Capstone Project – Diagnostic Questions

Udacity Data Science Nanodegree

Audris Ločmelis
July 26th, 2020

Project Definition

Project Overview

This capstone project is a first attempt at predicting answers to diagnostic questions in a recently released machine learning competition *Diagnostic Questions: The NeurIPS 2020 Education Challenge*[1].

> *"Digital technologies are becoming increasingly prevalent in education, enabling personalized, high quality education resources to be accessible by students across the world. Importantly, among these resources are diagnostic questions: the answers that the students give to these questions reveal key information about the specific nature of misconceptions that the students may hold. Analyzing the massive quantities of data stemming from students' interactions with these diagnostic questions can help us more accurately understand the students' learning status and thus allow us to automate learning curriculum recommendations."*
>
> *-- Diagnostic Questions: The NeurIPS 2020 Education Challenge*[2]

The data set contains over 20 million recorded answers to diagnostic questions in mathematics from students all around the world. The project data is sourced from an educational platform Eedi with thousands of students interacting with the platform in school year 2018/2019. The data is accessible from the computer science competitions platform CodaLab[3].

Problem Statement

The provided data consists of records of student-question pairs with an indication of correct/incorrect answer. The aim of the project is to predict whether a student will answer an unseen question correctly or not based on the previous answers and the responses of other users.

The problem of predicting the answers to diagnostic questions from a data perspective looks very similar to predicting a like/dislike [1/0] reaction for movies or articles. In order to model the outcome a common collaborative filtering recommendation approach, i.e., SVD (singular value decomposition) will be used. Similar to other real world problems, the user-item matrix will be very sparse, for that reason the primary method for SVD will be Funk SVD that can handle the missing values. As an alternative attempt, the missing values will be imputed with zeros to try making the predictions with closed form SVD that can't handle missing values.

---

[1] https://neurips.cc/Conferences/2020/CompetitionTrack

[2] https://arxiv.org/abs/2007.12061

[3] https://competitions.codalab.org/competitions/25449#learn_the_details

Metrics

In order to determine the quality of the predictions, two main performance measurements will be used

1. MSE (mean squared error) for the quality of SVD
2. F1 score for the binary predictions of correctness

Analysis

Data Exploration

The original dataset for the answer prediction challenge consists of 15'867'850 records, each of those is a unique user-item (student-question) pair. In total there are 118'971unique users and 27'613 unique questions that could create a maximum of 3'285'146'223 unique user-item pairs.

In other words, only 0.48% of the possible user-item combinations are present in the data, making the user-item matrix 99.52% sparse.

General statistics about the core data set of Diagnostic Questions challenge:

```
Unique user count: 118971
Unique question (item) count: 27613
Max combinations: 3285146223
Existing combinations: 15867850
Existing / max combinations: 0.00483
```

To limit the exploration time for this project, the data set will be reduced to the first 100'000 records. This is a very small subset of the total records available, but will at the very least reflect the structure of the available data so that more powerful processing can handle the larger data sample.

The original core dataset has six columns for unique identifiers of User (student), Question (item), Answer, as well as the *AnswerValue* for the exact answer selected, the *CorrectAnswer*, and the *IsCorrect* indicator. Only the *QuestionId*, *UserId*, and *IsCorrect* are of interest for this particular project, but the *AnswerId* was also used for proper time ordering.

Table 1. Illustration of the core data set train_task_1_2.csv.

QuestionId	UserId	AnswerId	IsCorrect	CorrectAnswer	AnswerValue
13880	32335	4300961	1	3	3
22467	32335	11888888	0	4	1
9307	32335	7218023	1	4	4

Data Visualization

No further visualizations were used for data exploration, however, data visualization plays a crucial role in understanding the classification performance issues.

Methodology

Data Preprocessing

In order to simulate a real world situation the answer data has to be chronological. It is necessary because a production system would make predictions of the future, therefore the validation should also follow the same pattern. The answer metadata data set contains a timestamp with precision up

to the minute of answer submission. This metadata was linked to the core data set to recreate the original chronological order of the responses.

