Multi-Stage Ensemble and Feature Engineering for MOOC Dropout Prediction

Jeong-Yoon Lee
Conversion Logic
jeong@conversionlogic.com

Andreas Toescher
Opera Solutions
andreas.toescher@commendo.at

Kohei Ozaki
Recruit Technologies
kohei_ozaki@r.recruit.co.jp

Mert Bay
Conversion Logic
mert@conversionlogic.com

Michael Jahrer
Opera Solutions
michael.jahrer@commendo.at

Tam T. Nguyen
Institute for Infocomm Research
nguyentt@i2r.a-star.edu.sg

ABSTRACT

In this paper, we present the winning solution of KDD Cup 2015, where participants are asked to predict dropouts in a Massive Open Online Course (MOOC) platform. Our approach demonstrates best practices in feature engineering dealing with complex real world data, and pushes forward the state-of-the-art ensemble technique. We began with feature engineering and extracted XXX and YYY features from raw student activity logs, course enrollment, and course material data. Then, we trained 64 classifiers with 8 different algorithms and different subsets of extracted features. Lastly, we blended predictions of classifiers with the multi-stage ensemble framework. Our final solution achieved AUC scores of 0.90918 and 0.90744 on the public and private leaderboards respectively, and put us to the 1st place out of 821 teams.

CCS Concepts

•**Computing methodologies** → **Supervised learning by classification; Ensemble methods;** *Cross-validation;*

Keywords

KDD Cup, Feature Engineering, Ensemble Learning

1. INTRODUCTION

Since 1997, KDD Cup has been one of the most prestigious competitions in knowledge discovery and data mining, where experts around the world from both industry and academia compete with each other with best modeling practices to solve real world challenges in complex data sets.

The task of KDD Cup 2015 was to predict dropouts of students in a Massive Open Online Course (MOOC) platform. MOOC platforms aim at providing the mass population with open access to quality education. Despite of their initial success in some courses, MOOC platforms have struggled with extremely high dropout rates. Perna et al. reported that the average completion rate is 4% among 1 million students across 16 Coursera courses offered by the University of Pennsylvania from June 2012 to June 2013 [14]. If we identify those who are likely to drop out, we can engage with and help them complete courses successfully. For this task, XuetangX, one of the largest MOOC platforms in China provides the student activity logs, course enrollment, and course material data.

Student dropout in MOOC can be viewed as customer churn, which is a prevailing problem in publishing, financial services, insurance, electric utilities, health care, banking, Internet, telephone, and cable service industries [11]. There are previous competitions related to churn prediction. The task of Teradata Center for CRM (TCC)-Duke Competition was to predict churn with 171 variables provided by a major wireless telecommunication company [11]. The winning solution of TCC-Duke Competition was the ensemble of TreeNet (or MART [6]) models [1]. The task of KDD Cup 2009 was to predict churn, appetency and up-selling with 15,000 variables provided by the French telecommunication company Orange [7]. The winning solution of KDD Cup 2009 was the ensemble of single models trained with 10 base algorithms [12].

Both competitions focused on predictive modeling rather than feature engineering by providing preprocessed variables instead of raw data as in KDD Cup 2015. However, in practice, feature engineering is inevitable and crucial in predictive modeling. For example, Morik and Köpcke showed significant improvements in churn prediction performance of 4 different algorithms when time intervals for the states of variables are added to original variables [10]. Furthermore, winning solutions of both competitions used simple ensemble methods: an average of single model predictions at TCC-Duke Competition [1] and ensemble selection [3], which averages single model predictions with stepwise greedy forward selection, at KDD Cup 2009 [12].

