

Contents

KDD CUP 2015	错误!未定义书签。
The Assignments	错误!未定义书签。
1.Introduction	2
1.1 Problem Description of KDD Cup 2015	2
1.2 Evaluation Standard	3
1.3 Overview of Data	4
2.Data Preprocess	5
3.Feature Extraction	10
3.1 Basic Features	10
3.2 Effective Features.....	11
3.3 Final Features	14
4. Choose models and learn classifiers	15
4.1 The introduction of chosen models	15
4.1.1 Logistic regression	15
4.1.2 Support vector machines	15
4.1.3 Random forests	16
4.1.4 Artificial neural networks.....	16
4.1.5 Gradient boosting	17
4.2 Learn classifiers	17
4.2.1 scikit-learn library	17
4.2.2 k-fold cross-validation	18
5. The performance of classifiers	18
5.1 The accuracy of classifiers	18
5.2 The ROC curve of classifiers	20
5.3 Other metrics of classifiers.....	21
6. Conclusion	22

1.Introduction

1.1 Problem Description of KDD Cup 2015

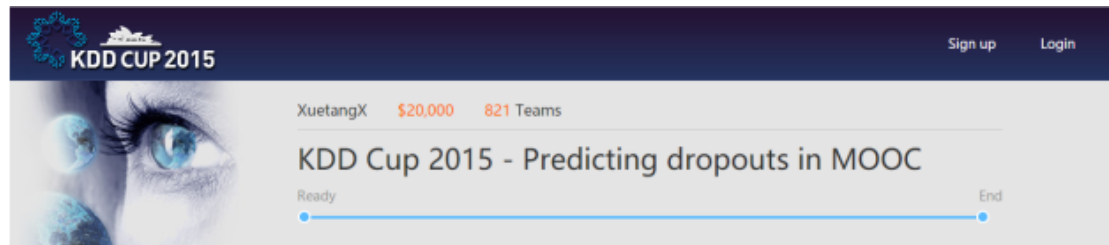


Figure 1.1 KDD Cup 2015

KDD is the abstract of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. The KDD Cup is an annual Data Mining and Knowledge Discovery and Data Mining. In 2015, the subject of KDD is predicting dropout in MOOC.



Figure 1.2 XuetangX

XuetangX, a Chinese MOOC learning platform initiated by Tsinghua University, was officially launched online on Oct 10th, 2013. In April 2014, XuetangX signed a contract with edX, one of the biggest global MOOC learning platform co-founded by Harvard University and MIT, to acquire the exclusive authorization of edX's high-quality international courses. In December 2014, XuetangX signed the Memorandum of Cooperation with FUN to make bilateral effort in course construction, platform development and other aspects. So far, there are more than 100 Chinese courses and over 260 international courses available on XuetangX.

Students' high dropout rate on MOOC platforms has been heavily criticized, and predicting their likelihood of dropout would be useful for maintaining and encouraging students' learning activities. Therefore, in KDD Cup 2015, we will predict dropout on XuetangX, one of the largest MOOC platforms in China.

1.2 Evaluation Standard

We already know the goal is to predict the probability that a student will drop out a course, but how to define whether a student drop the course? The sponsor defined that a user will drop a course within next 10 days based on his or her prior activities. If the user C leaves no records for course C in the log during the next 10 days, it will be regard that he dropout from course C .So we should value the data attribute.

Since the true value is either 1 means drop, or 0, means not drop, will be used to evaluate your binary classifier. So, the accuracy of a submission can be measured by the AUC (Area Under the ROC Curve).AUC is a powerful index of measuring the effectiveness of binary classifier as it has a really extreme excellent characteristic: it remains invariant during the class imbalance occasion.

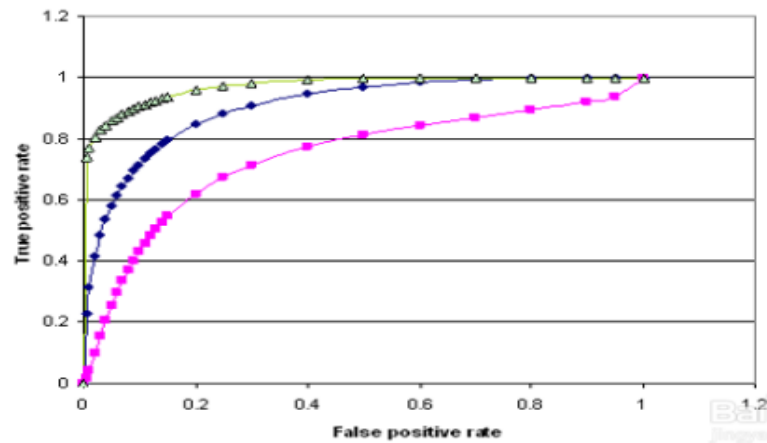


Figure 1.3 A Sample on AUC

In the KDD Cup 2015, at that time, the winner list of top-10 is shown as below.

Rank	Team name	Score
1	Intercontinental Ensemble	0.9091817339587759
2	FEG&NSSOL@DataVeraci	0.9088631699682458
3	CLMS	0.9085724159294324
4	Data Sapiens	0.9079240957270056
5	FirstTimeEver	0.907797205557665
6	KDDILABS&Keiku	0.9077939997455148
7	ttlbb	0.9077348384507644
8	xiaochuan	0.907345041689206
9	Donquote	0.9071736582217601
10	NLP Logix	0.9070665859706881

Figure 1.4 Top-10 Winner List

1.3 Overview of Data

We can get 5 compressed files from the official website, and the type is 7z.

Table 1.1 Data Files

File Name	Size
Test.7z	26.04MB
Data.csv.7z	0.02MB
Object.csv.7z	0.74MB
SampleSubmission.csv.7z	0.05MB
Train.7z	39.13MB

After extracting all the above files, we can find some of them contains sub files. The result of extracting is shown as below:

Table 1.2 Details of Files

File Name	Sub Files
Date.csv	NULL
Object.csv	NULL
SampleSubmission.csv	NULL
Test	enrollment_test.csv
	log_test.csv
Train	enrollment_test.csv
	log_test.csv
	truth_train.csv

Then, we make a new table of the details of data in order to better understand the structure of the data.

