데이터 마이닝 보고서

4조

김종휘
박성진
지성인
안수이

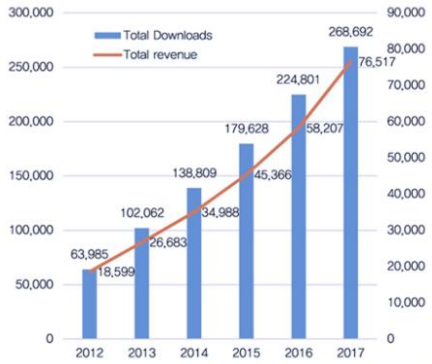
KU KONKUK UNIVERSITY

목차

Ⅰ. 주제 선정

(1) 주제 선정 배경

끊임없이 변화하는 모바일 환경은 탐색하기 어려운 공간이다. 특히, 데스크 탑보다 모바일의 사용 비율이 끊임없이 증가하고 있는데, Android는 스마트 폰 시장의 53.2%를 차지하고, iOS는 43%를 차지하고 있다. 우리는 모바일 어플리케이션의 지속 가능한 성장과 그 규모를 보아 시장 규모, 경쟁 현황분석 및 전략 수립을 통한 체계적인 시장분석이 필요하다고 생각하였다. 또한, 모바일 어플리케이션 분석은 미래 사용자의 성장과 유지(이탈 방지)를 이끌어 내기 위해 기존의 전략을 이해할 수 있는 좋은 방법이다. 따라서 모바일 어플리케이션 사용에 영향을 주는 변수를 선택하여 사용자들의 어플리케이션의 평점을 예측할 수 있다면 어플리케이션 관련 현상을 보다 정확하게 예측하고, 효율적으로 활용할 수 있다는 사고를 바탕으로 본 프로젝트를 시작하였다.



[그림-1] 모바일 앱스토어 다운로드 현황

Ⅱ. 데이터 수집 및 분석

1. 데이터 수집

(1) 데이터 수집

2017년 7월동안, Apple Inc 웹 사이트의 iTunes Search API에서 추출된 데이터를 사용하였다. 본 데이터 set에는 약 7,000개 이상의 Apple iOS 모바일 어플리케이션에 관한 세부정보가 들어있었고 총 16개의 변수가 존재하였다. 본 데이터 set에 관한 자세한 설명은 분량 관계상 Kaggle[1]을 통해서 보기로 한다.

(2) 파생변수 생성

16개의 변수들을 통해, 어플 평점에 영향을 줄 수 있는 부가적인 데이터를 생성하였다. 현재 버전의 총 평점 수를 나타내는 "Rating_count_ver", 현재 버전의 평균 평점을 나타내는 "User_rating_ver"의 변수를 활용하여, 현재 버전의 총 평점을 나타내는 "Sum_rate_ver" 이라는 파생 변수를 생성하였다.

2. 분석

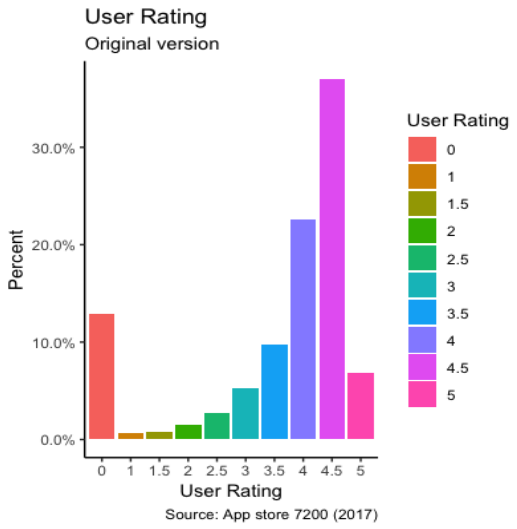
(1) 데이터 전처리 및 탐색적 자료 분석

- User_rating (타겟 변수)

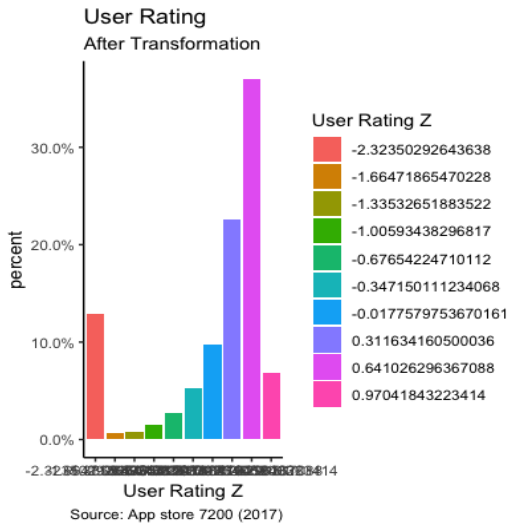
: Z 변환을 사용하여 User_rating이라는 변수를 연속형 변수로 활용하기로 하였다. 또한, 타겟 변수(User_rating)에 대해 Z 변환 이후 4 분위수를 활용하여 1분위수와 3분위수를 기준으로 "low"," middle"," high"의 3 가지 범주형 변수로 나타내었다. 이러한 과정을 통해 회귀분석과 범주형 자료 분석이 가능하게끔 만들었다. 다시 말해, 'User_rating' 이라는 타겟 변수에서 'User_rating_scaled' 라는 새로운 타겟 변수를 만들어 회귀분석이 가능하게 하였고, 'User_rating_scaled' 에서 다시 'User_rating_cls' 라는 새로운 타겟 변수를 만들어 범주형 자료 분석이 가능하게 만들었다.

---

[1] https://www.kaggle.com/ramamet4/app-store-apple-data-set-10k-apps



[그림-2] User_rating 분포



[그림-3] Z 변환 이후 User_rating 분포

- ID

: ID는 어플의 일련번호에 대한 변수이므로 결과에 본 프로젝트 분석에서 제외하였다.

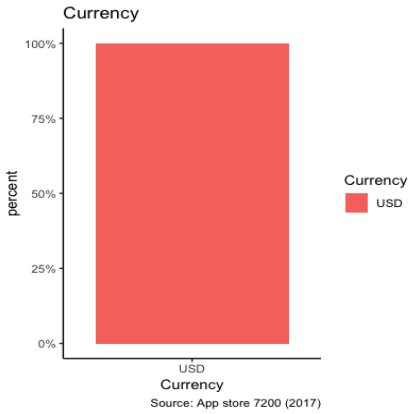
- App name

: App name은 어플의 이름에 대한 변수이므로 본 프로젝트 분석에서 제외하였다.

