Predictive modeling for STAAR test scores

An Jiang

January 2019

***Abstract*—We build a machine learning model to predict students' performance in the State of Texas Assessments of Academic Readiness (STAAR) standardized tests. The final prediction is based on students' previous STAAR scores, demographic information, ILL Benchmark test scores and various ILL usage information. We make predictions on students' actual scores as well as classification of At Risk/Not At Risk depending on the below/above the grade level performance assumption.**

I. Introduction

State of Texas Assessments of Academic Readiness (STAAR) is a series of standardized tests used in the state of Texas. Imagine Learning has collected the STAAR test scores of the 16-17 and 17-18 school years for the students in the Houston school districts as well as their ILL Benchmark test scores, ILL usage information and etc. We build a machine learning regression model to predict the STAAR test scores of the 17-18 school year. We will discuss how we established the regression model and improved its accuracy from two aspects – data preprocessing and model selection in the following two sections. Then, we make visualization about the prediction outputs and errors. Next, we transform the regression outputs to the "AtRisk" and "NotAtRisk" indicators according to a below/above the grade level performance assumption. Finally, we analyze the prediction results by grade level, do an interval estimation of the prediction error and discuss the correlation between the internal and external scores. All the datasets and codes are stored in the IL GitHub repository [1].

II. Dataset overview

State of Texas Assessments of Academic Readiness (STAAR) is a collection of different tests for various subjects and concepts. For the purpose of the assessment for Imagine Language and Literacy, we only care about the test "Reading 3-8" which evaluates the students' ability in English reading. Our model predicts students' STAAR test scores during the 17-18 school year while uses the scores during the 16-17 school year as one of the inputs. The dataset also contains students' demographic information including school name, grade, gender, English language learner (ELL) indicator, ethniciy and attendance rate. It also contains the overall and the item level scores for the three ILL Benchmark tests which were held at the beginning of the year (BOY), the middle of the year (MOY) and the end of the year (EOY) respectively. Each Benchmark test consists of 512 item questions. Usage information contains things like total minutes that a student spent on ILL and whether a student has completed or passed a specific learning activity in ILL. The Benchmark test scores have 2 granularity levels - the overall scaled scores and the item question scores. Similarly, usage time also have 2 granularity levels - the overall total minutes and weekly usage minutes that students spent on ILL respectively. We build 2 models accordingly. Furthermore, we also make comparison between the model with data imputation and the model that is able to handle the missing data automatically.

III. Data preprocessing

A. Load the dataset as dataframes

We load the dataset as Pandas dataframes and only keep the student rows with $"Grade" = 3, 4, ..., 8$ since higher grades take different type of reading test other than "Reading 3-8" and lower grades don't take STAAR test at all. In this way, we have 28594 student data samples.

B. Feature Engineering

1) Missing Data and Imputation: Table I shows the missing data ratio for the model with the overall scaled Benchmark test scores and the total minutes that a student spent on ILL.

TABLE I
MISSING DATA - OVERALL SCALED SCORE AND TOTAL MINUTES

	Missing Ratio (%)
Benchmark Vocab EOY Score	73.823
Benchmark Literacy EOY Score	65.045
Benchmark EOY Date	65.014
Benchmark BOY Date	61.793
Benchmark Vocab BOY Score	61.793
Benchmark Literacy BOY Score	61.793
Benchmark Vocab MOY Score	46.429
Benchmark Literacy MOY Score	40.578
Benchmark MOY Date	40.418
Reading.Scale.Score 1617	36.973
Reading.Scale.Score 1718	6.386
Attendance.Rate 1718	0.112
Ethnicity	0.045
SPED 1718	0.045
ELL 1718	0.045
Gender	0.045

We group the student samples by their demographic features and impute the missing values with the mean in each group. This is based on the assumption that students with similar demographic attributes should have similar test performances. We establish 3 level of student attributes for group-by imputation to make it more accurate and complete.

