

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,
Inc. (AIG)

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 (South Tower), 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

2nd Floor, Chuangye 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

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. DATASET

We got history from 200k enrollments, from 120k we know the labels. bla bla
bla bla

3. FEATURE ENGINEERING

All features.

3.1 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).

3.2 Feature RW

Peng and Xiacong feature.

3.3 Feature TN

Tam feature.

3.4 Feature MJ

Michael feature.

3.5 Feature MB

Mert feature.

3.6 Feature JL

Jeong feature.

4. INDIVIDUAL 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 Individual Models

- Model 1: RandomForest(R). Dataset: X
- Model 2: Logistic Regression(scikit). Dataset: Log(X+1)
- Model 3: Extra Trees Classifier(scikit). Dataset: Log(X+1) (but could be raw)
- Model 4: KNeighborsClassifier(scikit). Dataset: Scale(Log(X+1))
- Model 5: libfm. Dataset: 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.

Finally we ended up with linear ensembling with 39 courseID correction factors. These 39 factors improved the score from 0.90910 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, enrollment_id
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, enrollment_id
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)
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)
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, enrollment_id
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, enrollment_id
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)
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)

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