Predictive Modeling

The explanation of codes, how to run the model is in the `ipynb` file.

This file is a guidance of how to open `ipynb` file and how to run python in our own laptop.

Steps for Windows

1. Visits Anaconda.com/downloads
2. Select Windows
3. Download the **.exe** installer
    1. Download Python 3.7 version (the most recent version)
4. Open and run the **.exe** installer
5. Open the Anaconda Prompt
    1. On your laptop, there is a function called `Search window`
    2. You can find `Anaconda prompt` there
6. Enter `jupyter notebook`

We have three models, they are

- Lifetime Conversion Model
    - predict the probability of converting to lifetime member next year for previous year **annual member**
- Churn Model
    - predict the probability of leaving next year for previous year **annual membe**r
- New Member Acquisition Model
    - predict the probability of subscribing membership next year for previous year **non-member**

File Usage

Data is stored in `5 Predictive Files`

- `predictive_modeling - lifetime - churn - 2019.ipynb`
    - Code for Lifetime Conversion and Churn Analysis
- `predictive_modeling - new member.ipynb`
    - Code for new member acquisition
- Data for retention
    - `Data_member_retention_future_prediction_final.csv`
    - `Data_member_retention_training_final.csv`
    - `Individual_Info_Cleaned`
- Data for acquisition

- `Data_new_member_future_prediction_final.csv`
- `Data_new_member_training_final.csv`
- `Individual_Info_Cleaned`

Why we have to multiple files for each model?

We use previous years data for training model and then apply the most updated data to predict the final result.

For example, let's say we want to what to do in 2020, then the training data will include `2018` and `2019` membership status, and then we predict their membership status in `2020`

Logic of Prediction

sequence might change a bit, but this is the general framework included in the predictive files.

1. import necessary files and information
2. merge demographic and behavior info
3. data engineer
    1. remove missing values
    2. select necessary columns
    3. convert type
4. train test split
    1. resample data
5. model preparation
6. use model for new data

Why we need train test split?

- Training: train the model
- Testing: Evaluate the model

After the model is ready, we use this model for new data set.

For further predictive details, please refer to files end with `ipynb`