Knowledge Tracing Project

Kaige Yang
University College London
Kaige.yang.11@ucl.ac.uk

1 Work Flow

This project we follow the following work flow.

1. Project Overview
2. Data Understanding
3. Brain Storming
4. Data Cleaning
5. Exploratory Data Analysis
6. Feature Engineering
7. Feature Selection
8. Models
9. Model Selection
10. Model Fine-Tune
11. Further Improvement

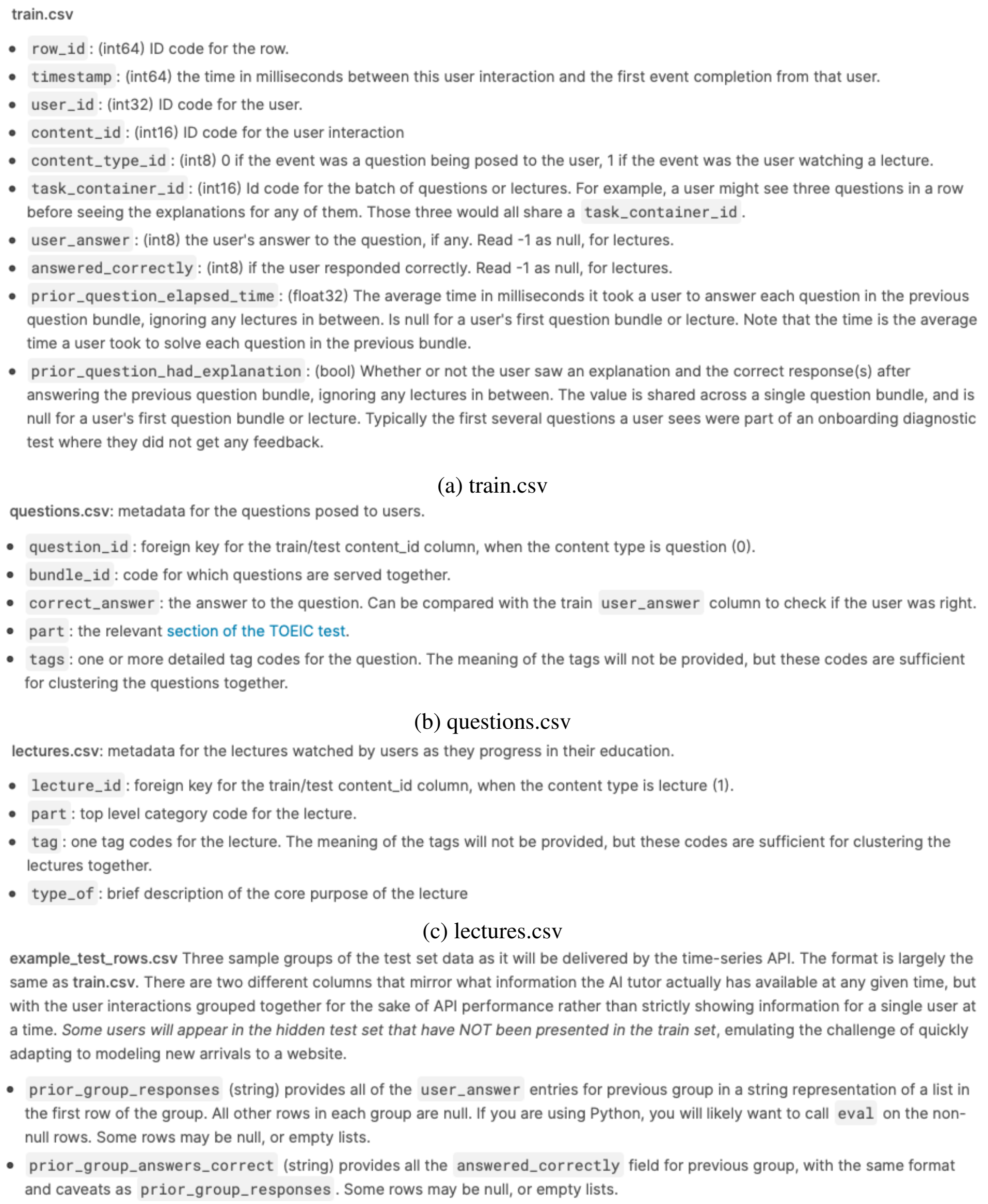
2 Project Overview

The goal is to predict whether students are able to answer correctly next question based on their previous learning experience. The dataset contains information regarding students' previous learning experience (lectures watched and questions answered), description and lectures. This is a time-series prediction problem.

