WSDM - KKBox Churn Prediction

Aaron Olson, Cameron Kennedy, Gaurav Khanna

This dataset is comprised of data collected by WSDM regarding a music streaming subscription available through KKBOX. The goal of this analysis is to predict customer churn.

The initial data set contains 24 variables (25 input variables and 1 variable to predict), these are spread across 4 tables. Listed below are the tables and variables or features available for study:

TABLE: Transactions: Transaction data for each user. 1.6GB, 21.5M X 9

Each row is a payment transaction

- **Msno**: User ID
- **Payment_method_id**: Payment Method
  - There doesn't appear to be a table that maps the values here to actual methods.
  - For ML purposes we don't need to know what a value of 41 means in the real world, however in order to make determinations from the data, understanding this map is meaningful
- **Payment_plan_days**: Length of membership plan in days
- **Plan_list_price**: Price for the plan in NTD currency
- **Actual_amount_paid**: Amount paid in NTD currency
- **Is_auto_renew**: T/F flag determining whether membership is auto-renew or not
- **Transaction Date**: Date of membership purchase in Year-Month-Day
- **Membership_expire_date**: Date of membership expiration in same format
- **Is_cancel**: T/F flag determining whether or not the user canceled service at end of membership term. This doesn't not perfectly correlate with is_churn because a user may upgrade/downgrade service which would flag is_cancel as true but is_churn as false.
- Consider adding:
  - **Max_Trans_Date**: Date of most recent user transaction, so we can see what they've done lately**.**
  - **User tenure:** Max transaction date – min transaction date by user
  - **Days to expiry:** Days until membership expires
  - **Number of transactions:** Simply a count by user
  - **Most_Recent_X:** Several variables for the most recent value of all of the above

TABLE: User_Logs: Logs of listening behavior and KKBOX activity per user.
29.1GB, 392M x 9

Each row is a unique user-date combination

- **Msno**: User ID
- **Date**: Date of the logged activity in Year-Month-Day
- **Num_25**: Number of songs played less than 25% of the way through the song length
- **Num_50**: Number of songs played between 25% and 50%
- **Num_75**: Number of songs played between 50% and 75%
- **Num_985**: Number of songs played between 75% and 98.5%
- **Num_100**: Number of songs played between 98.5% and 100%
- **Num_unq**: Number of unique songs played
- **Total_secs**: Total seconds played
- Considerations to Add:

- **Max_Trans_Date**: Date of most recent user transaction, so we can see what they've done lately (e.g., having their listening habits changed). Actually add this to user instead? Maybe, but we still need to calculate from it.

TABLE: Members: Information regarding the members of the music service. 0.4GB, 6.8M X 6

Each row is a unique user.

- **Msno**: User ID
- **City**: City of the user
- **BD**: Age of the user (caution has some outliers)
- **Gender**
- **Registered_via**: Registration method
  - There doesn't appear to be a table that maps the values here to actual methods.
  - For ML purposes we don't need to know what a value of 41 means in the real world, however in order to make determinations from the data, understanding this map is meaningful
- **Registration_init_time**: Initial time of registration in Year-Month-Day
- **Expiration_date**: Expiration of membership in Year-Month-Day

TABLE: Train: Used for our predictor variable. 45MB, ~1.0M X 2.

Each row represents a unique user.

- **Msno**: User ID
- **Is_churn**: T/F flag variable we are trying to predict.

Notes:

- Msno appears to be a hash in order to anonymize user id or name
- The datasets are fairly large: user_logs is 28 GB which can't be loaded into memory fully and brings a host of big data challenges which will be outside the scope of work related to this project. In order to filter the size of the data down, we propose the following scenarios:
  - Looking at only a specific time: for example only looking at 2016 data. This could introduce bias as there may be features that were added/removed from KKBOX service that are outside the data available for analysis and therefore bias would be introduced by only looking at a time subset.
  - Randomly select a chunk of the data: The challenge here is that there are 4 tables of data and if we randomly delete rows, we may not have a complete dataset for each user
  - Deleting random users: This introduces bias in the form of reducing users in the dataset, however due to the size of the dataset, by keeping a large number of users will reduce the bias introduced and may be the best solution moving forward
- In the user_log file, the number categories are presumed to be total song count played by that user on that particular day (rather than on a particular listening session). Will need to see if there are any duplicate entries of the combination of user id and date (if no duplicates exist then the integer is total for the day).
- There are two types of membership: manual renew and auto-renew
- The dataset comes from users whose membership is set to expire within a defined window of time Feb-April 2017 which may introduce bias into the dataset (only looking at subset of users, however due to the size the bias shouldn't be large or significant).
- Immediately Relevant Categories/Features:

    - Membership Expiration Date
    - Is_Cancel
- The is_cancel category is strongly correlated with the is_churn category (though as noted previously isn't identical). Need to ensure that model doesn't overly rely on this category when also analyzing other categories.