Table 1.3 Details of each File

File	Counts	Header
enrollment_train.csv	120,543	enrollment/id/username/course_id
enrollment_test.csv	80,363	
log_train.csv	8,000,000+	enrollment_id/time/source/event/object
log_test.csv	5387848	
Truth_train.csv	120543	enrollment_id/isDropout
Date.csv	40	course_id/from/to
SampleSubmission.csv	80,362	enrollment_id/isDropout

After the above operations, we can easily find that the file “enrollment_train.csv” and “enrollment_test.csv” have the same structure, so does the file “log_train.csv” and “log_test.csv”. The enrollment file defined the corresponding relationship between the id of course and user name, and use the unique id of enrollment. The log file is the largest file and probably the most important. It defines the action of user in the courses. The “truth_train.csv” file shows whether the user dropout the class. The “object.csv” file shows some basic information about the course like module, category and children object id. The “date.csv” file shows the start time and end time of courses. It is obvious that each course is 30 days. The “sampleSubmission.csv” file composed by a 80362*2 matrix shows the format of submission.

By the way, the concrete details of preprocessing is done by another teammate.

2.Data Preprocess

Through the data analysis we can find that the columns of username and course_id in the enrollment_train.csv and enrollment_test.csv are shown with encrypted sequence. It is difficult to process, we need to map these sequence to different digit. We preprocessed these fields by dictionary in python, every encrypted sequence corresponding to a number, then save them in files enrollment_train#.csv and enrollment_test#.csv. The results are show in the table below.

表 (3 个字段, 120,542 条记录) #1

	enrollment_id	username	course_id
1	1	0	28
2	3	1	9
3	4	2	28
4	5	3	9
5	6	4	22
6	7	5	9
7	9	6	28
8	12	7	28
9	13	8	4
10	14	9	28
11	16	10	28
12	18	11	28
13	20	12	28
14	22	13	9
15	23	12	22
16	26	14	28
17	28	12	10
18	30	15	28
19	31	16	28
20	32	17	22

表 注解

确定

Figure 2.1 enrollment_train#.csv

表 (6 个字段, 120,542 条记录) #1

	enrollment_id	username	course_id	course_num	nondropout_num	dropout
1	1	0	28	6	4.000	0
2	3	1	9	3	3.000	0
3	4	2	28	2	1.000	0
4	5	3	9	1	1.000	0
5	6	4	22	1	1.000	0
6	7	5	9	4	0.000	1
7	9	6	28	2	1.000	1
8	12	7	28	1	1.000	0
9	13	8	4	3	3.000	0
10	14	9	28	1	0.000	1
11	16	10	28	6	4.000	0
12	18	11	28	1	1.000	0
13	20	12	28	6	4.000	0
14	22	13	9	3	1.000	1
15	23	12	22	6	4.000	0
16	26	14	28	3	2.000	0
17	28	12	10	6	4.000	1
18	30	15	28	1	1.000	0
19	31	16	28	2	0.000	1
20	32	17	22	7	5.000	0

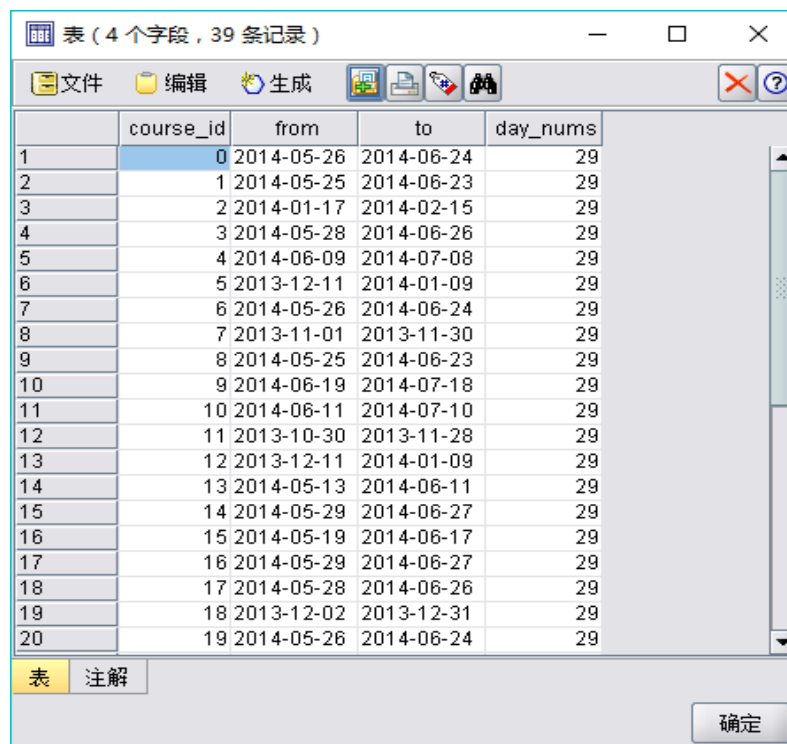
表 注解

确定

Figure 2.2 enrollment_test#.csv

The date.csv is about the first day and the last day of the course in log. The dictionary about the course we used is as the same enrollment_train.csv. We add a column(day_nums) to represent how long the course lasts(to - from). Then save it in

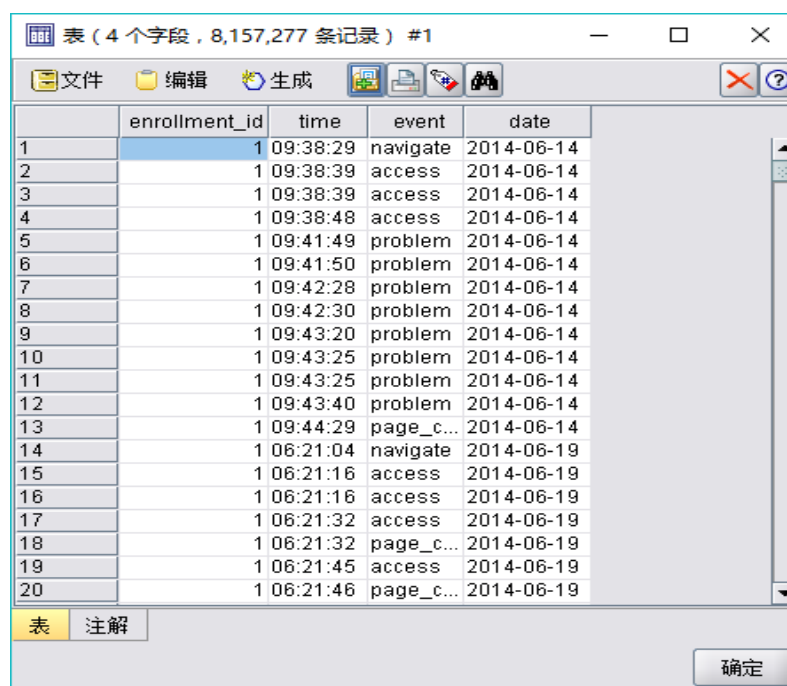
files date#.csv. The result is shown below.



