Georgetown Data Science Certificate

Personalizing an Audio Content Stream

Authors: Nicole Donnelly, Sujit Ray, Anthea Watson Strong
May 21, 2016

Github Repository
Deck

Abstract

A legacy media organization with a nationwide audience recently released a mobile app in an attempt to capture audience share among listeners who access audio content through digital distribution channels. The team signed an NDA with this organization, and will refer to this partner as the "Broadcaster" within our published materials.

The Broadcaster's app surfaces a stream of audio content to users. Users can hear one of two types of content.

> (1) News-- including the top of the hour newscasts, local and national news, and stories from the Broadcaster's flagship news programs.
> (2) Podcast-- including podcasts created by the Broadcaster and also independently created content like "Another Round" from Buzzfeed.

In app, users can skip, thumbs-up, share, or search for content. The Broadcaster has provided user data gathered by this app to our team. In this paper, the team describes our work building a model that will allow the Broadcaster to determine, for any given user, at any given hour, whether the app should surface news or a podcast to the user.

Problem Framing

The Broadcaster is a legacy media organization that produces and distributes audio content to radio stations around the country. Over the past two decades, the Broadcaster has been losing audience share on radio. While the Broadcaster is hoping to stem its audience losses through improved programming, leadership is also looking to develop new audience share among younger listeners who increasingly access content through digital platforms. If the Broadcaster can attract new audiences to its projects across nontraditional platforms, it could make up for the loss of terrestrial listeners.

While the Broadcaster is hoping to stem its audience losses through improved programming, leadership is also looking to develop new audience share among younger listeners who increasingly access content through digital platforms.

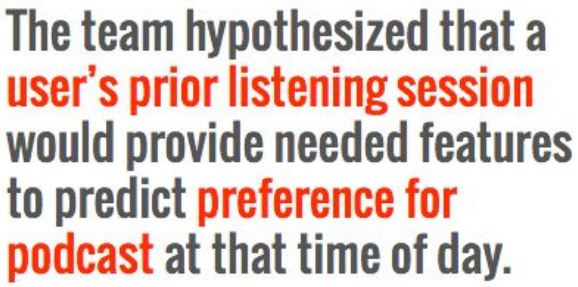
The app allows users to access content created by the Broadcaster, as well as other podcasts, available in a digital format. The app is available on

all the major mobile operating systems as well as Android Auto and Apple's CarPlay. Whereas previous versions of this app requires listeners to make playlists of segments they wanted to hear, this version immediately begins streaming the latest national newscast and the app follows this initial story with an auto generated stream of audio content.

The Broadcaster would like to increase user satisfaction with the app by accurately predicting whether, for any given hour, a user would prefer to hear news content or a podcast and serving that kind of content to the user.

Hypothesis

We hypothesize user preference can be inferred from user interactions with the app during their previous listening session. By looking at both implicit and explicit actions, we can determine whether a user preferred podcast content over news content. Implicit signals included whether or not they listened to the entire story ("completed"). Explicit signals included whether the user shared, searched, skipped, or thumbs-upped a story.



Process

Ingestion and Wrangling

The Broadcaster provided user records from 8/2014 through 2/2016. Our raw data included 614K+ unique users and 98M+ records. Data was delivered in 171 files totalling 7.56 gigabytes.

Each row in our data represented a story a user had started, along with a history of their interaction with that story. (Table of Fields Provided to Team)

The team wrote code to merge all the csv files into a single database table. Data was initially staged to a MySQL database on DreamHost. However, lag experienced with querying the data led to exploration of other options and ultimately migration of the data to MySQL on Google Cloud Platform.

During ingestion and wrangling, following tasks were performed:

- Anonymizing the data by eliminating references to the Broadcaster.
- Normalizing the platform data to the device name (some csv files used the name and some the number).
- Splitting the timestamp into separate date and time columns.
- Creating a table with the story_ID, and max values for elapsed time and story duration as duration was not reliably populated in the original data.
- Creating a table for provided topic information (topic_info).
- Creating a table with the number of entries for each user and the MIN/ MAX timestamp of their records (user_records).

