고려대학교
빅데이터 연구회

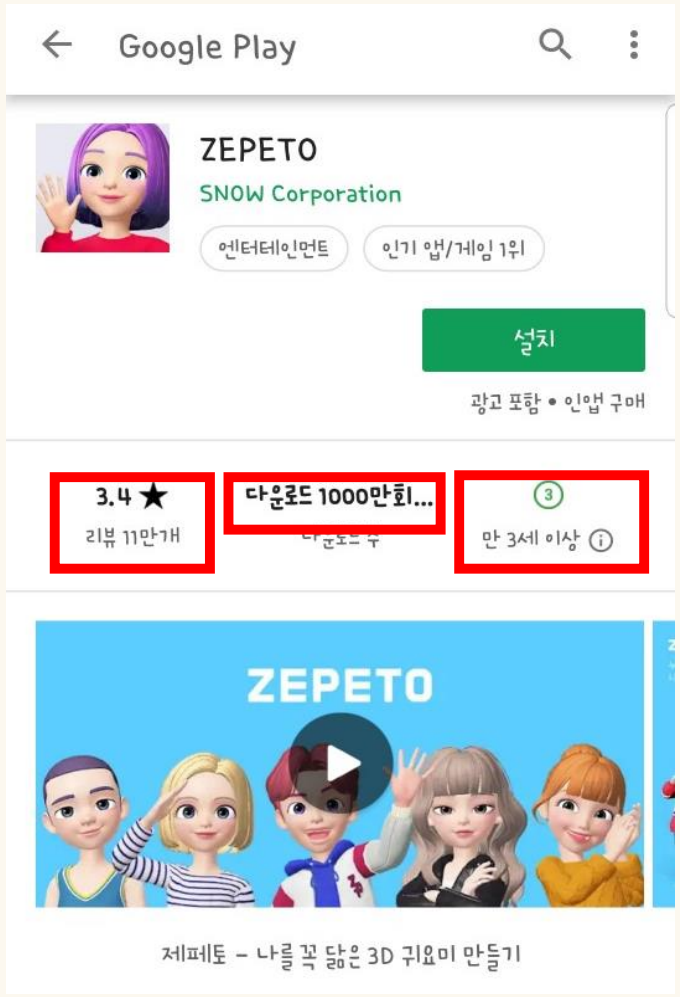
KU-BIG

지도 1조: Google Playstore Apps

문재원 김민주 양정윤 우경민

목차

1. 문제정의



앱개발 플랫폼 `빅스비 캡슐`…AI생태계 정조준
매일경제 | A6면1단 | 2018.11.08. | 네이버뉴스
◆삼성 폴더블폰 전격 공개 ◆삼성전자가 자체 인공지능(AI) 브랜드 빅스비의 영향력을 확대하기 위해 '빅스비 캡슐'이라는 서비스를 내놓기로 했다. 또 그동안 갤럭시 앱스(스마트폰), 기어 스토어(웨어러블), 빅스비...

중기부, 고교생 **앱개발** 경진대회 시상식 개최
이데일리 | 2018.11.05. | 네이버뉴스
(사진=중소벤처기업부) 중소벤처기업부가 SK플래닛과 공동으로 5일 경기도 판교 SK 플래닛 본사에서 '스마틴 앱 챌린지 2018' 시상식을 개최했다고 밝혔다. 스마틴 앱 챌린지는 2011년부터 개최해온 국내 고교생...

- 앱 개발에 대한 관심도 ↑
- 현실에서 쉽게 확인할 수 있는 정보를 이용해서 데이터 분석을 해볼 수 없을까?

1. 문제정의

실제 앱 개발을 할 때 도움이 될만한 정보에 대한 insight 도출

↓

기존의 앱데이터를 바탕으로 성공적인 앱이 갖추고 있는 요소는 무엇인가

↓

앱의 평점, 설치수에 대한 예측모델을 만들어보자!