- Size_bytes

: Size_bytes는 어플의 용량을 나타내는 변수이다. Byte 단위를 Megabyte로 스케일링 하였고, 최소 용량은 0.5625 Megabyte , 최대 용량은 3839.464 Megabyte이었다.

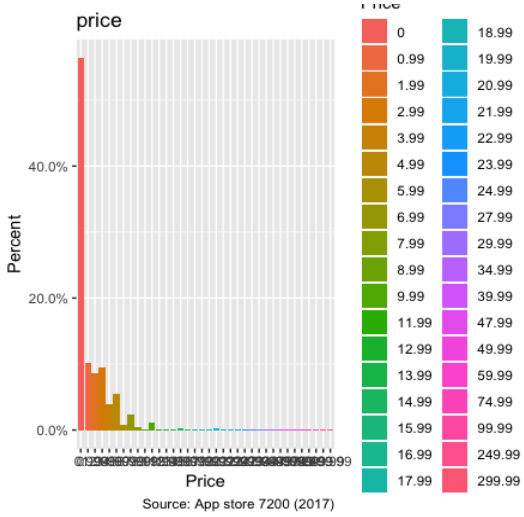
- Currency



[그림-4] Currency 분포

: Currency는 어플 사용에 대해 화폐 사용을 나타내는 변수이다. 모든 어플에 대해 USD라는 동일한 화폐를 사용했으므로 본 프로젝트 분석에서 제외하였다.

- Price

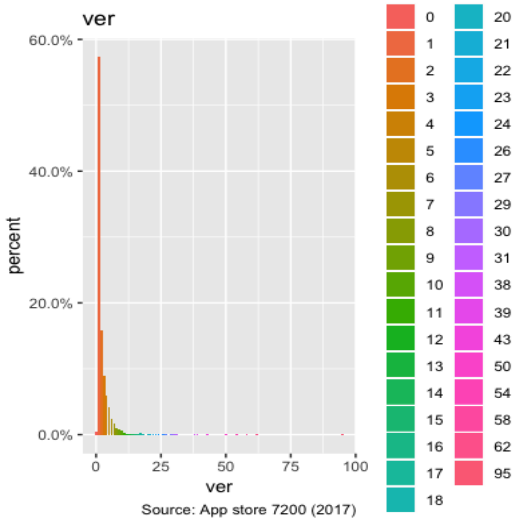


[그림-5] Price 분포

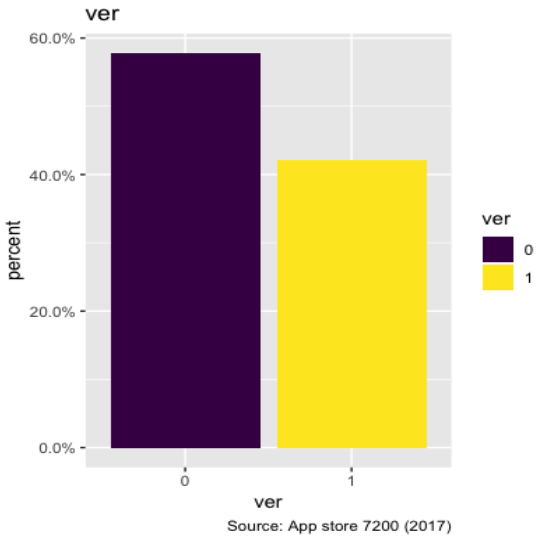
: 매우 치우친 오른쪽 꼬리 형태를 보였다.

- Ver

: 어플의 버전을 나타내는 변수이다.



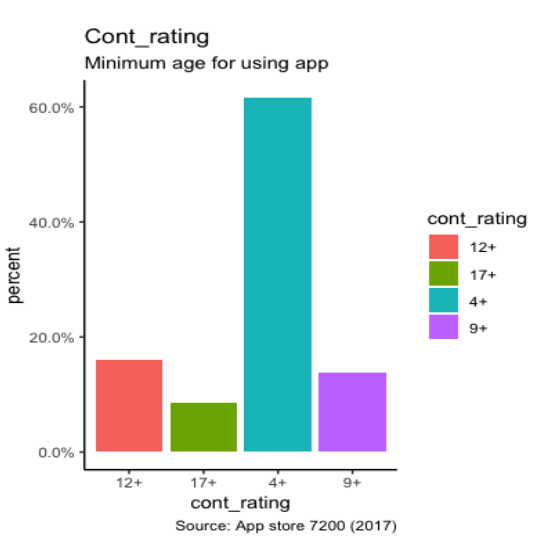
[그림-6] Ver 분포



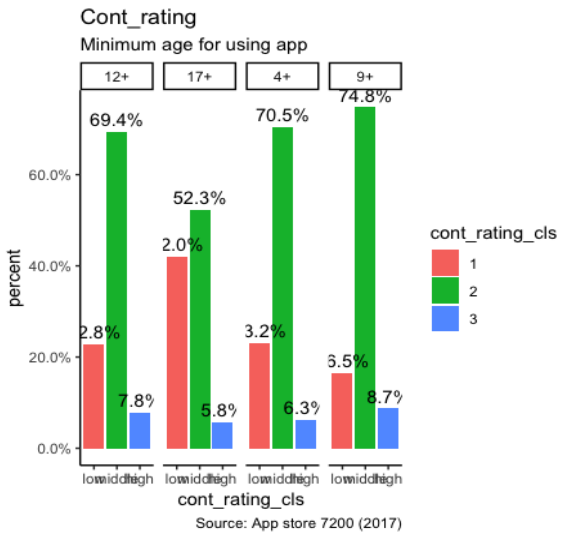
[그림-7] Ver 구분

- Cont_rating

: 어플 사용 가능 연령을 나타내는 변수이다.

