AgData 621 Final Project: Churn Modeling

CUNY SPS MSDS

Professor Nasarin

Group 4: Chi Pong, Euclid Zhang, Jie Zou, Joseph Connolly, Leticia Cancel

Spring 2022

Table of Contents

Abstract

Today, we see many subscription services that offer a wide variety of services to consumers. Within minutes, one can sign up for a subscription regarding grocery delivery, toiletries, supplements, therapy, prepared meals, boxes of random trinkets, news, perfumes, clothes, all sorts of media, and educational services. The most known of all to consumers is media streaming services; particularly, film and music streaming.

These services usually offer free and paid subscriptions, the latter of which a customer is typically charged monthly, and the former a customer has to endure advertisements. If all customers were committed to staying subscribed to a service, particularly of a "paid" tier, then these services could be more valuable than gold and oil combined. However, because people are unpredictable, not necessarily always easy to satisfy, and may be faced with external uncontrollable factors, customer churn can potentially become a problem.

In this paper, we wish to investigate customer churn within the Sparkify data set, which was made by Udacity as part of their data science course. This data set is a fictitious music streaming service that mimics Spotify data. Applying our skills and the knowledge we've obtained and learned in our MSDS program and this class, we will build a model to predict customer churn.

Key Words

Customer Retention, Music Streaming Service, Churn Model, Machine Learning, Modeling, Data Mining, Exploratory Data Analysis

Introduction

Churn analysis is a useful tool for companies that offer subscription services to review customer behavior and predict customer retention. Subscription cancellation can be either voluntary or involuntary. Involuntary cancellations are sometimes due to factors outside of the company's control such as customer job loss or customer geography. If a customer moves to an area where the service is not provided, they are then forced to cancel their subscription and explore the competitors. Voluntary cancellations are attributed to customer dissatisfaction. Therefore, it's imperative companies maintain customer happiness, thus increasing brand loyalty, thereby reducing voluntary cancellation. We are going to investigate different applications of churn analysis in the literature review section.

Literature review

In an article published by Steven Norton in the Wall Street Journal, he writes about the fitness company, Equinox, that built an AI model to predict customer churn. By doing this, the model assists "sales representatives to spot at-risk customers [who] intervene before they decide to leave." The data scientists at Equinox considered 70 features that contributed to customer churn, like the frequency of customers logging onto the customer portal, the frequency of orders placed, and an increase or decrease in power consumption. By building a deep neural network, the data scientists built a churn prediction model to determine the probability of customer churn over 30, 60, or 90 day periods. While the model was slow to gain traction and trust within the company, it eventually predicted churn with 90% accuracy within months of its launch. When giving predictions, the model identifies who is likely to churn and why. Now that sales representatives have the insight to customer churn, Equinox has seen a decrease in their cancellations. CIO of Equinox Milind Wagle claimed "...customers would churn because they didn't hear from reps…. that behavior is changing". [1]

In another business example, the Wall Street Journal itself is utilizing churn modeling on its subscribers. Anne Powell, the associate director of member engagement at Dow Jones, claims there is a direct correlation between churn and engagement. The key factor in their churn analysis was the average active days a customer engages in WSJ content. Utilizing and considering this discovery, they developed a strategy that effectively increased habit, and eventually the number of average active days.[2]

Researchers Latifah Almuqren, Fatma S. Alrayes, and Alexandra I. Cristea from Abdulrahman University in Riyadh, Saudi Arabia, identified an issue with most churn models where the data that is most readily available is historical customer satisfaction data which does not always match or help solve current issues. So they turned to Twitter data feeds to collect real-time customer satisfaction data. This presents a new challenge of text mining and transforming into usable data to upload into a model. Sentiment analysis was used to categorize tweets based on customer mood as Positive or Negative. The customer mood data combined with the structured customer data from the telecommunications company helped to create a predictive model for customer churn. The advantage of this method is that the collection of real time data via Twitter comes directly from the customer. The disadvantage is most people tend to only post

[1] Norton, S. (2018, April 2). *Equinix builds AI model to predict customer churn*. The Wall Street Journal. Retrieved May 15, 2022, from https://www.wsj.com/articles/equinix-builds-ai-model-to-predict-customer-churn-1522705279

[2] "How the WSJ Is Examining Engagement to Reduce Churn." *Talking Biz News*, 18 Dec. 2018, https://talkingbiznews.com/they-talk-biz-news/how-the-wsj-is-examining-engagement-to-reduce-churn/.

their opinions when they are dissatisfied and satisfied customers tend to be more silent, thus potentially creating skewed data. [3]

In this paper written by Sherendeep Kaur at the Asia Pacific University of Technology and Innovation in Kuala Lumpur, Malaysia, the “fuzzy algorithm” was used to clean "noisy data" analyzing telecommunications data for predicting customer churn. While this paper is quite advanced in its technique and methodology, it provides good insight into the available methods used for churn modeling, and provides a good perspective as to how to clean noisy data, which will be a problem we could be faced with in the future as data scientists. The conclusion does mention the use of social factors as a variable in a future model which could impact the accuracy rating like the idea of people only tweeting negative sentiment as exemplified above. [4]

In this churn analysis research paper published by Iowa State University, the researcher, known as Kriti, used data from IBM's “Using Customer Behavior Data to Improve Customer Retention” database. Python was used with the package FeatureTools to create new features for analysis which resulted in 724 features from the original 19 features. The xGBoost model had the best performance in identifying churned vs not churned and the top features for this classification. LIME prediction method (Local Interpretable Model-agnostic Explanation) was also used because it makes the model easier to interpret by non-experts. Three LIME prediction models were created to display churning, non-churning, and moderate customers in a way that is easier to digest than typical model visualizations. [5]

This churn analysis explored telecommunication customer data like the researchers from Riyadah, Saudi Arabia. The models used were Multilayer Perceptions (MLP) and Negative Correlation Learning (NCL). This analysis makes a great point in the challenge of churn analysis and the size of the churn population. The size of the non-churn group is usually much larger than the size of the churn group so that can impact the accuracy of the models. [6]

---

[3] Almuqren, Latifah, et al. “An Empirical Study on Customer Churn Behaviours Prediction Using Arabic Twitter Mining Approach.” *MDPI*, Multidisciplinary Digital Publishing Institute, 5 July 2021, https://www.mdpi.com/1999-5903/13/7/175/htm.

[4] *Literature Review of Data Mining Techniques in Customer Churn ...* https://jati.sites.apiit.edu.my/files/2018/07/2017_Issue2_Paper3.pdf.

[5] Kriti. “Customer Churn: A Study of Factors Affecting Customer Churn Using Machine Learning.” *Iowa State University*, Iowa State University, 2019, https://dr.lib.iastate.edu/server/api/core/bitstreams/963c8e0d-4209-4137-9d05-ac20968963f9/content.

[6] Rodan, Ali, et al. “Negative Correlation Learning for Customer Churn Prediction: A Comparison Study.” *TheScientificWorldJournal*, Hindawi Publishing Corporation, 2015, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4386545/.

