

Data Science II midterm project report

Group 5 members: Eleanor Zhang(zz2602), Mengran Ma (mm5354), Bingyu Sun (bs3142)

Introduction

Motivation

Mobile applications have been on the rise among smartphone users in the past decade. Two business tycoons, Android and iOS, hold over 95 percent of the market share. This dataset contains more than 7000 Apple iOS top trending mobile apps details in 2017. The data was extracted and cleaned by Ramanathan on Kaggle from the iTunes Search API at the Apple Inc website (1). Therefore, this data set is not liable to missing detailed information on each app. There are 16 variables for each 7197 mobile app in the raw dataset. We will explore and select a subset variables to implement statistical analysis.

After preliminary exploratory analysis, we fitted linear and nonlinear model in order to investigate which covariates are associated with the customer rating of the mobile app and which prediction model fit data best.

Data preparation and cleaning

The response variable is average user rating for all app versions on a scale from 1 to 5, which is highly discretized into values such as 1, 1.5, 2, etc. Although it might not be the best option to treat them as continuous variables, our data analysis treated them as continuous variable. We focused on 9 covariates including size in megabytes, price, average user rating for current version, content rating (categorical), primary genre (categorical), number of supporting devices, number of screenshots shown for display, number of supported languages, and Vpp Device Based Licensing Enabled (binary). After checking linear correlation and independency, we removed the binary variable about Vpp Device License because one of the two levels is overly sampled, making this variable as near zero variance. To limit user rating to those apps actually received scores, we removed apps without any user rating by filtering out zero total rating count. In addition, one modification of our predictors is that we made the primary genre into a binary variable. Originally, primary genre is a categorical variable with 23 different categories with “game” having the highest counts. We combined categories except game as “non-game”. Finally, since we have sufficient observations, we partitioned data into training and testing sets by 75/25. The splitting of train and test dataset is for model selection and evaluating selected model performance on “new” predictor observations.

Exploratory data analysis and visualization

Exploratory analysis is based on the entire dataset.


Firstly, we draw a feature plot (Figure. 1) and we see that little correlation between user_rating and size bytes, price and vpp licence. The only high correlation is between user rating of current version with user rating for all versions, which is not unexpected. Moderate correlation revealed between number of screenshots displayed and language number supported. The correlation between two categorical variables, primary genre and content rating, are illustrated in the boxplots. Apps that in the top ranks fall into the category of Productivity, Music, Photos, Business, Health&Fitness, Games, weather and shopping. Content rating of “17+” have a wide range of user rating, so this category technically contains no information on the content rating effect on app rating. Outliers of user rating are present in both variables. Games and Health&Fitness are overly sampled in the dataset as well as in each content rating level.


Furthermore, a correlation plot (Figure. 2) to show any correlation pattern between predictors. And we observe certain negative correlations between content rating “4+” level versus both “9+” and “17+” levels, and some positive correlations between primary genre versus size, average user rating value (for current version), content rating “9+”, and number of screenshots for display. However, all the above correlations are relatively moderate and even weak, therefore we would say we do not see strong correlations between covariates.

Methods

For all models, we used training data to implement feature selection and model selection. To select the best tuning parameter for ridge, lasso, elastic net, pca, pls, and mars, 10-fold cross-validation (cv) with 5 repeats was used.

Ridge, Lasso, and elastic net regression (shrinkage methods)

In order to get insight on the linear relationship between each covariate and user rating (outcome), we fitted a linear model using all eight predictors. User rating for current app version, Genre of Games, number of screenshots displayed and number of languages supported may have relatively strong association with user rating of all version. However, the linear model is not a good fit in terms of a R-squared of 0.25, consistent with our exploratory analysis that showed correlation between covariates and non-linear trend. 

Since there is correlation among our predictors, we can use either ridge, lasso, or elastic net to control variance. The optimal tuning parameters λ selected by repeated-cv are 0.03, 0.005, and 0.004 for ridge, lasso, and elastic net respectively. All three models are relatively flexible because of small lambda. Elastic net selected alpha equal to 1, meaning lasso is used in elastic net model. To select the best model, both ridge and lasso have the same mean RMSE, revealing they possess similar model fitting ability. We decide to focus on lasso result because it imposes constraints on model fitting that leads to variable selection. All variables except the dummy variable content rating of 9+ are selected by lasso, indicating content rating of 9+ is probably highly correlated to some covariates. In terms of predictive ability, both ridge and lasso have similar small test MSE, suggesting both models have no overfitting problem. 