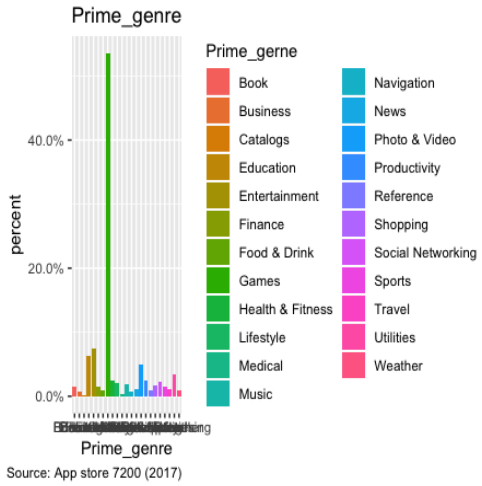


[그림-8] Cont_rating 분포

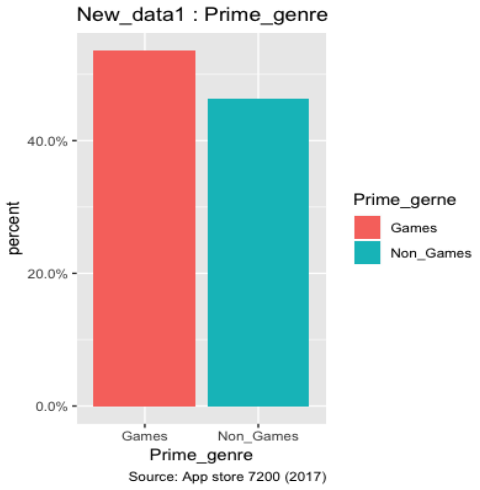


[그림-9] Cont_rating_cls 분포

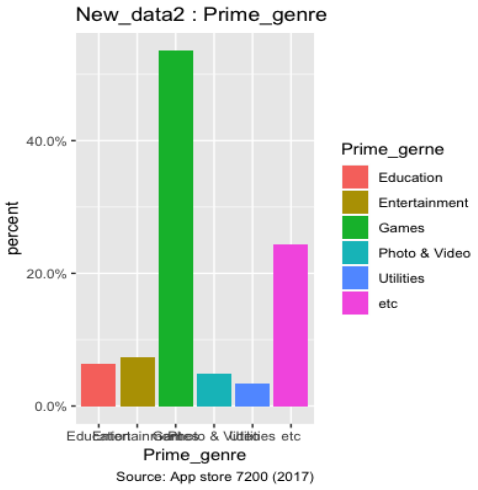
- Prime_genre



[그림-10] Prime_genre 분포



[그림-11] Games vs Non_Games 분포



[그림-12] 상위 5개의 카테고리+etc

- Sup.device.num

: 지원가능 디바이스 수를 나타내는 변수이다. 최소 9, 최대 47이라는 수치를 보였다.

- ipadSc_num

: 앱 설명 스크린샷 수를 나타내는 변수이다. 최소 0, 최대 5라는 수치를 보였다.

- Lang.num

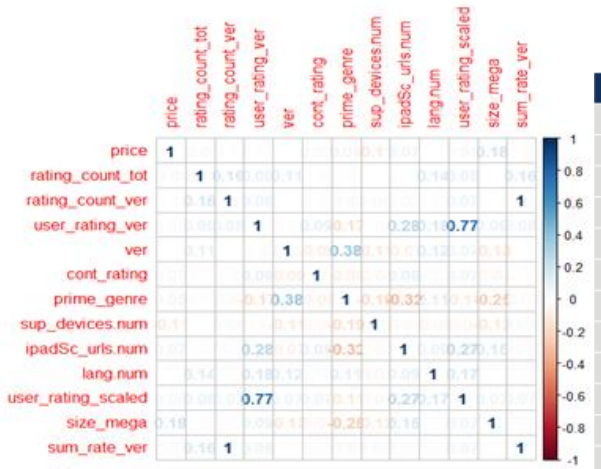
: 지원가능 언어 수를 나타내는 변수이다. 최소 0, 최대 7.5라는 수치를 보였다.

- Sum_rate_ver

: 현재 버전 총 평점을 나타내는 변수이다.

(2) Regression

1. Non-Penalty



[그림-13] 변수들의 상관계수

	GVIF	Df	GVIF^(1/(2*Df))
Price	1.0673	1	1.0331
Rating_count_tot	1.0753	1	1.0370
Rating_count_ver	100.3800	1	10.0190
User_rating_ver	1.1540	1	1.0742
Ver	1.2173	1	1.1033
Cont_rating	1.2221	3	1.0340
Prime_genre	1.5340	1	1.2385
Sup_devices.num	1.0950	1	1.0462
IpadSc_urls.num	1.2190	1	1.1041
Lang.num	1.0914	1	1.0447
Size_mega	1.1842	1	1.0882
Sum_rate_ver	100.2625	1	10.0131

[표-1] VIF

"Sum_rate_ver"과 "Rating_count_ver"의 상관성이 높고, 다중공선성이 존재한다. 따라서 파생변수인 "Sum_rate_ver"을 제거한 후 분석을 진행할 것이다. Data1 과 Data2 에 대해서 기본적으로 모든 변수를 사용하여 만든 모델과 변수선택법[2] 을

[2] Forward Selection method: 반응변수와 상관관계가 가장 큰 설명변수부터 시작하여 하나씩 설명변수를 선택하는 방법

Backward Elimination method: 설명변수를 모두 포함한 full model에서 설명력이 가장 작은 설명변수부터 하나씩 설명변수를 제거하는 방법

사용하여 만든 모델들(Forward, Backward, Stepwise)을 비교해볼 것이다. 또한 모델들을 진단하기 위해 6 가지의 Measure[3]들을 사용할 것이다.

