WSDM - KKBox's Churn Prediction

1. Problem definition:

a) What is the problem all about?

In 2018, KKBox, a popular music streaming service based in Taiwan, released a dataset consisting of a little over two years of (anonymized) customer transaction and activity data to predict which customers would churn after his/her subscription expires. Specifically, it's a forecast problem whether a user makes a new service subscription transaction within 30 days or not after the current membership expiration date.

b) Why is this an important problem to solve?

"It takes months to find a customer and only seconds to lose one"

In today's competitive market, customer retention is a challenge faced by every other business. Retaining existing customers is important as it is less time consuming than acquiring new customers. Customer churn would help a subscription business such as KKBox in creating substantial difference in their revenue stream.

c) Business impact of solving this problem

KKBOX, a music streaming service provider, is a subscription and advertisement-based business model. Monthly subscription is the primarily major source of revenue and the users have option to cancel their subscription whenever they see fit. Hence, improving the churn prediction is the key for KKBOX's growth.

2. Dataset:

a) Source of the dataset

The dataset is obtained from the WSDM - KKBox's Churn Prediction Challenge on Kaggle.

b) Explanation of each feature

Five datasets ('train', 'sample_submission', 'transaction', 'user log' and 'members_v3') stored in .csv format.

train.csv	msno	A unique identifier indicating each user
	is_churn	A Binary Target variable is_churn = 1 means churn is_churn = 0 means renewal

Table-1

sample_submission_zero.csv	msno	A unique identifier indicating each user
	is_churn	A Binary Target variable is_churn = 1 means churn is_churn = 0 means renewal

Table-2

transactions.csv	msno	user id
	payment_method_id	Payment method
	payment_plan_days	Length of membership plan in days
	plan_list_price	In New Taiwan Dollar (NTD)
	actual_amount_paid	In New Taiwan Dollar (NTD)
	is_auto_renew	Whether the user auto-renew the subscription
	transaction_date	Format %Y%m%d
	membership_expire_date	Format %Y%m%d
	is_cancel	Whether or not the user cancelled the membership in this transaction

Table-3

user_logs.csv	msno	user id
	date	Format %Y%m%d
	num_25	# of songs played less than 25% of the song length
	num_50	# of songs played between 25% to 50% of the song length
	num_75	# of songs played between 50% to 75% of the song length
	num_985	# of songs played between 75% to 98.5% of the song length
	,num_100	# of songs played over 98.5% of the song length
	num_unq	# of unique songs played
	total_secs	Total seconds played

Table-4

members_v3.csv	msno	user id
	city	City name
	bd	Age
	gender	User's gender
	registered_via	Registration method
	registration_init_time	Format %Y%m%d

Table-5

c) Data Size and any challenges you foresee to process it?

Size of each dataset (30.5 GB):

train.csv : 45 MB

sample_submission_zero.csv : 43.5 MB

transactions.csv : 1.61 GB

user_logs.csv : 28.4 GB

members_v3.csv : 708 MB

Challenge: user_logs.csv file size is lot larger than my computer RAM.

d) Tools that will be used to process this data

For train.csv, sample_submission_zero.csv, transactions.csv and members_v3.csv - Pandas is used to read.

For user_logs.csv - Dask is used to read.

I also tried to read the transactions.csv with dask, following are the results:

With Dask: 0.03 second.
With Pandas: 36 seconds.

Observation: For Larger files, dask is preferred.

e) Data Acquisition

Open-Source Data is provided by Kaggle (KKBox's Churn Prediction Challenge).
More data can be acquired with the scala file that is provided in KKBox's Churn Prediction Challenge.

3. Key metric (KPI):

a) Business Metric definition

This problem is Binary Classification problem and to penalize misclassification loss, best metric that can be used as log-loss aka Binary Cross entropy.

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

where yi's are the actual class labels (ground truth) and pi's are the predicted probabilities. The log-loss is between 0 and infinite. Our goal is to reduce loss near to 0.

b) Why is this metric used? Why not others?