2. 데이터 소개

- 구글 플레이 스토어 앱의 속성에 대한 데이터
- 10778개 앱, 13개의 column 으로 이루어짐
- 이름, 카테고리, 평점, 리뷰 개수, 크기, 다운로드수, 유료여부, 가격, 이용연령, 최신 업데이트 날짜, 현재 버전, 최소 안드로이드 버전
- 출처: Kaggle

App	Category	Rating	Reviews	Size	Installs	Type	Price	Content R	Genres	Last Updated	Current V	Android Ver
Photo Edi	ART_AND	4.1	159	19M	10,000+	Free	0	Everyone	Art & Des	07-Jan-18	1.0.0	4.0.3 and up
Coloring b	ART_AND	3.9	967	14M	500,000+	Free	0	Everyone	Art & Des	15-Jan-18	2.0.0	4.0.3 and up
U Launche	ART_AND	4.7	87510	8.7M	5,000,000	Free	0	Everyone	Art & Des	01-Aug-18	1.2.4	4.0.3 and up
Sketch - D	ART_AND	4.5	215644	25M	50,000,00	Free	0	Teen	Art & Des	08-Jun-18	Varies wit	4.2 and up
Pixel Draw	ART_AND	4.3	967	2.8M	100,000+	Free	0	Everyone	Art & Des	20-Jun-18	1.1	4.4 and up
Paper flow	ART_AND	4.4	167	5.6M	50,000+	Free	0	Everyone	Art & Des	26-Mar-17	1	2.3 and up
Smoke Eff	ART_AND	3.8	178	19M	50,000+	Free	0	Everyone	Art & Des	26-Apr-18	1.1	4.0.3 and up

2. 데이터 소개

Installs vs. Rating?

-> 평점, 리뷰개수, 최신 업데이트 날짜 등 이용자들이 설치해야 얻을 수 있는 사후변수가 다수이므로 예측변수로 설치 수(Installs)는 적합하지 않다고 판단

3. 전처리 방법

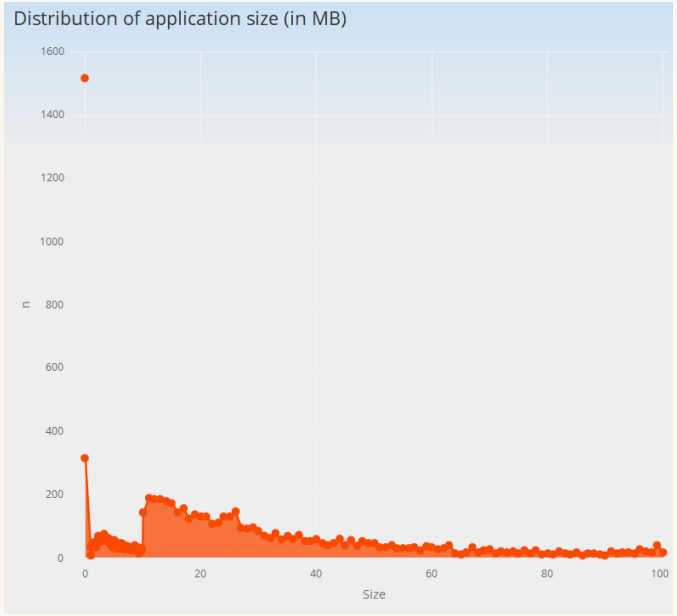
1) 1차 Data 처리

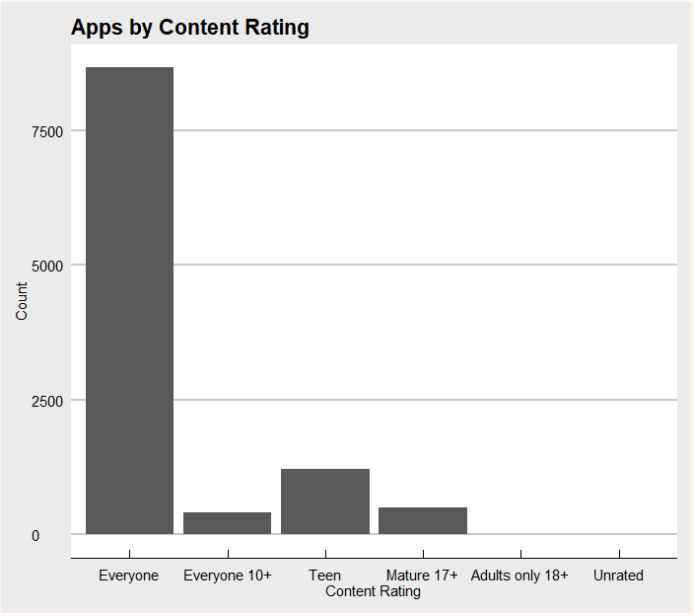
- 분석에 불필요한 기호 및 문자 제거 : + , $, MB, K
- 불필요하게 Factor로 처리된 변수를 Numeric으로 변경
- 오류 데이터 및 중요 변수의 결측 행 제거 – 총 9400개 OBS

```
> data.clean <- data %>%
+   mutate(
+     # Eliminate M to transform Size to numeric
+     Size = gsub("M", "", Size),
+     # Replace cells with k to 0 since it is < 1MB
+     Size = ifelse(grepl("k", Size), 0, as.numeric(Size)),
+     # Transform reviews to numeric
+     Reviews = as.numeric(Reviews),
+     # Transform Rating to numeric
+     Rating = as.numeric(as.character(Rating)),
```