1) 모델링

		AE	MAE	MAPE	MSE	RMSE	R^2_{adj}
Data1	DF_1	0.0127	0.4173	0.0281	0.3982	0.6311	0.6170
	FOR_1	0.0124	0.4168	0.0279	0.3984	0.6312	0.6171
	BACK_1	0.0124	0.4168	0.0279	0.3984	0.6312	0.6171
	STEP_1	0.0124	0.4168	0.0279	0.3984	0.6312	0.6171
Data2	DF_2	0.0123	0.4178	0.0282	0.3983	0.6311	0.6169
	FOR_2	0.0124	0.4168	0.0279	0.3984	0.6312	0.6171
	BACK_2	0.0124	0.4168	0.0279	0.3984	0.6312	0.6171
	STEP_2	0.0124	0.4168	0.0279	0.3984	0.6312	0.6171

[표-2] 변수제거 전

	AE	MAE	MAPE	MSE	RMSE	R^2_{adj}
DF_1	0.0127	0.4173	0.0281	0.3982	0.6311	0.6171
FOR_1	0.0124	0.4168	0.0279	0.3984	0.6312	0.6171
BACK_1	0.0124	0.4168	0.0279	0.3984	0.6312	0.6171
STEP_1	0.0124	0.4168	0.0279	0.3984	0.6312	0.6171
DF_2	0.0123	0.4178	0.0282	0.3983	0.6311	0.6170
FOR_2	0.0124	0.4168	0.0279	0.3984	0.6312	0.6171
BACK_2	0.0124	0.4168	0.0279	0.3984	0.6312	0.6171
STEP_2	0.0124	0.4168	0.0279	0.3984	0.6312	0.6171

[표-3] 변수제거 후

변수제거 전 데이터를 가지고 모델들을 만들어 6measure 로 진단한 것과 변수제거 후 데이터를 가지고 모델들을 만들어 6measure 로 진단한 것과 큰 차이가 없다는 것을 알 수 있다. 소수점 5 번째, 6 번째부터 차이가 있으므로 거의 비슷하다고 볼 수 있다. 하지만 변수제거 후의 모형이 변수제거 전 모형보다 미미한 차이지만 더 설명력이 있다.

Stepwise selection method : 각 단계에서 이미 선택된 변수들의 중요도를 다시 검사하여 중요하지 않은 변수를 제거하는 방법

[3] 1. $Average\ error = \frac{1}{n}\sum_{i=1}^{n}(y-\acute{y})$ 2. $MAE = \frac{1}{n}\sum_{i=1}^{n}|y-\acute{y}|$ 3. $MAPE = 100\% \times \frac{1}{n}\sum_{i=1}^{n}\frac{|y-\acute{y}|}{|y|}$

4. $MSE = \frac{1}{n}\sum_{i=1}^{n}(y-\acute{y})^2$ 5. $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y-\acute{y})^2}$ 6. $R^2_{adj} = 1-\left[\frac{n-1}{n-(p+1)}\right]\frac{SSE}{SST} \leq 1-\frac{SSE}{SST} = R^2$

Coefficients:

	Estimate	Std.Error	t value	Pr(> |t|)	
(Intercept)	-1.4456	0.0285	-50.647	< 2e-16	***
User_rating_ver	0.4213	0.0048	87.347	< 2e-16	***
Ver.L	0.1301	0.0121	10.730	< 2e-16	***
Ipadsc_urls.num	0.0269	0.0044	6.179	6.9e-10	***
Price	0.0051	0.0016	3.309	0.0009	***
Cont_rating 17+	-0.1028	0.0352	-2.922	0.0035	**
Cont_rating 4+	-0.0390	0.0231	-1.687	0.0917	.
Cont_rating 9+	-0.0063	0.0305	-0.207	0.8357	
Lang.num	0.0022	0.0011	2.057	0.040	*

[표-4] Data1 의 Forward Summary

Coefficients:

	Estimate	Std.Error	t value	Pr(> |t|)	
(Intercept)	-1.4456	0.0285	-50.647	< 2e-16	***
User_rating_ver	0.4213	0.0048	87.347	< 2e-16	***
Ver.L	0.1301	0.0121	10.730	< 2e-16	***
Ipadsc_urls.num	0.0269	0.0044	6.179	6.9e-10	***
Price	0.0051	0.0016	3.309	0.0009	***
Cont_rating 17+	-0.1028	0.0352	-2.922	0.0035	**
Cont_rating 4+	-0.0390	0.0231	-1.687	0.0917	.
Cont_rating 9+	-0.0063	0.0305	-0.207	0.8357	
Lang.num	0.0022	0.0011	2.057	0.040	*

[표-5] Data1 의 Backward Summary

Data1 과 Data2 를 나누는 기준인 "Prime_genre"변수가 Forward Selection 과 Backward Elimination 두 방법 다 선택이 되지 않아 Summary 값이 다 똑같이 나오게 되고, [표-2]처럼 Forward, Backward, Stepwise 의 measure 결과값이 같은 것을 볼 수 있다. 따라서 Data1 과 Data2 를 나눌 필요가 없어졌고, Data1 을 가지고 최종 모형을 결정해도 무관하다.

변수제거 전 후의 기준인 "Sum_rate_ver"도 변수선택이 안되므로 [표-2]와 [표-3]처럼 Forward, Backward, Stepwise 의 measure 결과값이 같은 것을 볼 수 있다.

2) Non-Penalty 최적모형 선택

파생변수인 "Sum_rate_ver"변수를 제거한 Data1 데이터로 Forward selection 방법을 사용해 최적 모형을 선택한다.

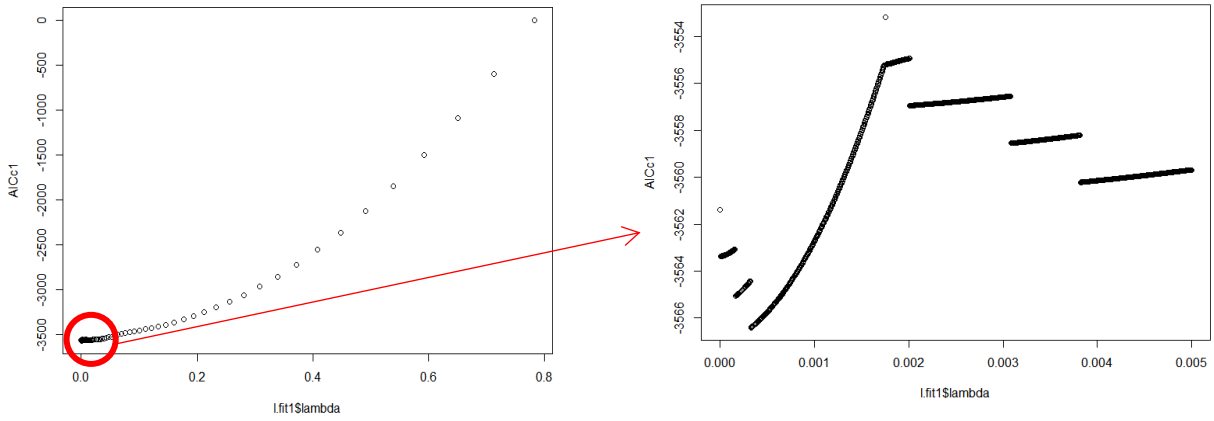
	AE	MAE	MAPE	MSE	RMSE	R^2_{adj}
DF_1	0.0127	0.4173	0.0281	0.3982	0.6311	0.6171
FOR_1	0.0124	0.4168	0.0279	0.3984	0.6312	0.6171
BACK_1	0.0124	0.4168	0.0279	0.3984	0.6312	0.6171
STEP_1	0.0124	0.4168	0.0279	0.3984	0.6312	0.6171