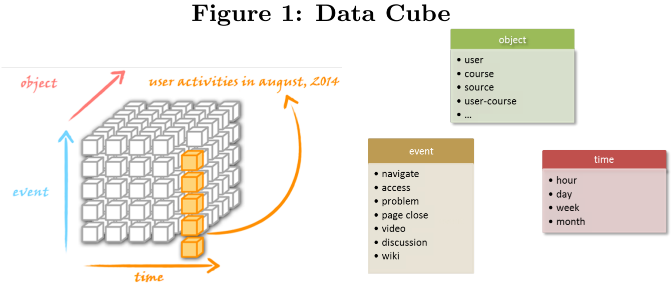
Our final solution is a joint work from 9 data scientists, distributed around the world. The pipeline from raw data to final solution is as follows:

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

KDD '16 August 13–17, 2016, San Fransisco, CA, USA

DOI: 10.1145/1235

Figure 1: Data Cube



- Stage-II ensemble (blends stage-I ensemble models)
- Stage-III ensemble (blends stage-II ensemble models)

The rest of the paper is organized as follows. Section 2 describes our feature engineering approach. Section 3 introduces various classification algorithms used to train single classifiers. Section 4 presents our multi-stage ensemble framework. Section 5 shows our final solution. Section 6 concludes the paper.

2. FEATURE ENGINEERING

Our team members extracted 7 feature sets, namely F1, F2, F3, F4, F5, F6, and F7 from raw data independently.

2.1 Data Sets

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. For each object, 3 additional fields of the category, children, and start date are provided. 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 [2]. In general, this data can be organized in three dimension space, object, time, and event as shown in Figure 1. Feature engineering tasks were carried out based on these views.

2.2 Common Features

Attributes available from raw data There are common features across 7 feature sets, F1 through F7 as follows:

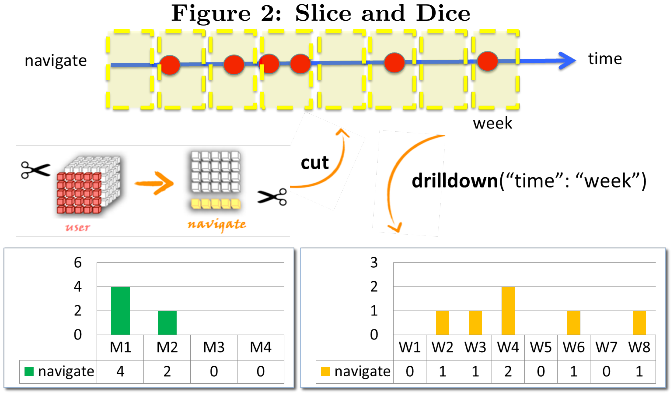
- Number of objects
- Number of events
- Aggregation features of features above

The common features can be generated using cube operations. Figure 2 shows an example how weekly and monthly count features are calculated. First, the data is cut using object dimension. In this case, we choose to generate feature for users. Next, we select an event ”navigate” in the event space to generate a time series presenting ”navigate” event over the time. Finally, the drill-down operation is used to generate monthly or weekly count features.

2.3 F1

The feature set F1 is generated by Chen and Ozaki and includes features as follows:

Figure 2: Slice and Dice



- 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)

Table A1 shows the full list of features in F1.

2.4 F2

The feature set F2 is generated by Yan and Zhou and includes features as follows:

- Visit time(hour, day) set features (including time span and max absent days)
- Act(event, object) counting features (some uses missed content counts)
- Course drop rate
- Number of courses the user enrolled
- Minimum time interval between time points(first visit, last visit, course begin, course end, 10 days after course end) of current course and another enrolled course
- Active days between course end and 10 days after course end
- Active days between last visit and course end
- Number of courses ended after current course end

Table A2 shows the full list of features in F2.

2.5 F3

The feature set F3 is generated by Nguyen, and are categorized into 3 groups of the count, aggregation, and date features.

2.5.1 *Count Features*

There are a few entities such as user, course, and object in the training dataset. Combining these entities together, we have user activities or events. The simplest way to generate features from these events is to count the number of times an entity engaging in the event. The motivation is that the more a user participate in course, the less likely she drop out the course. Table 1 shows the list of count features in F1.

No.	Description
1	The log count of each user
2	The log count of each course
3	The log count of each event
4	The log count of each user per week
5	The log count of each user per two weeks
6	The log count of each user per weekday
7	The log count of each user per month
8	The log count of each course per week
9	The log count of each course per two weeks
10	The log count of each course per weekday
11	The log count of each course per month
12	The log count of each event per week
13	The log count of each event per two weeks
14	The log count of each event per weekday
15	The log count of each event per month

Table 1: List of count features in F3.