3. 전처리 방법

2) EDA 진행 후 데이터 처리





Size : 1MB 이하는 size = 0으로 처리

Content Rating:

기존의 5개의 범주 → 3개의 범주

3. 전처리 방법

1) EDA 진행 후 데이터 처리

- Size : 1MB 이하는 0으로 처리
- Content Rating : 기존의 5개의 범주 → 3개의 범주
- Last updated : 마지막 업데이트 날짜부터 데이터 수집 시기까지 개월 수
- Android Version : 안드로이드 버전 9.0 에서 몇 단계나 떨어져 있는지

* Numeric 변수들은 scaling 후 분석진행

전처리를 통해 생성된 최종 변수들

Rating : 1.0~5.0 사이의 점수

Reviews : 0~1000000개

Size : 1MB 이하는 0으로 취급

Installs : 범주형? or 연속형?

전처리를 통해 생성된 최종 변수들

Type : Paid =1, Free = 0

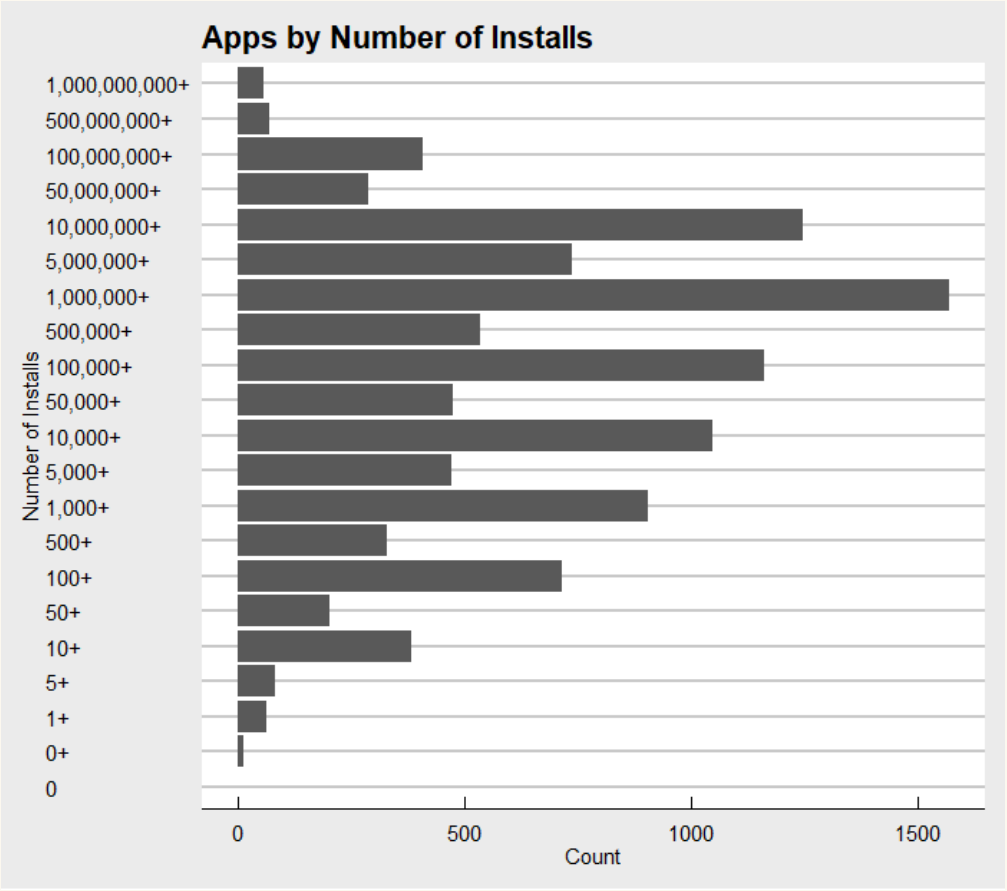
Price : 유료 앱 분석 관련하여 사용

Content Rating : Everyone / Teen / Mature+17

Last Updated : 2018.8. 로부터의 개월 수

Android ver. : Ver.9.0 - 사용가능 최소 Ver. 차이

문제점 인식



Installs(앱 설치수)