[표-6] regression non-penalty 최적모형

2. Penalty

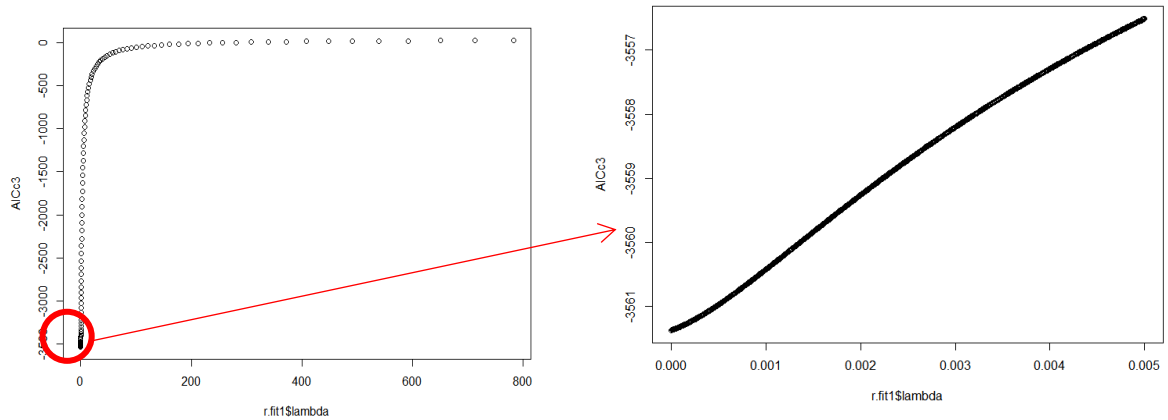
Penalized estimation 에서는 가장 대표적인 Penalty 함수[4]인 LASSO, RIDGE, SCAD 를 이용하여 각 penalty 에서 AIC 를 최소로 만들어주는 모형을 탐색해보았다.

1) 모델링



[그림-14] Lasso 의 AIC

-Lasso 에서 최소의 AIC 값을 가지기 위한 최적 lambda 는 0 에 가까운 값임을 확인하였다.

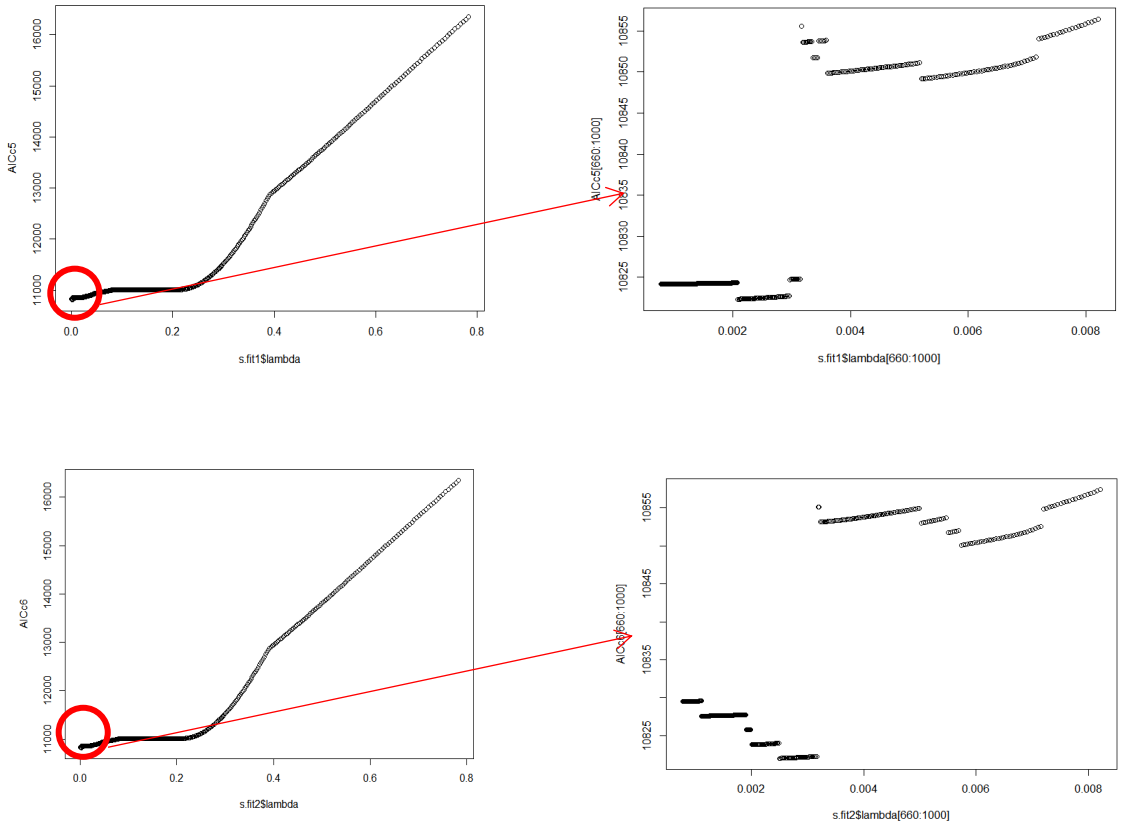


[그림-15] Ridge 의 AIC

-Ridge 에서 최소의 AIC 값을 가지기 위한 최적 lambda 는 0 임을 확인하였다.

---

[4] Lasso : $\min(\|Y - X\theta\|_2^2 + \lambda\|\theta\|_1)$ Ridge : $\min(\|Y - X\theta\|_2^2 + \lambda\|\theta\|_2^2)$



[그림-16] SCAD 의 AIC

-SCAD 에서 최소의 AIC 값을 가지기 위한 최적 lambda 는 0 에 가깝다.