2.5.2 *Aggregation Features*

Aggregation features are calculated based on count features. Usually, each course would have a fixed schedule for users to study. Therefore, students enrolled in the course must have consistent activity patterns. Aggregation features would measure the stability of course engagement. These features are minimum, mean, median, maximum, and standard deviation of count features on date basis such as weekly, monthly, etc.

2.5.3 *Date Features*

To capture how often users participate in a certain course, we generated date features. Date features can be time span among user activities as well as time span from last activity and last course date. Table 2 shows the list of date features in F3.

No.	Description
1	Min time span among activities
2	Max time span among activities
3	Mean time span among activities
4	Time span from the last activity and last course date
5	The number of unique activity days of each user

Table 2: List of date features in F3.

2.6 F4

The feature set F4 is generated by Jahrer, and includes features as follows:

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

2.7 F5

The feature set F5 is generated by Bay, and includes features as follows:

- Course ID - One-hot-encoded course_id
- Source time counts by enrollment - The log count of each source type per day for each enrollment
- Source time counts by course id - The log count of each source type per day for each course id
- Event time counts by enrollment - The log count of each event type per day for each enrollment
- Event time counts by course id - The log count of each event type per day for each course id

2.8 F6

The feature set F6 is generated by Lee, and includes features 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 hour_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.

2.9 F7

The feature set F7 is generated by Ozaki, and encodes target variables for each days.

- 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 an username.

- 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 an username.
- 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 the course are also counted for active days), which target variables are 1 in the training set, enrolled by an username.
- 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 the course are also counted for active days), which target variables are 0 in the training set, enrolled by an username.
- 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 by an username.
- 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 by an username.
- 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 of the course are also counted for active days), which target variables are 1 in the training set, enrolled by an username.
- 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 of the course are also counted for active days), which target variables are 0 in the training set, enrolled by an username.

3. CLASSIFICATION ALGORITHMS

We selected algorithms that achieve good predictive performance, process large sparse data sets efficiently (with exception of K-Nearest Neighbors) and differ from other algorithms. The 8 classification algorithms selected are as follows:

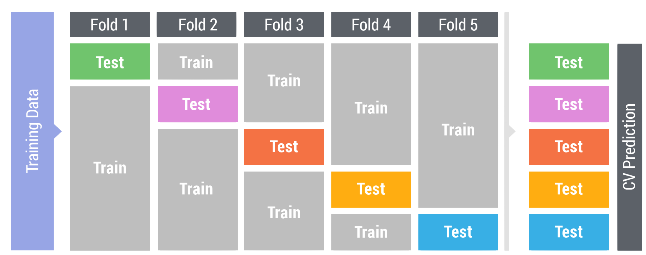
- Gradient Boosting Machine (GBM): We trained GBM classifiers using the Scikit-Learn Python package [13] and XGBoost [4]. We used various tree structures of 4 to 10 maximum depths and 0.004 to 0.05 shrinkage rates.
- Neural Networks (NN): We trained NN classifiers with the dropout [16], rectified linear unit (ReLU) [5] transfer function and sigmoid activation function. We used various network architectures of 1 to 3 hidden layers and 16 to 500 hidden units per layer. We wrote our own C++ NN implementation optimized for sparse data sets.
- Factorization Machine (FM): We trained FM classifiers using libFM [15] and libFFM [8]. We used 2-way interaction dimensions of 4 to 20. We transformed count variables x into $\log(1+x)$.
- Logistic Regression (LR): We trained LR classifiers using the Scikit-Learn Python package [13] and Vowpal Wabbit [9]. We used the regularization parameter $C = 0.01$. We transformed count variables x into $\log(1+x)$.
- Kernel Ridge Regression (KRR): We trained KRR classifiers using our own C++ implementation. We used the ridge regression constant $\lambda = 1.5e-3$ with the Gaussian kernel. We transformed count variables x into $\log(1+x)$.
- Extremely Randomized Trees (ET): We trained ET classifiers using the Scikit-Learn Python package [13].
- Random Forests (RF): We trained RF classifiers using the Scikit-Learn Python package [13].
- K-Nearest Neighbors (KNN): We trained KNN classifiers using our own C++ implementation. We used $k = 124$ with the Euclidean distance. We transformed count variables x into $\log(1+x)$.