In this article published by Victory Sharaf on towardsdatascience.com, she describes her experience in building a churn model on the full version of the Sparkify dataset, which is 12 GB in size. She found that her Logistic Regression model performed the best compared to Random Forest, Gradient Boosted Trees, and SVM methods, utilizing Python libraries. Due to our limited resources and processing power, we're using a smaller dataset of 148 MB in size. Like our model, she created a binary classifier to evaluate customer churn within Sparkify. [7]

Finally, to exemplify another real-life business churn strategy, we turn to this article published on Digiday.com, by Lucia Moses. Here, she details how the New York Times increased their approach in maintaining customers and preventing churn. While this article does not detail specifics and/or data science modeling, it does offer practical solutions that pertain to the business itself. In 2015, NYT hired Clay Fisher to be the SVP of Consumer Revenue, who developed a "retention-centric" approach in maintaining customers. The first 90 days appear to be the most critical for customer retention. The NYT will reach out to the customer via email highlighting their reporting, as well as special exclusive events for subscribers. According to Clay Fisher, this approach makes customers feel like they're a part of a "whole new world once they've become a subscriber". On the converse, NYT has also determined when customers are likely to leave the service and have implemented strategies to retain them. For instance, once there's an indication a subscriber has been spending less time on the Times' website, they'll "spend more to show articles in people's Facebook feeds and promoting service journalism or interactive stories…". This has proven to be a very effective method in retaining customers, and in the long run, decreasing the cost of acquisition. [8]

Initial Data Observation

The raw data set is made up of 543,705 observations of user web activities for 448 users. Each observation records the time stamp, user ID, user information, the type of activity and the information of the song if the activity is listening to a song (see Appendix A). Important findings from the raw data include:

- The records of all users span from October 1st 2018 until December 1st 2018
- The user agent is static for each user, which doesn't necessarily reflect the reality of real-world data where people use different devices, browsers, and OSs.
- The location of the users is static.
- By examining on the time stamps of the activities and the length of the songs, we found that there is no partially listened songs.

[7] Sharaf, Victory. "Sparkify-Churn Prediction with PySpark on Big Data." *Medium*, The Startup, 3 Feb. 2021, https://medium.com/swlh/sparkify-churn-prediction-with-pyspark-on-big-data-c50157ee491c.

[8] Moses, Lucia. "To Get to 10 Million Subscribers, the New York Times Is Focusing on Churn." *Digiday*, 26 Oct. 2017, https://digiday.com/media/get-10m-subscribers-new-york-times-focusing-churn/.

- There are “add to Playlist” activities in the record, but there is no “remove from Playlist” activities.
- Only users who canceled the service are considered as churned. Free users with no recent activities are not considered as churned.

Data Processing and Feature Engineering

Before building our predictor features, there was a minor error in the data set that needs to be fixed. For the few users registered after October 1st 2018, the time of registration is incorrect. We corrected this information by identifying the time stamp of the *Submit Registration* according to corresponding web sessions (see Appendix B). General user account information such as gender, user account level (free / paid) and user streaming device (Windows, iPhone, iPad, etc.) were easily retrieved from the raw data.

We expanded our collection of predictor variable by the aggregating counts of each type of user activities: songs played, thumbs up or down, and errors encountered. Since new registrations and churned users have shorter period of observation, we normalized the numbers of activities by the total number of membership hours (regardless of paid or free) for all users to improve unbiasedness.

We also calculated additional special features based on the aggregated activities. For example, the ratio of the number of thumbs-ups to the number of thumbs-downs, the average number of advertisements per song listened, and the average number of activities per session. A complete list of generated predictor variables is showed in Appendix C.

Exploratory Data Analysis

A total of 24 predictor variables (3 categorical and 21 numerical) were generated from our feature engineering. Variables that are highly correlated can lead to inaccurate model inference. To reduce multicollinearity of our data and reduce the complexity of our model, we generated a correlation matrix plot (see Appendix D) of the numerical variables to identify the ones to be removed.

We determined variables that had correlations greater than 0.8 were considered highly correlated. For the variables that were highly correlated to each other, the variable of the most interest was kept, and the remaining ones were excluded from analysis. A total of 8 variables were removed by this method. The updated correlation matrix plot after the variables were removed is showed in Appendix E.

Next, we visualized our predictor variables against the target variable, churn. The distributions of the categorical variables (see Appendix F) revealed that users who subscribed to

paid account statuses seemed to have a higher churn rate. Gender seemed to have no effect on the likelihood of churn. iPhone users appeared to be more likely to churn than other device users. It could be due to the possibility of the fact that the iPhone (iOS) app has problems. But it could also just be a high number of users on iPhone than other device types.

The distributions of some numerical variables are right skewed. To have a better visualization, the variables were log-transformed in the distribution plots (no variable was transformed in the data for our modeling) (see Appendix G). From the distribution plots, we found that the number of songs listened, the number of errors encountered, the number of account level switching have positive effect to the churn rate. The increase in the frequency of going to go to the next song among all the activities also increases the likelihood of churn. Users who do not change their account level (the ones staying 100% of the time free or 100% of the time paid) seem to have lower churn rate. They seemed to be content with service they are using. The ones who switched their account level seem to show unsatisfactory sentiment toward the service. Other variables show little or no obvious patterns about churning. The above findings are just ideas we get from the plots, the actual effect would need to be confirmed by our model analysis.

Methodology

Before we started building our model, we noticed that our data was imbalanced. There was a significantly difference between the number of non-churn users and the number of churned users. In order to build a model with higher unbiasedness, we would use the method of up-sampling (see Appendix H). The method adds duplicate records of the minor class to the sample so that the size of the minor class is adjusted to be close to the size of the major class.

The objective of our analysis is not only to predict the users that may churn, but also finding the factors that have significant effects on the rate of churn. Among all modeling options, we decided to use logistic regression as it has a relatively higher interpretability.

We first began the model with all defined features. After checking the marginal plots, we found that the variable for the ratio of new songs did not fit well with our model. To fix this, we squared that variable and included it in the model. (see Appendix I). We rebuilt our model with the added term, we observed some features to be insignificant (see Appendix J). In order to find the most important factors and reduce the complexity of our model, we performed feature selection using the method of backward elimination based on the AIC score (see Appendix K).

Experimentation and Results

Our final model contained nine predictor variables, and the model was verified to be valid based on the residual plot (see Appendix M). Confirmed by the marginal model plots, there is no lack of fit of the model (see Appendix L). Finally, we evaluated the performance of our model. The model produced an AUC of 0.9445 (see Appendix N), which is exceptional. The calculated optimal threshold that maximized the sum of sensitivity and specificity based on the ROC is 0.574. Using this threshold, both model evaluations of the up-sampled data and the pre-up-sampled data (see Appendixes O, P respectively) gave a balanced accuracy of over 91%. The F1 scores were 0.92 and 0.86 respectively.

Looking at our final summary of our logistic model (see Appendix Q), we gathered the following conclusions. The coefficient of songs per hour is positive. This implies that the longer the service is used, the user feels more unsatisfied and has a higher propensity to churn. The coefficients of users who have paid accounts and maintained a paid account until the end of the survey are unsatisfied with the service. The coefficients regarding advertisements indicated that the more advertisements a user had to endure, the more likely a user churned. It's also important to note that the advertising coefficient regarding paid-tier users is 6X larger than the same coefficient for free-tier users, which indicates a deep discontentment among paid-tier users. Finally, the devices used by each user is not statistically significant. This contradicts with our initial findings that iPhone users are the most likely to churn among all device types.