2) Penalty 최적모형 선택

	AE	MAE	MAPE	MSE	RMSE
LASSO_1	0.0145	0.4179	0.2859	0.4030	0.6348
LASSO_2	0.0142	0.4183	0.2862	0.4029	0.6347
RIDGE_1	0.0149	0.4186	0.2894	0.4072	0.6381
RIDGE_2	0.0146	0.4190	0.2897	0.4073	0.6382
SCAD_1	0.0151	0.4187	0.2903	0.4087	0.6393
SCAD_2	0.0148	0.4190	0.2903	0.4087	0.6393

[표-7] Penalty 함수들의 최적 모델들

대체적으로 LASSO 방법론이 5 가지 test error 에서 모두 가장 좋은 것으로 보여졌다.

3. Regression 결론 및 한계

	AE	MAE	MAPE	MSE	RMSE
Optimal By Non Penalty (For_1)	0.0124	0.4168	0.0279	0.3984	0.6312
Optimal By Penalty (LASSO_2)	0.0142	0.4183	0.2862	0.4029	0.6347

[표-8] Non-Penalty 와 Penalty 의 최적 모델 비교

Penalty 함수를 이용해 모수를 추정한 회귀 모형과 그렇지 않고 단순히 best subset selection(non-penalty regression)을 통해 유의미한 변수를 뽑아낸 모형 중 5 가지 model assessment measure 에서 모두 후자가 더 낫다는 결론을 시사하였다.

현재 우리는 그림-3 에서 볼 수 있듯 종속변수의 분포가 쌍봉(bimodal)의 형태임을 알 수 있다. 이를 인지하였지만 사용 할 수 있는 분포가 정규분포로 한정됐었기에 Ridge regression 에서 lambda 값이 0 이 나오는 등의 현상이 발생한 것으로 파악된다.

쌍봉 분포의 형태를 갖는 분포를 종속변수에 가정함으로써 regression 의 성능을 높일 여지가 남아있다고 결론을 내릴 수 있겠다.

(3) Classification

1. Quantile 을 활용한 low/middle/high 데이터

1) Multinomial Logistic regression & RF

첫 번째로, 두 모델 모두 default parameter 을 사용하였으며, measure 로는 test acc 와 test hum 을 사용하였다. Hum 은 hypervolumn under manifold 로써, AUC 의 다차원 버전이다.

	multinom	rf
test_acc	0.8040446	0.8668061
test_hum	0.6011467	0.7892026

[표-9] classes 분류 모델 결과(Default)

2) RF-tuning

Randomforest 모델의 경우 default parameter 임에도 불구하고 상대적으로 높은 성능을 보이므로, tuning 을 진행하였다. RF 의 tuning parameter 중 mtry, ntree, nodesize 만을 조정하였고, mtry 5, ntree 60, nodesize 3 에서 최적 모형을 얻었다.

	default	mtry:3	mtry:5	mtry:7	mtry9	mtry:11
mtry	3.0000000	3.0000000	5.0000000	7.0000000	9.0000000	11.0000000
ntree	500.0000000	60.0000000	60.0000000	30.0000000	30.0000000	270.0000000
node	1.0000000	3.0000000	3.0000000	2.0000000	1.0000000	2.0000000
test_acc	0.8668061	0.8709902	0.8730823	0.8709902	0.8709902	0.8709902

[표-10] Randomforest parameter tuning

3) Multinom-tuning

Multinom model 경우 parameter를 조정하지 않고, 변수에 대한 선택을 진행하였다. 위의 RF 모델의 importance 기준으로 상위 5개에 속하는 변수만 사용하여 최적 모형을 얻었다. 상위 5개의 변수는 rating_count_tot, rating_count_ver, user_rating_ver, sum_rate_tot, size_mega이다.

4) 모델평가

	multinom	rf	tune_multi	tune_rf
test_acc	0.8040446	0.8668061	0.7977685	0.8730823
test_hum	0.6011467	0.7892026	0.6145244	0.7916897

[표-11] 3 classes 분류 모델 결과(Default/Tuning)

두 모델 모두 80%(acc)의 높은 성능을 보였고, RF 모델의 성능이 더 높았다. 하지만 label 자체를 quantile 을 사용하여 임의로 만들어주었기 때문에 위 와 같은 결과가 나왔다고 판단하여, 실제 "User_rating"변수를 가지고 재실험을 하였다.

2. "User_rating" 사용

"User_rating"은 0 부터 5 까지 0.5 단위의 10 classes 로 구성되어 있다. 전처리를 통해 dataset 을 추가 구성하였다. Data1 은 10 classes 로 구성된 "User_rating"이고, Data2 는 반올림을 통한 5 classes(1,2,3,4,5)로 구성된 "User_rating"이다.

앞서 "Prime_genre"에 대한 두가지 데이터를 생성하였다. 그중에서 Games, Non-game 으로 전처리한 데이터에서" User_rating"을 5classes 로 나눈 경우와 10classes 로 나눈 두가지 데이터가 존재한다. 두가지 데이터에 모두 Mutilnomial logistic regression, SVM(Radial, Linear, Sigmoid, Polynomial), RF 모델을 default 로 적용하여 평가하였다.

A matrix: 1 × 6 of type dbl

	Multinom	SVM.radial	SVM.linear	SVM.polynomail	SVM.sigmoid	Rndomforest
test Accuracy	0.6178522	0.6680614	0.5829847	0.5822873	0.5069735	0.735007

[표-12] Game/Non-Game 5classes

A matrix: 1 × 6 of type dbl

	Multinom	SVM.radial	SVM.linear	SVM.polynomail	SVM.sigmoid	Rndomforest
test Accuracy	0.4986053	0.5209205	0.4937238	0.4965132	0.4372385	0.6283124

[표-13] Game/Non-Game 10classes

두 데이터셋에 대하여 모두 RF가 우수한 성능을 보였고, 추가적으로 "Prime_genre"에 대해서 6가지 카테고리로 전처리한 데이터에서도 같은 결과를 보였다.