4. LEARNING FRAMEWORK

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.

4.1 Model Validation

Figure 3: 5-fold CV



We use stratified 5-fold cross validation (CV) for model validation and ensemble. 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 as inputs for ensemble model training as well as validation score calculation.

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

4.2 Multi-Stage Ensemble

We use the multi-stage ensemble with stacked generalization [18, 17] to blend predictions of multiple models. As shown in Figure 2, in each stage, we train ensemble models with 5-fold CV, and use the CV and test predictions of models in the previous stage as inputs. Then, we pass the CV and test predictions of the ensemble models to the next stage as inputs.

Figure 4: 5-fold CV stacked generalization ensemble

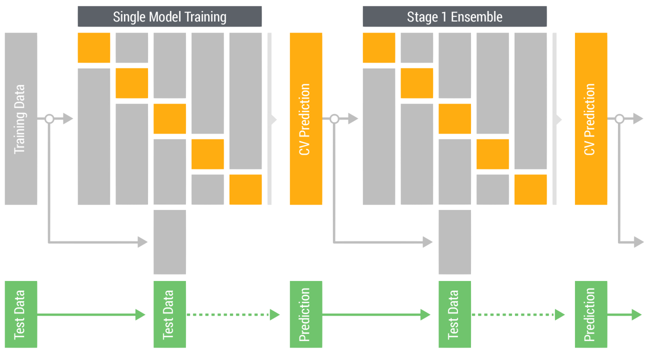
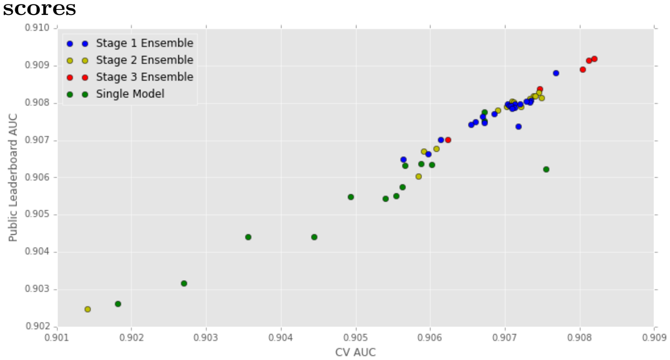


Figure 5: 5-fold CV vs. public leaderboard AUC scores



5. FINAL SOLUTION

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.

We train 64 single models with the 8 different algorithms and different subsets of 7 feature sets.

We trained 15 stage-I ensemble classifiers with different subsets of CV predictions of 64 single classifiers.

At KDD Cup 2015, GBM outperforms other algorithms. Our top 8 single models as well as top 2 stage-1 ensemble models are trained with GBM.

We trained 2 stage-II ensemble classifiers with different subsets of CV predictions of 15 stage-I ensemble classifiers.

We trained a stage-III ensemble classifier with CV predictions of 5 classifiers: 1 stage-II ensemble, 3 stage-I ensemble, and 1 individual classifiers.

Performance improvement diminishes as we add more ensemble stages. The stage-1 ensemble improves the CV AUC score by XXXX from 0.906721 to 0.907688. The stage-2 ensemble improves the CV AUC score by XXXX to 0.907968. The stage-3 ensemble improves the CV AUC score by XXXX to 0.908194.

We choose subsets of predictions from the previous stage for ensemble model training,

A linear combination of the 5 models from table 5 results

ID	Stage	Algorithm	5-CV	Weight
S1	Single	GBM	0.906721	1.1703
E4	I	GBM	0.907878	1.9626
E8	I	NN	0.907567	0.7871
E18	I	ET	0.906207	0.4580
E2	II	LR	0.907968	1.6146

Table 3: List of models selected in the stage-3 ensemble.

