

Multi-Stage Ensemble and Feature Engineering for MOOC Dropout Prediction

Jeong-Yoon Lee
Conversion Logic

12300 Wilshire Blvd. Los Angeles,
CA 90025, USA
jeong@conversionlogic.com

Kohei Ozaki
AIG Japan Holdings KK

Kamiyacho MT Bldg 18F, 4-3-20
Toranomon, Minato-ku, Tokyo
105-0001, Japan
ozaki.kohei@aig.co.jp

Song Chen
American International Group

175 Water Street, New York, NY
10038, USA
song.chen@aig.com

Andreas Toescher
Opera Solutions

Hauptplatz 12, 8580 Koeftach,
Austria
andreas.toescher@commendo.at

Mert Bay
Conversion Logic

12300 Wilshire Blvd. Los Angeles,
CA 90025, USA
mert@conversionlogic.com

Tam T. Nguyen
Institute for Infocomm
Research, A*STAR

1 Fusionopolis Way, #21-01
Connexis, Singapore 138632
nguyentt@i2r.a-star.edu.sg

Michael Jahrer
Opera Solutions

Hauptplatz 12, 8580 Koeftach,
Austria
michael.jahrer@commendo.at

Peng Yan
NetEase Youdao

Chuanyue Building, Tsinghua
Science Park, Beijing, 100084, China
yanpeng@rd.netease.com

Xiaocong Zhou
Tsinghua University

Haidian District, Beijing, 100084,
China
infinitezxc@gmail.com

ABSTRACT

This paper describes the winning solution of KDD Cup 2015. The competition aims to predict dropouts in Massive Open Online Courses (MOOCs). Our approach begins with feature engineering to extract predictive features from activity logs of students and meta data. Then, we train sixty three individual classifiers with different subsets of features and seven algorithms. Lastly, we blend predictions of individual classifiers with the multi-stage ensemble framework. Our solution achieves AUC scores of 0.90918 and 0.90744 on the public and private leaderboards respectively.

Categories and Subject Descriptors

I.5.4 [Pattern Recognition]: Application

General Terms

Application

Keywords

KDD Cup, Feature Engineering, Ensemble Learning

1. INTRODUCTION

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...\$15.00.

The task of KDD Cup 2015 is to predict the likelihood of dropout for students on XuetangX, one of the largest Massive Open Online Course (MOOC) platforms in China.

Activity logs of 200,906 enrollments from 112,448 students across 39 courses are provided. Each activity is described by 6 fields of the username, course ID, timestamp, source, event, and object. The training set consists of 8,157,278 logs from 120,543 enrollments with the target variable indicating if a student dropped out. The test set consists of 5,387,848 logs from 80,363 enrollments. The full description of the data sets is available in [?]

Dropout predictions for the test set are submitted and AUC scores are calculated. AUC scores on the 50% of the test set are available in the public leaderboard.

Our final solution is a joint work from 9 data scientists, distributed around the world. The pipeline from raw data to final solution is as follows:

- Hand crafted feature engineering (most of hard work)
- Automatic feature design (autoencoder)
- Individual models (gbm, nn, factor model,..)
- Stage-I ensemble (blends individual models)
- Stage-II ensemble (blends stage-I ensemble models)
- Stage-III ensemble (blends stage-II ensemble models)

2. FEATURE ENGINEERING

All features.

2.1 Feature SK

Features generated by Song and Kohei can be classified as follows:

- Enrollment-based features (No.1-8)
- Username-based features (No.9-18)

- Username-based features for each courses (No.19-25)
- Features based on 10 days after the end date of course (No.26-35)
- Features based on 1 day after the end date of a course (No.36-45)
- Day-level features (No.46)
- Day-level features using target variables (No.47-58)

Full list of features generated by Song and Kohei are described in Table 1. (just listing them for now. TBD in detail).

2.2 Feature RW

Peng and Xiaocong feature.

2.3 Feature TN

Tam feature.

2.4 Feature MJ

Features generated by Michael Jahrer are in sparse format:

- uID (0-112,447)
- cID (112,448-112,486)
- uIDcnt (112,487-112,487)
- eIDcnt (112,488-112,488)
- eID → sID (112,489-112,490)
- eID → evID (11,2491-112,497)
- eID → oIDCnt (112,498-139,443)
- eID → tIDCnt (139,444-139,635)
- uID: $\text{floor}(\log(\text{dateSpan}^2 + 1))$ (139,636-140,635)
- uID → $\log(\text{time diff to obj start} + 1)$ (140,636-140,636)
- eID → dateVec diff stats (140,637-140,649)

2.5 Feature MB

Mert feature.

