CS150A Database
Course Project

Yubo Liu
ID: 2020533187
liuyb@shanghaitech.edu.cn

Junlei Wu
ID: 2020533017
wujl@shanghaitech.edu.cn

Guideline

Compared with developing a novel machine learning algorithm, building a machine learning system is less theoretical but more engineering, so it is important to get your hands dirty. To build an entire machine learning system, you have to go through some essential steps. We have listed 5 steps which we hope you to go through. Read the instructions of each section before you fill in. You are free to add more sections.
If you use PySpark to implement the algorithms and want to earn some additional points, you should also report your implementation briefly in the last section.

1 Explore the dataset

Instruction:
Explore the given dataset, report your findings about the dataset. You should not repeat the information provided in the 'Data Format' section of project.pdf. Instead, you can report the data type of each feature, the distribution of different values of some important features(maybe with visualization), is there any missing value, etc

Your work below:
From the *data_exploration.ipynb* file, we have a sense of the dataset. The data type of each feature are shown in table 1:

Row	int64
Anon Student Id	object
Problem Hierarchy	object
Problem Name	object
Problem View	int64
Step Name	object
Step Start Time	object
First Transaction Time	object
Correct Transaction Time	object
Step End Time	object
Step Duration (sec)	float64
Correct Step Duration (sec)	float64
Error Step Duration (sec)	float64
Correct First Attempt	int64
Incorrects	int64
Hints	int64
Corrects	int64
KC(Default)	object
Opportunity(Default)	object

Table 1: data type of each feature

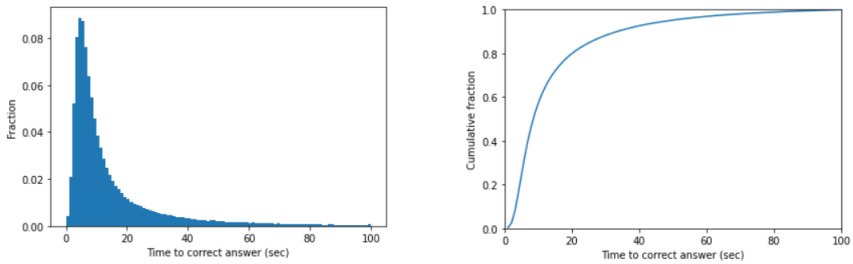
In order to find out how long does it take a student to solve any problem step on average, we see the

description of column "Correct Step Duration". The result is shown in table2.

count	mean	std	min	25%	50%	75%	max
181599	17.924024	35.179534	0	5	8	17	1067

Table 2: Description of column "Correct Step Duration"

So ignoring all the students that did not solve a problem step correctly, the average duration for any problem step was about 18 seconds. We visualize the distribution of column "Correct Step Duration" to understand it more intuitively (shown in figure1).



(a) Distribution (b) Cumulative distribution function

Figure 1: The distribution of column "Correct Step Duration"

2 Data cleaning

Instruction:
Some people treat data cleaning as a part of feature engineering, however we separate them here to make your work clearer. In this section, you should mainly deal with the missing values and the outliers. You can also do some data normalization here.

Your work below:
Since the values of columns from "Step Start Time" to "Corrects" are not provided, we decide to delete these columns in train dataset.
Then, we focus on "Correct First Attempt", "Anon Student Id", "Step Name", "Problem Name", "KC(Default)" since we decide to calculate Correct First Attempt rate (CFAR) of student, step, problem and KC. However, sometimes these columns contains no value, which makes it impossible for us to calculate the CFAR of the column with no value. To solve it, we use mean of CFAR of other features to replace it.
In addition, we calculate the KC number for each row and normalize opportunity by calculating the mean.

3 Feature engineering

Instruction:
In this section, you should select a subset of features and transform them into a data matrix which can be feed into the learning model you choose. Report your work with reasons.

Your work below:
As stated in part 2, we focus on "Correct First Attempt", "Anon Student Id", "Step Name", "Problem Name". The reason why we choose these columns is because we think that the CFA is affected by the ability of the person, the difficulty of the problem and the method selected and these indicators can be expressed by these columns. We define a new feature Correct First Attempt Rate(CFAR), which means the ratio of all correct first attempts to all first attempts. And we calculate the CFAR of these features to measure their performance in these areas.
What's more, we calculate the number of KC a step needs and the oppurtunity for the KCs because

we think the number KC used in the problem step may also affect the students' behavior, thus we generate another feature KC_number.
We delete "Problem Hierarchy", "ProblemName" and "Step Name" when feed the data to machine learning algorithm since they can be represented by the CFAR of them and we don't want it to disturb the prediction.

4 Learning algorithm

Instruction:
In this section, you should describe the learning algorithm you choose and state the reasons why you choose it.

Your work below:
We've tried different machine learning algorithms including Decision Tree Classifier, Logistic Regression, LightGBM, Gradient Boosting Classifier, Random Forest Classifier to predict the test data. We use these algorithm and tune the parameter to find a relatively good assignment, and then compare the performance of each algorithm by calculating the RMSE to find the best algorithm, the results are shown in table 3.

ML model	RMSE
Decision Tree Classifier	0.4977
Logistic Regression	0.4191
LightGBM	0.4046
Gradient Boosting Classifier	0.4155
Random Forest Classifier	0.3989

Table 3: Performance of different algorithm

From the table, we can find that Random Forest Classifier has best performance. So it is naturally that we decide to use the result of Random Forest Classifier.

5 Hyperparameter selection and model performance

Instruction:
In this section, you should describe the way you choose the hyperparameters of your model, also compare the performance of the model with your chosen hyperparamters with models with sub-optimal hyperparameters

Your work below:
We tune the parameters of each algorithm to find a relatively good assignment and then compare them.
Since Decision Tree algorithm shows a obviously high RMSE, so we we don't tune the parameters of it because we didn't think it will be a good choice.
In LightGBM, learning rate's default value is 0.1, and we find that when the value is 0.01 with fixed other parameter, the RMSE decreases from 0.4173 to 0.4046, so we choose learning rate to be 0.11. We also change the other parameters like n_estimators, min_child_weight, num_leaves, subsample, but did not find a lower RMSE.
In LogisticRegression, we change the value of penalty, tol, intercept_scaling, solver, max_iter to find a better assignment of parameter and find out that the default parameter is good enough.
In GradientBoostingClassifier, we change the value of penalty, tol, intercept_scaling, solver, max_iter to find a better assignment of parameter and find out that the default parameter is good enough.
In Random Forest Classifier, we change learning_rate, n_estimators, criterion, min_samples_split, min_samples_split, criterion to find a better choice. Since we choose Random Forest Classifier, we will show the detail of parameter selection of this algorithm.

n_estimators	criterion	max_depth	min_samples_split	random_state	RMSE
100	gini	None	2	7	0.4262
100	entropy	None	2	7	0.4245
200	entropy	None	2	7	0.4209
300	entropy	None	2	7	0.4209
200	entropy	None	0.1	7	0.4064
200	entropy	None	0.01	7	0.4064
200	entropy	30	0.01	7	0.4064
200	gini	30	0.01	7	0.3989

Table 4: Performance of different parameter selections

So finally, the parameters we choose for the algorithm is:

n_estimators=200
max_depth=30
min_samples_split=0.01
criterion = gini
random_state = 7

And the RMSE of our implement is 0.3989.