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. ADDITIONAL AUTHORS

Additional authors: Xiaocong Zhou (Tsinghua University, email: `infinitezxc@gmail.com`), Song Chen (AIG, email: `song.chen@aig.com`) and Peng Yan (NetEase Youdao, email: `yanpeng@rd.netease.com`).

8. REFERENCES

[1] https://www.salford-systems.com/resources/case-studies/119-teradata-center-for-crm-at-duke-competition.

[2] http://kddcup2015.com/submission-data.html.

[3] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes. Ensemble selection from libraries of models. In *Proceedings of the twenty-first international conference on Machine learning*, page 18. ACM, 2004.

[4] T. Chen and T. He. xgboost: eXtreme gradient boosting. *R package version 0.4-2*, 2015.

[5] G. E. Dahl, T. N. Sainath, and G. E. Hinton. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8609–8613. IEEE, 2013.

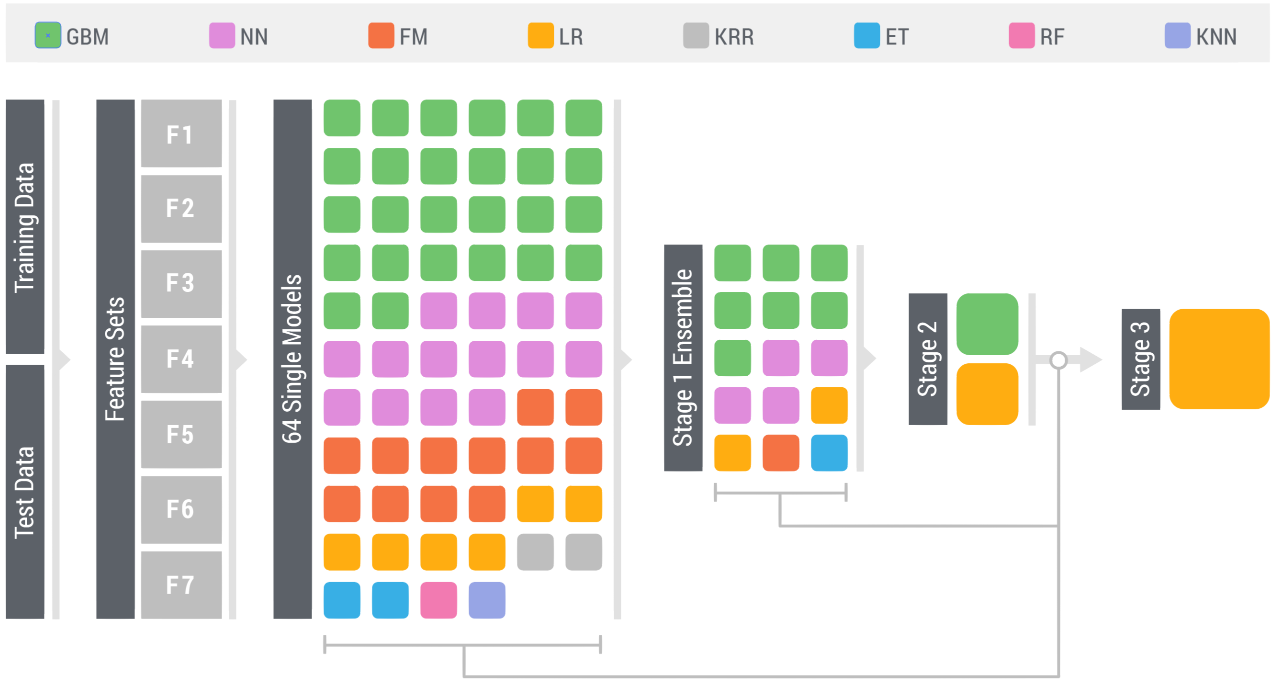
[6] J. H. Friedman and J. J. Meulman. Multiple additive regression trees with application in epidemiology. *Statistics in medicine*, 22(9):1365–1381, 2003.