Evaluation Metric: Area under the ROC curve between the predicted probability and the observed target.

Report Draft.

3 Data Understanding



train.csv

- `row_id` : (int64) ID code for the row.
- `timestamp` : (int64) the time in milliseconds between this user interaction and the first event completion from that user.
- `user_id` : (int32) ID code for the user.
- `content_id` : (int16) ID code for the user interaction
- `content_type_id` : (int8) 0 if the event was a question being posed to the user, 1 if the event was the user watching a lecture.
- `task_container_id` : (int16) Id code for the batch of questions or lectures. For example, a user might see three questions in a row before seeing the explanations for any of them. Those three would all share a `task_container_id` .
- `user_answer` : (int8) the user's answer to the question, if any. Read -1 as null, for lectures.
- `answered_correctly` : (int8) if the user responded correctly. Read -1 as null, for lectures.
- `prior_question_elapsed_time` : (float32) The average time in milliseconds it took a user to answer each question in the previous question bundle, ignoring any lectures in between. Is null for a user's first question bundle or lecture. Note that the time is the average time a user took to solve each question in the previous bundle.
- `prior_question_had_explanation` : (bool) Whether or not the user saw an explanation and the correct response(s) after answering the previous question bundle, ignoring any lectures in between. The value is shared across a single question bundle, and is null for a user's first question bundle or lecture. Typically the first several questions a user sees were part of an onboarding diagnostic test where they did not get any feedback.

(a) train.csv

questions.csv: metadata for the questions posed to users.

- `question_id` : foreign key for the train/test content_id column, when the content type is question (0).
- `bundle_id` : code for which questions are served together.
- `correct_answer` : the answer to the question. Can be compared with the train `user_answer` column to check if the user was right.
- `part` : the relevant section of the TOEIC test.
- `tags` : one or more detailed tag codes for the question. The meaning of the tags will not be provided, but these codes are sufficient for clustering the questions together.

(b) questions.csv

lectures.csv: metadata for the lectures watched by users as they progress in their education.

- `lecture_id` : foreign key for the train/test content_id column, when the content type is lecture (1).
- `part` : top level category code for the lecture.
- `tag` : one tag codes for the lecture. The meaning of the tags will not be provided, but these codes are sufficient for clustering the lectures together.
- `type_of` : brief description of the core purpose of the lecture

(c) lectures.csv

example_test_rows.csv Three sample groups of the test set data as it will be delivered by the time-series API. The format is largely the same as **train.csv**. There are two different columns that mirror what information the AI tutor actually has available at any given time, but with the user interactions grouped together for the sake of API performance rather than strictly showing information for a single user at a time. *Some users will appear in the hidden test set that have NOT been presented in the train set*, emulating the challenge of quickly adapting to modeling new arrivals to a website.

- `prior_group_responses` (string) provides all of the `user_answer` entries for previous group in a string representation of a list in the first row of the group. All other rows in each group are null. If you are using Python, you will likely want to call `eval` on the non-null rows. Some rows may be null, or empty lists.
- `prior_group_answers_correct` (string) provides all the `answered_correctly` field for previous group, with the same format and caveats as `prior_group_responses` . Some rows may be null, or empty lists.

(d) test.csv

Figure 1: Description of Datasets

4 Brain Storming

Before diving into data processing, it is better to have a deep thought on the problem from the first principle. We ask the following questions:

- What information is necessary to solve the problem?
  - What is the question?
  - Has the user watched the related lectures?

  - Has the user answer related questions? Answered correctly or wrongly?
  - Has the user got any feedback?
  - How long has the user learned?
  - How about other users? Closed related users might perform similarly.
  - The difficulty of the question.

  In summary, we need the learning experience of each user, the relationship between users, the connection between lecture and questions. The relation between questions. The relation between lectures.

- What information is provided by the dataset?

  - Users: How long has studied? What lectures has watched? what questions are answered, How long does it takes to answer the questions? Whether read the explanation of the correct answer?
  - Questions: id, tag, category, answer, group.
  - Lectures: id, type, category, tag.

