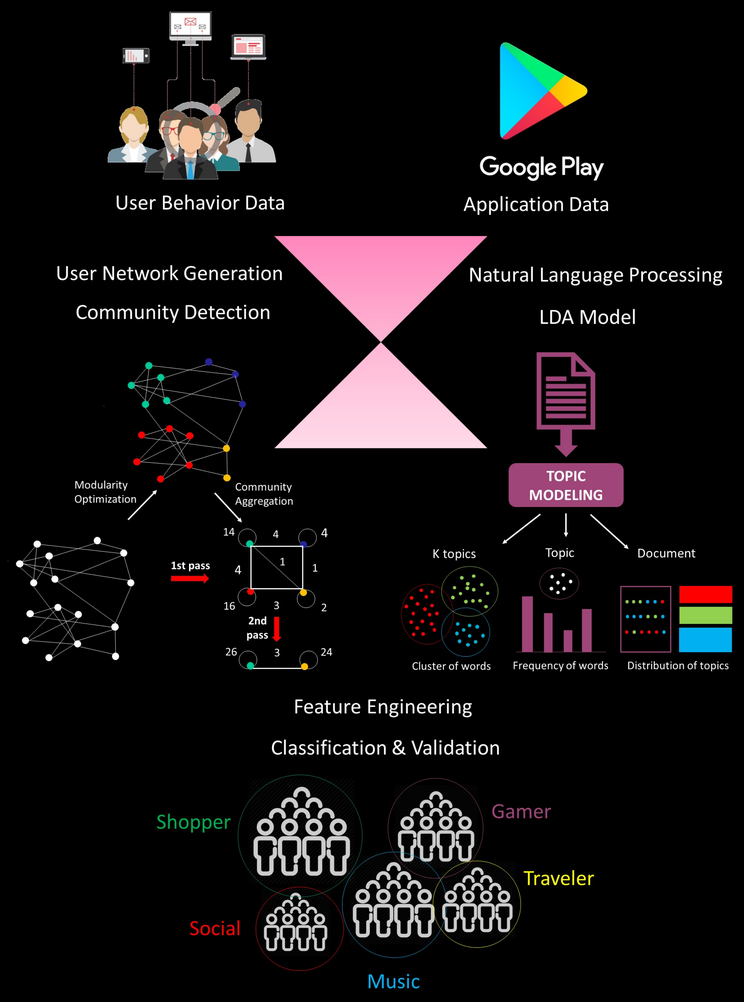
T-Mobile X *SmartCurve*

Automation of Apps Segmentation and Persona Labeling

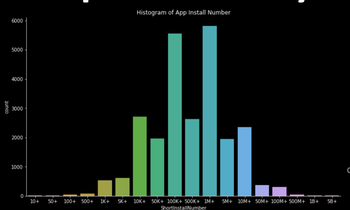
Problem/Background

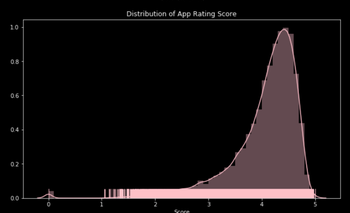
T-Mobile recently acquired a mobile marketing start-up which is aimed to provide a strong foundation for mobile advertising. Dynamic segmentation of app data and segment labeling will tell a better story about customer preferences and optimize labeling of application traffic.
Therefore, they are seeking for help that someone can cluster the applications that T-Mobile customers are using, segment user groups based on their shared applications and label the customer personas that will support their advertisement audience search engine eventually.