Discussion and Conclusion

From our findings, we discovered that the more the user uses the service, the more unsatisfied they feel. Users of the paid tier were especially unsatisfied with the service since they endured advertisements. In general, the more one has to sit through advertisements, the more unsatisfied the customer is. It also seems the users who churned most from Sparkify were all iPhone users, which implies a bad design on the iPhone platform. We also discovered that the more customers find songs they enjoy and repeat often, the more satisfied they are.

Though the model itself does not improve user retention, it does unveil a plethora of issues that may be subtle and easy to miss. Therefore, it can provide a blueprint for the company to follow and improve its service, thus increasing user retention.

By offering user-targeted benefits or incentives for retention, as exemplified by Equinox and The New York Times' method, Sparkify can reach out to customers before they churn. For instance, a notification system could be built for the company that indicates an increased likelihood in user churn. This would allow them to reach out to the user with offers, such as a discounted upgrade for less advertisements. But overall, Sparkify should decrease the number of advertisements each user has to endure. It could also benefit Sparkify to develop a better or improve upon their recommendation algorithm for music to users in the form of playlists. As discovered by our analysis, we are confident that customers who find music they truly enjoy will

want to hear it repeatedly, and thus remain on the platform. In addition, it seems evident the developers of this platform need to focus on improving their service so customers will be more satisfied. For instance, it would be a worthwhile investment for the company to improve their experience on iPhones (iOS).

If these changes can be implemented and the customer experience is personalized, Sparkify will certainly improve upon its customer retention rate. Applying machine learning and data wrangling, we were able to conclude this paper with valuable findings that could potentially save and/or transform a business for the better. The tools and resources we as data scientists have at our disposal give us an advantage for viewing and transforming data in ways that are not clear to most.

References

Almuqren, Latifah, et al. "An Empirical Study on Customer Churn Behaviours Prediction Using Arabic Twitter Mining Approach." *MDPI*, Multidisciplinary Digital Publishing Institute, 5 July 2021, https://www.mdpi.com/1999-5903/13/7/175/htm.

"How the WSJ Is Examining Engagement to Reduce Churn." *Talking Biz News*, 18 Dec. 2018, https://talkingbiznews.com/they-talk-biz-news/how-the-wsj-is-examining-engagement-to-reduce-churn/.

Kriti. "Customer Churn: A Study of Factors Affecting Customer Churn Using Machine Learning." *Iowa State University*, Iowa State University, 2019, https://dr.lib.iastate.edu/server/api/core/bitstreams/963c8e0d-4209-4137-9d05-ac20968963f9/content.

Literature Review of Data Mining Techniques in Customer Churn ... https://jati.sites.apiit.edu.my/files/2018/07/2017_Issue2_Paper3.pdf.

Norton, Steven. "Equinix Builds AI Model to Predict Customer Churn." *The Wall Street Journal*, Dow Jones & Company, 2 Apr. 2018, https://www.wsj.com/articles/equinix-builds-ai-model-to-predict-customer-churn-1522705279.

Rodan, Ali, et al. "Negative Correlation Learning for Customer Churn Prediction: A Comparison Study." *TheScientificWorldJournal*, Hindawi Publishing Corporation, 2015, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4386545/.

Sharaf, Victory. "Sparkify - Churn Prediction with PySpark on Big Data." *Medium*, The Startup, 3 Feb. 2021, https://medium.com/swlh/sparkify-churn-prediction-with-pyspark-on-big-data-c50157ee491c.

Moses, Lucia. "To Get to 10 Million Subscribers, the New York Times Is Focusing on Churn." *Digiday*, 26 Oct. 2017, https://digiday.com/media/get-10m-subscribers-new-york-times-focusing-churn/.

Code

https://github.com/ezaccountz/DATA_621/blob/main/Final Project/Final_Project_draft_v4.2.Rmd

RPubs Link

RPubs - Data 621 Final Project v4.2

Appendices

Appendix A

Raw Data variables

- ts: Time Stamp (in milliseconds) of the user activity, range from 1538352000000 (the beginning of 10/1/2018) to 1543622400000 (the end of 12/1/2018) Greenwich Mean Time (GMT).
- userId: The user's unique ID. NA if for guest activities.
- sessionId: Identifier of a connection session. The Id is not unique and may be used by another user at a different time.
- page: The page corresponding to a user's action. For example, NextSong indicates a user starts listening to a new song and Roll Advert indicates an advertisement is loaded.
- auth: Indicates the user's status (logged in, logged out, guest, canceled).
- method: method of the user's web request, PUT or GET.
- status: status of a web request. For example, 404 indicates the requested resource is not found.
- level: the user's account level at the time of the activity. 2 levels: paid or free.
- itemInSession: the number of cumulative activities during a web session.
- location: geometric location (city and state) of the user.
- userAgent: user agent of the user, which includes the type of device, operating system and browser version that the user is using.
- lastName: last name of the registered user
- firstName: first name of the registered user
- registration: the time stamp of the time that a user submitted his/her registration
- gender: the gender of the user
- artist: the artist of the song that the user is listening. Different artists may have songs with the same title / name.
- song: the title / name of the song.
- length: the length of the song in seconds

Appendix B

CORRECTING TIME OF REGISTRATION

```
regist_df <- filter(df,df$page=="Submit Registration")

for (i in c(1:nrow(regist_df))) {
  temp_df <- df %>%
                filter(sessionId==regist_df$sessionId[i]) %>%
                filter(!is.na(userId)) %>%
                mutate(delta=abs(ts-regist_df$ts[i])) %>%
                arrange(delta,desc=FALSE)

  df[!is.na(df$userId) & df$userId==temp_df$userId[1],"registration"] <- regist_df$ts[i]
}
```

Appendix C

FEATURE ENGINEERING

A user's observation period is defined as following:
For users who registered after October 1st, 2018, the beginning of the observation period is the time stamp of the *Submit Registration* activity. Otherwise, the beginning of the observation period is October 1st, 2018.
For users who churned, the end of the observation period is the time stamp of the *Cancellation Confirmation* activity. Otherwise, the end of the observation period is December 1st, 2018.
The *duration_in_hours* is the total number of hours from the beginning to the end of the observation period for each user.