2.6 Feature JL

Features generated by Jeong-Yoon Lee are as follows:

- User ID (20,113) - One-hot-encoded username. Usernames appearing less than 100 times in training log data are grouped together as one user ID.
- Course ID (39) - One-hot-encoded course_id.
- Source Event (10) - One-hot-encoded combination of source and event.
- Object ID (3,554) - One-hot-encoded object. Objects appearing less than 100 times in training log data are grouped together as one object ID.
- Count (1) - Number of log entries for an enrollment_id.
- Object Category (6) - Number of log entries with an object category for an enrollment_id.
- Number of Children Objects (7) - One-hot-encoded total number of object's children for an enrollment_id.
- Object Timespan (10) - One-hot-encoded timespan in days between object's start date and last day of the class

- Day of Class (30) - One-hot-encoded day of the class
- Week of Class (4) - One-hot-encoded week of the class
- End Month of Class (7) - One-hot-encoded end month of the class
- Object Started in Dropout Period (2) - Binary variable that is 1 if object started after but before 10 days after last day of the class and 0 otherwise.

3. MODEL VALIDATION

We use stratified 5-fold cross validation to estimate the performance of single and ensemble models: Training data are split into five folds while the sample size and dropout rate are preserved across folds.

For validation, each of single and ensemble models is trained five times. Each time, one fold is held out and the remaining four folds are used for training. Then, predictions for the hold-out folds are combined and form the model's CV prediction. CV predictions are used in AUC score calculation and/or as inputs in ensemble model training.

For test, each of single and ensemble models is retrained with whole training data. Then predictions for test data are used for submission and/or as inputs in ensemble model prediction.

4. SINGLE MODELS

something like this.

4.1 Learning Algorithms

- Logistic Regression (LR)
- Kernel Ridge Regression (KRR)
- Factorization Machine/Field-aware Factorization Machine (FM/FFM)
- Neural Networks (NN)
- Extreme Trees (ET)
- Gradient Boosting Decision Trees (GBDT)

4.2 Single Models

- Model 1: RandomForest(R). Dataset: X
- Model 2: Logistic Regression(scikit). Dataset: $\text{Log}(X+1)$
- Model 3: Extra Trees Classifier(scikit). Dataset: $\text{Log}(X+1)$ (but could be raw)
- Model 4: KNeighborsClassifier(scikit). Dataset: $\text{Scale}(\text{Log}(X+1))$
- Model 5: libfm. Dataset: $\text{Sparse}(X)$. Each feature value is a unique level.

5. MULTI-STAGE ENSEMBLE

Stratified 5-fold cross validation (CV). We used xgboost [?], neural nets and linear regression for stage-II ensembling.

5.1 Stage-I Ensemble

We trained 20 stage-I ensemble classifiers with different subsets of CV predictions of 63 individual classifiers.

5.2 Stage-II Ensemble

We trained 2 stage-II ensemble classifiers with different subsets of CV predictions of 20 stage-I ensemble classifiers.

5.3 Stage-III Ensemble

We trained a stage-III ensemble classifier with CV predictions of 5 classifiers: 2 stage-II ensemble, 2 stage-I ensemble, and 1 individual classifiers.

| id | name | type | 5CV | linear weight |
|----|---------------------------------------|----------|----------|---------------|
| 1 | trn.final.90788 | Stage-I | 0.907878 | 1.96267 |
| 2 | esb58v5+magic.dae+nn.validCV.0.907567 | Stage-I | 0.907567 | 0.787138 |
| 3 | et_esb58v5_rank.val.0.906207 | Stage-I | 0.906207 | 0.458095 |
| 4 | lr_forward_0.01_esb.esb15v3.val.yht | Stage-II | 0.907968 | 1.61461 |
| 5 | xgb_rf.ko_new_feat.txt.valCV.0.906721 | Single | 0.906721 | 1.1703 |

A linear combination of the 5 models from table 6.3 results in train AUC=0.908072 and accuracy=0.887334. Which leads to 0.90910 public leaderboard score. By adding 39 courseID correction factors train AUC=0.908194 and public score improved to 0.90918.