A matrix: 1 × 6 of type dbl

	Multinom	SVM.radial	SVM.linear	SVM.polynomail	SVM.sigmoid	Rndomforest
test Accuracy	0.6150628	0.665272	0.5669456	0.5760112	0.525802	0.7433752

[표-14] 6 category 5classes

A matrix: 1 × 6 of type dbl

	Multinom	SVM.radial	SVM.linear	SVM.polynomail	SVM.sigmoid	Rndomforest
test Accuracy	0.5041841	0.5083682	0.4930265	0.4958159	0.4630404	0.6213389

[표-15] 6 category 10classes

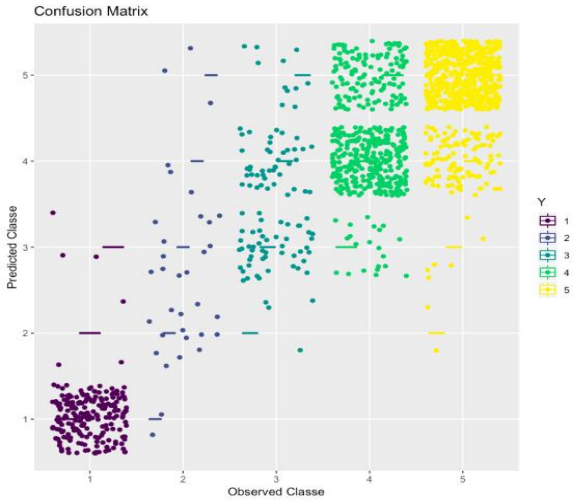
3. "User_rating" 5classes and RF

최종 classification은 "User_rating"(1,2,3,4,5)에 대해 RF 모델을 사용하였다. 마찬가지로 "Prime_genre"에 대한 전처리가 두 가지였고, 이에 따른 다른 데이터 셋에 대해 각기 실험을 진행하였다.

"Prime_genre"를 두가지 범주로 나눈 경우 test accuracy 0.7545의 성능을 보였고, 6가지 범주로 나눈 경우 test accuracy 0.7580의 성능을 보였다. 또한, 6가지 범주로 처리한 경우 "Prime_genre"에 대한 importance가 상대적으로 높게 나왔고, 위 실험결과를 종합하여 최종 모델과 최종 데이터 셋을 결정하였다.

4. Classification 결론

최종 모델은 "Prime_genre"를 6개의 카테고리화 하고, "User_rating"을 5classes로 전처리한 데이터에 대해서, Randomforest로 결정하였다. Parameter tuning을 통해 default model에 비해 test acc 0.015 높은 결과를 얻었다.

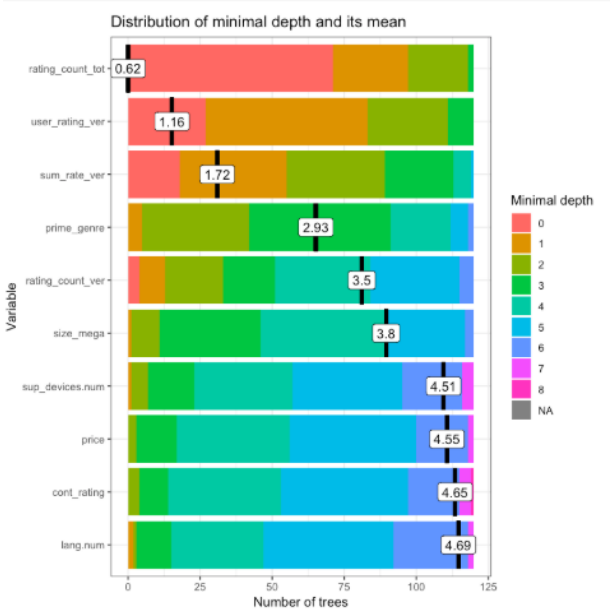


[그림-17] Confusion matrix

class 1 과 4, 5에 대한 분류는 상대적으로 잘 이루어 지나, class 2, 3 에 대한 오분류가 높다. 이는 randomforest model에서 제공하는 통계치들을 확인한 결과이다.

```
Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
Sensitivity            0.9901 0.608696  0.53846   0.6497   0.7778
Specificity            0.9951 0.985826  0.95681   0.8423   0.8438
Pos Pred Value         0.9710 0.411765  0.45794   0.6540   0.8119
Neg Pred Value         0.9984 0.993571  0.96835   0.8398   0.8141
Prevalence            0.1416 0.016039  0.06346   0.3145   0.4644
Detection Rate         0.1402 0.009763  0.03417   0.2043   0.3612
Detection Prevalence   0.1444 0.023710  0.07462   0.3124   0.4449
Balanced Accuracy      0.9926 0.797261  0.74764   0.7460   0.8108
```

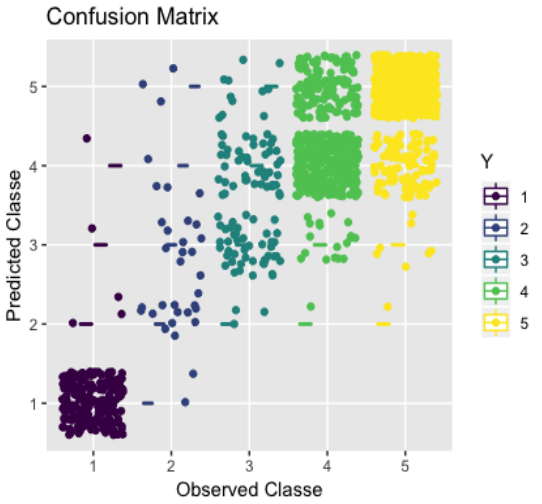


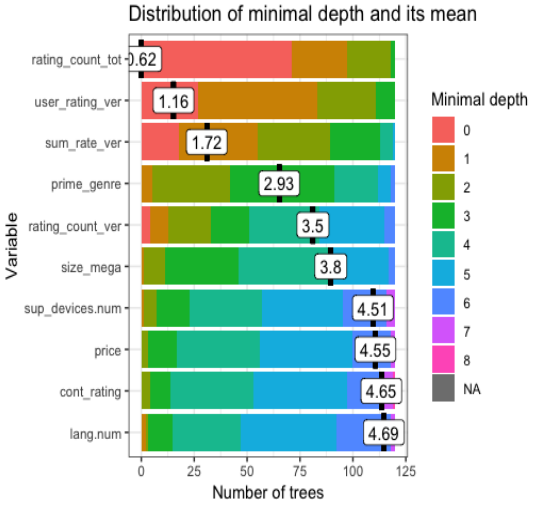
[그림-18] Distribution of minimum depth and its mean

Randomforest은 여러개의 tree를 bagging 알고리즘을 변형한 앙상블 모델이다. 따라서, 개별 tree를 활용하여 설명변수들의 min_depth distribution을 구할 수 있다. 그림에서의 수치가 낮을수록 상위 node의 split에 사용된 경우이며, 더 중요한 변수라고 해석할 수 있다.

