CÁC YẾU TỐ ẢNH HƯỞNG ĐẾN NGƯỜI DÙNG ĐỐI VỚI CÁC ỨNG DỤNG TRÊN GOOGLE PLAY STORE

By
Group 2

Class RPAN233577_21_1_01
Instuctor Quach Dinh Hoang
December 19th, 2021
Faculty of Information Technology
HCMC University of Teachnology and Education

ABSTRACT

Thị trường ứng dụng trực tuyến trên smartphone ngày càng phát triển. Việc Phân tích ứng dụng dành cho thiết bị di động là một cách tuyệt vời để có thể khai thác lợi nhuận. Bằng phương pháp trực quan hóa dữ liệu và phân tích thống kê giúp ta có hiểu hơn về thị trường ứng dụng. Từ đó, xây dựng các mô hình học máy giúp đưa ra các quyết định trong tương lai. Đó thực sự là một tiềm năng to lớn để thúc đẩy các doanh nghiệp phát triển ứng dụng hiệu quả và nắm xu hướng ứng dụng trong tương lai của thị trường.

1. INTRODUCTION

Trước khi phát hành một ứng dụng trên Google Play Store luôn là giải đoán khó khăn đối với các nhà phát triển. Nhất là trong thời kỳ bùng nổ về số lượng người dùng và thỏa mãn được hết yêu cầu luôn là vấn đề khó khăn. Việc tìm kiếm các giải pháp giải quyết luôn là vấn đề được mọi người quan tâm. Với mục tiêu là giúp phát triển một ứng dụng thành công nên output được xác định là *điểm đánh giá, số lượt bình luận*. Input sẽ là các thuộc tính còn lại (phiên bản, giá, thể loại,). Sau khi phân tích và đánh giá trên tập dữ liệu, chúng ta sẽ sử dụng thuật toán học máy (linear regression, k-nn, …) để đưa ra dự các giá trị tương lai.

2. DATA

Bài toán sẽ thực hiện phân tích sẽ dựa trên tập dữ liệu các ứng dụng trên Google Play Store. Tập dữ liệu này crawl bằng Jquery với hơn 10.800 quan sát được ghi nhận và được chia sẻ trên diễn đàn kaggle https://www.kaggle.com/lava18/google-play-store-apps

Dữ liệu mà ta có được vẫn còn khá thô sơ, để có thể phục vụ cho việc phân tích, ta tiền xử lý dữ liệu.

2.1. Transform

`App` là tên của ứng không mang nhiều ý nghĩa nên ta sẽ loại bỏ

Installs (lượt cài đặt) và **Reviews** (Bình luận) chuẩn hóa về kiểu numeric.

Genres (thể loại) ta sẽ đếm lấy số loại của App

Android.Ver và **Current**.**Ver** Ta sẽ lấy chữ số đầu của Version (hoặc bản cập nhật).

Price loại bỏ symbols "$" và **Size** (kích thước ứng dụng) ta chuyển về cùng đơn vị kilobyte.

Last.**Updated** (ngày vập nhật cuối): ta sẽ tách thành **Days** (số ngày lần cuối cập nhật hoặc thêm), **Month**.**Added**, **Year**.**Added** (lần lượt là tháng và năm mà ứng dụng được thêm hoặc cập nhật).

Sau khi xử lý ta thu được dữ liệu ở định dạng sau:

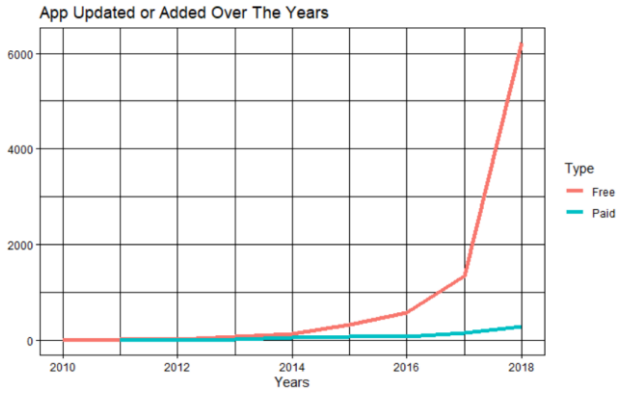
Biến liên tục: Rating, Price, Size, Min.Android Ver
Biến phân loại: Category, Type, Content.Rating
Biến rời rạc: Reviews, Installs, Month.Added, Year.Added, Days, Genres, Current Ver