	course_id	from	to	day_nums
1	0	2014-05-26	2014-06-24	29
2	1	2014-05-25	2014-06-23	29
3	2	2014-01-17	2014-02-15	29
4	3	2014-05-28	2014-06-26	29
5	4	2014-06-09	2014-07-08	29
6	5	2013-12-11	2014-01-09	29
7	6	2014-05-26	2014-06-24	29
8	7	2013-11-01	2013-11-30	29
9	8	2014-05-25	2014-06-23	29
10	9	2014-06-19	2014-07-18	29
11	10	2014-06-11	2014-07-10	29
12	11	2013-10-30	2013-11-28	29
13	12	2013-12-11	2014-01-09	29
14	13	2014-05-13	2014-06-11	29
15	14	2014-05-29	2014-06-27	29
16	15	2014-05-19	2014-06-17	29
17	16	2014-05-29	2014-06-27	29
18	17	2014-05-28	2014-06-26	29
19	18	2013-12-02	2013-12-31	29
20	19	2014-05-26	2014-06-24	29

Figure 2.3 date#.csv

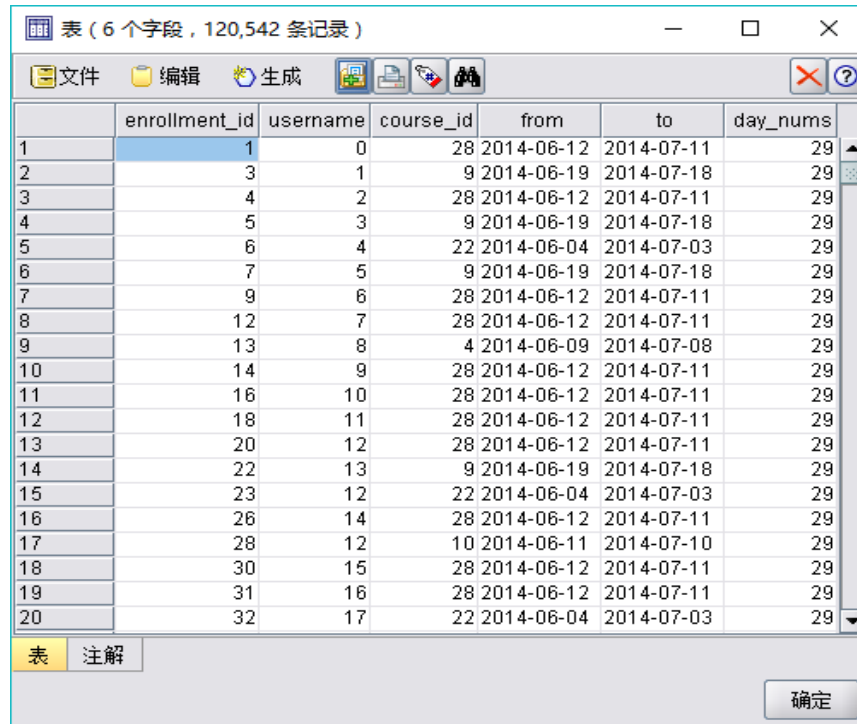
Then we need to process log_train.csv. First we delete some attributes(source, object) which regarded useless for our data mining. Then, we split the attribute of time to time and date. Finally save it in files log_train#.csv. The result is shown below.



	enrollment_id	time	event	date
1	1	09:38:29	navigate	2014-06-14
2	1	09:38:39	access	2014-06-14
3	1	09:38:39	access	2014-06-14
4	1	09:38:48	access	2014-06-14
5	1	09:41:49	problem	2014-06-14
6	1	09:41:50	problem	2014-06-14
7	1	09:42:28	problem	2014-06-14
8	1	09:42:30	problem	2014-06-14
9	1	09:43:20	problem	2014-06-14
10	1	09:43:25	problem	2014-06-14
11	1	09:43:25	problem	2014-06-14
12	1	09:43:40	problem	2014-06-14
13	1	09:44:29	page_c...	2014-06-14
14	1	06:21:04	navigate	2014-06-19
15	1	06:21:16	access	2014-06-19
16	1	06:21:16	access	2014-06-19
17	1	06:21:32	access	2014-06-19
18	1	06:21:32	page_c...	2014-06-19
19	1	06:21:45	access	2014-06-19
20	1	06:21:46	page_c...	2014-06-19