PCR & PLS

Similar to ridge, lasso, and elastic net, PCR and PLS are useful when a large number of predictors are correlated. Given that our covariates have small or no correlation, we did not expect PCR and PLS to improve the model fitting or prediction. Both PCR and PLS have similar mean RMSE and test MSE compare to shrinkage methods respectively, confirming that there is no improvement using derived input methods.

Polynomial for single covariate

From the scatterplot (Figure.1), fitting linear model on each covariate is not sufficient to capture their relationship and make prediction on outcome. Thereby, we added more flexibility in the model. At this point to maintain interpretability of the model, we added nonlinear nature on three variables. Number of languages supported, size megabytes and user rating of current app version may be in nonlinear relationship with app user rating. The tuning parameter d is the highest degree on the predictor. Choice of d from 1 to 4 on each covariate was determined by 10-fold cross-validation. The RMSE comparison shows $d = 1$ on size

megabytes, $d = 4$ on user rating on current version and $d = 1$ on number of languages supported. High degree on covariates is bad for interpretation because we cannot fix all other covariates at the same time. For the purpose of understanding the association between covariates and user rating, we will keep all variables on the first degree. Another way to extend the linear form of prediction function is to use spline methods.

Generalized Additive Model (GAM)

We primarily used smoothing splines because it does not require specifying degree of freedom and the placements of knots, which we did have prior knowledge of. Therefore smoothing spline method with one tuning parameter is desired for our analysis. Additionally, the penalty term in the model will alleviate overfitting issues that could possibly arise when selecting for the unknown smooth function. The result is equivalent to natural spline method placing knots at every unique values of covariates. The main drawback of this increasingly flexible model is that it is not convenient to interpret the relationship between covariates and app user rating. Thus in this section, we focused on the prediction performance of model.

As we have seen before, there is nonlinear nature between number of languages supported, user rating of current version and size bytes. We incorporated these smoothing features into the generalized additive model using *mgcv* package in R and explore what works best for the data. ANOVA test is used to compare nested models of linear regression, generalized linear regression with adding one smoothing component at a time out of those three covariates. The result shows we should include all three smoothing features in the model. In Figure.3, the smoothing component has trumpet shape in credential region toward increasing value of size byte and language supported. The reason is very likely due to the sparse response data towards the high end of covariates, such that the variation of outcome is magnified toward the right tail region. The smoothing component of user rating for current app version remains stable in U shape.

MARS

Now we further relaxed on the functional form of prediction function with GAM. Multivariate Adaptive Regression Splines (MARS) was used for fitting model with and without interactions between covariates. We set the prune parameter number from 2 to 11 (maximum number of covariate component in linear model). For the purpose of prediction, the cross validation result demonstrates 11 components without interaction terms (product degree = 1) leads to the minimum RMSE, which also outperforms model with interaction (product degree = 2). The final MARS model is a combination of piecewise regression using new features of variables in the form of hinge functions, including user rating for current version, genre of games, number of languages supported, size of megabytes and price.

Model Selection and Prediction Performance

Based on cross validation result of train data (Figure.5), MARS model has obvious improvement on reducing prediction error at least on train data. Therefore, we chose the MARS model as our prediction model for user rating. Considering that increasing flexibility of model will always reduce train MSE, we presented the prediction error on test data (Table 1). As expected, train error will generally underestimate the test error. Nonetheless, the train error and test error are at least consistent. Thus these regression models did not overfit the data. So our final model is the stated following:

$$E(\text{user_rating}) = 3 + 0.28 h(\text{user_rating_ver} - 1) + 1.35 h(1 - \text{user_rating_ver}) + 0.12 \text{prime_genre} - 0.3 h(\text{user_rating_ver} - 4.5) + 0.35 h(\text{user_rating_ver} - 2) - 0.0033 h(30 - \text{lang_num}) + 0.005 h(\text{size_bytes} - 118) - 0.005 h(118 - \text{size_bytes}) - 0.005 h(\text{size_bytes} - 44) - 0.015 h(5.99 - \text{price})$$