The following predictor features are generated from the raw data:

- *end_level*: The user's account level at the end of the observation period (paid account or free account
- *gender*: The gender of the user
- *userAgent*: The type of device that the user is using (Windows, Iphone, Ipad, etc.)
- *tot_act_phour*: The total number of user activities / Total number of hours in the observation period
    - $tot\_act\_phour = \frac{Total\_Activities}{duration\_in\_hours}$
- *songs_phour*: The total number of songs listened / Total number of hours in the observation period
    - $songs\_phour = \frac{NextSong}{duration\_in\_hours}$
- *tot_tu_phour*: The total number of thumbs-ups / Total number of hours in the observation period
    - $tot\_tu\_phour = \frac{Thumbs\_Up}{duration\_in\_hours}$
- *tot_td_phour*: The total number of thumbs-downs / Total number of hours in the observation period
    - $tot\_td\_phour = \frac{Thumbs\_Down}{duration\_in\_hours}$
- *frds_added_phour*: The total number of friends added / Total number of hours in the observation period
    - $frds\_added\_phour = \frac{Add\_Friend}{duration\_in\_hours}$
- *tot_add2PL_phour*: The total number of songs added to the play list / Total number of hours in the observation period

- $tot\_add2PL\_phour = \frac{Add\_to\_Playlist}{duration\_in\_hours}$

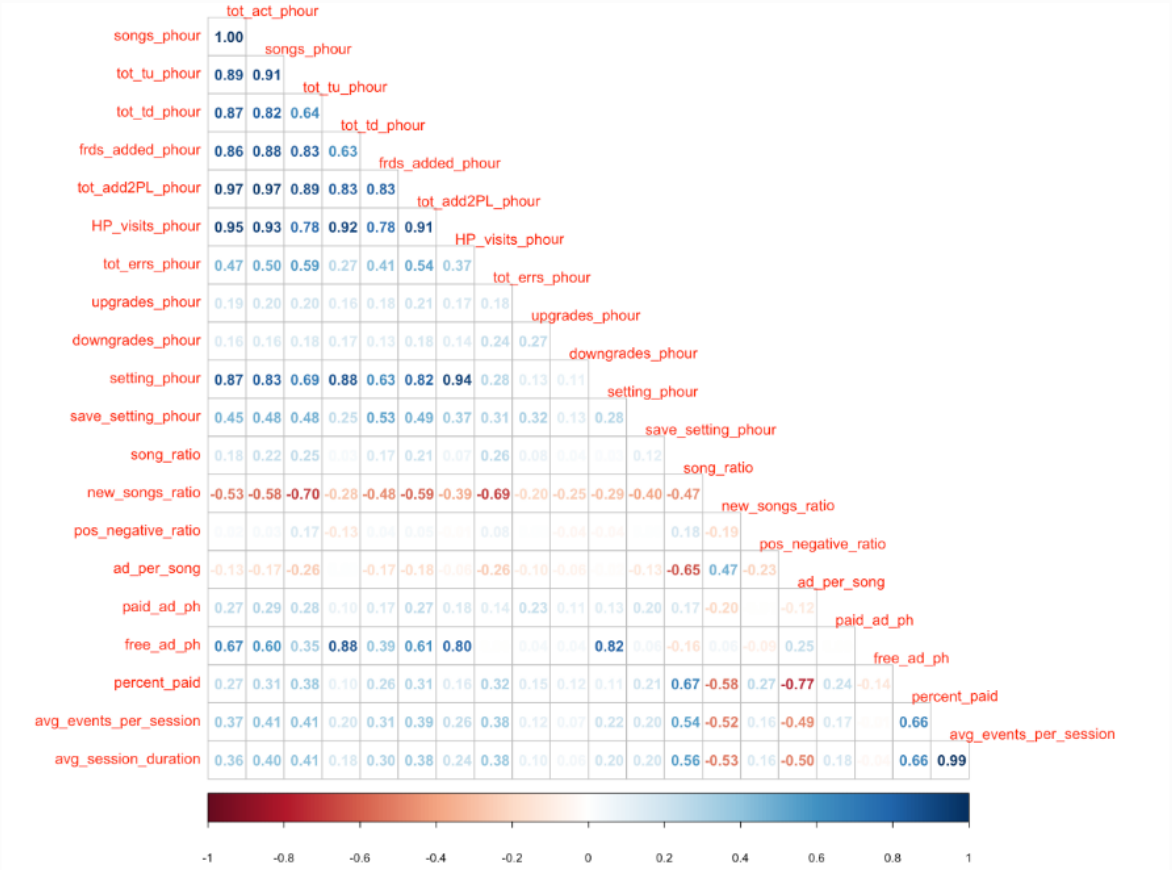
- *HP_visits_phour*: The total number of home page visit / Total number of hours in the observation period
  - $HP\_visits\_phour = \frac{Home}{duration\_in\_hours}$
- *tot_errs_phour*: The total number of error page encountered / Total number of hours in the observation period
  - $tot\_errs\_phour = \frac{Error}{duration\_in\_hours}$
- *upgrades_phour*: The total number of account level upgrading submitted / Total number of hours in the observation period
  - $upgrades\_phour = \frac{Submit\_Upgrade}{duration\_in\_hours}$
- *downgrades_phour*: The total number of account level downgrading submitted / Total number of hours in the observation period
  - $downgrades\_phour = \frac{Submit\_Downgrade}{duration\_in\_hours}$
- *setting_phour*: The total number of setting updates attempted / Total number of hours in the observation period
  - $setting\_phour: = \frac{Settings}{duration\_in\_hours}$
- *save_setting_phour*: The total number of setting updates submitted / Total number of hours in the observation period
  - $save\_setting\_phour = \frac{Save\_Settings}{duration\_in\_hours}$
- *song_ratio*: The percentage of the activities that are NextSong (start listening to a song)
  - $song\_ratio = \frac{NextSong}{duration\_in\_hours}$
- *new_songs_ratio*: The percentage of the songs listened that the user has not listened before (which are the non-repeated songs)
  - $new\_songs\_ratio = \frac{new\_songs\_listened}{NextSong}$
- *pos_negative_ratio*: The ratio of the number of thumbs-ups to the number of thumbs-downs. To handle the issue of dividing by zero, the ratio is modified to (thumbs-ups + 1) / (thumbs-downs + 1)
  - $pos\_negative\_ratio = \frac{Thumbs\_Up+1}{Thumbs\_Down+1}$
- *ad_per_song*: The average number of advertisements per song listened
  - $ad\_per\_song = \frac{Roll\_Advert}{NextSong}$
- *paid_ad_ph*: The total number of advertisement listened when the user account level = paid / Total number of hours in the observation period
  - $paid\_ad\_ph = \frac{paid}{duration\_in\_hours}$
- *free_ad_ph*: The total number of advertisement listened when the user account level = free / Total number of hours in the observation period
  - $free\_ad\_ph = \frac{free}{duration\_in\_hours}$

- *percent_paid*: The percentage of the song-listening time that the user's account is in the paid level
- *avg_events_per_session*: The average number of activities per session
- *avg_session_duration*: The average duration per session in hours

