가장 낮은 cv test error를 보였습니다. 셋다 AUC가 0.7이상으로 모두 높은 예측력을 갖고 있지만 party 패키지가 가장 유의미한 예측력을 지닌다고 할 수 있습니다.

3. 결론

다양한 기법들을 통해 저희는 두가지 결론을 내릴 수 있었습니다. 첫째로 영화 흥행을 잘 예측하는 기법은 party 패키지를 사용한 의사결정나무입니다. 로지스틱, KNN, LDA, QDA, 의사결정나무들 모두 Accuracy 값이 0.7을 넘기기 때문에 예측력이 대체적으로 좋은 편이었습니다. 그 중에서도 특히 party 의사결정나무 모델은 Accuracy의 값이 0.849로 가장 높고, 10-fold-cv-test error의 값이 0.181 값으로 꽤 낮은 값을 보였습니다. 따라서 이후 새로운 영화에 대한 정보들이 주어졌을 때, 흥행 여부를 예측하기 위해 party 의사결정나무를 사용하는 것을 추천할 수 있습니다.

두번째로는 영화 흥행에 영향을 끼치는 주요 변수들을 찾아낼 수 있었습니다. 로지스틱 회귀분석에서 계수가 유의미한 값을 보여주는 변수들은 review 개수와 imdb_score였습니다. 이들은 모두 사용자에 대한 평가항목의 일부라는 공통점을 가집니다. 또한 세가지 의사 결정나무에서 공통적으로 중요한 노드로써 역할을 한 변수들은 budget, num_critic_for_reviews, num_user_for_reviews였습니다.

로지스틱과 의사결정나무 결과를 바탕으로 영화 흥행에 관해 크게 영향을 미치는 것은

사용자를 비롯한 비평가들의 review와 budget이라는 것을 알 수 있었습니다. 이를 토대로 영화제작사에 영화 흥행을 위해 제작 단계에서는 투자를 탄탄히 받고, 개봉 이후에는 전문가 혹은 일반 관람객 review를 활성화하는 방법을 모색하도록 제언할 수 있습니다.