2.2. Missing Value

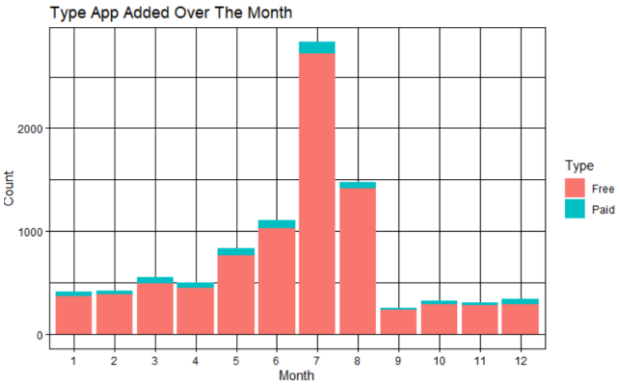
Các trường hợp NA trong tập dữ liệu ta chủ yếu tập trung vào thuộc tính Rating có 1474 trường hợp. để đơn giản ta lựa chọn giải pháp bỏ hết chúng.

3. DATA VISULIZATION

Cửa hàng trực tuyến Google Play Store ra mắt với hình thức miễn phí vào năm 2008 (2009 với ứng dụng trả phí), số lượng ứng dụng tăng dần qua các năm và **bùng nổ kể từ 2016**. Nhưng ứng dụng trả phí có lẽ không mang lại nhiều lợi ích cho nhà phát hành, ta thấy số lượng các ứng dụng trả phí phát triển rất là khiêm tốn.

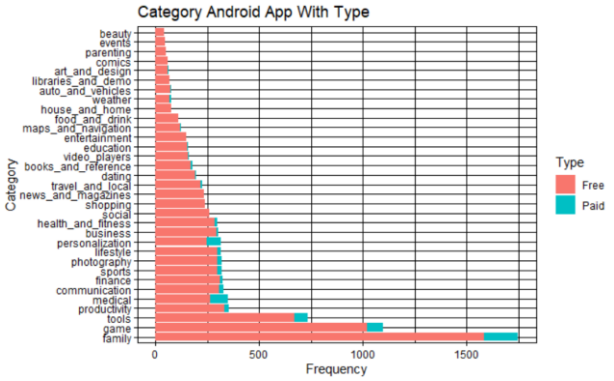


Thêm 1 điều khá thú vị nữa**, tháng 7 và 8 là tháng yêu thích** của các nhà phát triển ứng dụng, có vẻ đây là 1 thời điểm kỳ nghỉ giữa năm thích hợp để ra mắt (hoặc cập nhật) ứng dụng.

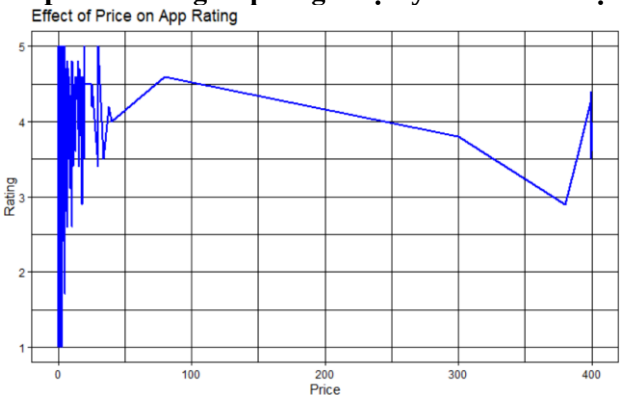


Tuy nhiên khi số lượng Apps tăng đáng kể, thì để trở nên được phổ biến đòi hỏi phải có sự khác biệt. Hàng loạt các danh mục ứng dụng đa dạng nổi lên với mục đích làm hài lòng người dùng. Trong 3 ứng dụng nhiều nhất được phát hành trên cửa hàng ta thấy nó thuộc về family, game và tools. Liệu có phải những mục này được người dùng yêu thích nên được các nhà phát triển mới phát hành nhiều ứng dụng như vậy.