[7] I. Guyon, V. Lemaire, M. Boullé, G. Dror, and D. Vogel. Analysis of the KDD cup 2009: Fast scoring on a large orange customer database. *JMLR: Workshop and Conference Proceedings, Volume 7*, pages 1–22, 2009.

[8] Y. Juan, Y. Zhuang, and W.-S. Chin. https://www.csie.ntu.edu.tw/ cjlin/libffm/.

[9] J. Langford, L. Li, and A. Strehl. Vowpal wabbit open source project. Technical report, Technical Report, Yahoo, 2007.

Figure 6: End-to-end pipeline for the final solution



[10] K. Morik and H. Köpcke. Analysing customer churn in insurance data–a case study. In *Knowledge Discovery in Databases: PKDD 2004*, pages 325–336. Springer, 2004.

[11] S. A. Neslin, S. Gupta, W. Kamakura, J. Lu, and C. H. Mason. Defection detection: Measuring and understanding the predictive accuracy of customer churn models. *Journal of marketing research*, 43(2):204–211, 2006.

[12] A. Niculescu-Mizil, C. Perlich, G. Swirszcz, V. Sindhwani, Y. Liu, P. Melville, D. Wang, J. Xiao, J. Hu, M. Singh, et al. Winning the KDD cup orange challenge with ensemble selection. *JMLR: Workshop and Conference Proceedings, Volume 7*, pages 23–34, 2009.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[14] L. Perna, A. Ruby, R. Boruch, N. Wang, J. Scull, C. Evans, and S. Ahmad. The life cycle of a million MOOC users. In *Presentation at the MOOC Research Initiative Conference*, 2013.

[15] S. Rendle. Factorization machines with libFM. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):57, 2012.

[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[17] K. M. Ting and I. H. Witten. Issues in stacked generalization. *J. Artif. Intell. Res.(JAIR)*, 10:271–289, 1999.

[18] D. H. Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

APPENDIX

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

Table A1: List of features in the feature set F1.

No.	Description
1	act counts
2	hourset length in last 2 days
3	last month
4	max absent days
5	day set length
6	hour set length
7	average hours per day
8	event wiki counts
9	event discussion counts
10	event access counts
11	event video counts
12	event problem counts
13	obj chapter not visited
14	obj chapter visited ratio
15	obj video not visited
16	obj video visited ratio
17	obj problem not visited
18	obj problem visited ratio
19	obj set length
20	total time span
21	days from last act to course end
22	course drop rate
23	number of courses enrolled
24	min days between first visit and next course begin
25	min days between 10 days after last visit and next course begin
26	min days between last visit and next course end
27	min days between previous course end and last visit
28	min days between 10 days after current course end and next course begin
29	min days between 10 days after current course end and next course end
30	min days between current course end and next visit
31	number of active days between last visit and course end
32	number of active days in 10 days after course end
33	number of courses ended after current course end

Table A2: List of features in the feature set F2.