Conclusions

Feature selection via lasso and elastic net shows that user rating for current app version, genre of Games, number of screenshots displayed and number of languages supported may have relatively strong association with user rating of all version. Intuitively speaking, more visual display and games app will be more likely to receive higher rating. More languages available in the app allow more diverse customer population. For the purpose of prediction, MARS final model is generated by the algorithm of recursive partitioning and automatic feature selection supported by generalized cross validation. Comparing the RMSE across all methods we mentioned above, MARS model could provide significant improvement on prediction while maintaining interpretability of the model. The final model suggests the important variable for prediction should include user rating for current version, genre of games, number of languages supported, size of megabytes and price. This is consistent with our expectation. These covariates except genre are partitioned into different segment with hinge functions in the MARS final model. Along with the scatter plot we observed in Figure.1, those covariates does not provide too much predictive capability on the outcome. The binary variable of genre being Games has positive coefficients 0.12, suggesting that Games are associated with higher user rating. The association between each of the rest four variables with overall user rating is conditional on their respective values.

Reference

1. Data website: <https://www.kaggle.com/ramamet4/app-store-apple-data-set-10k-apps>

Supplementary

Predictor pool:

- **Size megabytes**: To improve computation efficiency, we changed the units of size bytes into megabytes; 96% apps take less than 1000 mb.
- **Price**: 99% apps cost less than \$10; some apps requires over hundreds of dollars which might be annual subscription fee.
- **User rating for all version** (outcome variable): user rating on the scale from 1 to 5; “0” means the app is not rated yet (removed for model fitting)
- **User rating for current app version**
- **Number of supporting device**
- **Number of screenshots shown for display**
- **Number of supported languages**: range from 0 to 75 with mean 5.
- **Vpp Device Based Licensing Enabled**: binary (removed due to near zero variance)
- **Content rating**: tells the age group suitable for use with 4 levels.
- **Primary genre**: it includes 23 types of app; Games genre is oversampled so changed to binary variable of being Games or not.