- Do we need external dataset?
  The hierarchy of lectures and Questions might be useful.

- How to process the dataset?
  We need to exploit the inter-connection between users, questions and lectures and the hierarchy of lectures and Questions

- What models are good at solving the problem?
  This is a time-series binary classification problem. Many models are candidates.

5 Data Transformation

From the brain storming above, we conclude that the probability of correctly answer a question is determined by the hardness of the question, the ability of the student, whether the student watches the lecture and/or explanation. Note that the ability and hardness should be updated as new data arrives. To make these information more explicitly, we process the data in the following steps:
Question:

- Measure the hardness of each question: the ratio of correctly answered across all students.

- Cluster questions based on tags and parts.

Lectures:

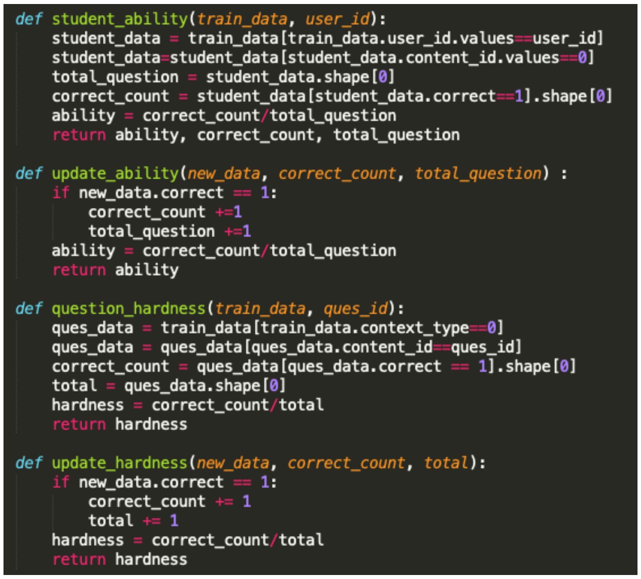
- Cluster lectures based on tags and parts.

Train:

- Measure the ability of each student by the correct ratio.

- Whether each student watch lectures.

We also design to functions to update the ability of students and hardness of questions.



```python
def student_ability(train_data, user_id):
    student_data = train_data[train_data.user_id.values==user_id]
    student_data=student_data[student_data.content_id.values==0]
    total_question = student_data.shape[0]
    correct_count = student_data[student_data.correct==1].shape[0]
    ability = correct_count/total_question
    return ability, correct_count, total_question

def update_ability(new_data, correct_count, total_question) :
    if new_data.correct == 1:
        correct_count +=1
        total_question +=1
    ability = correct_count/total_question
    return ability

def question_hardness(train_data, ques_id):
    ques_data = train_data[train_data.context_type==0]
    ques_data = ques_data[ques_data.content_id==ques_id]
    correct_count = ques_data[ques_data.correct == 1].shape[0]
    total = ques_data.shape[0]
    hardness = correct_count/total
    return hardness

def update_hardness(new_data, correct_count, total):
    if new_data.correct == 1:
        correct_count += 1
        total += 1
    hardness = correct_count/total
    return hardness
```

Figure 2: Update Hardness and Ability

6 Data Cleaning

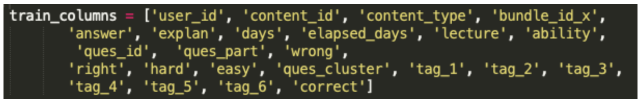
We handle missing values.

- The missing hardness is replaced by average hardness.
- For new student, the ability is set as average level.

Categorical variables are processed by LabelEncoder()

- Explanation is converted to $0/1$.
- Lecture types are encoded as $0, 1, 2$.