Table II shows the missing data ratio for the model with granular data, i.e., item question scores and students' weekly time usage on ILL.

We firstly delete those subtests that have a 100% missing ratio because they don't provide any information to the

TABLE II
MISSING - ITEM SCORES AND WEEKLY TIME USAGE

	Missing Ratio (%)
Word.Recognition.Form.C.2021.q99	100
Academic.Vocabulary.Easy.Form.C.6021.q249	100
Readables.Comprehension.Form.C.4021.leveled.book.c.3.2	100
...	...
Word.Recognition.2001.q65	63.052
Word.Recognition.2001.q62	63.052
Benchmark BOY Date	61.793
Benchmark MOY Date	40.418
Reading.Scale.Score 1617	36.973
Reading.Scale.Score 1718	6.386
Attendance.Rate 1718	0.112
Ethnicity	0.045
SPED 1718	0.045
ELL 1718	0.045
Gender	0.045

machine learning models. However, for item score models, the overall missing ratio is too high ($> 90\%$), which will cause a high bias with imputation. Thus, we decide not to apply imputation on item score models and use models that are able to handle the missing data autonomously like XGBoost and LightGBM algorithms.

2) One-Hot Encoding for Categorical Features: We use one-hot encoding to handle the categorical features. However, we need to apply the label encoding on some categorical variables that may contain information in their information in their ordering set before one-hot encoding.

3) Data Normalization for Numerical Features: Data Normalization is used to standardize the range of independent features of data. Otherwise gradient descent for loss function optimization converges slow or may even diverge. In this model, we use the most widely used data normalization method in machine learning - standardization. The general method of calculation is to determine the distribution mean and the standard deviation for each feature. Next we subtract the mean from each feature. Then we divide the values of each feature by its standard deviation.

$$x' = \frac{x - \bar{x}}{\sigma}$$

Where x is the original feature vector, $\bar{x}$ is the mean of that feature vector, and σ is its standard deviation. Feature standardization makes the values of each feature in the data have zero-mean and unit-variance.

C. Getting the training and test sets

We split the training and test set randomly with a training-test size ratio equal to 8:2. We apply cross validation on the training set, so there is no explicit validation set.

IV. MODEL SELECTION

Firstly, we define a cross validation strategy as the criterion to compare the performance in accuracy for machine learning models. The strategy is to shuffle the dataset prior to the cross validation with 5 folds, and compute the root mean squared error between the labels and predictions as the measure of loss.

We compare different types of models using the cross validation strategy mentioned above. Our base models include generalized linear models (Ridge, Lasso, Elastic Net, Kernel Ridge), Random Forest, Extremely Randomized Trees, Gradient boosting, XGBoost and LightGBM. Among these models, XGBoost and LightGBM have the best performance. Then we further boost predictive accuracy by building a stacked model with base models and a meta learner [2]. For the model with feature imputation, we select ENet, Gradient Boosting, KRR and Lasso as base models and Ridge as the meta model. On the other hand, for the models without feature imputations, we select XGBoost and LightGBM as base models and Ridge as the meta model.

We list the results of the top 3 algorithms for the overall score and usage model as well as the item score and weekly usage model. The overall score and usage model has two sub-cases respectively depending on whether the imputation is applied. Item score and weekly usage model has no imputation and use the algorithms that can handle the missing data autonomously. Here CV stands for cross validation, CV error is the cross validation strategy mentioned above, test error is root mean squared error on the test set. Note that from Table III to V, the error is for the standardized labels.