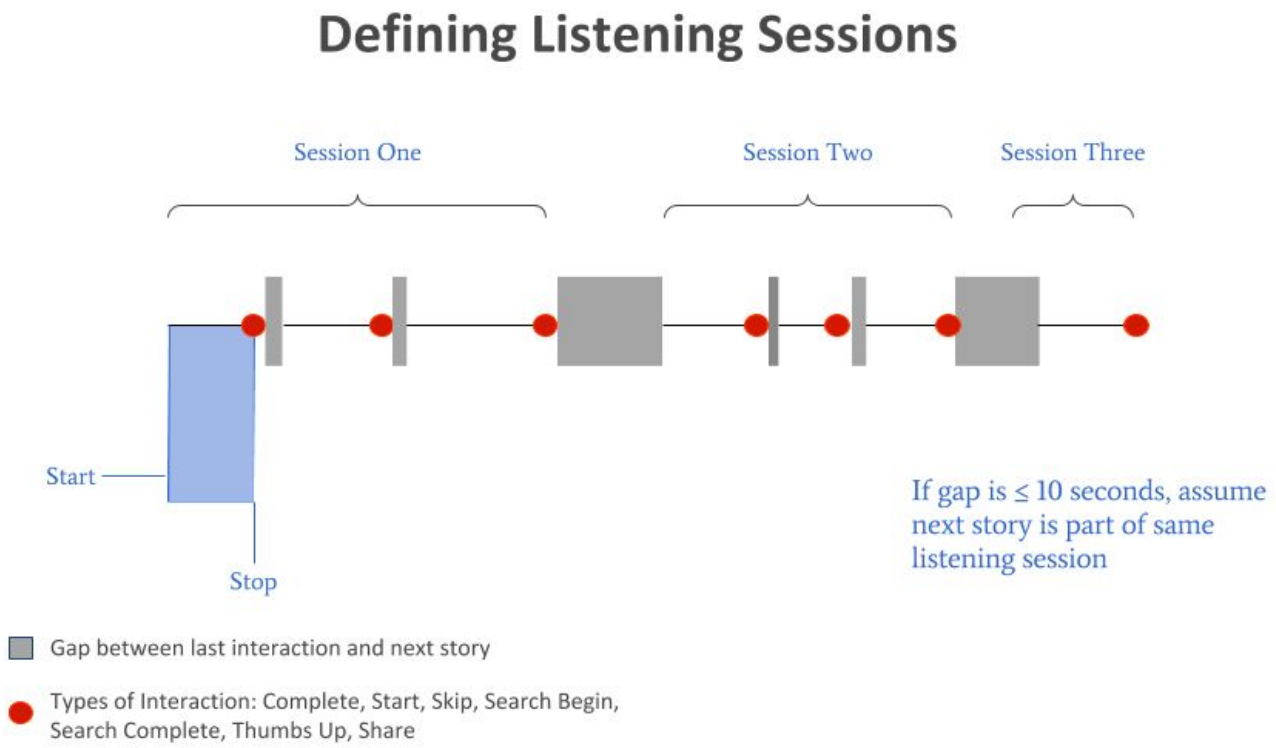
After ingesting and wrangling the data, we conducted exploratory visualization. We were interested in identifying relevant user trends over time. First, we graphed the number of app users overtime. Based on the

explosion of usage in the months preceding our analysis, as well as advice from the Broadcaster's team, we chose to concentrate our work on the last six months of data, September 2015 through February 2016.

In addition to overall usage, preliminary investigation showed users were more likely to interact with the app during the workweek and were more likely to interact with the app in the morning. However, most users were not interacting with the app at all, failing to provide any signals of preference.

Methodology

The team hypothesized that a user's prior listening session would provide needed features to predict preference for podcast at that time of day. We believed a user skipping content and then listening to a different type of content sent a stronger predictive signal than a user who completes all the content served during a session. In order to capture these active signals, we set out to calculate listening sessions and roll up the individual ratings by user session.



The listening sessions were calculated through the following process:

- The team ordered all stories a user had ever started by time.
- Using the timestamp of the interaction, we calculated the duration of the story.
- Then we looked at the start time of the user's next story and calculated the gaps between the last user interaction and the next story.
- If a gap was ≤ 10 seconds, we assumed the next user story was part of the same listening session as the last story. If the gap was >10 seconds, we assumed the next story was part of a new session.