1. 실제로는 연속형 변수이나 데이터에서는 구간별로 나누어진 범주형 변수로 주어짐
2. 구간간격이 점점 커짐

Ex. 10 ~ 50 vs 10000 ~ 50000

해결방안1 구간별 난수 생성

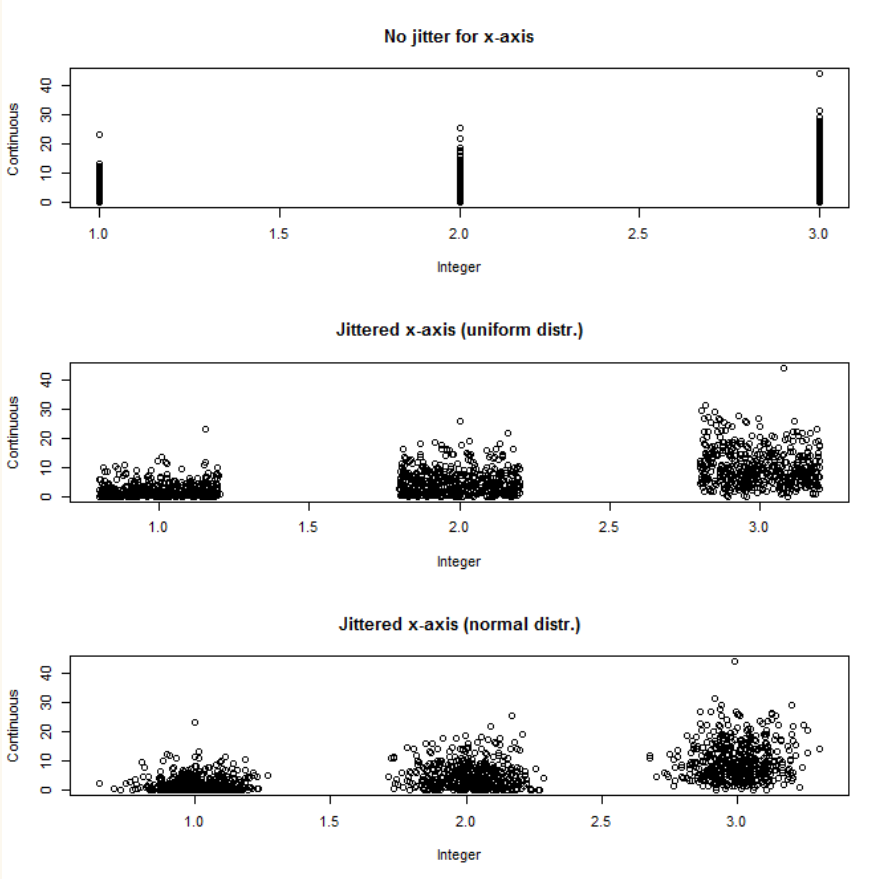
구간별로 100~500 사이의 난수 생성,

1000,000~5000,000 사이의 난수 생성하는

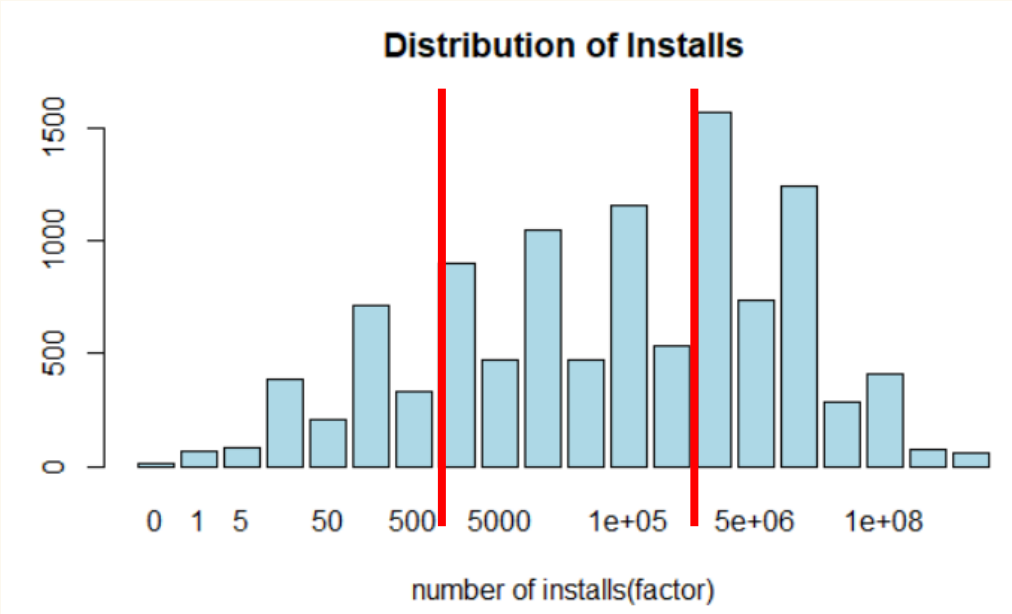
방식으로 Random noise 만들기

-> 구간 간격이 점점 커짐

-> 데이터 왜곡이 심함



해결방안2 Installs 구간별로 Rating 예측 모델링



Installs 구간별 모델링?

Rating에 영향을 미칠 것으로 생각되는 가장 중요한 변수를 구간으로 나눔으로써 x변수에서 제외시키는 것은 의미 있는 분석이라고 할 수 없음. -> **Installs을 numeric으로 보고 분석 진행**

4. 모델링 - WLS Regression

IDEA

Install 후 Review를 남긴 사람들이 많은 앱의 Rating을 중요하게 반영

EX.

Installs = 100 : Rating = 4.0

Installs = 1000000 : Rating = 4.0

같은 Rating이지만 다르지 않을까? – WLS시도

4. 모델링 - WLS Regression

가중치 위한 파생변수 생성

- Install 대비 Review 수가 많을수록, 즉 참여도가 높을 수록 의미 있는 Rating

(Install이 1000인데 review가 10개인 앱과 Install이 10,000,000인데 review가 10개인 앱을 동일하게 볼 수 없다)

- Review/install를 가중치로 참여도의 지표로 사용

4. 모델링 - WLS Regression

```
Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)              4.0412543  0.0366449 110.282  < 2e-16 ***
log(Reviews)             0.0409046  0.0017205  23.775  < 2e-16 ***
google3$Size            -0.0006398  0.0001966  -3.254  0.00114 **
how.month               -0.0032046  0.0004208  -7.616 3.00e-14 ***
as.numeric(Android.Ver) -0.0208351  0.0064854  -3.213  0.00132 **
TypePaid                 0.2037560  0.0134834  15.112  < 2e-16 ***
Content.RatingMature 17 -0.0969616  0.0202149  -4.797 1.65e-06 ***
Content.RatingTeen       0.0062203  0.0112100   0.555  0.57899
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.06234 on 6413 degrees of freedom
Multiple R-squared:  0.1329,    Adjusted R-squared:  0.132
F-statistic: 140.5 on 7 and 6413 DF,  p-value: < 2.2e-16
```