Figure 1. Feature Plot of Response versus Predictors

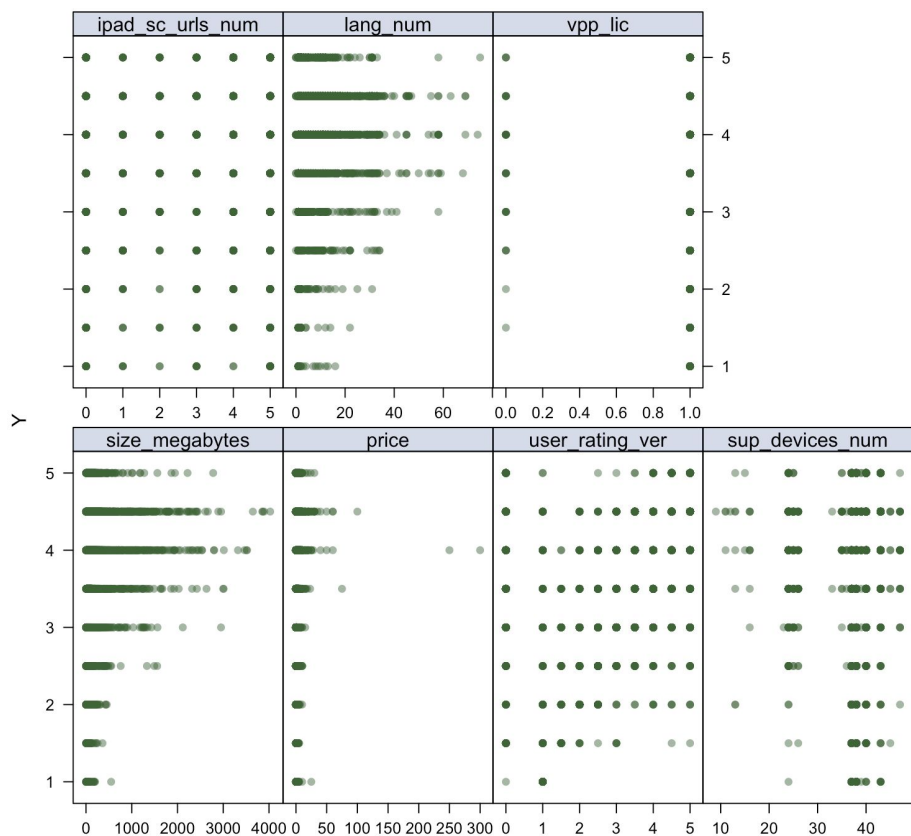


Figure 2. Correlation Plot of predictors

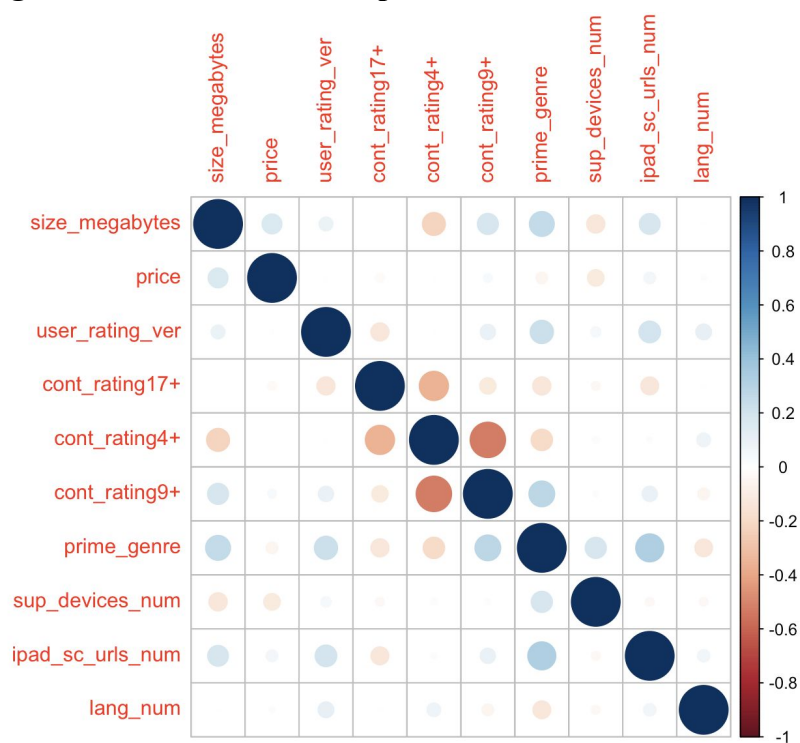


Figure 3. Smoothing spline component in GAM

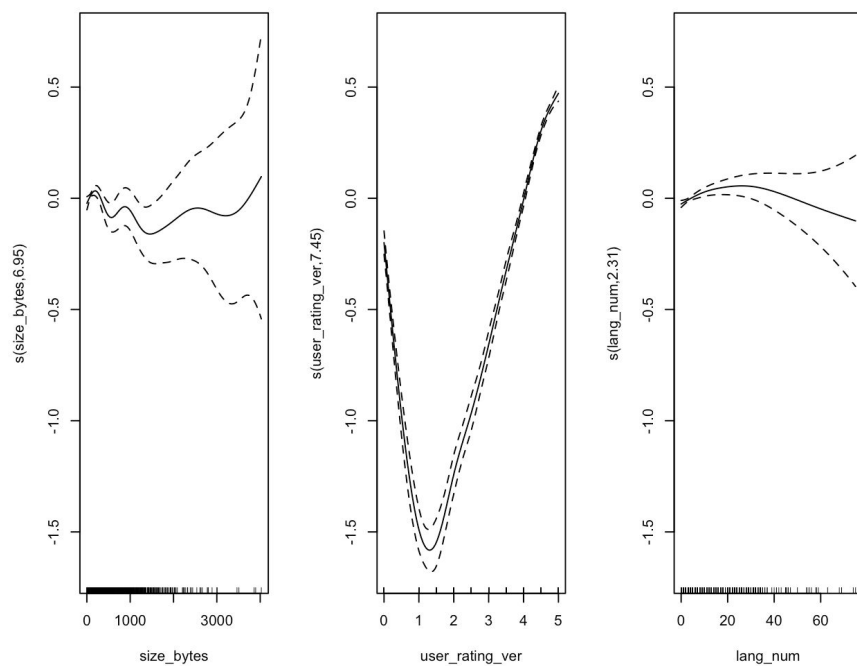
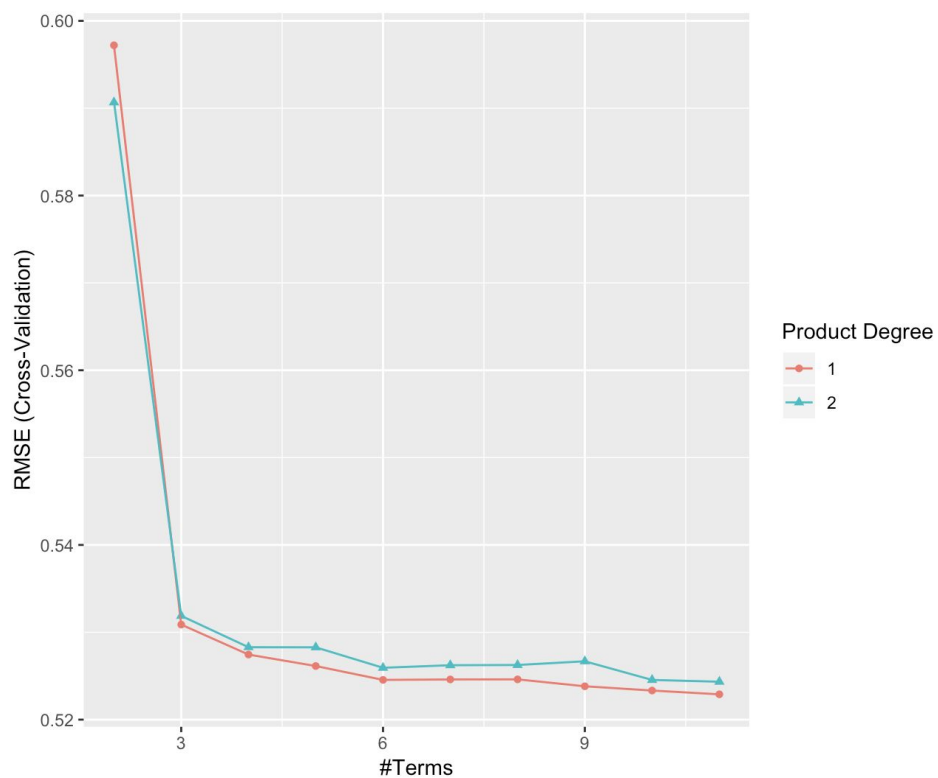