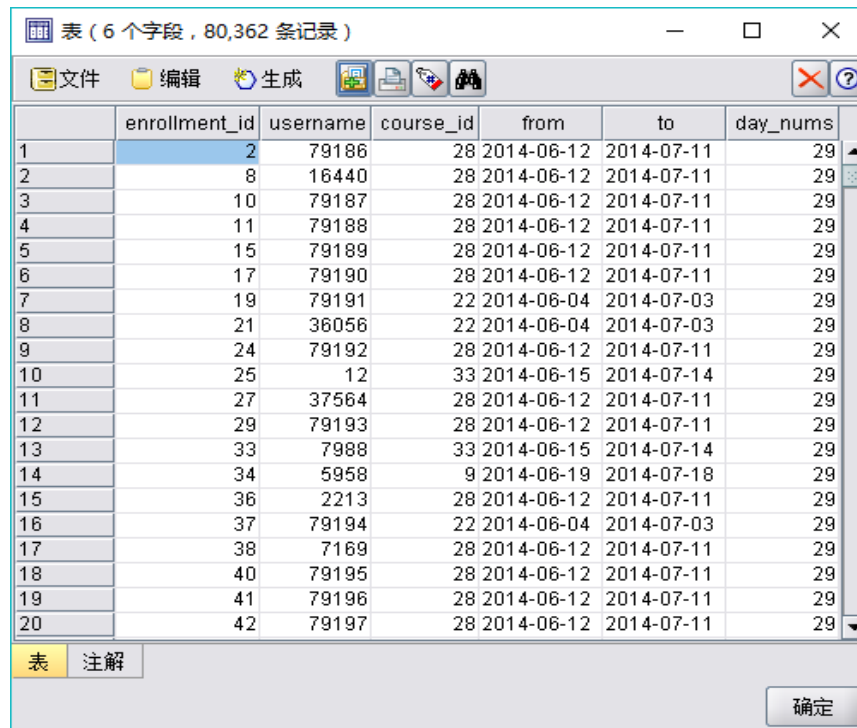
Figure 2.4 log_train#.csv

We make date#.csv left link to the enrollment_train#.csv. At the same time, we need to process date#.csv and enrollment_test#.csv at the same way. Then save them in files course_enrollment_train#.csv and course_enrollment_test#.csv. The result is shown below.



	enrollment_id	username	course_id	from	to	day_nums
1	1	0	28	2014-06-12	2014-07-11	29
2	3	1	9	2014-06-19	2014-07-18	29
3	4	2	28	2014-06-12	2014-07-11	29
4	5	3	9	2014-06-19	2014-07-18	29
5	6	4	22	2014-06-04	2014-07-03	29
6	7	5	9	2014-06-19	2014-07-18	29
7	9	6	28	2014-06-12	2014-07-11	29
8	12	7	28	2014-06-12	2014-07-11	29
9	13	8	4	2014-06-09	2014-07-08	29
10	14	9	28	2014-06-12	2014-07-11	29
11	16	10	28	2014-06-12	2014-07-11	29
12	18	11	28	2014-06-12	2014-07-11	29
13	20	12	28	2014-06-12	2014-07-11	29
14	22	13	9	2014-06-19	2014-07-18	29
15	23	12	22	2014-06-04	2014-07-03	29
16	26	14	28	2014-06-12	2014-07-11	29
17	28	12	10	2014-06-11	2014-07-10	29
18	30	15	28	2014-06-12	2014-07-11	29
19	31	16	28	2014-06-12	2014-07-11	29
20	32	17	22	2014-06-04	2014-07-03	29

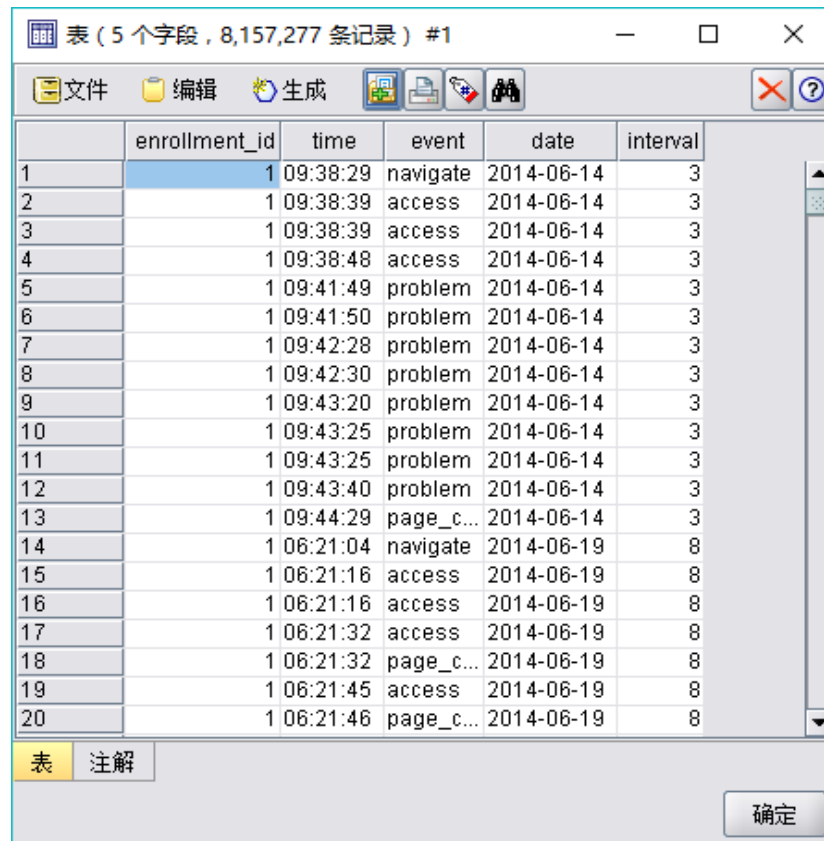
Figure 2.5 course_enrollment_train#.csv



	enrollment_id	username	course_id	from	to	day_nums
1	2	79186	28	2014-06-12	2014-07-11	29
2	8	16440	28	2014-06-12	2014-07-11	29
3	10	79187	28	2014-06-12	2014-07-11	29
4	11	79188	28	2014-06-12	2014-07-11	29
5	15	79189	28	2014-06-12	2014-07-11	29
6	17	79190	28	2014-06-12	2014-07-11	29
7	19	79191	22	2014-06-04	2014-07-03	29
8	21	36056	22	2014-06-04	2014-07-03	29
9	24	79192	28	2014-06-12	2014-07-11	29
10	25	12	33	2014-06-15	2014-07-14	29
11	27	37564	28	2014-06-12	2014-07-11	29
12	29	79193	28	2014-06-12	2014-07-11	29
13	33	7988	33	2014-06-15	2014-07-14	29
14	34	5958	9	2014-06-19	2014-07-18	29
15	36	2213	28	2014-06-12	2014-07-11	29
16	37	79194	22	2014-06-04	2014-07-03	29
17	38	7169	28	2014-06-12	2014-07-11	29
18	40	79195	28	2014-06-12	2014-07-11	29
19	41	79196	28	2014-06-12	2014-07-11	29
20	42	79197	28	2014-06-12	2014-07-11	29

Figure 2.6 course_enrollment_test#.csv

We make log_train# left link to the course_enrollment_train#.csv. We add a column(interval) to represent the interval of the data of log and the start time of the course. The interval equal date(log_train#.csv) minus (course_enrollment_train#.csv). Then save it in files log_train_final#.csv. The result is shown below.