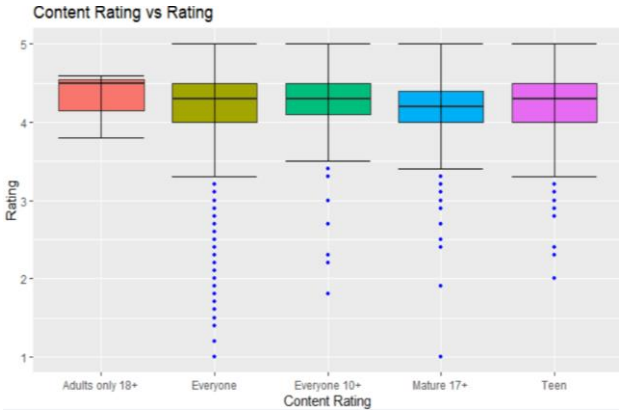
Nhưng **`Rating` trung bình không khác biệt nhiều.** Có sự khác biệt không đến từ các danh mục.



Các ứng dụng trả phí cũng là một định hướng cho các nhà phát triển. Nhưng cần cân nhắc, vì có vẻ như các Apps với mức **giá càng cao** thì mong muốn từ người dùng cũng sẽ tăng và **`Rating` sẽ thấp nếu không đáp ứng được yêu cầu của họ**.

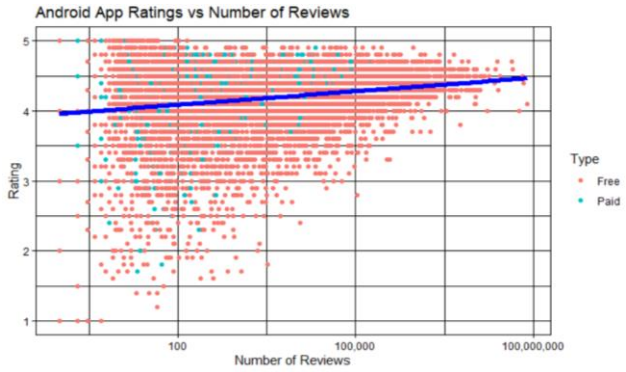


Những yếu tố khác như nội dung của ứng dung không ảnh hưởng quá nhiều đến `Rating`, nhưng **ứng dụng 17+** có vẻ như chúng **nhận được `Rating` thấp hơn** các ứng dụng khác.

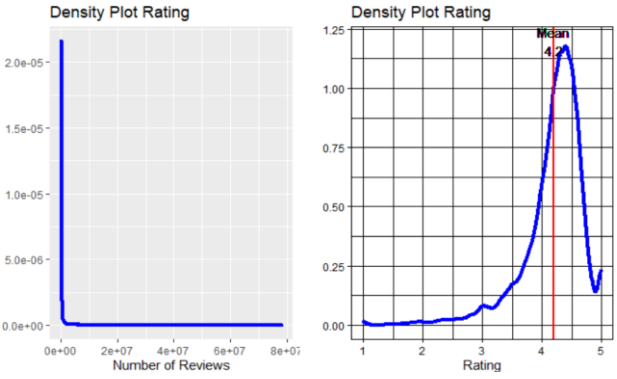


Số lượng `Reviews` là yếu tố quan trọng ảnh hưởng tới tối ưu hóa app và quyết định tải ứng dụng của người dùng. Và 1 ứng dụng có nhiều `Reviews` cũng giúp cho app đó trở nên hoàn thiện và có thể sẽ nhận được nhiều đánh giá tích

cực hơn trong tương lai. Có thể thấy lượng **`Reviews`** càng cao thì **`Rating`** cũng tăng.



Một điều cũng cần lưu ý, phân bố của Reviews khá là lệch, có lẽ ta cũng nên điều chỉnh lại ở phần sau:



4. DATA MODELING

Sau khi xem phân tích dữ, ta thấy phân bố của chúng khá lệch so với phân bố chuẩn, ta cần chuẩn hóa trước khi đưa vào mô hình.