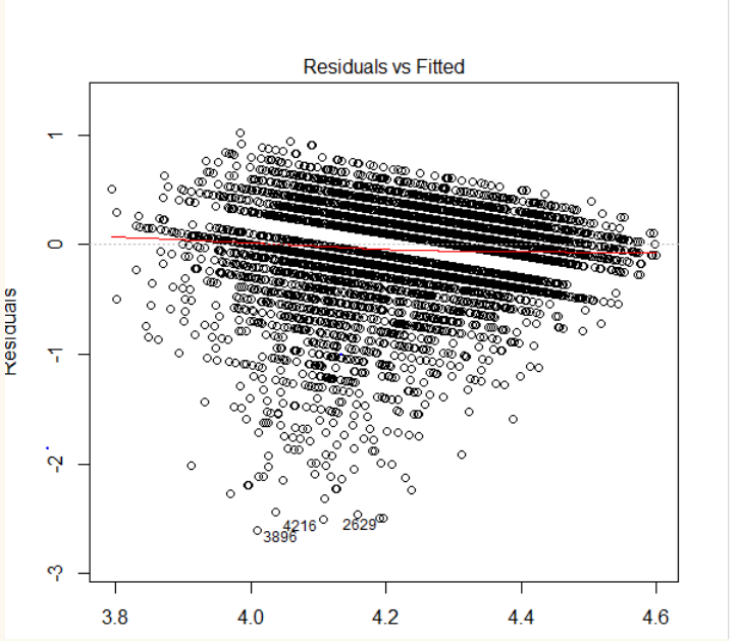
- R-square 값..?
- Content Rating 유의하지 않음

4. 모델링 - Regression

Content Rating 변수 조정

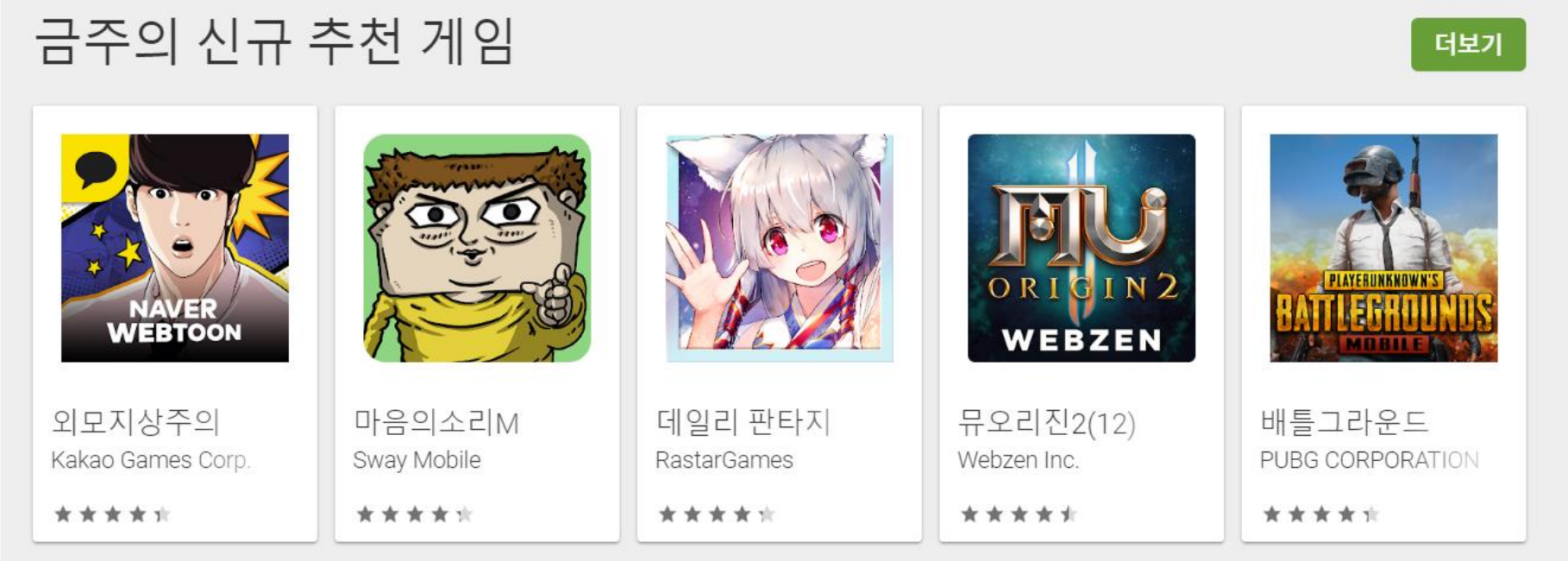
```
Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)           4.0306214  0.0366382 110.011  < 2e-16 ***
log(Reviews)          0.0411691  0.0017225  23.901  < 2e-16 ***
google3$Size         -0.0005221  0.0001954  -2.673  0.00754 **
how.month            -0.0031578  0.0004214  -7.494 7.57e-14 ***
as.numeric(Android.Ver) -0.0200557  0.0064942  -3.088  0.00202 **
TypePaid              0.2074631  0.0134835  15.386  < 2e-16 ***
Content.NEW2         -0.0143527  0.0103592  -1.386  0.16595
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.06245 on 6414 degrees of freedom
Multiple R-squared:  0.1299,    Adjusted R-squared:  0.1291
F-statistic: 159.6 on 6 and 6414 DF,  p-value: < 2.2e-16
```