TABLE III
ERROR - OVERALL SCORE AND USAGE - IMPUTATION

Model	CV error	Test error
XGBoost	0.6230	0.6069
LightGBM	0.6342	0.6203
Stacked model	0.6188	0.6025

TABLE IV
ERROR - OVERALL SCORE AND USAGE - NO IMPUTATION

Model	CV error	Test error
XGBoost	0.6107	0.5925
LightGBM	0.6213	0.6080
Stacked model	0.6077	0.5928

TABLE V
ERROR - ITEM SCORE AND WEEKLY USAGE - NO IMPUTATION

Model	CV error	Test error
XGBoost	0.6102	0.5937
LightGBM	0.6232	0.6119
Stacked model	0.6075	0.5919

Observing the tables listed above, we can tell that the stacked model is superior than the other two models no matter what type of scores is used. Moreover, using more granular data is a little bit better than using overall data. The stacked model errors in Table V is smaller than those in Table IV. Finally, applying the algorithms that can handle the missing values automatically without imputation is a little bit better then data imputation. The stacked model errors in Table IV is smaller than those in Table III.

V. Output visualization

Firstly, we transform the all the standarized labels back to their original range for all granularity levels of scores and compare the percentage error between the target labels and the predictions of the STAAR test scores in Figure 1 to Figure 3.

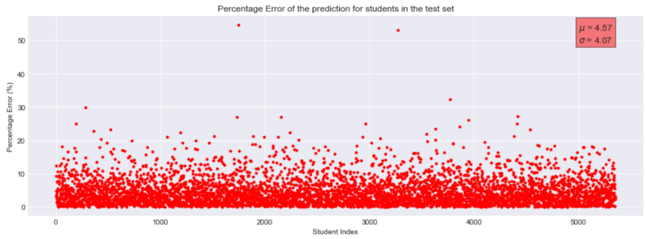

Fig. 1. Percentage Error - overall score and usage - Imputation

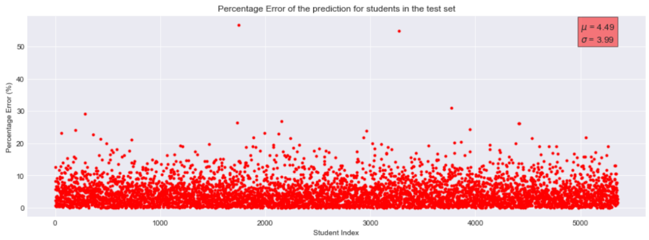

Fig. 2. Percentage Error - Overall score and usage - No imputation

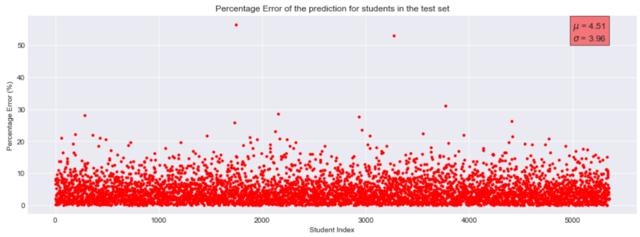

Fig. 3. Percentage Error - Item score and weekly usage - No Imputation

We summarize the percentage errors' mean and standard deviation in Table VI. From this table, we can tell that the item score and usage model without imputation is better than the other two models. On the other hand, imputation helps improve the accuracy in terms of the PE mean a little bit. However, we must realize that imputation is not feasible for item scores because the missing ratio is too high and also it is too difficult to do that. In conclusion, our best model is the stacked model using the item score and weekly usage with no imputation.

VI. The categorization of At Risk/Below Grade Level Performance based on regression outputs

Table VII is the criterion for ”At Risk” and ”Not AtRisk” categorization. In the context of STAAR test score prediction, ”At Risk” is equivalent to ”Below the grade level performance”. In other words, a student is considered ”at risk” or ”below the grade level performance” if her/his STAAR score is less than the criterion number listed in Table VII.

TABLE VI
Percentage errors' mean and standard deviation

	PE mean (%)	PE std (%)
Overall score and usage - Imputation	4.57	4.07
Overall score and usage - No Imputation	4.49	3.99
Item score and weekly usage - No Imputation	4.51	3.96

TABLE VII
Criterion of At Risk

Grade	At Risk Max. STAAR Score
3	1345
4	1434
5	1470
6	1517
7	1567
8	1587

Table VIII shows the classification accuracy of the ”AtRisk” predictions transformed from the regression outputs.