The main preprocessing step in order to do the singular value decomposition is the creation of user-item matrix. In addition, since the closed-form SVD approach can't handle missing values, a modified user-item matrix was created where all the missing values were replaced by zeroes.

For limited computation burden the core data set is limited to the first 100'000 records after chronological sorting. In addition, for reproducibility purposes, the accompanying repository holds this data export.

Implementation

Two methods for singular value decomposition were used in the modeling process:

1. The FunkSVD algorithm that uses gradient descent to iteratively update the U, Σ, V^T matrices. A custom coded method was used to perform FunkSVD.
2. The closed-form SVD was used as an alternative on the modified user-item matrix. The standard *numpy* implementation was used.

For comparison purposes both methods had a common number of latent features, in this case it was arbitrarily set to 15.

Refinement

The two methods were used in their original forms. Since the FunkSVD wasn't achieving very accurate predictions, the standard SVD was used to test if this approach could be used for faster processing.

Results

Model Evaluation and Validation

The model performance for FunkSVD has proven to be underwhelming. First of all, the mean squared and mean absolute errors (MSE, MAE) are very high for the user-item pairs where the comparison was possible.

Deviation from the ground truth on validation set:

```
MSE: 1.2877163295958245
sqrt(MSE): 1.1347758939966184
MAE: 0.8483938817363773
```

When converting the predicted value to a classification output (a simple ≥ 0.5 rule is used) the classification results demonstrate underperformance on the minority class (`IsCorrect==0`). In Table 2 the recall value for the non-correct answers is just 4% indicating that the FunkSVD algorithm wasn't able to handle class imbalance.

Table 2. FunkSVD classification performance evaluation metrics.

	precision	recall	f1-score	support
0	0.32	0.04	0.06	1733
1	0.67	0.96	0.79	3582
accuracy			0.66	5315
macro avg	0.50	0.50	0.43	5315
weighted avg	0.56	0.66	0.56	5315

Furthermore, the precision for the minority class (`IsCorrect==0`) is also considerably lower than that of the majority class leading to a very low balanced F score of 0.06.

To understand the reasons behind the poor classifications, the distributions for the predicted values were plotted against the ground truth in Figure 1. The conclusion is that the FundSVD predictions are unable to distinguish between the two classes since the distributions of predicted values are almost identical (figure 1-B).

Fig. 1. FunkSVD distribution comparison for predicted values against ground truth (IsCorrect) indicator for validation set.

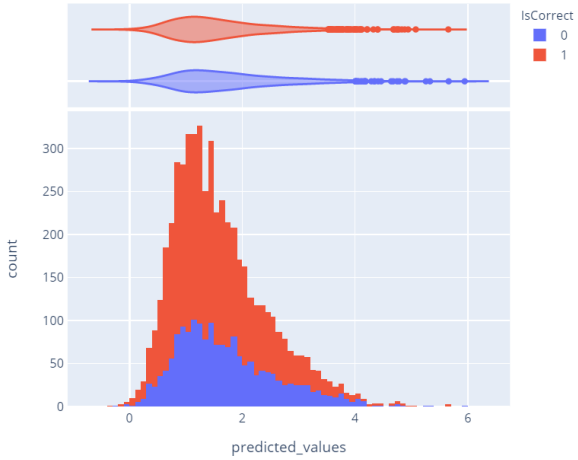

Fig. 1-A. Stacked and violin distributions

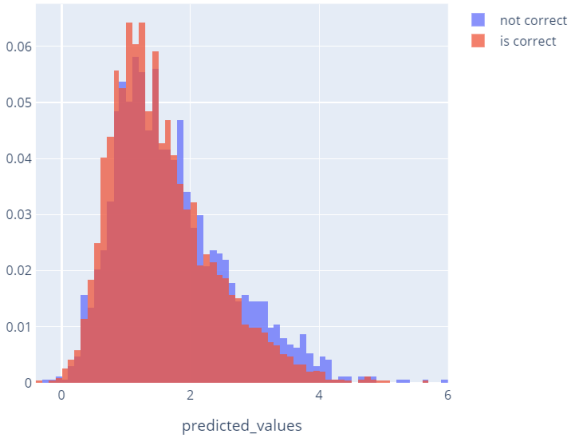

Fig. 1-B. Normalized distribution comparison

To examine if the issue with the identical distributions is a failure to discriminate among the two classes completely a qualitative look at the training data was necessary. It turns out that the FunkSVD performance on the training data is indeed slightly better.