Data Samples

Given the difficulties of working with the total dataset, the team created 4 different samples to work with.

Our data samples were as follows:

- Sample One: All sessions related to randomly selected 10K users. The set therefore represents complete behavioral spectrum of these users.
- Sample Two: Randomly chosen 20K sessions selected from 22M sessions. The set did not necessarily include the complete behavioral spectrum of any user.
- Sample Three: A set of 20K sessions that reflected the total population's behavior.
- Sample Four: A set of 50K sessions that reflected the total population's behavior.

Feature Analysis and Modeling

First Pass

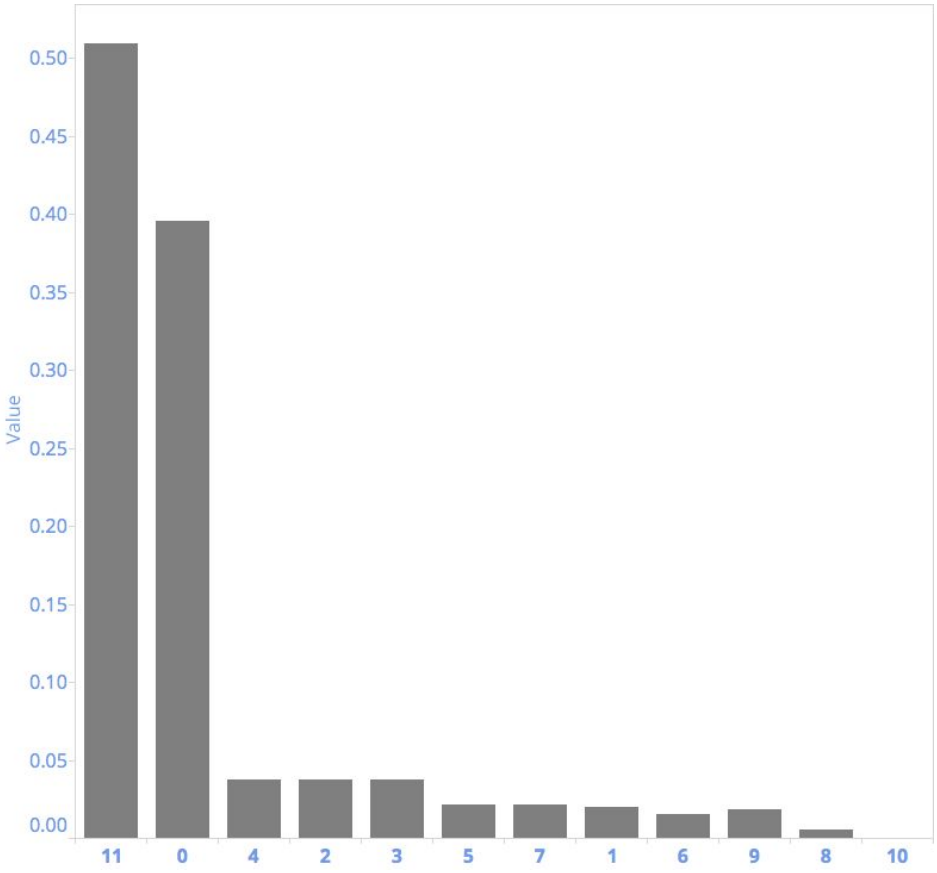
The user listening sessions resulted in twelve potential features to model. In our first pass at model development, we used all twelve features. (Full Results)

We chose algorithms for modeling based on guidance received from our advisors. We began with the following models from scikit-learn: ExtraTreesClassifier, RandomForestClassifier, GaussianNB (naive bayes), BernoulliNB, SVM (support vector machine), and LogisticRegression.