- Everyone / Otherwise로 분류
- R-square 더 낮아짐



5. 새로운 모델링 - Logistic Regression

"Featured": 구글 스토어의 선택으로 첫 페이지 등에 표시되는 것. '스토어의 추천'



5. 새로운 모델링 - Logistic Regression

IDEA: Featured

- 평점 4.0이상은 피처드 되는 앱들이 갖춰야 할 필수요건
- 어플 개발자의 입장에서 4.0이 새로운 기준이 될 수 있다.

  -> Rating 4.0 이상 = 1 / Rating 4.0 미만 으로 high.rating 변수 생성
- 개발자 입장에서 평점 4.0이 될 수 있을지 여부를 예측하는 모델
- 개발자가 자신의 앱을 과대평가(0인데 1이라고 판단)하는 것이 더 문제라고 생각되어 specificity를 늘리는 방향으로 cutoff 조정

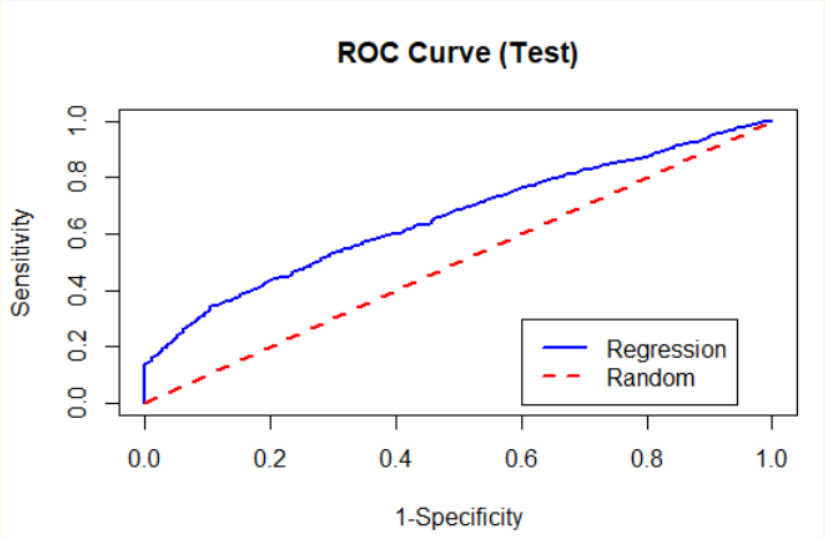
5. Logisitic Regression

1) 7:3으로 data partition

- Cutoff=0.75

```
> ctable
        Predicted
Actual    0    1
     0  233  340
     1  524 1600
```

```
> miss.err = 1-sum(diag(ctable))/sum(ctable) # Misclassification Rate
> miss.err
[1] 0.320356
>
> pred.acc = 1 - miss.err #Prediction Accuracy
> pred.acc
[1] 0.679644
>
> diag(ctable)[2]/apply(ctable, 1, sum)[2] # Sensitivity
        1
0.7532957
> diag(ctable)[1]/apply(ctable, 1, sum)[1] # Specificity
        0
0.4066318
```



AUC= 0.655345

5. Logisitic Regression

2) 5-fold CV

- Cutoff=0.75

```
> Reg.cv.sns = (Reg.sns)/5
> Reg.cv.spc = (Reg.spc)/5
> cv.err.train = miss.err.train/ V; cv.err.train # CV training error
[1] 0.315472
> cv.err.test = miss.err.test/ V;cv.err.test # CV test error 오분류율
[1] 0.3255943
> Reg.cv.sns; Reg.cv.spc
          1
0.2948546
          0
0.1788499
```