Deviation from the ground truth on training set:

```
MSE: 0.49885076234855896
sqrt(MSE): 0.7062936799579612
MAE: 0.4538409171597077
```

Also, the classification results are better than those for the validation set.

Table 3. FunkSVD classification performance evaluation metrics on training data.

	precision	recall	f1-score	support
0	0.57	0.04	0.07	23837
1	0.71	0.99	0.82	56163
accuracy			0.70	80000
macro avg	0.64	0.51	0.45	80000
weighted avg	0.67	0.70	0.60	80000

This is due to the slight differences in distributions for different ground truth (IsCorrect) indicator values as seen in Figure 2-B.

Fig. 2. FunkSVD distribution comparison for predicted values against ground truth (IsCorrect) indicator for training set.

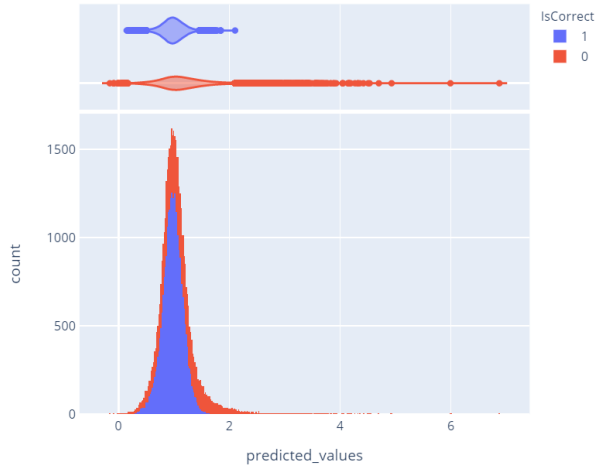

Fig. 2-A. Stacked and violin distributions

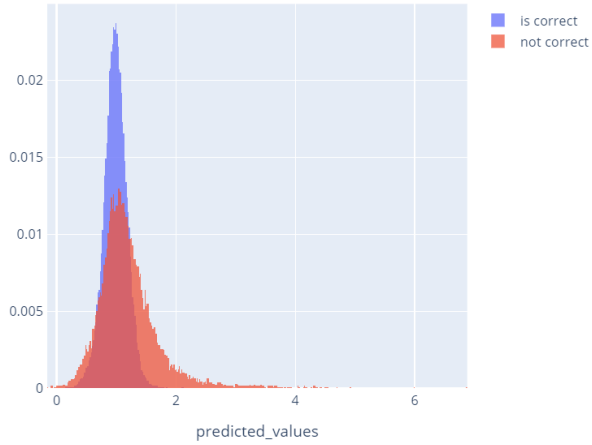

Fig. 2-B. Normalized distribution comparison

Even though the (`IsCorrect==1`) predictions are clerly centered around 1, still, the FunkSVD predictions are not constrained to the interval $[0,1]$ and produce mathematically sound but unjustifiable predictions for classification purposes. In fact, the (`IsCorrect==0`) even has a higher median than the other class (Fig. 2-A) which is not indicating a good reconstruction of the ground truth with the FunkSVD.

Since the FunkSVD results were underwhelming, another experiment to do a regular SVD on a modified training matrix. The aim of it was to test if such a simplified approach could be used and would offer better or at least not-worse results.

As seen in Figure 3, the predicted values for a modified matrix are underperforming by any measure. There are in fact no predictions close to 1 with all the predictions forming a spike around value 0, consequently, not replicating the 30/70% split in the actual training data but rather mirroring the sparsity of less than 1% of positive answers in all possible user-item pairs.

