# Model Documentation

Username: threecourse
Location: Tokyo, Japan
Competition: Coupon Purchase Prediction

## 1. Summary

This problem was considered binary classification and logistic regression was applied.
Train and test data records were each user and coupon pairs with features.
Vowpal Wabbit and Xgboost were used for training.
Ensemble method was simple average of predicted values from the two.
For each user, coupons were ordered by predicted values and top 10 coupons were chosen.

## 2. Features Selection / Extraction

i. Terms are defined as below for clear explanation.

period : weeks to the date test period started, -1 for the test week
pref : coupon's prefecture
user_pref : user's prefecture
non-spot genres : genres of Gift, Lesson, Other, Delivery. They are assumed to have little relation to location.
spot genres : genres except non-spot genres. They are assumed to have strong relation to location.
spot-pref : coupon's prefecture considered only if the coupon is of spot genres.
genre-price : genre and price range combination. genres divided by price range depending on their characteristic and price diversity.
couponkey : combination of CATALOG_PRICE, DISCOUNT_PRICE, small_area, capsule. This is for capturing re-issue of coupon from the same shop.

Probability of purchase:
- Numerator is count of purchased coupons by the user.
- Denominator is count of coupons which are in active periods.
    - Active periods are periods where any purchase by the user occured.
- Here, exclude the information in the period where the coupon is.

probability of visiting:
- Numerator is count of visited coupons by the user.
- Denominator is count of coupons which are in active periods.
    - Active periods are periods where any visit by the user is observed.
- Here, exclude the information in the period where the coupon is.

popularity of couponkey
- average purchases of coupons with the same couponkey
- if there are no other coupons with the same couponkey, average purchases of coupons with the same genre

purchased couponkey
- purchases of coupons with the same couponkey by the user
- here, exclude the information in the period where the coupon is.

ii. Features are below:
- genre, pref, user_pref
- sex, age
- popularity of couponkey
- purchased couponkey
- distance between pref and user_pref
- binned distance between pref and user_pref
- standardized log price in the genre
- if pref is the same with user_pref
- if pref is in 2nd to 4th closest prefectures to user_pref
- if user_pref is null
- if user_pref is in coupon areas (in coupon_area_train.csv, coupon_area_test.csv)
- probablity of purchase - each genres and each spot-prefs
- probablity of visiting - each genres and each spot-prefs
- probablity of purchase - same genre, spot-pref, small_area and genre-price with the coupon
- probablity of visiting - same genre, spot-pref, small_area and genre-price with the coupon
- sum of probability of purchase - each binned distance between pref and user_pref
- sum of probability of visiting - each binned distance between pref and user_pref

## 3. Modeling Techniques and Training

Train data and test data:

Record is (user, coupon) pair with features.
- Train data is all purchased records and sampled non-purchased records.
  Here, non-spot genres are excluded from training data.
- Test data is all records in the test period.
- Weight is adjusted to equalize the positive and negative samples.

Training:

i. Logistic Regression by Vowpal Wabbit
Generalized Linear model with feature interactions and L1 reguralization

ii. Logistic Regression by XGBoost
Gradient Boosted Decision Trees

Ensembling:
  Simple average of the two predicted values (value is averaged, not probability)

Prediction:
  For each user, predict coupons as below.
  i. Test coupons appeared in the user's visit log is automatically chosen in the first.
    Here, non-spot genres are included.
  ii. Test coupons are ordered by ensembled predicted values and chosen as the order.
    Here, non-spot genres are excluded.