Appendix D

Correlation Matrix Plot – All Variables

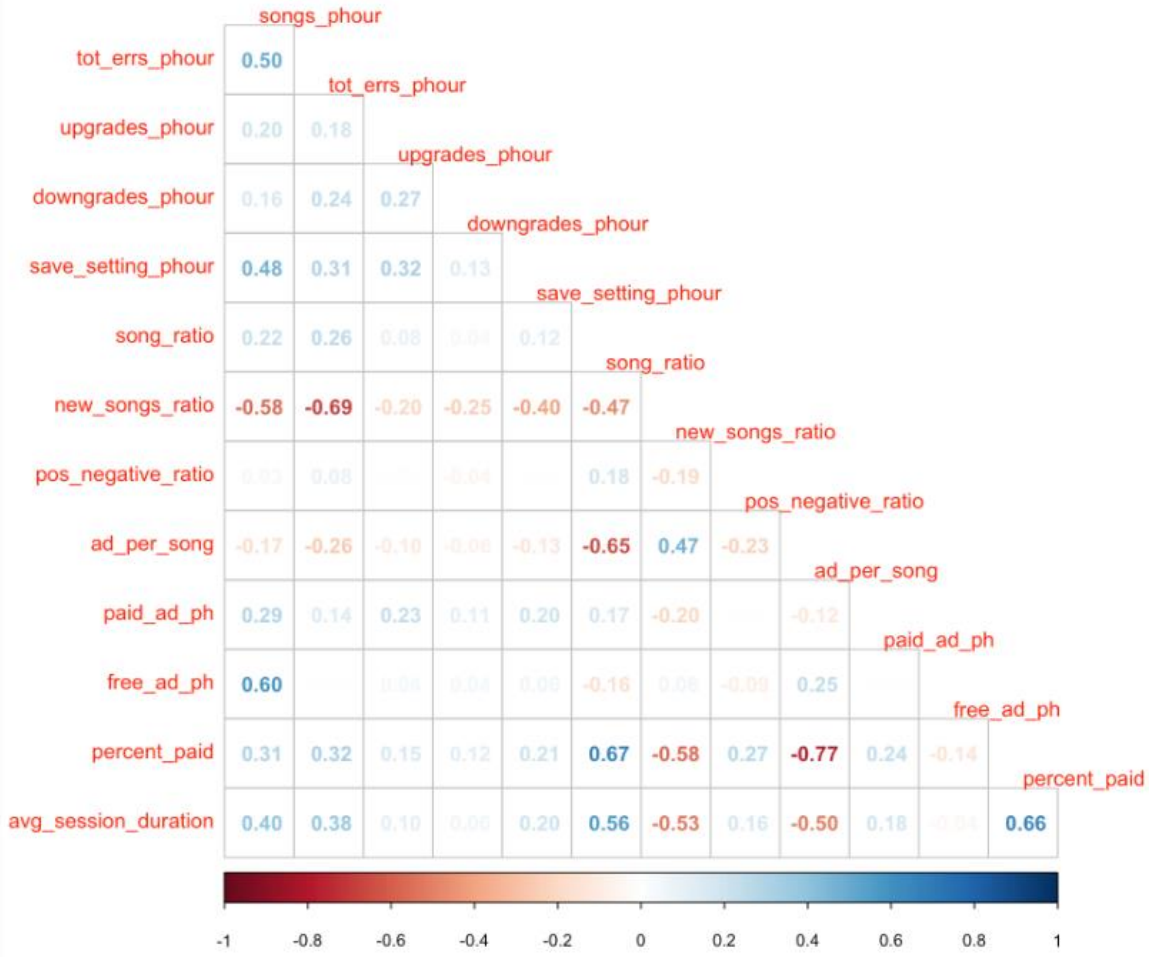
We plugged in all the predictors, or independent variables, into this correlation matrix to visualize if there are any variables constitute multicollinearity.



Appendix E

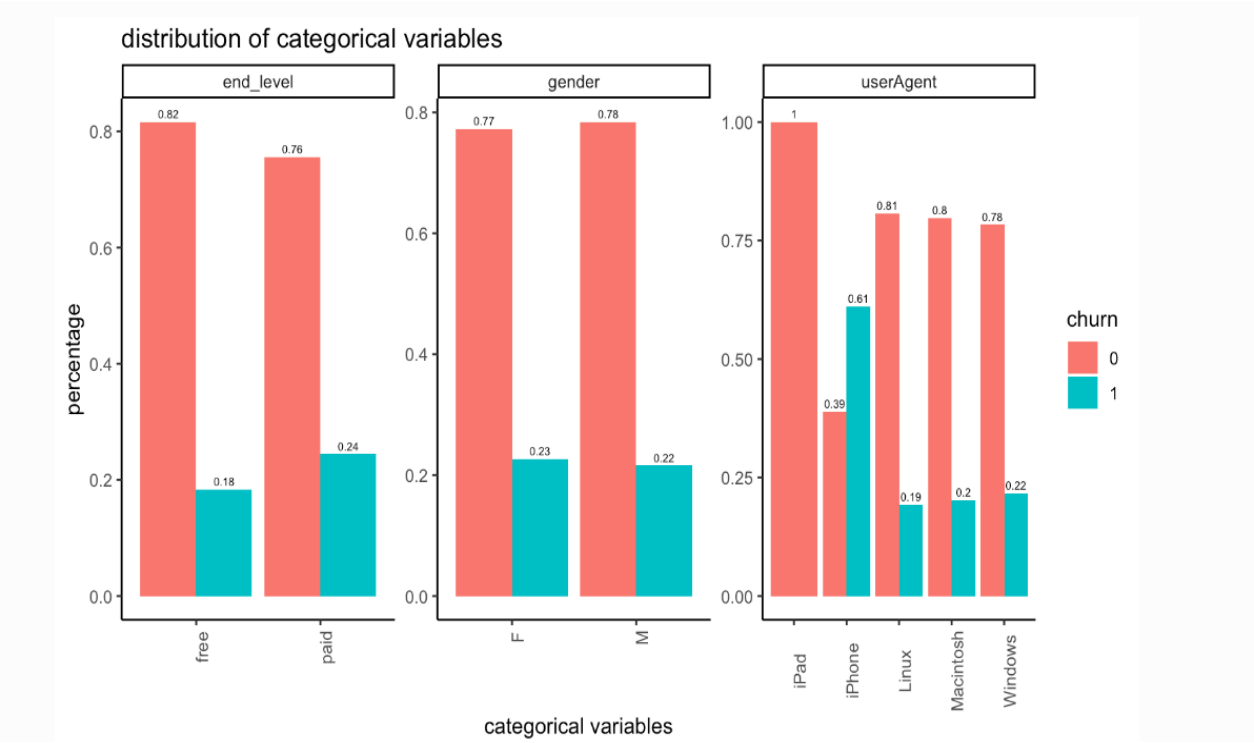
Correlation Matrix Plot – Highly Correlated Variables Removed

We have the following correlations after the highly correlated ones are removed.