Figure 4. MARS Cross Validation result of RMSE with product degree 1 and 2



Product degree 1 means no interaction, product degree 2 has interaction added to the model.

Figure 5. Boxplot comparing training RMSEs between different models

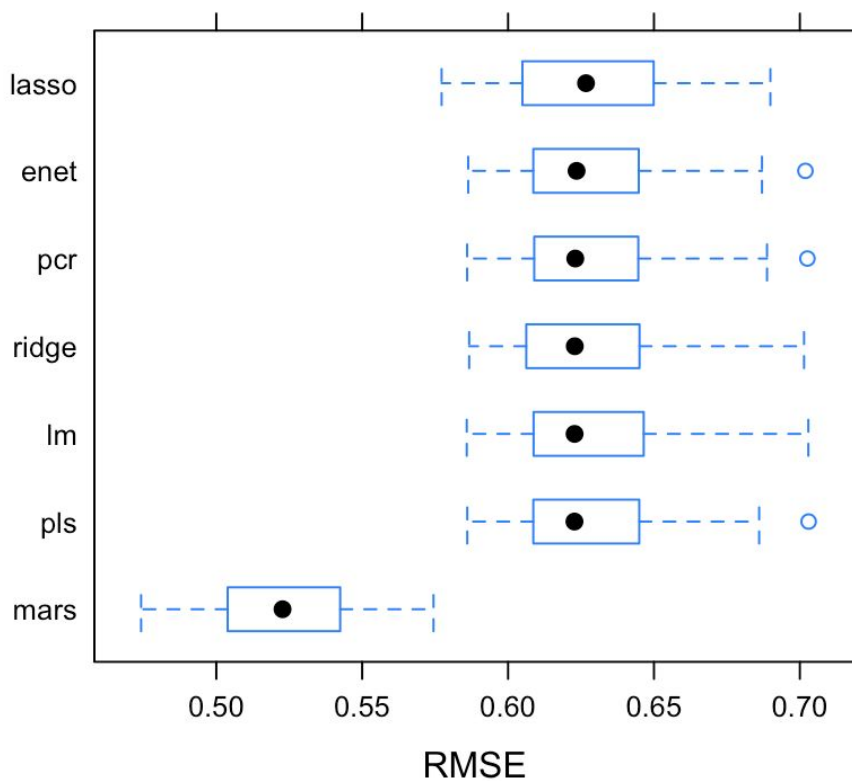


Table 1. Summary of testing MSEs and training MSEs for different models

	Test MSE (Mean Square Error)	Train MSE (mean of CV MSE)
Linear model	0.407	0.396
Lasso	0.407	0.396
Ridge	0.406	0.396
Elastic Net	0.407	0.396
PLS	0.407	0.395
PCR	0.407	0.395
MARS	0.283	0.273