	20K Randomly Selected Sessions	20K Reflecting Total Population Behavior	10K Users with All Sessions	50K Reflecting Total Population Behavior
Extra Trees Cl	f1 0.191344 precision 0.291667 recall 0.142373	f1 0.190164 precision 0.305263 recall 0.138095	f1 0.250842 precision 0.330313 recall 0.202195	f1 0.221719 precision 0.316129 recall 0.170732
Random Forest Cl	f1 0.170213 precision 0.271186 recall 0.124031	f1 0.170648 precision 0.308642 recall 0.117925	f1 0.253954 precision 0.369869 recall 0.193357	f1 0.204947 precision 0.390135 recall 0.138978
GaussianNB	f1 0.293173 precision 0.287402 recall 0.299180	f1 0.257757 precision 0.248848 recall 0.267327	f1 0.311551 precision 0.305672 recall 0.317661	f1 0.288052 precision 0.294807 recall 0.281600
BernoulliNB	f1 0.285156 precision 0.295547 recall 0.275472	f1 0.234987 precision 0.258621 recall 0.215311	f1 0.300412 precision 0.313837 recall 0.288088	f1 0.280107 precision 0.317172 recall 0.250799
SVM	f1 0.030651 precision 0.666667 recall 0.015686	f1 0.000000 precision 0.000000 recall 0.000000	Cannot Compute	f1 0.015723 precision 0.555556 recall 0.007974
LR	f1 0.040268 precision 0.400000 recall 0.021201	f1 0.010101 precision 0.142857 recall 0.005236	f1 0.050543 precision 0.441729 recall 0.026805	f1 0.032949 precision 0.714286 recall 0.016863

Feature Analysis

Based on our initial results, we looked at methods to improve our models through feature selection and feature scaling. For feature selection, we chose to use random forests for feature ranking "since they are so easy to apply: in general they require very little feature engineering and parameter tuning and mean decrease impurity is exposed in most random forest libraries."[1]. We implemented this with ExtraTreesClassifier and RandomForestClassifier. (Full Results)



Feature 11 time_diff_hr
Feature 0 prev_duration
Feature 4 prev_shift
Feature 2 prev_avg_rating_news
Feature 3 prev_avg_rating_podcast
Feature 5 prev_num_news

Both models resulted in fairly consistent feature ranking across all data samples. We chose to use the top five ranked features for our future attempts at modeling.

We also chose to reduce the number of models to run. Both NaiveBayes models appeared the most promising. However, we were not using binary/boolean features so favored the GaussianNB model over BernoulliNB. SVM "...is...generally fast to compute[2]" as long as the number of instances in a data population number in the few tens of thousands. With our population numbering millions of instances, use of SVM was untenable particularly since it could not handle all our population samples. We also chose to pursue LogisticRegression in order to see if we could achieve more promising results.

[1] "Diving into Data," Ando Saabas
[2] Thoughtful Machine Learning, Matthew Kirk pg. 100

Scaled Data, Reduced Features

We used the five most predictive features in our next pass at modeling (Full Results). We scaled the features using scikit learn preprocessing.scale function. We used the same data samples and ran GaussianNB and Logistic Regression again. The LogisticRegression results were still not acceptable.

Model Refinement and Tuning

Next we focused on tuning the LogisticRegression model to see if we could improve results. (Full Results) LogisticRegressionCV was implemented with 15 values for C and five folds, resulting in a suggested value of .0001 for C. This led to some improvement in values. We made the business decision to optimize recall so ultimately discarded LogisticRegression. GaussianNB has no hyperparameters so no further refinement was needed. Final results for each model are shown below.

	20K Randomly Selected Sessions	20K Reflecting Total Population Behavior	10K Users with All Sessions	50K Reflecting Total Population Behavior
GaussianNB	f1 0.308000 precision 0.331897 recall 0.287313	f1 0.359091 precision 0.389163 recall 0.333333	f1 0.292221 precision 0.313606 recall 0.273565	f1 0.269928 precision 0.290448 recall 0.252115
LR	f1 0.084507 precision 0.461538 recall 0.046512	f1 0.090535 precision 0.354839 recall 0.051887	f1 0.046577 precision 0.423762 recall 0.024643	f1 0.037037 precision 0.354839 recall 0.019538

Broadcaster Data without Session

In order to further explore our hypothesis that user preference would be determined by user session information, we also ran the data as provided by the Broadcaster through modeling, limiting it to the most recent 3 months. We mapped categorical variables to integers, scaled the data, and used six features for modeling. Data was subsetted as train, test, and validation by the Broadcaster and we kept those splits.