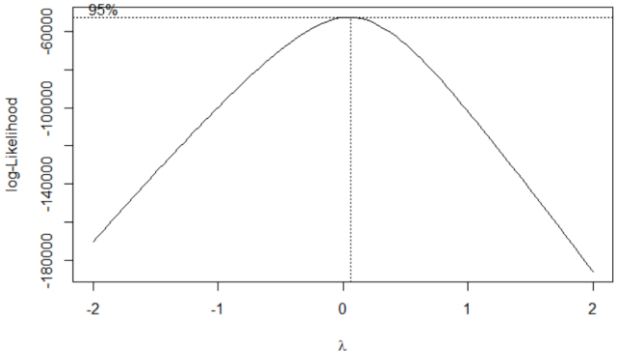
Ta sẽ sử dụng phương pháp hoán chuyển box-cox để xác định hàm hoán chuyển phù hợp. Phương pháp **box-cox có mục đích là hoán chuyển dữ liệu thành phân bố chuẩn**. Phương pháp này có thể tóm tắt qua phương trình sau đây tìm hàm g(y) với:

$$\begin{cases} \frac{y^{\lambda}-1}{\lambda} & \text{nếu } \lambda \neq 0 \quad [1] \\ \log(y) & \text{nếu } \lambda = 0 \quad [2] \end{cases}$$

Tham số λ được tối đa hóa qua log-likelihood:

$$L(\lambda) = \frac{n}{2} log(\frac{RSS}{n}) + (\lambda - 1) \sum log(y_i) \qquad [3]$$

Từ [3] ta thu được kết quả λ sau:



Từ đồ thị trên ta thấy giá trị $\boldsymbol{\lambda}$ tối ưu của [3]:

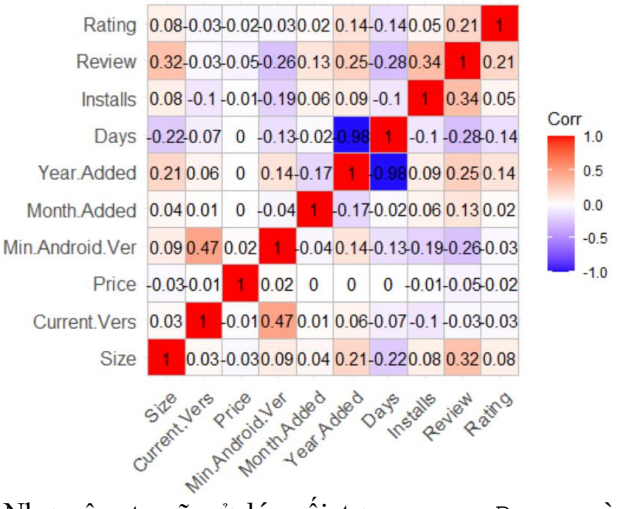
$\boldsymbol{\lambda}$ = **0.06** tương ứng [1] ➔ $\frac{y^{0.06}-1}{0.06}$

Sau khi chuẩn hóa dữ liệu, các giả định hiện tại khá phù hợp để xây dựng **mô hình hồi quy tuyến tính**:

$$y = \alpha + \beta x_i + \varepsilon$$

Mô hình này cho rằng các biến có mối quan hệ tuyến tính với nhau. Để đánh giá mô hình hồi quy tuyến tính có đại diện cho dữ liệu, chúng ta sử dụng **thang đo R^2 và RMSE** (root mean squared error). Mô hình có R^2 cao có nghĩa là mô hình giải thích được nhiều phương sai, và giảm độ bất định, nên RMSE sẽ thấp. Và ngược lại R^2 thấp.

Ngoài ra, những thuộc tính có mối tương quan mạnh với nhau sẽ phát sinh vấn đề **đa cộng tuyến** khi xây dựng mô hình hồi quy đa biến. Ở ma trận tương quan này ta thấy có 2 mối liên hệ khá chặt:



Như vậy, ta sẽ xử lý mối tương quan `Days` và `Year.Added` bằng cách tìm một **hàm số** có thể **tóm tắt môi liên quan giữa 2 biến** và dùng biến số