As this dataset is highly imbalanced, log-loss is preferred over ROC-AUC.
We can use F1 score too as business metric but simply using Accuracy or Precision and Recall won't be preferred because same, they need to be grouped together like F1 score. Log-loss measures probabilistic prediction while F1 ignores this, so log-loss needs to be used.

c) Where does this metric fail?

Log Loss is difference between ground truth and predicted score for every observation and average those errors over all observations. If we predict a probability 0.99 to an observation that is negative, log-loss will become high. (-log0.01 = 2). So sometimes we need to clip predictions to decrease the risk of that happening. Also log-loss is hard to interpret, with just magnitude value hard to predict how good model is doing but its better if it near to 0.

d) What type of problems is this metric used elsewhere in ML and Data Science?

Log Loss is mainly used for Classification Problem such as

Customer behaviour prediction: Customers can be classified into different categories based on their buying patterns, web store browsing patterns etc.

Document classification: A multinomial classification model can be trained to classify documents in different categories.

Image Classification: A multinomial classification model can be trained to classify images into different categories.

Web text classification: Classifies web text or assign tag to web text based on pre-determined categories learned from the past data.

Credit card fraud detection, Sentiment analysis, Product categorization, Malware classification etc.

4. Real world challenges and constraints

a) What real world constraints do you have while solving this problem?

There is no strict latency requirement as company can analyse first with the model results and then can apply into in their production. Some kind of interpretability is required so that company can understand on what features churning is taking place and misclassification needs to be penalized, that's the ultimate goal of the project.

b) What are the requirements that your solution must meet?

Objective of the problem is that the company can predict whether the user will churn or not, if our model is predicting churn for the specific user, company can do several steps to save the subscriber not to churn based on model interpretability.

5. How are similar problems solved in literature?

a) How is the problem mapped to an existing ML methodology?

It's a classification problem where we have to predict whether user is churned or not.

b) List all solution approaches that you think are relevant to your problem?

In regards with ML: we can use models like logistic Regression, Random Forest, Naïve Bayes and gradient Boosting.

DL: MLP (Multi-Layer Perceptron)

Logistic Regression: Widely used ML model for predicting binary classifier based on one or more features. Advantage of Logistic Regression is it can be fast trained but can't deal with missing data, well there are some hacks like imputation which we can apply for missing data.

Random Forest: Ensemble method using bootstrap aggregating which integrates multiple decision tree models. Each individual tree employs a structure of decision nodes and branches. The tree model starts with a root node, and each internal node is a binary split labelled with an input feature. Decision tree is a greedy algorithm because it finds the best split at each step by finding the best improvement. In order to evaluate misclassification, Gini index for each node will be calculated to evaluate the purity: The bagging method can reduce model variance without increasing bias.

Naïve Bayes: simple classification method which constructs a conditional probability model using Bayes' Theorem. It assumes that the effect of the value of each feature on a given class is independent from the value of all the others (class conditional independence).

Gradient Boosting: Model which ensembles multiple weak prediction models such as decision trees to produce a strong classifier. Boosters are learned sequentially with early learners fitting simple models to the data and then analysing the data for errors. Those errors identify problems or particular instances of the data that are hard to fit examples. Then later models focus primarily on those hard examples trying to get the prediction right. Boosting used to reduce bias without increasing variance.

Multi-Layer Perceptron: Hidden layers with activation function such as relu / tanh. Output layer having activation function as sigmoid. Would use dropout and batch-normalization after hidden layers to reduce overfitting as it will bring regularization.

c) References:

i. https://www.kaggle.com/c/kkbox-churn-prediction-challenge.

ii. https://arxiv.org/pdf/1802.03396

iii. http://cs229.stanford.edu/proj2017/final-posters/5147439.pdf

iv. https://blog.hubspot.com/service/how-to-reduce-customer-churn

v. https://medium.com/analytics-vidhya/kaggle-top-4-solution-wsdm-kkboxs-churn-prediction-fc49104568d6

vi. https://medium.com/swlh/kkbox-churn-prediction-solution-6c85f7ae07f8