6. CONCLUSIONS

Our final AUC score of 0.90918 results from a complex pipeline from raw data to final score. Every part of that pipe needs to be (sub-)optimal implemented by our team to get the best score at the end. The first part “feature design” is the most important one and needs expertise, experience and of course a bit luck to capture all signals in the data.

7. ACKNOWLEDGEMENTS

Thanks to dropbox, github and skype to enable easy communication around the globe.

| No. | Description |
|-----|--|
| 1 | Course_id encoded by 1-of-N coding |
| 2 | Number of requests by an enrollment_id |
| 3 | Number of unique object by an enrollment_id |
| 4 | Number of unique problem object of event by an enrollment_id |
| 5 | Number of active days by an enrollment_id |
| 6 | Number of active hours by an enrollment_id |
| 7 | Time of first access in hours by an enrollment_id |
| 8 | Time of last access in hours by an enrollment_id |
| 9 | Number of enrollments by an username |
| 10 | Number of requests by an username |
| 11 | Number of unique objects by an username |
| 12 | Number of unique problem object of event by an username |
| 13 | Number of active days by an username |
| 14 | Number of active hours by an username |
| 15 | Time of first access in hours by an username |
| 16 | Time of last access in hours by an username |
| 17 | Time of first problem access in hours by an username |
| 18 | Time of last problem access in hours by an username |
| 19 | For each course, number of requests by an username |
| 20 | For each course, number of unique object by an username |
| 21 | For each course, number of unique problem object by an username |
| 22 | For each course, number of active days by an username |
| 23 | For each course, number of active hours by an username |
| 24 | For each course, time of first access in hours |
| 25 | For each course, time of last access in hours |
| 26 | Number of enrollment_ids during 10 days after the end date of course by an username |
| 27 | For each course, number of access logs during 10 days after the end date of course by an username |
| 28 | For each course, number of unique objects during 10 days after the end date of course by an username |
| 29 | For each course, number of unique problem objects during 10 days after the end date of course by an username |
| 30 | For each course, number of active hours during 10 days after the end date of course by an username |
| 31 | For each course, difference between first and last access during 10 days after the end date of course by an username |
| 32 | For each course, time of first access in hours during 10 days after the end date of course by an username |
| 33 | For each course, time of last access in hours during 10 days after the end date of course by an username |
| 34 | For each course, time of first access to an problem object in hours during 10 days after the end date of course by an username |
| 35 | For each course, time of last access to an problem object in hours during 10 days after the end date of course by an username |
| 36 | Number of enrollment_ids during 1 day after the end date of course by an username |
| 37 | For each course, number of access logs during 1 day after the end date of course by an username |
| 38 | For each course, number of unique objects during 1 day after the end date of course by an username |
| 39 | For each course, number of unique problem objects during 1 day after the end date of course by an username |
| 40 | For each course, number of active hours during 1 day after the end date of course by an username |
| 41 | For each course, difference between first and last access during 1 day after the end date of course by an username |
| 42 | For each course, time of first access in hours during 1 day after the end date of course by an username |
| 43 | For each course, time of last access in hours during 1 day after the end date of course by an username |
| 44 | For each course, time of first access to an problem object in hours during 1 day after the end date of course by an username |
| 45 | For each course, time of last access to an problem object in hours during 1 day after the end date of course by an username |
| 46 | For each days of the course, which date is provided in date.csv, number of unique active courses by an username |
| 47 | For each 10 days after the end date of the course, number of active enrollment_id, which target variables are 1 in the training set, enrolled by |
| 48 | For each 10 days after the end date of the course, number of active enrollment_id, which target variables are 0 in the training set, enrolled by |
| 49 | For each 10 days after the end date of the course, number of active enrollment_id (in this case, days between last access and the end date of t |
| 50 | For each 10 days after the end date of the course, number of active enrollment_id (in this case, days between last access and the end date of t |
| 51 | For each 14 days before the end date of the coruses, number of active enrollment_id, which target variables are 1 in the training set, enrolled |
| 52 | For each 14 days before the end date of the coruses, number of active enrollment_id, which target variables are 0 in the training set, enrolled |
| 53 | For each 14 days before the end date of the coruses, number of active enrollment_id (in this case, days between last access and the end date o |
| 54 | For each 14 days before the end date of the coruses, number of active enrollment_id (in this case, days between last access and the end date o |

Table 1: List of features generated by Song and Kohei.