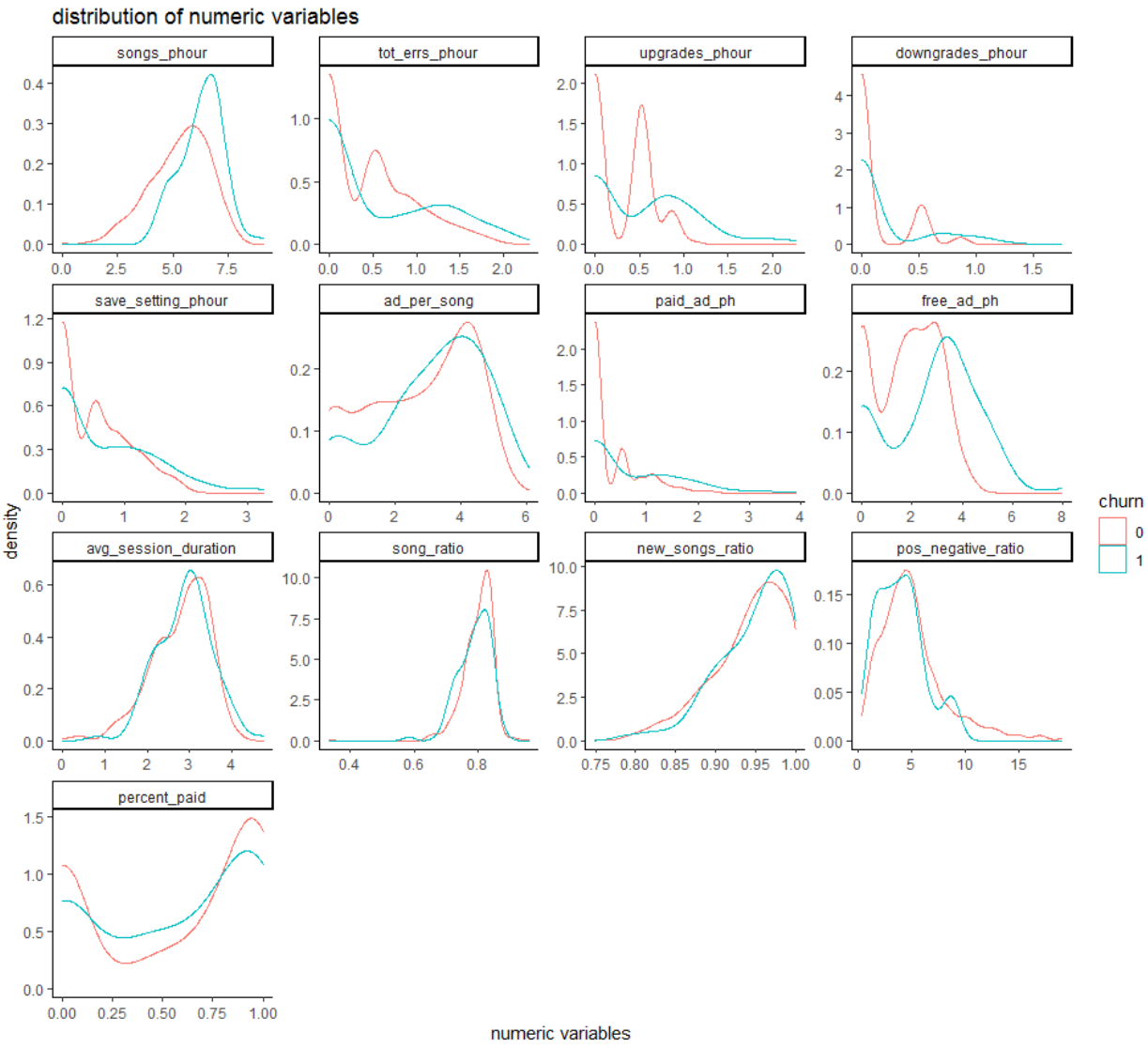
Appendix F

Distribution Plots – Categorical Variables



Appendix G

Distribution Plots – Numerical Variables



Appendix H

UP-SAMPLING

```
temp <- train_df %>% filter(churn == 1) %>%
      slice(rep(1:n(),
            round(nrow(filter(train_df, churn == 0))/
                    nrow(filter(train_df, churn == 1)),0)-1))
train_df2 <- bind_rows(train_df, temp)
```

Appendix I

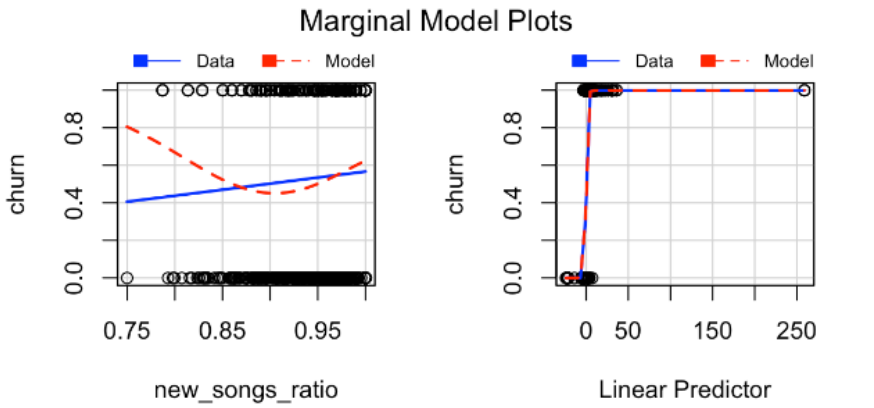
Model Lack of Fit

```
model_logi <- glm(churn~.,family = binomial, train_df2)
```

Checking the marginal model plots, we find that the variable *new_songs_ratio* does not fit well to our model.

Hide

```
marginalModelPlots(model_logi,~new_songs_ratio)
```



In order to fit the curvature, we will add a squared term of *new_songs_ratio* to our model

Hide

```
model_logi <- glm(churn~.+I(new_songs_ratio^2),family = binomial, train_df2)
```

Appendix J

Full Model Summary

```
## Call:
## glm(formula = churn ~ . + I(new_songs_ratio^2), family = binomial,
##     data = train_df2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.0027  -0.4708   0.0000   0.2581   2.4211
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -6.290e+02  7.097e+02  -0.886  0.37541
## end_levelpaid         9.017e-01  4.897e-01   1.841  0.06556 .
## genderM               8.163e-02  2.642e-01   0.309  0.75736
## userAgentiPhone       1.829e+01  7.050e+02   0.026  0.97931
## userAgentLinux        1.609e+01  7.050e+02   0.023  0.98179
## userAgentMacintosh    1.601e+01  7.050e+02   0.023  0.98188
## userAgentWindows      1.597e+01  7.050e+02   0.023  0.98193
## songs_phour           8.935e+00  1.054e+00   8.480  < 2e-16 ***
## tot_errs_phour       -1.990e+01  1.373e+02  -0.145  0.88472
## upgrades_phour        1.094e+02  2.475e+02   0.442  0.65843
## downgrades_phour     -2.055e+02  3.942e+02  -0.521  0.60220
## save_setting_phour    3.036e+01  1.070e+02   0.284  0.77658
## song_ratio           -4.434e+00  3.872e+00  -1.145  0.25216
## new_songs_ratio       1.163e+03  1.640e+02   7.091 1.34e-12 ***
## pos_negative_ratio   -1.506e-01  5.470e-02  -2.753  0.00591 **
## ad_per_song           9.938e+00  5.395e+00   1.842  0.06544 .
## paid_ad_ph            2.782e+02  9.471e+01   2.938  0.00331 **
## free_ad_ph            4.491e+01  1.376e+01   3.263  0.00110 **
## percent_paid          2.241e+00  8.144e-01   2.751  0.00594 **
## avg_session_duration  1.811e-03  7.150e-02   0.025  0.97979
## I(new_songs_ratio^2) -5.495e+02  8.246e+01  -6.664 2.66e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1029.82  on 744  degrees of freedom
## Residual deviance:  428.78  on 724  degrees of freedom
## AIC: 470.78
##
## Number of Fisher Scoring iterations: 15
```