ID	Stage	Algorithm	Feature	Note	5-CV	Public Leaderboard
S1	Single	GBM	F2 + F3 + F6 + F7	Feature selection with RF	0.906721	0.907765
S2	Single	GBM	F2 + F1 + F6 + F7	-	0.906729	0.907525
S3	Single	GBM	F2 + F1 + F3	Feature selection with RF	0.905875	0.906361
S4	Single	GBM	F2 + F3 + F6	Feature selection with RF	0.905543	0.905516
S5	Single	GBM	F2 + F1 + F3 + F5	Feature selection with RF	0.905356	-
S6	Single	GBM	F2 + F1	-	0.905312	-
S7	Single	GBM	F2 + F3	Feature selection with RF	0.904935	0.905480
S8	Single	GBM	F2 + F1	-	0.904914	-
S9	Single	NN	F2 + F1 + F3 + F4	DAE features added	0.904235	-
S10	Single	NN	F2 + F1 + F3 + F4	-	0.903736	-
S11	Single	NN	F2 + F1 + F3	DAE features added	0.903669	-
S12	Single	FM	F1 + F2	-	0.903560	0.904411
S13	Single	NN	F2 + F1 + F3	-	0.903428	-
S14	Single	GBM	F2 + F6	-	0.903385	-
S15	Single	GBM	F1	-	0.902918	-
S16	Single	GBM	F2	-	0.902287	-
S17	Single	FM	F2	-	0.901983	-
S18	Single	NN	F2	DAE features added	0.901846	0.902614
S19	Single	KRR	F2 + F1 + F3 + F4	DAE features added	0.901522	-
S20	Single	NN	F2 + F1 + F3	-	0.900982	-
S21	Single	GBM	F1	-	0.900906	-
S22	Single	GBM	F3	Feature selection with RF	0.899239	-
S23	Single	GBM	F3	Feature selection with RF	0.899167	-
S24	Single	GBM	F3	Feature selection with RF	0.898969	-
S25	Single	GBM	F3	Feature selection with RF	0.898890	-
S26	Single	GBM	F3	-	0.898749	-
S27	Single	FM	F2 + F6	-	0.898308	-
S28	Single	GBM	F2 + F6	-	0.897968	-
S29	Single	GBM	F3	-	0.897912	-
S30	Single	NN	F2 + F6	-	0.897143	-
S31	Single	NN	F2 + F5	-	0.896748	-
S32	Single	LR	F2 + F1 + F3	-	0.896435	-
S33	Single	FM	F2 + F6	-	0.896160	-
S34	Single	GBM	F3	-	0.895754	-
S35	Single	KRR	F4	-	0.894524	-
S36	Single	GBM	F3	-	0.893507	-
S37	Single	GBM	F6	-	0.892364	-
S38	Single	GBM	F6	-	0.892253	-
S39	Single	NN	F4	-	0.891217	-
S40	Single	LR	F3	-	0.890580	-
S41	Single	NN	F4	-	0.890565	-
S42	Single	FM	F3	-	0.888418	-
S43	Single	FM	F3	-	0.888381	-
S44	Single	RF	F3	-	0.887583	-
S45	Single	ET	F3	-	0.887768	-
S46	Single	FM	F6	-	0.887116	-
S47	Single	FM	F6	-	0.886866	-
S48	Single	NN	F4	-	0.886705	-
S49	Single	NN	F6	-	0.886109	-
S50	Single	GBM	F6	-	0.885184	-
S51	Single	GBM	F6	-	0.885124	-
S52	Single	FM	F6	-	0.885037	-
S53	Single	FM	F6	-	0.884697	-
S54	Single	GBM	F4	-	0.882441	-
S55	Single	ET	F2 + F6	-	0.881539	-
S56	Single	FM	F6	-	0.880878	-
S57	Single	FM	F6	-	0.880366	-
S58	Single	NN	F6	-	0.880219	-
S59	Single	KNN	F4	-	0.877652	-
S60	Single	NN	F6	-	0.876905	-
S61	Single	LR	F4	-	0.821332	-
S62	Single	LR	F6	-	0.804150	-
S63	Single	LR	F6	-	0.802138	-
S64	Single	LR	F6	-	0.800225	-

Table A3: List of single models.

ID	Stage	Algorithm	Note	5-CV	Public Leaderboard
E1	III	LR	Stepwise greedy forward selection	0.908194	0.909181
E2	II	LR	Stepwise greedy forward selection	0.907968	-
E3	II	GBM	-	0.907379	0.908187
E4	I	GBM	-	0.907878	-
E5	I	GBM	-	0.907734	-
E6	I	LR	Stepwise greedy forward selection	0.907716	-
E7	I	GBM	-	0.907668	0.908796
E8	I	NN	DAE features added	0.907567	-
E9	I	GBM	-	0.907353	0.908060
E10	I	GBM	-	0.907283	0.908043
E11	I	NN	-	0.907076	-
E12	I	GBM	-	0.907036	0.907977
E13	I	NN	DAE features added	0.906956	-
E14	I	LR	-	0.906746	-
E15	I	NN	-	0.906714	-
E16	I	GBM	RF feature selection	0.906689	-
E17	I	FM	-	0.906537	-
E18	I	ET	-	0.906200	-

Table A4: List of ensemble models.