-> Cross validation을 했더니 specificity가 확 떨어짐

5. Logisitic Regression

2) 5-fold CV

- Cutoff=0.768

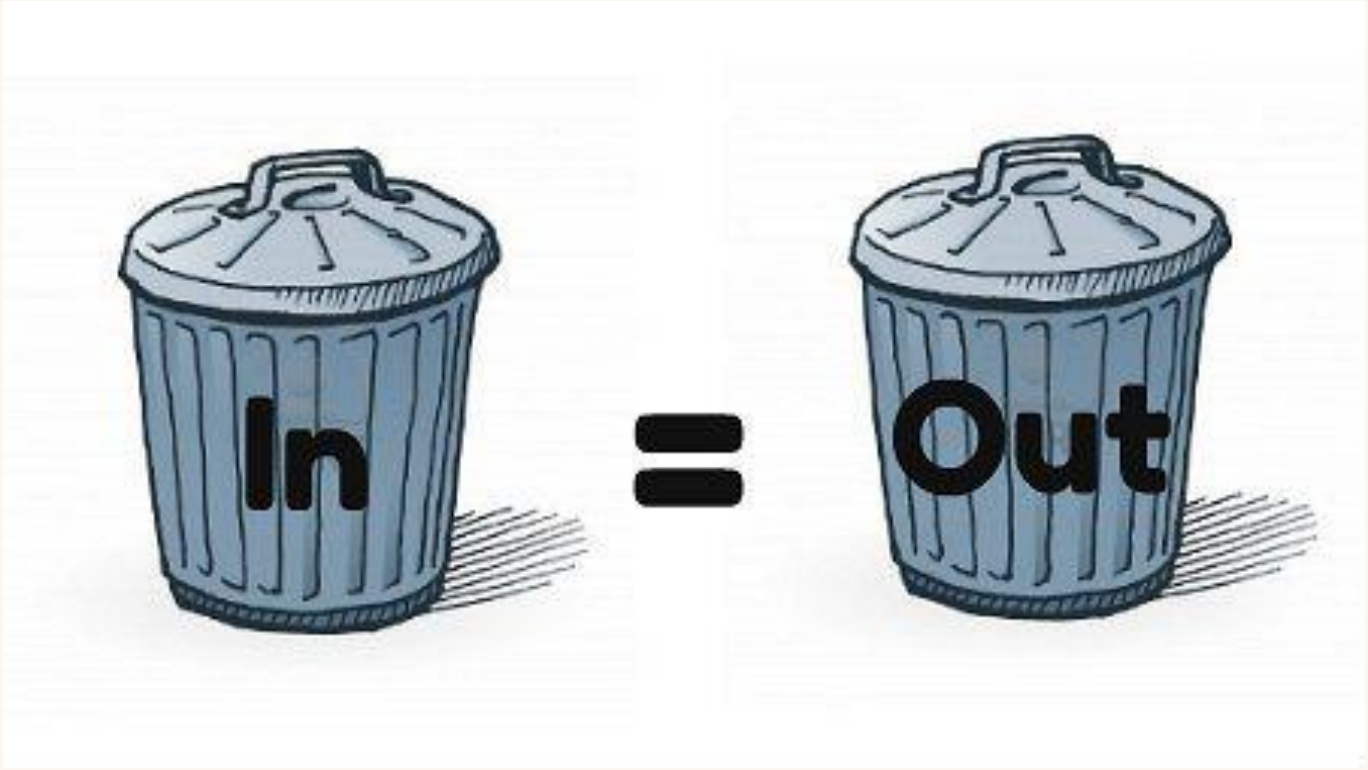
```
> cv.err.train = miss.err.train/ V; cv.err.train # CV training error
[1] 0.3945961
> cv.err.test = miss.err.test/ V;cv.err.test # CV test error 오분류율
[1] 0.3994583
> Reg.cv.sns; Reg.cv.spc
          1
0.2343891
          0
0.2613273
```

-> Error rate를 희생하더라도 specificity 좀 더 높일 수 있도록 cutoff를 조정해보았으나 크게 유의미한 결과를 얻지는 못했다.

6. 의의 및 한계

- Installs 변수를 넘어서지 못했다.
- 데이터의 x변수가 Y변수를 충분히 설명하지 못해 모든 모델의 성능이 떨어진다.
- 다양한 알고리즘을 적용해보지 못했다.

6. 의의 및 한계




Do you have
any
question?
Thank you
for your attention.