COMS - W4995 Applied Machine Learning Final Deliverable

Gokul Sunilkumar	Lei Liu	Rachel Peng	Xiaolin Sima	Ziyao Zhou
gs3214	ll3442	xp2197	xs2483	zz2915

December 4, 2022

1. **Introduction:** This project aims to predict US K-12 student academic performance using supervised regression models. NAEP, an educational assessment, is a widely used metric for student performance, which reports the average math and reading scores in grades 4 and 8. The quality of education is highly correlated with the resources allocated to schools, which are a combination of local, state, and federal expenditures and revenues that vary significantly among and within states. This creates disparities in student performance. We obtained a diverse dataset containing features that allowed models to predict academic performance from several perspectives, such as the state's financials and enrollment status. We hypothesized that higher expenditure per enrollment yields higher reading and math scores in both grades 4 and 8. Regression models, both linear and tree-based, were trained. R-squared (R^2) and mean square error (MSE) were used as our main evaluation metrics to compare the performance among models.

2. **Data Preprocessing and Exploration:** This project uses data from U.S. Education Datasets: Unification Project, scripted from the U.S. Census Bureau and the National Center for Education Statistics (NCES) and recorded from 1986 to 2019. It has 1K+ rows and contains missing values in several columns, which are dropped as our first step in data preprocessing. Highly correlated features were dropped to decrease the model complexity. A Robust scaler was used to get the mean and standard deviation of numerical features while excluding the impact of outliers. To fairly represent the financial resources that each state allocates to students, we created a new column feature, expenditure per enrollment. To convert the state column from categorical to numerical values, we applied target encoding with the mean of the expenditure per enrollment for each of the corresponding states. The total score was created by summing the math and reading scores in grades 4 and 8, which better represents student performance in both math and reading and serves as the target column.

3. **Modelling:**
   Table below shows the evaluation metrics (MSE and R^2) across all 6 models:

	Cross-Validation		Test Set	
	MSE	R^2	MSE	R^2
Linear Regression	668.01	0.12	560.24	0.16
Ridge Regression	668.01	0.12	560.24	0.16
Lasso Regression	682.63	0.11	582.72	0.13
Elastic Net	667.84	0.12	559.85	0.16
Random Forest	86.04	0.89	81.99	0.88
XGBoost	118.42	0.84	49.5	0.93

   (a) **Linear Models:**
   First, to establish a performance baseline, we trained a set of linear models to predict the total score given that it was a supervised regression task. Linear models generally had poor and unstable performance in our primary

analysis where the target was the total score, which was the sum of 4th and 8th grade math and reading scores. In our secondary analysis, we were curious to see if we could predict 4th grade math score using 4th grade reading score alone while dropping total score. Details of each linear model for these two analyses are shown below:

i. Plain vanilla linear regression performed poorly in our primary analysis. The dominant features in terms of importance were Year, State and Expenditure per Enrollment. Year was positively correlated with total score while State and Expenditure per Enrollment were negatively correlated. This would suggest the more resources spent on students, the worse they perform, which we found counterintuitive. Moreover, the model performance was highly dependent on the features we selected: If we included the average math and reading scores in our features, then our MSE would go to 0 and R^2 to 1, suggesting that those features could almost perfectly predict total scores. In our secondary analysis, MSE and R^2 improved dramatically from 560 to 18 and from 16% to 62%, respectively. This would suggest that knowing a student's reading score would help us predict his or her math score.

ii. Ridge regression achieved slightly better results after we tuned the hyper-parameter alpha in the primary analysis. We used GridSearch to test out 100 alphas from 0 to 1 with 0.01 as the step size, and selected the alpha=0.99 that yielded the best performance. The most significant features were Expenditure per Enrollment, State, and Year, similar to the linear regression. In the secondary analysis, MSE and R^2 further improved to 14 and 69%, respectively.

iii. Lasso regression was used to check if feature selection could improve the performance for the specific task and dataset at hand and similarly poor performance. Feature importance chart showed that federal revenue and total revenue were the primary contributors to the total score, but they had opposite signs. Intuitively they both should be positively correlated to the total score. In the secondary analysis, MSE and R^2 further improved to 11 and 69%, respectively.

iv. Elastic net with default parameters was initiated with alpha = 1 and l1-ratio = 0.5 and later tuned using GridSearch. The candidates for alpha are 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 0.0, 1.0, 10.0, and 100.0. For l1-ratio, we are tuning it from 0 to 1 with 0,01 as the step size. The best parameters proposed by GridSearch were alpha = 10.0 and l1-ratio = 0.0, which was actually a Ridge Regression. The outstanding features were still Expenditure per enrollment, Year, and State. In the secondary analysis, MSE and R^2 improved to 559.85 and 0.16 respectively.