Appendix K

Backward Elimination

```
model_logi <- step(model_logi, trace=0)
```

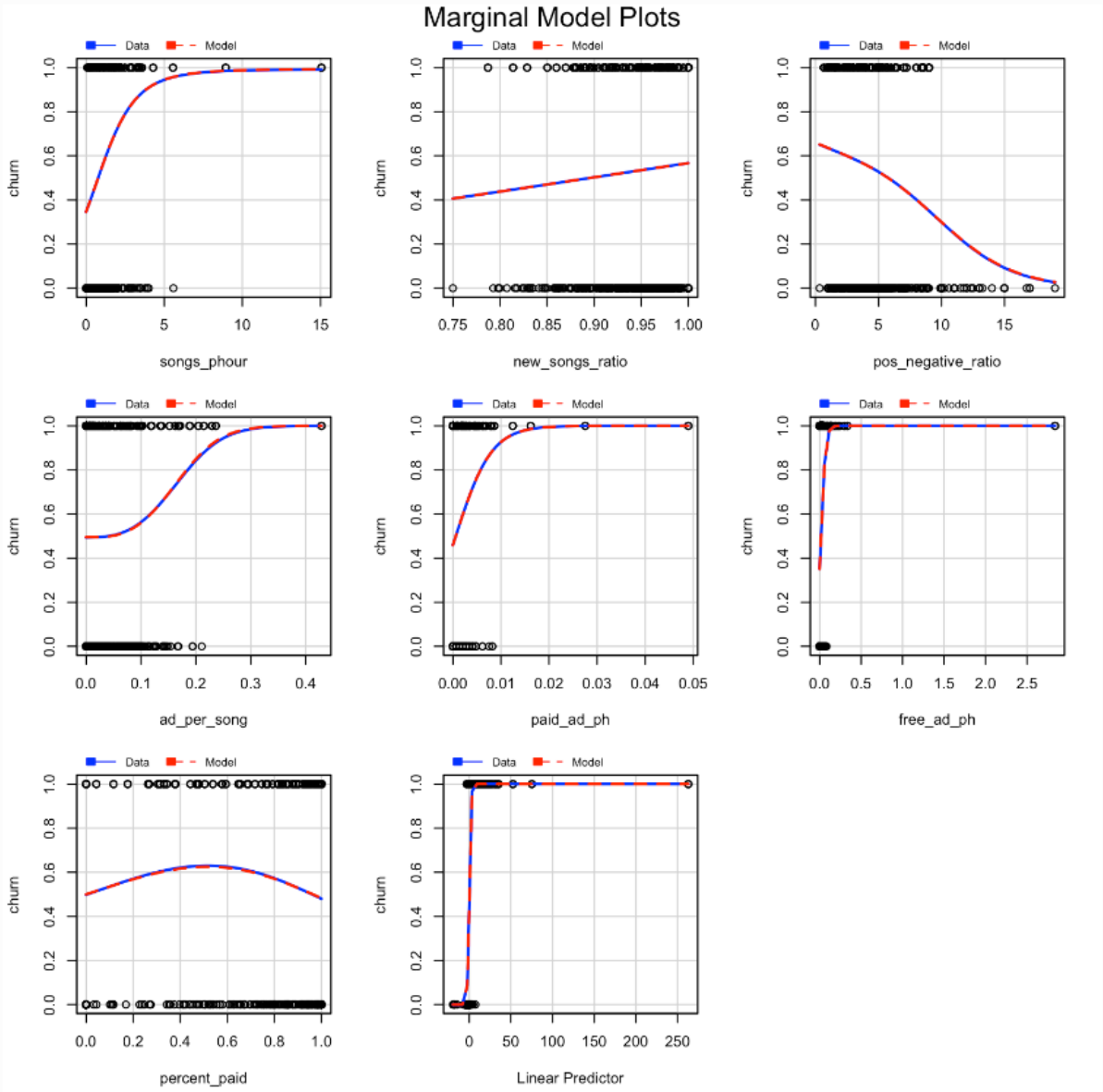
```
summary(model_logi)
```

```
##
## Call:
## glm(formula = churn ~ end_level + userAgent + songs_phour + new_songs_ratio +
##     pos_negative_ratio + ad_per_song + paid_ad_ph + free_ad_ph +
##     percent_paid + I(new_songs_ratio^2), family = binomial, data = train_df2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.8177  -0.4831   0.0000   0.2596   2.4452
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -628.87451  692.48295  -0.908 0.363802
## end_levelpaid           1.10537    0.38790   2.850 0.004377 **
## userAgentiPhone        18.36385  688.03763   0.027 0.978707
## userAgentLinux         16.16706  688.03760   0.023 0.981254
## userAgentMacintosh     16.07003  688.03745   0.023 0.981366
## userAgentWindows       16.02738  688.03744   0.023 0.981415
## songs_phour             8.93534    0.97286   9.185  < 2e-16 ***
## new_songs_ratio      1153.47860  156.71905   7.360 1.84e-13 ***
## pos_negative_ratio     -0.14576    0.05143  -2.834 0.004593 **
## ad_per_song            11.74476    5.15449   2.279 0.022694 *
## paid_ad_ph            277.22956   90.61246   3.060 0.002217 **
## free_ad_ph             45.63953   12.18794   3.745 0.000181 ***
## percent_paid            1.96790    0.74120   2.655 0.007930 **
## I(new_songs_ratio^2) -543.96482   78.58793  -6.922 4.46e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1029.8  on 744  degrees of freedom
## Residual deviance:  430.5  on 731  degrees of freedom
## AIC: 458.5
##
## Number of Fisher Scoring iterations: 15
```

Appendix L

Marginal Model Plots

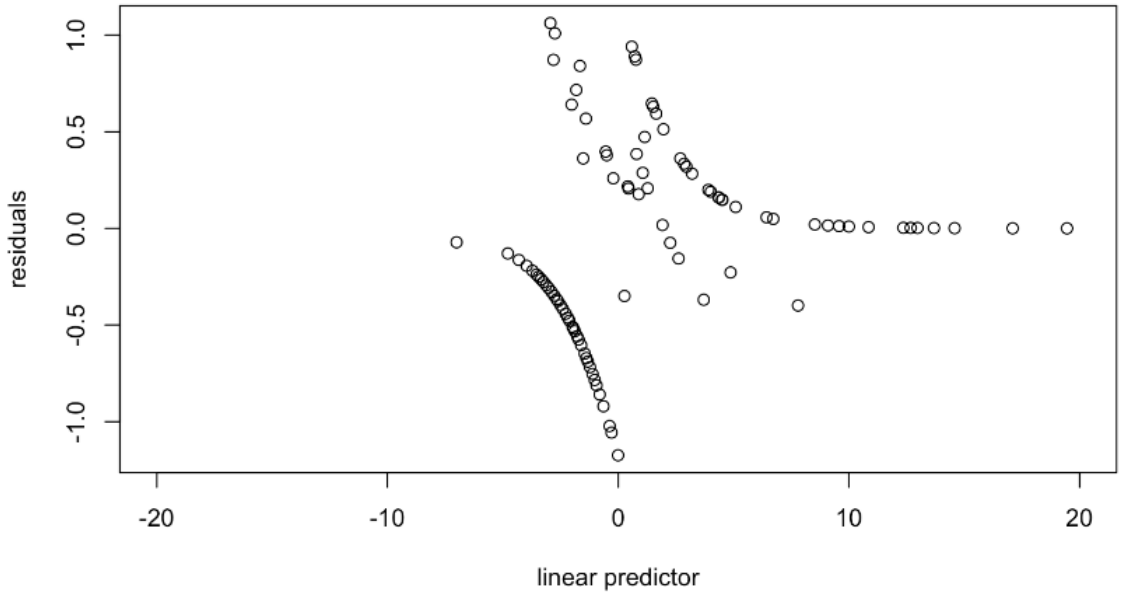
```
marginalModelPlots(model_logi,~songs_phour+new_songs_ratio+pos_negative_ratio+ad_per_song+
                   paid_ad_ph+free_ad_ph+percent_paid, layout =c(3,3))
```