	enrollment_id	time	event	date	interval
1	1	09:38:29	navigate	2014-06-14	3
2	1	09:38:39	access	2014-06-14	3
3	1	09:38:39	access	2014-06-14	3
4	1	09:38:48	access	2014-06-14	3
5	1	09:41:49	problem	2014-06-14	3
6	1	09:41:50	problem	2014-06-14	3
7	1	09:42:28	problem	2014-06-14	3
8	1	09:42:30	problem	2014-06-14	3
9	1	09:43:20	problem	2014-06-14	3
10	1	09:43:25	problem	2014-06-14	3
11	1	09:43:25	problem	2014-06-14	3
12	1	09:43:40	problem	2014-06-14	3
13	1	09:44:29	page_c...	2014-06-14	3
14	1	06:21:04	navigate	2014-06-19	8
15	1	06:21:16	access	2014-06-19	8
16	1	06:21:16	access	2014-06-19	8
17	1	06:21:32	access	2014-06-19	8
18	1	06:21:32	page_c...	2014-06-19	8
19	1	06:21:45	access	2014-06-19	8
20	1	06:21:46	page_c...	2014-06-19	8

Figure 2.7 log_train_final#.csv

Final we merge truth_train.csv and enrollment_train#.csv. We add columns course_num,and nondropout_num. course_num means how many course the user choose. nondropout_num means the numbers of course the user continuing study. If more than half of this user's course is on, we make the dropout equal 0, else make drop equals 1. Then save it in files enrollment_dropout#.csv". The result is shown below.

	enrollment_id	username	course_id	course_num	nondropout_num	dropout
1	1	0	28	6	4.000	0
2	3	1	9	3	3.000	0
3	4	2	28	2	1.000	0
4	5	3	9	1	1.000	0
5	6	4	22	1	1.000	0
6	7	5	9	4	0.000	1
7	9	6	28	2	1.000	1
8	12	7	28	1	1.000	0
9	13	8	4	3	3.000	0
10	14	9	28	1	0.000	1
11	16	10	28	6	4.000	0
12	18	11	28	1	1.000	0
13	20	12	28	6	4.000	0
14	22	13	9	3	1.000	1
15	23	12	22	6	4.000	0
16	26	14	28	3	2.000	0
17	28	12	10	6	4.000	1
18	30	15	28	1	1.000	0
19	31	16	28	2	0.000	1
20	32	17	22	7	5.000	0

Figure 2.8 enrollment_dropout#.csv

3.Feature Extraction

After we analyze the data, we should find the useful information from the data provided by the sponsor. What's more, by the preprocessing operation, we can get the following files: “log_train_final#.csv”, “course_enrollment_train#.csv” and “enrollment_dropout#.csv”.

3.1 Basic Features

The most important information is the event in the log file. The log record the action rule of the user. As we all know that a user will finish the course if he have a large number of events. Because large number of events means the user must be interested in the course and complete the course according to the course schedule. On the contrary, if the user has few number of events, it means the user has little interest in the course, so there is a great possibility that the user will drop the course. The user just enroll the course to see whether the content of the course is suitable for him.

According to the analysis above, we contract 4 basic features to describe a user:

Table 3.1 Basic features

Feature name	Feature description
course_num	the total number of courses a user have enrolled.
nondropout_num	the total number of courses a user don't drop or he passed.
dropout	the user dropout a course, corresponding to the concrete course. The value can only be 0 and 1.1 means the user drop the course, 0 is stand for the userpass the course.day to thirtieth day
nondrop_precent	the ratio of a user don't drop courses, calculated by nondropout_num divided course_num.

In this part, we can get 4 basic features. As we can see in the below picture.

表 (7 个字段, 120,542 条记录) #1

	enrollment_id	username	course_id	course_num	nondropout_num	dropout	nondrop_precent
1	1	0	28	6	4.000	0	0.667
2	3	1	9	3	3.000	0	1.000
3	4	2	28	2	1.000	0	0.500
4	5	3	9	1	1.000	0	1.000
5	6	4	22	1	1.000	0	1.000
6	7	5	9	4	0.000	1	0.000
7	9	6	28	2	1.000	1	0.500
8	12	7	28	1	1.000	0	1.000
9	13	8	4	3	3.000	0	1.000
10	14	9	28	1	0.000	1	0.000
11	16	10	28	6	4.000	0	0.667
12	18	11	28	1	1.000	0	1.000
13	20	12	28	6	4.000	0	0.667
14	22	13	9	3	1.000	1	0.333
15	23	12	22	6	4.000	0	0.667
16	26	14	28	3	2.000	0	0.667
17	28	12	10	6	4.000	1	0.667
18	30	15	28	1	1.000	0	1.000
19	31	16	28	2	0.000	1	0.000
20	32	17	22	7	5.000	0	0.714

表 注解

确定

Figure 3.1 Features describe users' actions.

3.2 Effective Features

There have 7 different kinds event. Each event means the student do the different

action and may visit the different contents. So, we go on extracting feature. This time, we count times of different event type of an enrollment_id, we also count the response of each event which names “browser_count” and “server_count”. To save memory and improve efficiency, we change the event type from string to integer. Such as, problem = 1, video = 2, access = 3, wiki = 4, discussion = 5, navigate = 6, page_close = 7, so 1-7 means the count of different event type. “problem”: working on course assignment; “video”: watching course videos; “access”: accessing other course objects except videos and assignments; “wiki”: accessing the course wiki; “discussion”: accessing the course forum; “navigate”: navigating to other part of the course. “page_close”: closing the web page.

Here, we define the attributes of problem, video, access, wiki and discussion (1,2,4,5) as the effective events, as the other events are transition event, without useful information in it.

Another effective definition is about the last time of special event.

Then, we define 3 attributes about the overall characteristic of event. From the log_train.csv file, we can see the event is consist of continuous events mark by the start moment. We denote a special event’s duration time by the next event’s start time minus present event’s start time. And the minus unit is minute. What’s more, if the operation isn’t video and the gap of two event is more than a hour, we omit the idle time and recalculate the duration time at the start time of the later event.