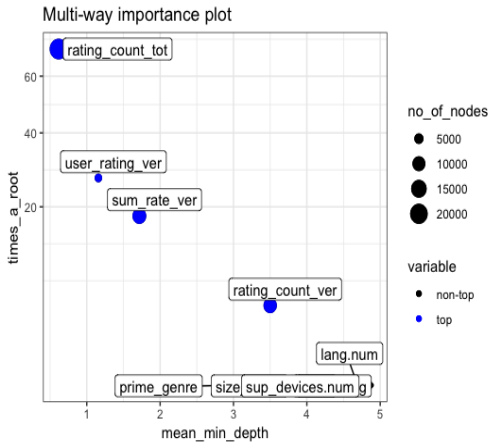
Ⅲ. 결론

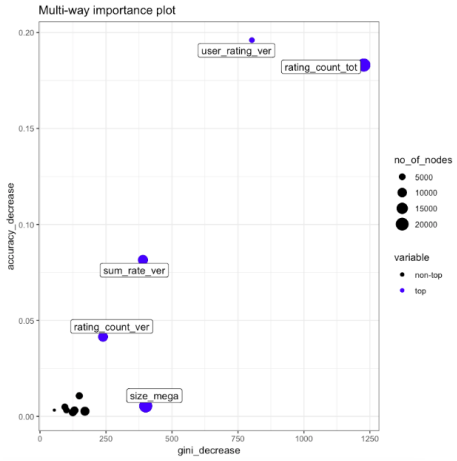
(1) 결과 해석





[그림-19] Confusion Matrix [그림-20] Distribution of Minimal depth and its mean





[그림-21] Multi_way importance plot 1 [그림-22] Multi_way importance plot 2

Var	Non	AD	GD	TAR
cont_rating	4923	0.0032	100	0
ipadSc_urls.num	5431	0.0049	94	0
lang.num	9428	0.0026	170	0
price	7268	0.0031	130	0
prime_genre	5846	0.0107	148	0
rating_count_tot	21169	0.1830	1226	71
rating_count_ver	10654	0.0414	238	4
size_mega	20949	0.0054	400	0
sum_rate_ver	11454	0.0815	390	18
sup_devices.num	7022	0.0020	124	0
user_rating_ver	3708	0.1960	803	27
ver	2786	0.0032	54	0

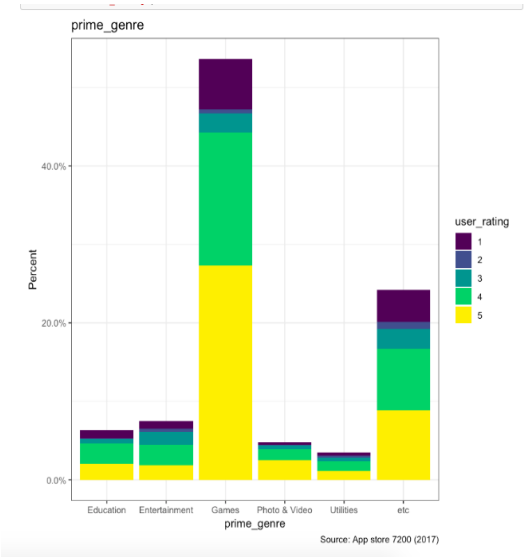
[표-16] Importance Frame[5]

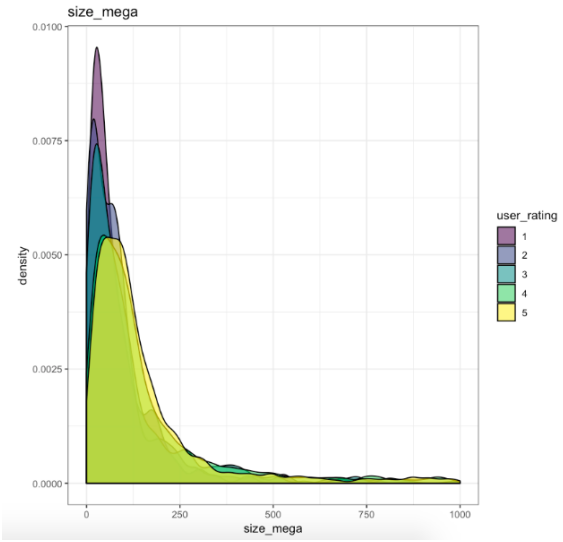
요약하자면, 범주형 변수 중에서는 "Prime_genre"가 유의했으며, 연속형 변수 중에서는 "Size_mega", "Rating_count_ver", "User_rating_ver", "Sum_rate_ver", "Rating_Count_tot" 변수가 유의하였다.

(2) 한계점

우리는 'Size_mega','Prime_genre' 두 변수가 유의하다고 판단하였다. 그러나 두 변수에 대한 Importance는 상대적으로 미미한 편에 속하였다. 특히, 상위 4개의 변수들을 살펴보면 모두 직접적으로 rating과 관련된 변수들인데 어플 용량과 어플의 장르(카테고리)가 중요한 변수로 등장하였다. 그림으로 살펴보면 다음과 같다.

[5] Non은 Split에 사용된 횟수를 의미하며, AD는 평균 Acc감소율을 뜻한다. GD는 평균 impurity 감소율을 나타내며, TAR은 root node로 사용된 횟수를 의미한다.





장르는 모든 범주에 고르게 분포되어 있었지만, "Size_mega"는 치우친 분포를 보였다. 즉, 상대적으로 높은 평점을 가진 어플리케이션들의 용량이 크다는 사실을 알 수 있었다.

Ⅳ. 팀원 역할 및 기여도

이름	김종휘	박성진	안수이	지성인
역할 및 기여도	모델링 보고서 PPT (25%)	모델링 보고서 PPT 데이터 전처리 코드 정리 (25%)	모델링 보고서 PPT (25%)	모델링 보고서 PPT (25%)

Ⅴ. 참고문헌

https://ettrends.etri.re.kr/ettrends/154/0905002058/0905002058.html - 주제 선정

https://www.kaggle.com/ramamet4/app-store-apple-data-set-10k-apps - 데이터 수집