Appendix M

Deviance Residual vs Linear Predictor Plot

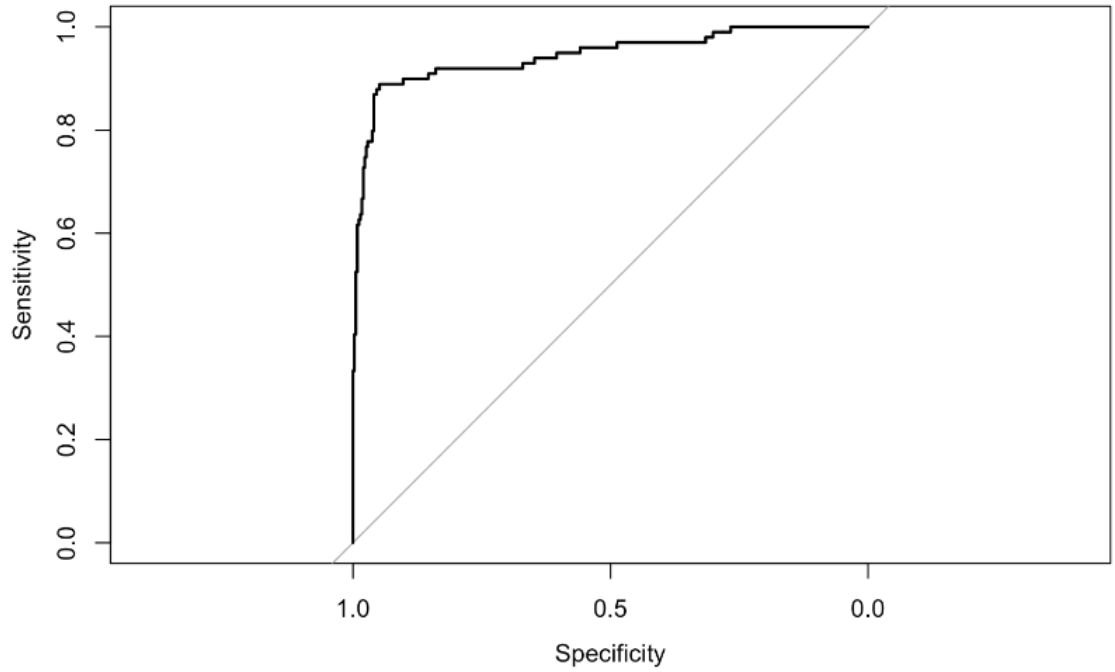
```
residual_df <- mutate(train_df2, residuals=residuals(model_logi,type="deviance"),
                      linpred=predict(model_logi,type = "link"))
gdf <- group_by(residual_df, cut(linpred, breaks=unique(quantile(linpred,(1:100)/101))))
diagdf <- summarise(gdf, residuals=mean(residuals), linpred=mean(linpred))
plot(residuals ~ linpred, diagdf, xlab="linear predictor",xlim=c(-20,20))
```



Appendix N

ROC Curve

```
rocCurve <- roc(train_df2$churn, model_logi$fitted.values)
plot(rocCurve)
```



Hide

```
rocCurve$auc
```

```
## Area under the curve: 0.9445
```

Appendix O

MODEL PERFORMANCE – UP-SAMPLED DATA

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 331  44
##          1  18 352
##
##                Accuracy : 0.9168
##                  95% CI : (0.8946, 0.9356)
##     No Information Rate : 0.5315
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8336
##
##  Mcnemar's Test P-Value : 0.001498
##
##             Sensitivity : 0.8889
##             Specificity : 0.9484
##          Pos Pred Value : 0.9514
##          Neg Pred Value : 0.8827
##               Precision : 0.9514
##                  Recall : 0.8889
##                      F1 : 0.9191
##              Prevalence : 0.5315
##          Detection Rate : 0.4725
##    Detection Prevalence : 0.4966
##       Balanced Accuracy : 0.9187
##
##        'Positive' Class : 1
##
```

Appendix P

MODEL PERFORMANCE – PRE-UP-SAMPLED DATA

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 331  11
##          1  18  88
##
##                Accuracy : 0.9353
##                  95% CI : (0.9084, 0.9562)
##     No Information Rate : 0.779
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8166
##
##  Mcnemar's Test P-Value : 0.2652
##
##             Sensitivity : 0.8889
##             Specificity : 0.9484
##          Pos Pred Value : 0.8302
##          Neg Pred Value : 0.9678
##               Precision : 0.8302
##                  Recall : 0.8889
##                      F1 : 0.8585
##              Prevalence : 0.2210
##          Detection Rate : 0.1964
##    Detection Prevalence : 0.2366
##       Balanced Accuracy : 0.9187
##
##        'Positive' Class : 1
##
```

Appendix Q

Final Model Summary

```
summary(model_logi)
```

```
##
## Call:
## glm(formula = churn ~ end_level + userAgent + songs_phour + new_songs_ratio +
##     pos_negative_ratio + ad_per_song + paid_ad_ph + free_ad_ph +
##     percent_paid + I(new_songs_ratio^2), family = binomial, data = train_df2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.8177  -0.4831   0.0000   0.2596   2.4452
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -628.87451  692.48295  -0.908 0.363802
## end_levelpaid           1.10537    0.38790   2.850 0.004377 **
## userAgentiPhone        18.36385  688.03763   0.027 0.978707
## userAgentLinux         16.16706  688.03760   0.023 0.981254
## userAgentMacintosh     16.07003  688.03745   0.023 0.981366
## userAgentWindows       16.02738  688.03744   0.023 0.981415
## songs_phour             8.93534    0.97286   9.185  < 2e-16 ***
## new_songs_ratio      1153.47860  156.71905   7.360 1.84e-13 ***
## pos_negative_ratio     -0.14576    0.05143  -2.834 0.004593 **
## ad_per_song            11.74476    5.15449   2.279 0.022694 *
## paid_ad_ph            277.22956   90.61246   3.060 0.002217 **
## free_ad_ph             45.63953   12.18794   3.745 0.000181 ***
## percent_paid            1.96790    0.74120   2.655 0.007930 **
## I(new_songs_ratio^2) -543.96482   78.58793  -6.922 4.46e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1029.8  on 744  degrees of freedom
## Residual deviance:  430.5  on 731  degrees of freedom
## AIC: 458.5
##
## Number of Fisher Scoring iterations: 15
```