Fig. 3. Linear SVD on modified user-item matrix, prediction distribution comparison for predicted values against ground truth (IsCorrect) indicator for validation set. The potential reason for that is briefly discussed in next section.

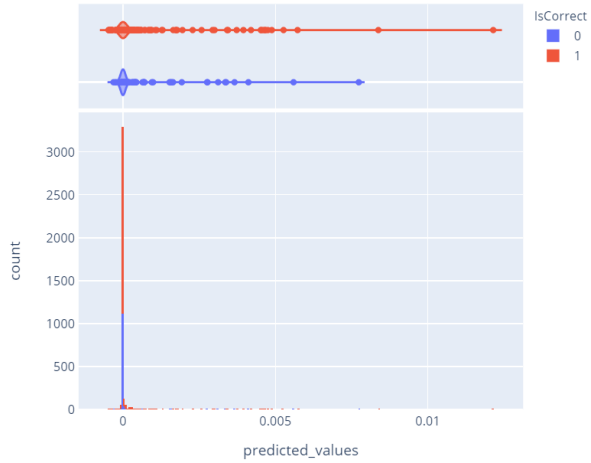

Fig. 3-A. Stacked and violin distributions

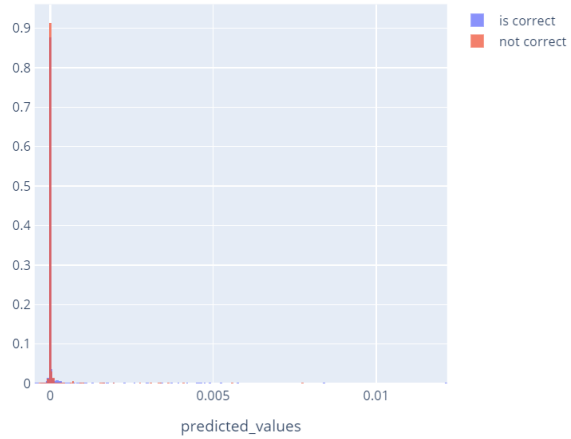

Fig. 3-B. Normalized distribution comparison

Justification

Due to the aforementioned sparsity with less than 1% of possible values present in the user-item matrix, the modified user-item matrix introduces a severe class imbalance that the SVD approach is not able to cope with. It predicts all answers to be 0 due to the overwhelming count of 0 values in the training data.

As for the FunkSVD, there are several problems with the approach for solving the problem at hand.

- The predictions are not constrained to the interval $[0,1]$, thus allowing for meaningless prediction outputs.
- The matrix factorization approach overtrains on the training data and soesn't generalize to the validation set.
- Class imbalance plays a major role in the quality of the predictions.

Conclusion

Reflection

The data set is a unique collection of student assessment data that I specifically waited to be published so I could use for this capstone project. It combines my personal passion for data science and also education technology, more specifically my interest in personalized education.

In conclusion, this could be considered as a first experiment on a newly released data set to test out the ideas of collaborative filtering for personalized education. The approach has many similarities with recommendations but also is very different in the actual nature of the problem.

The problem at hand seems to be quite challenging to solve but nevertheless, the available data set and the announced ML competition are excellent vehicles for promoting the exciting opportunities that digital assessment can bring to the students and educators.

Improvement

To improve on the first attempt to classify the question answering outcomes there are two main things to be considered in the future. First, the data available is much richer than was possible to process with the available computing resources, therefore more attention should be put on computational efficiency and brute force computational power. Second, other recommendation techniques that can handle missing values should be tested before a credible conclusion about the suitability of recommender techniques for the problem at hand can be made.

In addition, there could be more relevant ways to improve on the existing methods by using the knowledge available in the question meta-data. In particular, to enrich the data with context on question affiliations based on competency areas. A latent or explicit model of student knowledge would be a better predictor for individual questions than an attempt to predict based on isolated items and a limited history of student interactions with them.

Closing notes

The sample data and code for the analysis is available on GitHub:
https://github.com/AudrisLocmelis/DSN/tree/master/diagnostic_questions_capstone