Finally, we merge the train dataset and question dataset. After the above data processing, we get the following features. All features are numerical. For modelling, 'user-id' is ignored as it contains no information.



```python
train_columns = ['user_id', 'content_id', 'content_type', 'bundle_id_x',
        'answer', 'explan', 'days', 'elapsed_days', 'lecture', 'ability',
         'ques_id',  'ques_part', 'wrong',
        'right', 'hard', 'easy', 'ques_cluster', 'tag_1', 'tag_2', 'tag_3',
        'tag_4', 'tag_5', 'tag_6', 'correct']
```

Figure 3: Features in the dataset

7 EDA

8 Feature Selection

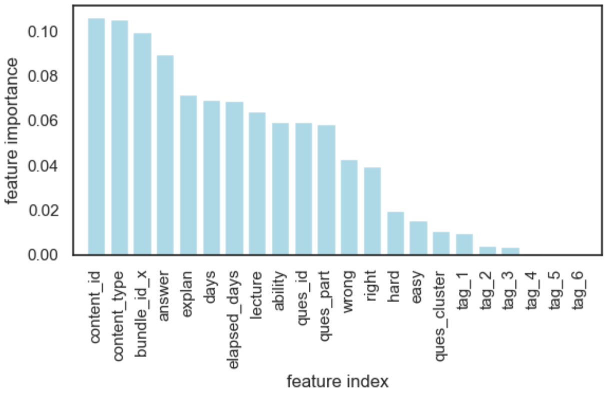


Figure 4: Feature Importance

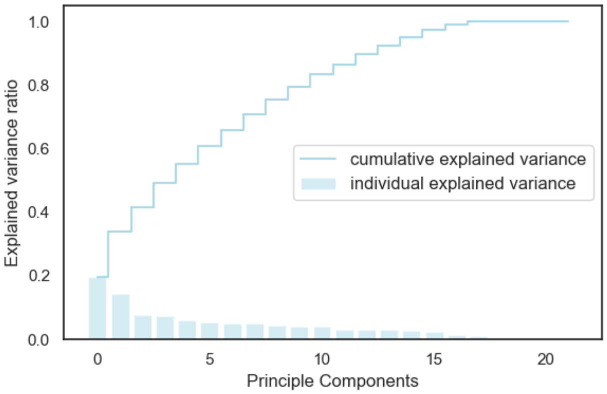


Figure 5: PCA Score

9 Models

The train dataset is ordered by ascending user-id and ascending timestamp. We split the dataset in to training and testing dataset by random sampling. This ensures that students appears in both subsets.

As this is a binary classification problem, many models can be used. We test the following models.

- Logistic Regression
- SVM
- KNN
- Decision Tree
- Random Forest
- Neural Network
- XGBoost

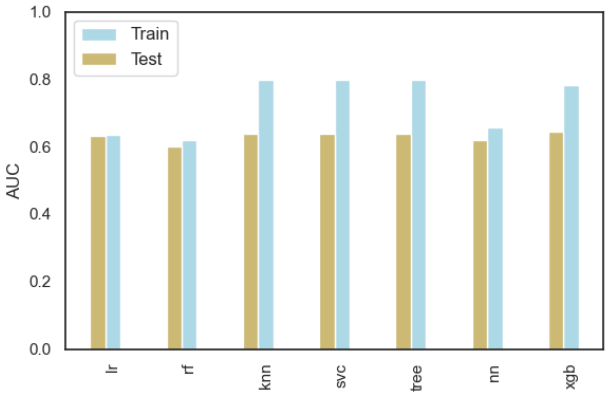


Figure 6: ROC AUC

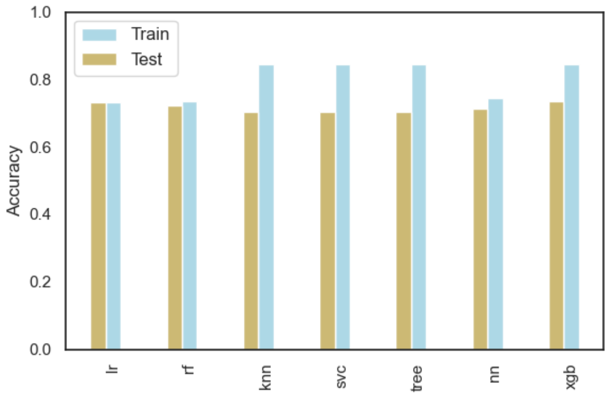


Figure 7: Accuracy

10 Model Selection

This is a big data and online project. When selecting models we need to consider the computational complexity and scalability of models. On the computational complexity side, KNN and SVC can not be trained incrementally. On the scalability side, KNN scales poorly with large data.

11 Model Fine-tuning

Models can be fine-tuned via grid search of hyperparameters. Due to the limited computational sources, we omit this step.

12 Further Improvement

The similarity between students would be useful in prediction the correct answer given other students' performance. The relationship between questions is also important.