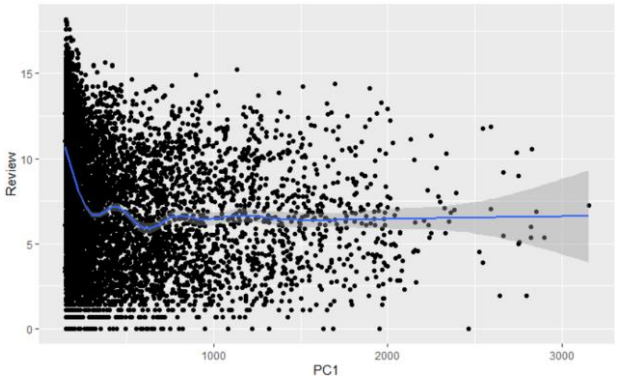
này cho mô hình hồi quy tuyến tính. Chẳng hạn x_1 và x_2 có mối tương quan chặt chẽ, ta sẽ tìm 1 hàm tuyến tính (còn gọi principal component hay PC):

$$PC = a_1 x_1 + a_2 x_2 \qquad [4]$$

```
                PC1           PC2
Days       0.999996269 -0.002731588
Year.Added -0.002731588 -0.999996269
Importance of components:
                            PC1    PC2
Standard deviation     395.9 0.2084
Proportion of Variance   1.0 0.0000
Cumulative Proportion    1.0 1.0000
```

Ta thấy biến PC1 giải thích 100% độ dao động của 2 biến và PC2 là 0%. Do đó, ta dùng PC1 để mô tả: [4] ➔ PC = `Days` - 0.0027*`Year.Added`

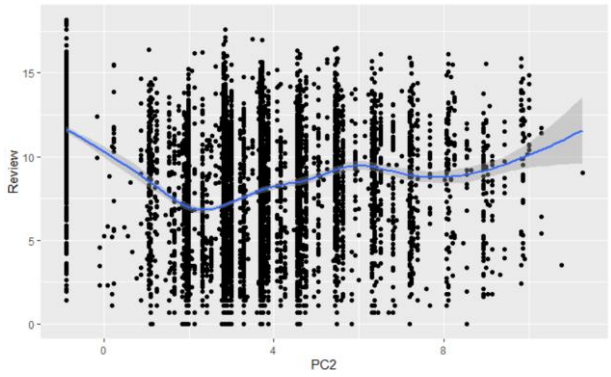
Xem qua đồ thị này ta thấy nó khá giống **phương trình bậc 5** (có 4 cực trị)**.** Không chắc chắn nhưng ta sẽ thử nghiệm nó trong mô hình:



Tương tự các bước trên `Min.Android.Ver` và `Current.Vers` ta có phương trình:

PC=0.4823*Min.Android.Ver+0.876*Current.Ver

Ta cũng xem nó như **hàm bậc 4** (có 3 cực trị):



Như vậy, xem như ta đã có một số thuộc tính để xây dựng mô hình hồi quy.

Ngoài mô hình trên, các bước chuẩn hóa phía trên cũng khá phù hợp để xây xây dựng **mô hình k-NN** (k Nearest Neighboor). KNN là thuật toán đi tìm đầu ra của một điểm dữ liệu mới bằng cách chỉ dựa trên thông tin của K điểm dữ liệu trong training set gần nó nhất.

5. BUILD MODEL

Để xây dựng mô hình hồi quy tuyến tính, ta sẽ phân vùng tập dữ liệu thành tập training (80%) và validation (20%). Sau khi phân vùng dữ liệu ta sẽ xây dựng mô hình trên *training set*. Và đây là kết quả khi ta chưa biến đổi Box – Cox cũng như chưa xử lý đa cộng tuyến. Như ta thấy R^2 vẫn khá là thấp chỉ giải thích 43% kết quả, và đa số các thuộc tính không có ý nghĩa thống kê (p – value > 0.05) :

```
Call:
lm(formula = Reviews ~ . - PC1 - PC2, data = train_data)

Residuals:
      Min        1Q    Median        3Q       Max
-20721292   -241408    -25134    137909  54738810

Residual standard error: 2385000 on 7442 degrees of freedom
Multiple R-squared:  0.4305,    Adjusted R-squared:  0.4269
F-statistic: 119.7 on 47 and 7442 DF,  p-value: < 2.2e-16
```