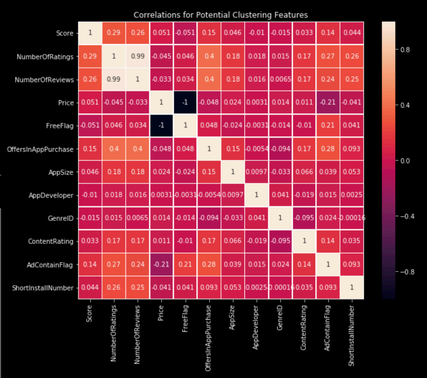


Google Play. (n.d.). Retrieved from http://play.google.com/

Exploratory Data Analysis



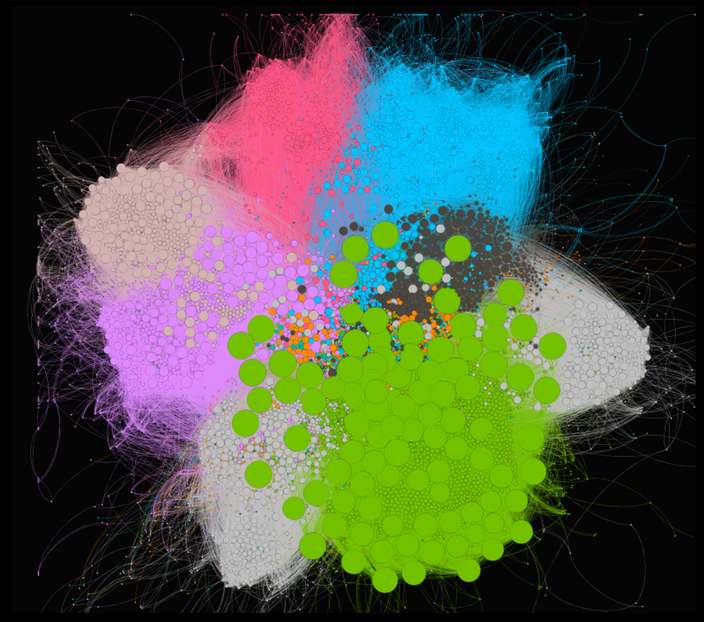




Social Network Analysis

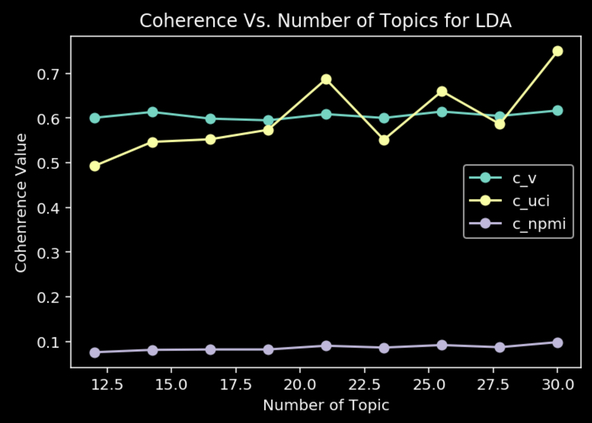
Based on the data about users' usage on applications, user network is generated to represent the relationship between users in terms of their common applications. The more similar apps two users are sharing, the stronger their relation. Considering the network structure, Louvain algorithm is used to detect user communities within the whole network where users with stronger relationships are clustered together as a segment.

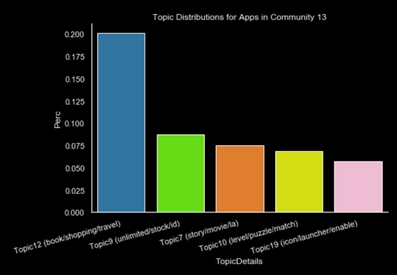
The following figure shows the user network where different colors represent different user communities.

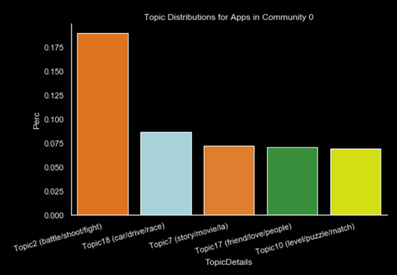


User Persona Generation

The process of generating user persona for each SNA community is composed of (1) performing Topic Modeling on app description and (2) labeling each SNA community with proper keywords.
Leveraging Latent Dirichlet allocation (LDA), 20 topics are generated from app description. This number of topic is selected considering the number of SNA communities and optimal topic coherence values. Then each community is assigned with the top first related topic as its user persona label.







Evaluation

In order to validate the accuracy of SNA communities, classification models based on app and user related features are built. XG Boosting is chosen with its great performance on predicting test dataset.

Table 1: XG Boosting Performance on Test Data

MSE	0.3709497136378481
RMSE	0.6090564125250206
Log Loss	1.294691652086312
Mean Per-Class Error	0.44792784092094023