According to the analysis above, we extract 3 effective features:

Table 3.2 Effective features 1

Feature name	Feature description
all_opnum	the total number of all events involved in an enrollment.
valid_opnum	the total number of valid events involved in an enrollment.
last_minutes	the last time of all events involved in an enrollment.

	enrollment_id	interval	last_minutes	valid_opnum	all_opnum
1	1	3	7.000	8	13
2	1	8	39.883	6	24
3	1	10	1.517	0	7
4	1	11	27.800	4	13
5	1	12	1.000	0	1
6	1	13	7.217	0	4
7	1	16	57.133	28	67
8	1	19	6.850	0	6
9	1	20	904.117	3	17
10	1	21	494.583	14	51
11	1	23	259.600	28	56
12	1	24	34.033	0	4
13	1	29	15.283	0	4
14	1	30	79.967	25	47
15	3	1	3.850	15	28
16	3	5	1.050	0	3
17	3	8	5.150	13	30
18	3	13	1.633	0	10
19	3	14	6.233	32	48
20	3	16	8.567	44	64

Figure 3.2 Effective Features 1

In order to make the feature more effective, we calculate the single day of the above feature during the 30 days.

Table 3.3 Single day of the above feature

Feature name	Feature description
all_opnum_1...30	The single day of all events involved in an enrollment.
valid_opnum_1...30	The single day of the total number of valid events involved in an enrollment.
last_minutes_1...30	The single day of the last time of all events involved in an enrollment.

Here, we can get another $3 \times 30 = 90$ effective features.

In order to improve the amount of effective features and make the prediction more accurately. Here, we split 30 days into 3 parts and 10 days as a signal. We use “pre” means the premier 10 days; “mid” means the middle 10 days; “last” means the last 10 days. What’s more, we can get the statistical characteristics of the above feature. “min” means the minimum value, “max” means the maximum value, “sum” means the summarize of data, “mean” means the data centralization trends, “std” reflects the degree of discretization of a data set.

Table 3.4 Effective features 2

Feature name	Feature description
pre_min/max/sum/mean/std	the statistical characteristics of the premier 10 days.
mid_min/max/sum/mean/std	the statistical characteristics of the middle 10 days.
last_min/max/sum/mean/std	the statistical characteristics of the last 10 days.
thirty_day_min/max/sum/mean/std	the statistical characteristics of the total 30 days.

Here, we can get another $4 \times 5 = 20$ effective features.

In this part, we can get 110 effective features. As we can see in the below picture.

	enrollment_id	all_opnum_1	all_opnum_2	all_opnum_3	all_opnum_4	all_opnum_5	all_opnum_6	all_opnum_7
1	1	0	0	13	0	0	0	0
2	3	28	0	0	0	3	0	0
3	4	0	0	0	20	11	0	0
4	5	2	0	145	0	0	0	0
5	6	0	0	0	0	0	0	0
6	7	20	0	125	0	0	0	0
7	9	0	0	0	0	0	0	0
8	12	0	0	0	0	0	0	0
9	13	0	0	0	9	0	34	0
10	14	0	0	0	0	0	0	0
11	16	0	0	0	0	48	17	11
12	18	0	0	0	24	0	6	0
13	20	0	0	30	33	5	0	0
14	22	0	0	0	0	0	0	0
15	23	0	0	0	0	0	0	0
16	26	0	0	0	0	0	0	0
17	28	0	0	0	0	0	11	0
18	30	0	0	0	0	0	0	0
19	31	0	0	26	0	0	4	0
20	32	0	0	0	0	0	0	0

Figure 3.3 Effective Features 2

3.3 Final Features

After the above analysis, in this part, we can summarize the basic features and the effective features as the final features. The size is 114(4+110) expect for the following 3 attributes: “enrollment_id”, “username”, “course_id”.

表 (117 个字段, 120,542 条记录)

	rty_day_mean	thirty_day_std	username	course_id	course_num	nondropout_num	dropout	nondrop_precent
1	10	18.700	0	28	6	4.000	0	0.667
2	9	18.360	1	9	3	3.000	0	1.000
3	3	5.890	2	28	2	1.000	0	0.500
4	21	40.730	3	9	1	1.000	0	1.000
5	0	3.200	4	22	1	1.000	0	1.000
6	15	31.320	5	9	4	0.000	1	0.000
7	3	7.420	6	28	2	1.000	1	0.500
8	4	6.790	7	28	1	1.000	0	1.000
9	15	35.760	8	4	3	3.000	0	1.000
10	3	6.390	9	28	1	0.000	1	0.000
11	11	22.200	10	28	6	4.000	0	0.667
12	5	9.750	11	28	1	1.000	0	1.000
13	4	8.520	12	28	6	4.000	0	0.667
14	6	25.580	13	9	3	1.000	1	0.333
15	3	10.550	12	22	6	4.000	0	0.667
16	8	23.050	14	28	3	2.000	0	0.667
17	3	9.160	12	10	6	4.000	1	0.667
18	0	1.620	15	28	1	1.000	0	1.000
19	8	14.720	16	28	2	0.000	1	0.000
20	2	6.410	17	22	7	5.000	0	0.714

表 注解

确定

Figure 3.4 Final Features

4. Choose models and learn classifiers

4.1 The introduction of chosen models

4.1.1 Logistic regression

Logistic regression was developed by statistician David Cox in 1958. The binary logistic model is used to estimate the probability of a binary response based on one or more predictor (or independent) variables (features). It allows one to say that the presence of a risk factor increases the odds of a given outcome by a specific factor.

In statistics, logistic regression, or logit regression, or logit model is a regression model where the dependent variable (DV) is categorical. The case of a binary dependent variable—that is, where the output can take only two values, "0" and "1", which represent outcomes such as pass/fail, win/lose, alive/dead or healthy/sick.

The case of mooc dropout prediction exactly is binary Classification. The Logistic regression should be valid.

4.1.2 Support vector machines

In machine learning, support vector machines (SVM) are supervised learning models with associated learning algorithms that analyze data used for classification

and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

4.1.3 Random forests

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

4.1.4 Artificial neural networks

An ANN is based on a collection of connected units called artificial neurons (analogous to biological neurons in an animal brain). Each connection (synapse) between neurons can transmit a signal to another neuron. The receiving (postsynaptic) neuron can process the signal(s) and then signal downstream neurons connected to it. Neurons may have a state, generally represented by real numbers, typically between 0 and 1. Neurons and synapses may also have a weight that varies as learning proceeds, which can increase or decrease the strength of the signal that it sends downstream. Further, they may have a threshold such that only if the aggregate signal is below (or above) that level is the downstream signal sent.