We think the poor performance of linear models was largely due to the fact that features in our dataset were somewhat correlated despite our effort to drop the most correlated ones, and multicollinearity can yield solutions that are wildly varying and possibly numerically unstable.

(b) **Tree-based Models:**

We then moved from linear models to tree-based models, as tree-based models typically deal better with multi-collinearity. By design, when we have highly correlated features, the structure of tree-based models will pick one of the features as the split, and the model will still work well. Tree-based models generally had good performance in both of our primary analysis and secondary analysis. Details of tree-based models for these two analyses are shown below:

i. Random Forest performed well on our data. A grid search 10 fold cross validation is used to find the best parameters. The two parameters that we focused on are max_depth, which is the maximum depth of the tree, and n_estimators, which is the number of trees in the forest. The candidates for max_depth are 3, 6, 9, 12, 15. The candidates for n_estimators are 10, 50, 100, 500, 1000. After evaluating on the R-squared score, the best parameters chosen are max_depth = 15 and n_estimators = 1000. We then used these parameters to run

our best model. The most influential feature is State, followed by Enroll, Local Revenue, and Grades_PK_G, which is the number of students in Pre-Kindergarten education. In the secondary analysis, MSE decreased to 10.77 and R^2 decreased to 0.81, which are still pretty good performance.

ii. XGBoost is our best model. It has achieved the highest R-squared score and the lowest MSE. When tuning the hyperparameters of our XGboost model, Bayesian Optimization is applied to enhance the tuning efficiency and bring better generalization performance on the test set. The chosen parameters are max_depth, gamma, reg_alpha, reg_lambda, colsample_bytree, and min_child_weight. The loss function is mean_squared_error. The tuning suggests that the best model should be initiated with max_depth = 12, gamma = 5.445, reg_alpha = 53, reg_lambda = 0.497, colsample_bytree = 0.593, min_child_weight = 10, and the performance of the test set is very promising as now the R-squared score is higher than 0.9 and the MSE suggests that the average difference between the actual and predicted test score is further reduced to around 7.03. In XGBoost, no feature is reduced to zero and the most significant ones are STATE, ENROLL and LOCAL_REVENUE.

4. **Best Model:** XGBoost

There are mainly three reasons why XGBoost is the best model:

(a) Firstly, XGBoost outperforms linear models when the features and the target variable do not preserve linear relationships. In the data exploration part, we have studied the relationships between the target variable and the features. XGBoost can make more complex decision boundaries that fit well with nonlinear data. Besides, in the data exploration part, we discovered that the distributions of the feature data are skewed to the right. XGBoost uses the "exact" split finding method which considers every possible split point when building a tree and it has its own heuristic upon each split. Therefore, XGBoost is better than the linear models in our project.

(b) Secondly, XGBoost is robust and powerful by its design. Compared to its prototype, the decision-tree model, XGBoost trains the model in a gradual, additive, and sequential manner. Each predictor corrects its predecessor's error: weights are assigned to all the independent variables which are then fed into the decision tree which predicts results. The weight of variables predicted wrong by the tree is increased and these variables are then fed to the second decision tree. These individual predictors then ensemble to give a strong and more precise prediction of the TOTAL_SCORE.

(c) Thirdly, XGBoost deals with multicollinearity by using both L1 and L2 regularization.The weights on L1 and L2 regularization are reg_alpha and reg_lambda respectively. In our optimal model, the reg_alpha is very high (=53). In this case, our model tends to shrink data values towards a central point as the mean. The TOTAL_SCORE data, however, is 'condensed' in nature. The standard deviation of the TOTAL_SCORE column is actually only 27. The formula of MSE and standard deviation showcase that the former computes the sum of the square of the difference between the actual and the predicted score and then takes the average; the latter computes the sum of the square of the difference between each actual score and their mean and then takes the average. A high reg_alpha pushes the predicted values toward the mean and therefore our MSE is actually pushed toward the standard deviation (since the dataset is large, we assume n = n-1) which is considerably low.

5. **Conclusion:**

To test our hypothesis that higher expenditure per enrollment, encoded as STATE, yields higher reading and math scores, we trained 6 models and found that tree-based models outperformed linear models due to their superiority in handling correlation in features. XGBoost achieved the best performance, where STATE was the most important feature by a wide margin. Therefore, we concluded that there was indeed a strong relationship between expenditure per enrollment and a student's math and reading scores.