Khi hoán chuyển Box – Cox và xử lý đa cộng tuyến, kết quả có chút cải thiện nhưng không nhiều. R^2 tăng nhẹ 8% nhưng hầu hết các thuộc tính đều có ý nghĩa thống kê và độ chính xác vẫn thấp. Điều này có thể do **mô hình** mà chúng ta **xây dựng quá đơn giản.**

```
Call:
lm(formula = Review ~ Category + Type + Content.Rating + Genres +
    Month.Added + Size + Installs + Rating + Price + poly(PC2,
    4) + poly(PC1, 5), data = train_data)
Residual standard error: 4.582 on 7437 degrees of freedom
Multiple R-squared:  0.5128,    Adjusted R-squared:  0.5094
F-statistic: 150.5 on 52 and 7437 DF,  p-value: < 2.2e-16
```

Vì kết quả trên *training set* khá thấp nên ta sẽ không triển khai thuật toán nà đối với bài toán này.

Tiếp theo, ta sẽ thử mô hình K-NN như đã đề cập ở trên. Đối với mô hình K-NN ta sẽ có thêm bước chọn siêu tham số (hyper–parameters). Và ta cũng tiếp tục sử dụng các tham số của mô hình trước cho mô hình KNN.

Nhưng trước tiên, tương tự như mô hình trên ta sẽ xây dựng mô hình trên tập train kết quả như sau:

```
7490 samples
  11 predictor

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 7490, 7490, 7490, 7490, 7490, 7490, ...
Resampling results across tuning parameters:

  k    RMSE      Rsquared   MAE
    1  2.180019  0.8925188  1.555187
   10  1.826583  0.9224041  1.398552
  100  1.806202  0.9239532  1.390404
  200  1.847543  0.9205612  1.415724

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was k = 100.
```

Kết quả khá cao khi ta áp dụng một thuật toán phức tạp hơn. Trên training set ta thấy kết quả tốt nhất là k = 100, R^2 = 92% tức giải thích được 92% phương sai của mô hình, khá là chính xác. Ta cũng áp dụng mô hình dự đoán trên vào tập validation:

```
k-Nearest Neighbors

1873 samples
  11 predictor

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 1873, 1873, 1873, 1873, 1873, 1873, ...
Resampling results across tuning parameters:

  k    RMSE      Rsquared   MAE
    1  2.055329  0.8715644  1.590156
   10  1.668224  0.9129406  1.304717
  100  1.722683  0.9093814  1.328009
  200  1.854365  0.9021913  1.418681

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was k = 10.
```

Ta thấy trên tập validation mô hình cho hệ số R^2 tốt nhất. Xét mô hình trên tập train ứng với hệ số k ta sử dụng cho mô hình này thì chúng khá tương đồng với nhau. Vậy nên ta chấp nhận dùng mô hình hiện tại để tiến hành dự đoán:

Review <dbl>	pre <dbl>
1.892664	2.044164
2.580891	3.022618
4.729971	4.309642
3.856596	4.133452
11.450973	12.088976
1.690547	2.023952

Từ mô hình trên chúng ta thấy được kết quả dự đoán tương đối tốt.

6. CONCLUSION

Sau khi thực nghiệm 2 mô hình khác nhau ta thấy thuật toán K-NN cho một kết quả tốt hơn rất nhiều so với linear regression. Ta có thể thể rút ra rằng, bài toán trên khá phức tạp so với giả định tuyến tính đơn giản. Nếu có thể chúng ta có thể thử nhiều mô hình khác nhau, để tìm mô hình tốt nhất cho bài toán.

7. APPENDICES

Phụ lục 1: Dữ liệu thô

```
Rows: 10,841
Columns: 13
$ App            <chr> "Photo Editor & Candy Camera & Grid & ScrapBook", "Colo…
$ Category       <chr> "ART_AND_DESIGN", "ART_AND_DESIGN", "ART_AND_DESIGN", "…
$ Rating         <dbl> 4.1, 3.9, 4.7, 4.5, 4.3, 4.4, 3.8, 4.1, 4.4, 4.7, 4.4, …
$ Reviews        <chr> "159", "967", "87510", "215644", "967", "167", "178", "…
$ Size           <chr> "19M", "14M", "8.7M", "25M", "2.8M", "5.6M", "19M", "29…
$ Installs       <chr> "10,000+", "500,000+", "5,000,000+", "50,000,000+", "10…
$ Type           <chr> "Free", "Free", "Free", "Free", "Free", "Free", "Free",…
$ Price          <chr> "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", …
$ Content.Rating <chr> "Everyone", "Everyone", "Everyone", "Teen", "Everyone",…
$ Genres         <chr> "Art & Design", "Art & Design;Pretend Play", "Art & Des…
$ Last.Updated   <chr> "January 7, 2018", "January 15, 2018", "August 1, 2018"…
$ Current.Ver    <chr> "1.0.0", "2.0.0", "1.2.4", "Varies with device", "1.1",…
$ Android.Ver    <chr> "4.0.3 and up", "4.0.3 and up", "4.0.3 and up", "4.2 an…
```