TABLE VIII
Classification accuracy

	Baseline accuracy	Classification accurac
Overall score and usage - Imputation	51.16%	79.23%
Overall score and usage - No imputation	51.16%	79.96%
Item score and weekly usage	51.16%	79.79%

VII. Analysis by grade

In this section, we only consider the model using item score and weekly usage with no imputation which has the best performance among all models.

Figure 4 and Figure 5 show the root mean squared log error and classification accuracy by grade level respectively.

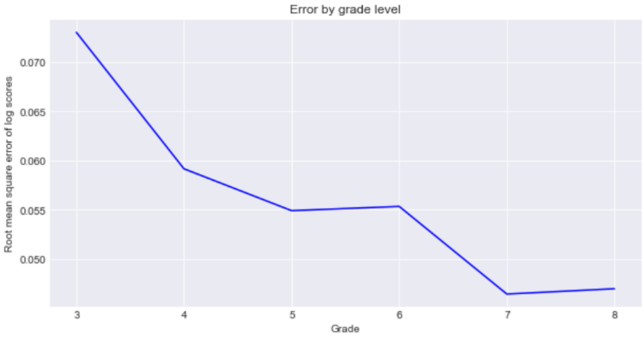

Fig. 4. Error by grade level

Figure 6 shows the comparison between the observed and the predicted mean scores by grade for students' STAAR

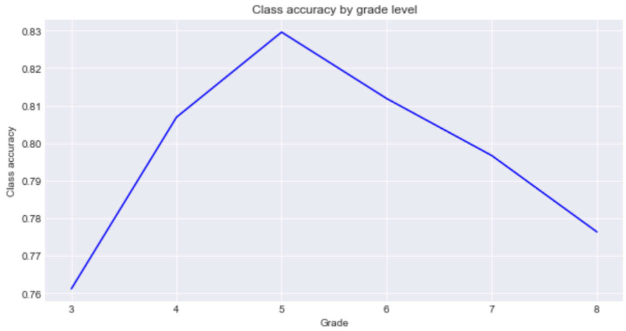

Fig. 5. Classification accuracy by grade level

test. The scores are rounded to integers. By comparison, the predicted mean scores are almost perfectly consistent with the observed mean scores.

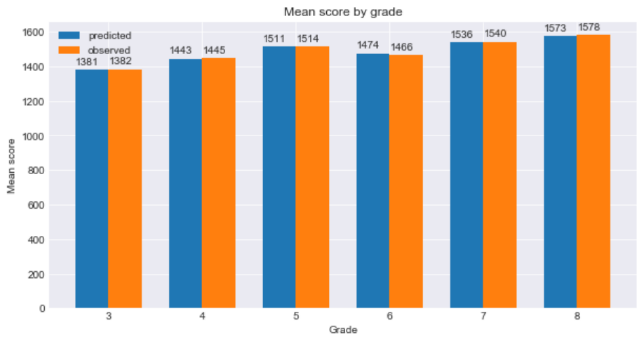

Fig. 6. Mean scores by grade

VIII. Interval estimate of the prediction error

To understand the prediction accuracy intuitively, we discuss the interval estimate of the percentage error mean mentioned in Table VI. For the item score and weekly usage model with no imputation, the 95% Confidence interval for the true PE mean of the percentage error is $[4.45\%, 4.67\%]$. The maximum STAAR reading 3-8 test score in the test set is 2017. Multiplying the maximum STAAR score with the upper bound of the corresponding confidence interval, we get an upper bound of the error in score value with a 95% probability. Thus, our prediction error is at most $2017 \times 4.67\% \approx 94$ for at least 95% experiments.

References

[1] IL Universal Sceener project GitHub repository. https://github.com/ImagineLearning/universal-screener.

[2] Ben Gorman. A Kaggler's Guide to Model Stacking in Practice. http://blog.kaggle.com/2016/12/27/a-kagglers-guide-to-model-stacking-in-practice/.