Typically, neurons are organized in layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first (input), to the last (output) layer, possibly after traversing the layers multiple times. In artificial networks with multiple hidden layers, the initial layers might detect primitives and

their output is fed forward to deeper layers who perform more abstract generalizations and so on until the final layers perform the complex object recognition.

4.1.5 Gradient boosting

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

Like other boosting methods, gradient boosting combines weak "learners" into a single strong learner in an iterative fashion. It is easiest to explain in the least-squares regression setting, where the goal is to "teach" a model F to predict values in the form $\hat{y} = F(x)$ by minimizing the mean squared error $(\hat{y} - y)^2$, averaged over some training set of actual values of the output variable y .

4.2 Learn classifiers

4.2.1 scikit-learn library

There are many open source library about machine learning. I mainly use the scikit-learn library, which work on the python.

Scikit-learn, also called sklearn, includes three modules: feature extraction, data processing and model evaluation and supports four machine learning algorithms, including classification, regression, dimension reduction and clustering. Sklearn is an extension of Scipy, built on the basis of the Numpy and matplotlib libraries. The advantages of these modules can greatly improve the efficiency of machine learning. Sklearn has a complete documentation, easy to get started, with a wealth of API, popular in academia. Sklearn has encapsulated a large number of machine learning algorithms, including LIBSVM and LIBLINEAR. At the same time sklearn built a large number of data sets, saving access to data collection and finishing time.

4.2.2 k-fold cross-validation

Cross-validation is a way to predict the fit of a model to a hypothetical validation set when an explicit validation set is not available. In k-fold cross-validation, the original sample is randomly partitioned into k equal sized subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k-1 subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be averaged to produce a single estimation. The advantage of this method over repeated random sub-sampling (see below) is that all observations are used for both training and validation, and each observation is used for validation exactly once. 10-fold cross-validation is commonly used. So in the process of training models, I apply 10-fold cross-validation.

5. The performance of classifiers

5.1 The accuracy of classifiers

I split the 70% samples into train dataset and 30% into test dataset. Train classifiers applying 10-fold cross-validation with train dataset. And test classifiers with test dataset. The performance of classifiers is as follow.

Table 5.1 The performance of classifiers

Classifier	The mean score of 10-fold cross-validation	The accuracy on test dataset	Training time
Logistic regression	0.944986	0.94345	3.85 mins
Random forests	0.950331	0.94859	3.87 mins
Artificial neural networks	0.945401	0.94505	300.95 mins
Support vector machines	0.941609	0.94016	707.75 mins
Gradient boosting	0.952962	0.95188	10.35 mins