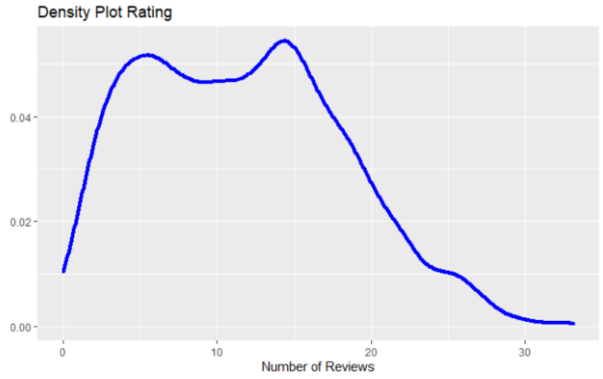
Phụ lục 2: Giá trị NA

```
     Rating          Reviews             Size             Installs          Type
 Min.   : 1.000   Length:10841       Length:10841       Length:10841       Length:10841
 1st Qu.: 4.000   Class :character   Class :character   Class :character   Class :character
 Median : 4.300   Mode  :character   Mode  :character   Mode  :character   Mode  :character
 Mean   : 4.193
 3rd Qu.: 4.500
 Max.   :19.000
 NA's   :1474
    Genres          Last.Updated       Current.Ver        Android.Ver
 Length:10841       Length:10841       Length:10841       Length:10841
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

Phụ lục 3: Sau khi làm sạch và biến đổi dữ liệu

```
Rows: 9,363
Columns: 14
$ Category        <chr> "art_and_design", "art_and_design", "art_and_design", …
$ Rating          <dbl> 4.1, 3.9, 4.7, 4.5, 4.3, 4.4, 3.8, 4.1, 4.4, 4.7, 4.4,…
$ Reviews         <dbl> 159, 967, 87510, 215644, 967, 167, 178, 36815, 13791, …
$ Size            <dbl> 19456.0, 14336.0, 8908.8, 25600.0, 2867.2, 5734.4, 194…
$ Installs        <dbl> 1e+04, 5e+05, 5e+06, 5e+07, 1e+05, 5e+04, 5e+04, 1e+06…
$ Type            <chr> "Free", "Free", "Free", "Free", "Free", "Free", "Free"…
$ Price           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
$ Content.Rating  <chr> "Everyone", "Everyone", "Everyone", "Teen", "Everyone"…
$ Genres          <dbl> 1, 2, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, …
$ Min.Android.Ver <dbl> 4.0, 4.0, 4.0, 4.2, 4.4, 2.3, 4.0, 4.2, 3.0, 4.0, 4.1,…
$ Current.Vers    <dbl> 1, 2, 1, -1, 1, 1, 1, 6, 2, 2, 1, 1, 3, 1, 1, -1, 3, 1…
$ Year.Added      <dbl> 2018, 2018, 2018, 2018, 2018, 2017, 2018, 2018, 2017, …
$ Month.Added     <dbl> 1, 1, 8, 6, 6, 3, 4, 6, 9, 7, 10, 7, 4, 6, 8, 6, 7, 11…
$ Days            <dbl> 358, 350, 152, 206, 194, 645, 249, 200, 467, 181, 430,…
```

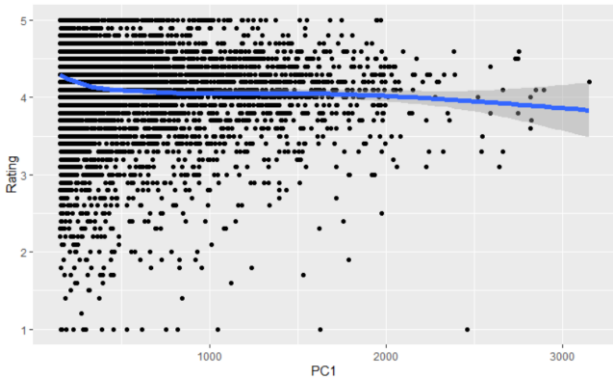
Phụ lục 4: Sau khi hoán chuyển Box – Cox cho Reviews



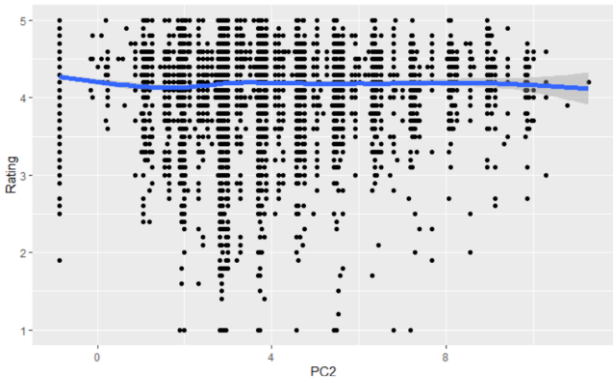
Phục lục 5: hàm tóm tắt `Current.Vers` và `Min.Android.Ver`