GaussianNB	f1 0.967882 precision 0.937957 recall 0.999780
LR	f1 0.968729 precision 0.940750 recall 0.998425

As illustrated above, LogisticRegression modeling performed strongly on the data in this format.

Conclusion: Data with Sessions vs Data without Sessions

Ultimately, the data in its provided format proved to create a stronger model. Validation scores for each GaussianNB model below.

Validation of extracted data with 10K sample and GaussianNB model

	precision	recall	f1-score	support
News	0.88	0.9	0.89	8497
Podcast	0.32	0.28	0.3	1435
Total	0.8	0.81	0.8	9932

Validation of provided data with the Broadcaster validation set and Broadcaster model

	precision	recall	f1-score	support
News	0.99	0.99	0.99	3656335
Podcast	0.92	0.85	0.89	323856
Total	0.98	0.98	0.98	3980191

Bibliography

Brownlee, Jason. "Feature Selection in Python with Scikit-Learn - Machine Learning Mastery." *Machine Learning Mastery*. MachineLearningMastery.com, 13 July 2014. Web. 20 May 2016.

Melville, Prem, and Vikas Sindhwani. "Recommender Systems." Encyclopedia of Machine Learning. New York: Springer-Verlag Berlin Heidelberg, 2011. N. pag. Print.

Kirk, Matthew. *Thoughtful Machine Learning*. Sebastopol: O'Reilly Media, 2014. Print.

Pedregosa *et al.*, *Scikit-learn: Machine Learning in Python*, JMLR 12, pp. 2825-2830, 2011.

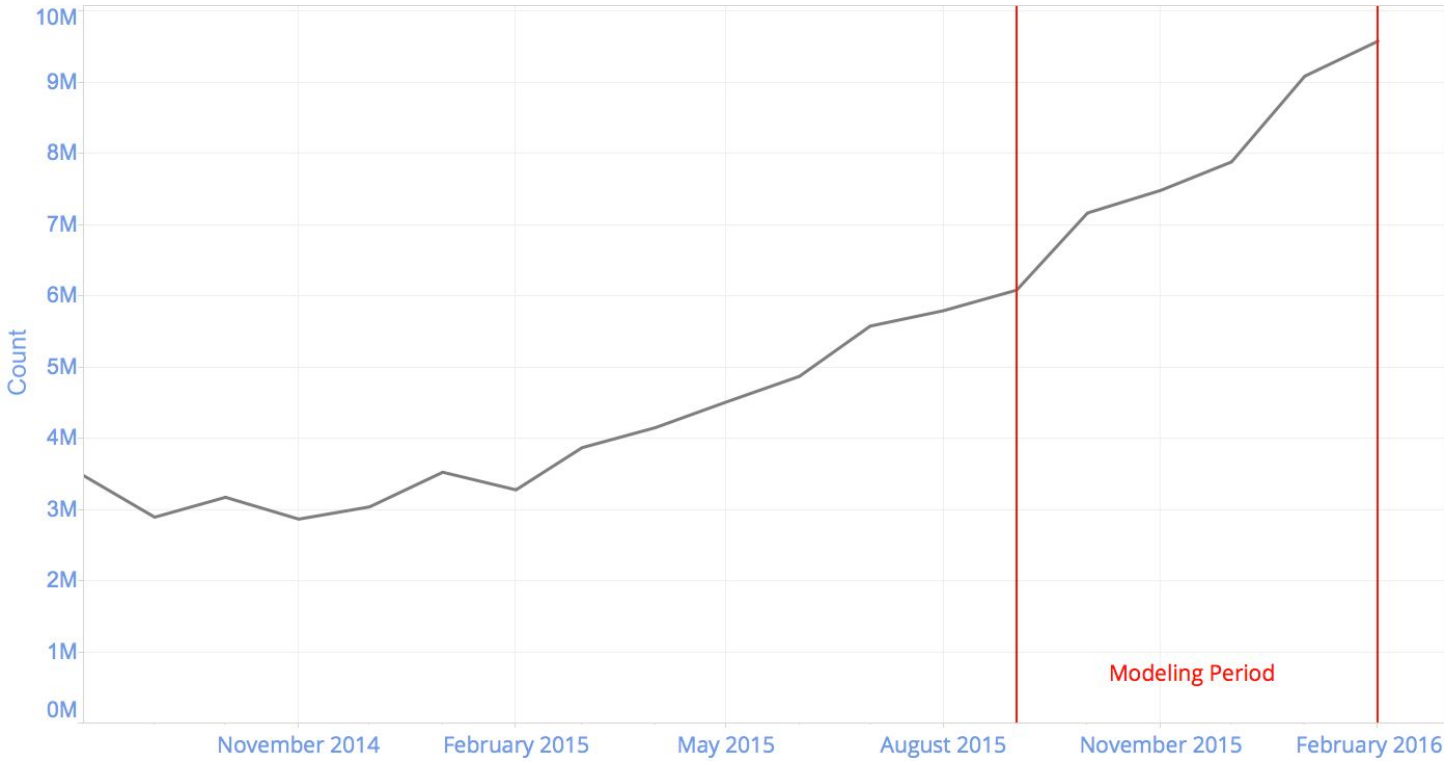
Saabas, Ando. "Diving into Data." *Diving into Data*. N.p., 1 Dec. 2014. Web. 20 May 2016.