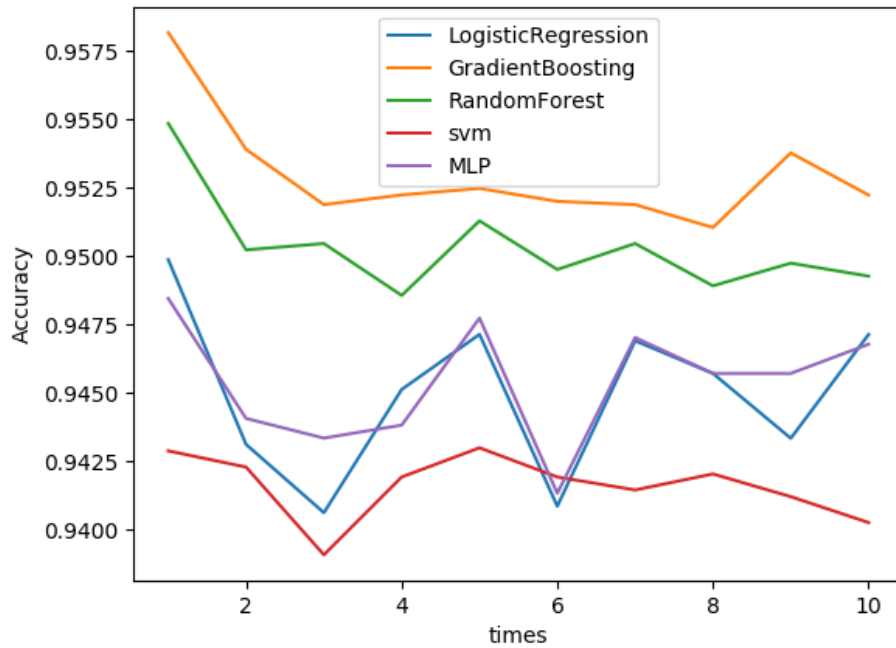


Figure5.1 The score of 10-fold cross-validation

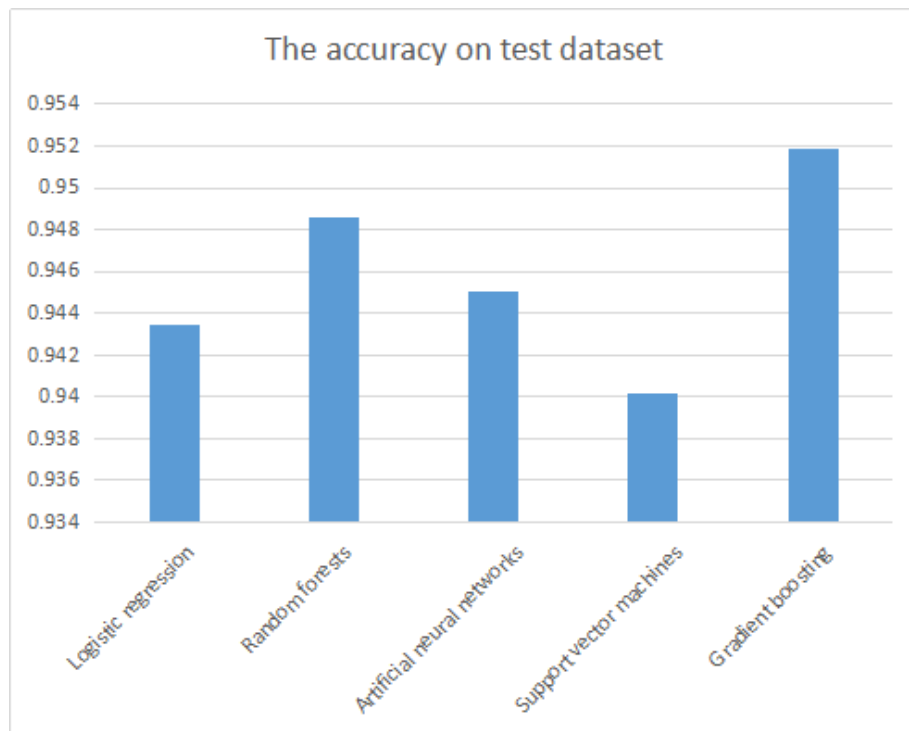


Figure 5.2 The accuracy on test dataset

From above table and histogram, we know the performance of Gradient Boosting is best. The mean score of 10-fold cross-validation is 0.952962, and the score of every time better than other classifiers. The accuracy on test dataset is 0.95188, and the training time is 10.35 minutes.

5.2 The ROC curve of classifiers

The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as sensitivity, recall or probability of detection in machine learning. The false-positive rate is also known as the fall-out or probability of false alarm and can be calculated as $(1 - \text{specificity})$. The ROC curve is thus the sensitivity as a function of fall-out.

When using normalized units, the area under the curve (often referred to as simply the AUC) is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (assuming 'positive' ranks higher than 'negative').

The area under the curve can reflect the accuracy of classification. The larger area represents higher accuracy. The ROC curve of classifiers are as follow.

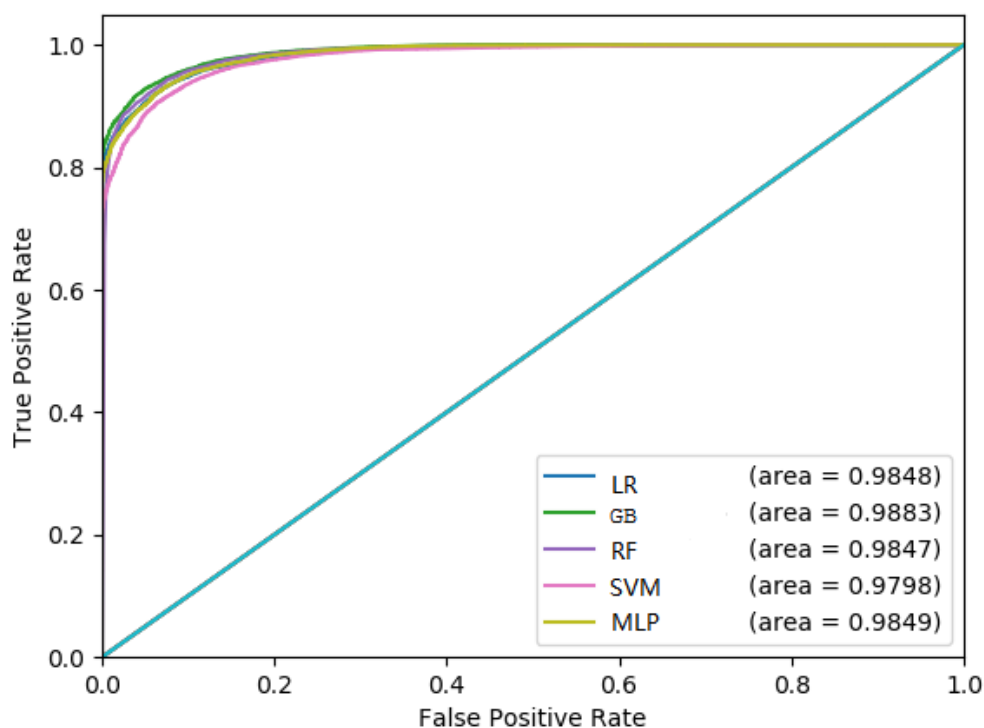


Figure 5.3 The ROC curve of classifiers

From ROC curve we can also know that the area under Gradient Booting curve is largest. So the Gradient Booting model have best performance.

5.3 Other metrics of classifiers

Some time, it may be invalid to evaluate classifiers with accuracy. So we also need to apply other metrics precision, F1 score and recall evaluating classifiers. The precision is the ratio of true positive samples in marked positive samples. The recall is the ratio of marked positive samples in true positive samples. F1 score is the combination of precision and recall. Its formula is as follows.

$$F = \frac{2 * precision * recall}{precision + recall}$$

The result of these metrics of classifiers are as follows.

Table 5.2 Other metrics of classifiers

classifiers	class	precision	recall	f1-score	support
LogisticRegression	0	0.89	0.83	0.86	7439
	1	0.96	0.97	0.96	28724
	avg/total	0.94	0.94	0.94	36163
GradientBoosting	0	0.90	0.86	0.88	7439
	1	0.96	0.98	0.97	28724
	avg/total	0.95	0.95	0.95	36163
RandomForest	0	0.88	0.87	0.87	7439
	1	0.97	0.97	0.97	28724
	avg/total	0.95	0.95	0.95	36163
SVM	0	0.87	0.83	0.85	7439
	1	0.96	0.97	0.96	28724
	avg/total	0.94	0.94	0.94	36163
ANN	0	0.89	0.84	0.86	7439
	1	0.96	0.97	0.97	28724
	avg/total	0.94	0.95	0.94	36163

From the table we know that all classifiers can recognize class 1 better, may because there are more samples for class 1. RandomForest have the highest precision 0.97, and GradientBoosting has the highest recall 0.98 for class 1. GradientBoosting, RandomForest, ANN have the same high f1-score 0.97 for class 1. The class 1 represents dropout. So our classifiers can recognize dropout well.

6. Conclusion

Through the project and the Data Mining course, I learn how to begin a data mining. First, data understand, data preprocess are necessary. Second, feature engineering is very important, and we should extract valid and efficient features. Final, we should select proper models. Moreover, I learned how to use scikit-learn library, how to evaluate a classifier. In addition, I learned the importance of cooperation with partners in work.

Data Mining is an interesting course. What's more lucky is that it taught by Professor Ye. Not only have I learned the base knowledge during this course, but also more other from Professor Ye. Thanks for Professor Ye and tutors.