```
                          PC1        PC2
Min.Android.Ver 0.4823026  0.8760047
Current.Vers    0.8760047 -0.4823026
Importance of components:
                            PC1    PC2
Standard deviation     2.2835 1.2568
Proportion of Variance 0.7675 0.2325
Cumulative Proportion  0.7675 1.0000
```

Phụ lục 6: Đồ thị hàm tóm tắt tương quan `Days` và `Year.Added` đối với `Rating`



Phụ lục 7: Đồ thị của hàm tóm tắt tương quan `Current.Vers` và `Min.Android.Ver` đối với `Rating`



Phụ lục 8: Huấn luyện mô hình hồi quy tuyến tính dự đoán Rating:

```
lm(formula = Rating ~ Category + Content.Rating + Type + Genres +
    Month.Added + Size + Installs + Review + Price + poly(PC2,
    3) + poly(PC1, 3), data = train_data)
Residual standard error: 0.4914 on 7440 degrees of freedom
Multiple R-squared:  0.1006,    Adjusted R-squared:  0.09466
F-statistic: 16.98 on 49 and 7440 DF,  p-value: < 2.2e-16
```

Phụ lục 9: Huấn luyện mô hình K-NN dự đoán Rating

```
k-Nearest Neighbors

7490 samples
  11 predictor

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 7490, 7490, 7490, 7490, 7490, 7490, ...
Resampling results across tuning parameters:

  k    RMSE       Rsquared    MAE
    1  0.6762018  0.01940146  0.4338237
   10  0.5383662  0.03001116  0.3705538
  100  0.5113882  0.04126658  0.3564659
  200  0.5116351  0.03829478  0.3578461

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was k = 100.
```

CONTRIBUTIONS

MSSV	STT	NAME	MAIN	PROGRESS
19133065	61	Nguyễn Thanh Tùng	Intro, Data	100%
20133036	18	Nguyễn Thành Đồng	Visualization	100%
19133022	20	Hồng Tiến Hào	Visualization Data Modeling	100%
18133024	31	Ngô Phi Lít	Build Model	100%

REFERENCES

[1] Nguyen, Van Tuan. *Mô Hình Hồi Quy Và Khám Phá Khoa Học*. Nhà xuất bản tổng hợp thành phố Hồ Chí Minh, 2020. Printed book

[2] Hadley Wickham, Danielle Navarro, and Thomas Lin Pedersen. *ggplot2: elegant graphics for data analysis*. Nhà xuất bản Springer 2020. Ebook from https://ggplot2-book.org

[3] Quach, Dinh Hoang. Lecture, 2021. Slide and video

[4] Chang, Winston. *R Graphic Cookbook 2nd*. NHÀ XUẤT BẢN O'reilly, 2021. EBOOK from https://r-graphics.org/