## 4. Code Description

batch.py
- Batch script to run all

a00_prepare.py
- Prepare data such as translate Japanese strings into English, replace hash strings to integer ids and add period to coupons.

b00 - b20 files is for feature engineering.

b00_price.py
- Create genre-price and standarized log price in the genre. Genre-price is genre divided by price range.

b10_location.py
- Create features about locations.

b11a_purchase.py, b11b_purchase_smallarea.py, b11c_purchase_genre_price.py
- Calculate probability of purchase for each user about genre, spotprefs, small_area and genre-price.

b12a_visit.py, b12b_visit_smallarea.py, b12c_visit_genre_price.py
- Calculate probability of visiting for each user about genre, spotprefs, small_area and genre-price.

b13_couponkey_popularity.py
- Infer popularity of each coupon by couponkey and genre.

b14_past_purchase_key.py
- Count purchases by each (user, couponkey) pairs.

b15_area.py
- Create (coupon, pref) pairs whose pref appeared in coupon_area_train or coupon_area_test.

b20_visit_log.py
- Create (user, coupon) pairs in test period which appeared in visit_log.

c00_selection.py
- Generate, select and sample train records and generate test records

dxx_dataframe_preprocess.py
- Function to create dataframes with features by merging features into records created in b00 - b20 files and processing some features.

d00_create_vwtxt.py
- Create Vowpal Wabbit format input file, calling function in dxx_dataframe_preprocess.py.

d01_create_xgbdata.py
- Create Xgboost format input file, calling function in dxx_dataframe_preprocess.py.

e00_vw.sh
- Run Vowpal Wabbit, train with train data and predict test data.

e01_xgb.py
- Run Xgboost, train with train data and predict test data.

f00_create_submission.py
- Create submission file based on predicted values from vowpal wabbit and xgboost.

g00_filemanage.py
- Move submission file and model files to submission folder.

## 5. Dependencies

OS: Amazon Linux AMI release 2015.03
Require 64GB RAM to run the codes.

Python 2.7.9
Vowpal Wabbit 8.0.0
Xgboost v0.40

used python packages are below:
ipython 3.2.1
pandas 0.16.2
numpy 1.9.2
scipy 0.16.2
scikit-learn 0.16.1

## 6. How To Generate the Solution (aka README file)

i. Add Data from Kaggle into "input" folder.
(except prefecture_locations.csv. BOM-removed file is already there.)
ii. Run by "ipython batch.py" in the src folder.

## 7. Additional Comments and Observations

I believe we don't have enough information for predicting non-spot genre coupon purchases such as Delivery. In other words, in Delivery genre, information of coupons we can use is only price and location. Here, location seems to have low predicting power because users can order delivery from any location.
That is why I gave up predicting non-spot genres.

As for hyper parameters, it seemed that tuning parameters do not affect very much.
As for ensemble, I tried another approach, applying xgboost again on the two predicted values. That was slightly better in local validation and public LB, but worse in private LB.

## 8. Simple Features and Methods

Only Xgboost model scored 0.009199 in private and only Vowpal Wabbit model scored 0 0.008910 in private.
I had to decide feature interactions and scale such as taking the logarithm for Vowpal Wabbit, but didn't have to for Xgboost, so Xgboost was easier and more efficient.

## 9. Figures

The table below is probability of purchase, by genre and if pref is the same with user_pref. This indicates that relation to location differes much by genres.

| | **if pref is the same with user_pref** | | |
|---|---|---|---|
| **Genre** | True | False | True/False odds |
| Beauty | 0.007661 | 0.000494 | 15.50 |
| Delivery | 0.004571 | 0.004416 | 1.03 |
| Food | 0.022426 | 0.001827 | 12.27 |
| Gift | 0.054706 | 0.030205 | 1.81 |
| Hair | 0.006117 | 0.000280 | 21.86 |
| Health | 0.003826 | 0.000548 | 6.97 |
| Hotel | 0.010492 | 0.001548 | 6.77 |
| Leisure | 0.046278 | 0.002250 | 20.57 |
| Lesson | 0.002052 | 0.000591 | 3.47 |
| Nail | 0.005854 | 0.000442 | 13.25 |
| Other | 0.014777 | 0.010485 | 1.40 |
| Relaxation | 0.006597 | 0.000526 | 12.54 |
| Spa | 0.006129 | 0.000442 | 13.85 |

## 10. References

There are no paticular references.