"User Guide." Scikit-learn 0.17.1 Documentation. N.p., n.d. Web. 20 May 2016.

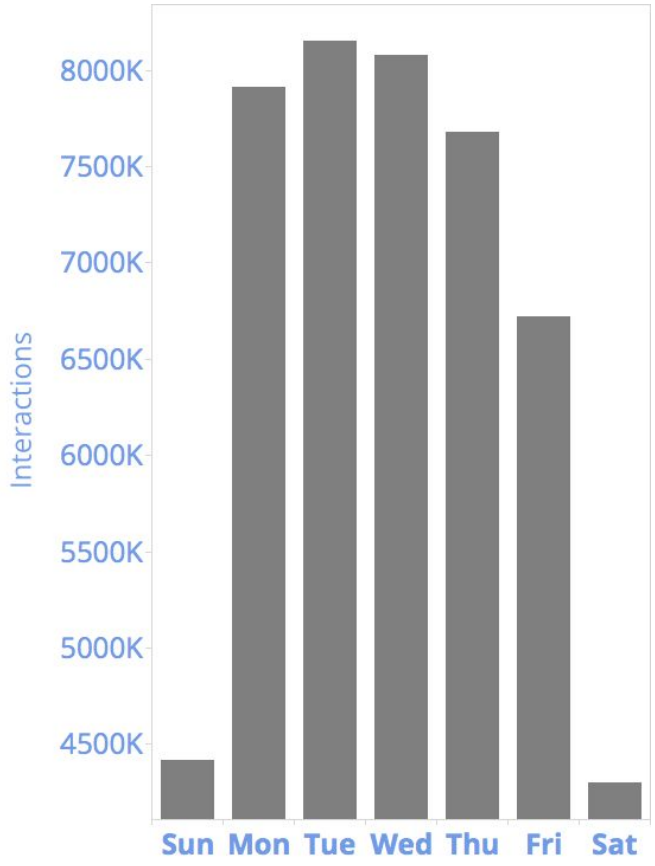
"MySQL 5.7 Reference Manual." *MySQL*. Oracle, n.d. Web. 20 May 2016.

Appendix

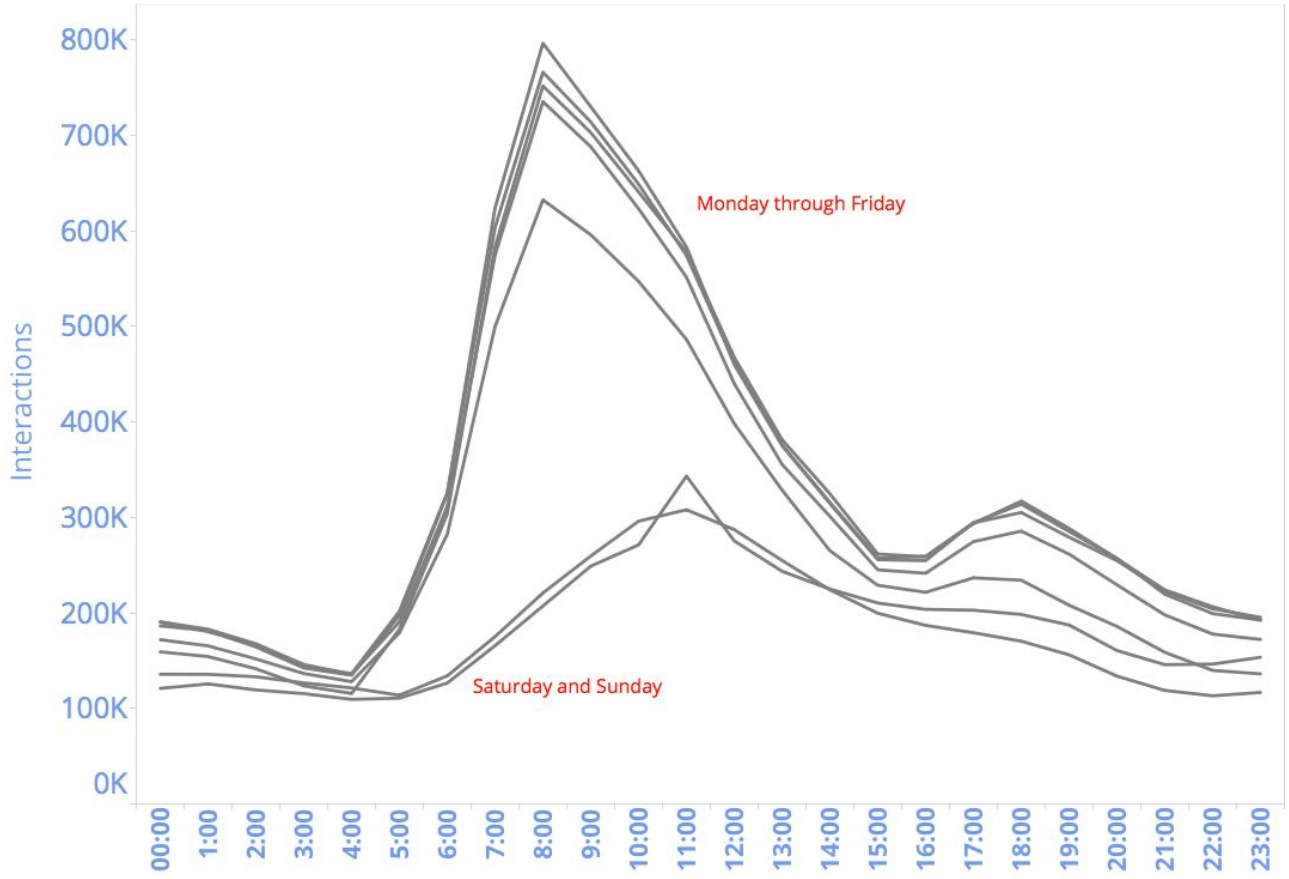
User Interactions by Month



User Interactions by Day of Week



User Interactions by Hour



Raw Data Provided to Team

Field	Description	Expected Format
user_id	unique identifier, consistent over time	NUM
story_id	unique identifier, consistent over time	NUM
rating_value	value in [0,1] relating to the rating (a measure of the quality of the interaction as assigned by the organization)	[0,1]
rating	action the user took in response to the piece	possible values: THUMBUP, COMPLETED, SKIP, START, SHARE, SRCHSTART, SRCHCOMPL
elapsed	seconds user listened	NUM
duration	seconds of the length of the piece	NUM
timestamp	Eastern time, not local time	Timestamp
channel	[REDACTED]	
origin	how the story was selected to give to the user	
platform	The user's mobile operating system	1 = IPHONE, 2 = ANDROID, 3 = WINDOWPH
thing_type_id		1 = news, 9 = story, 15 = podcast

No. of Users by Total Listening Time in Seconds