Interpretable Machine Learning for Student Performance Prediction

Rachel Nguyen rmn1@williams.edu
Computer Science Department
Williams College

1. Introduction

The COVID-19 pandemic has caused unprecedented setbacks to society on many fronts, especially education. It is estimated that pandemic-induced school closures will cause a student who attended primary or secondary school in 2020 to lose approximately $\$16,000$ in lifetime earnings (Azevedo et al., 2020). More disturbingly, the educational impacts of the pandemic around the world are disproportionate. Unlike the majority of high-income countries, many low-income nations lack resources to implement measures that increase students' access to remote learning (Azevedo et al., 2022). Understanding factors that can affect student performance is of great importance to offset this widening educational inequality. Machine learning (ML)—a field devoted to studying methods that automatically learn from data to improve performance on certain tasks—offers a fertile ground to achieve this goal thanks to the explosive growth of data in the last few decades. In Cortez and Silva (2008), the authors used several ML models to predict academic performance of students from two secondary schools in Portugal based on over thirty factors such as parents' education, alcohol consumption, etc. While the models achieved high predictive accuracy, they found that only a small number of the input variables seemed to be relevant. In Gan et al. (2022), the authors proposed an efficient procedure for computing confidence intervals when quantifying the effects of input features on the predictions. Not only is this method fast, but it is also model-agnostic and relies on minimal assumptions. In this paper, I first apply several ML algorithms on the dataset used in Cortez and Silva (2008) to obtain a baseline. Then, I train models using the same algorithms on a subset of the original dataset after removing irrelevant features based on the method given in Gan et al. (2022) and compare how they perform against the baseline. My hope is to develop interpretable models for predicting student performance. Narrowing down the most influential features on the predictions in this problem is meaningful, especially for low-income countries as it helps them efficiently invest their limited resources in areas crucial for improving student academic performance, thereby bridging the growing global learning gap caused by the pandemic. My findings concur with the conclusion drawn in Cortez and Silva (2008) that the majority of input features are unimportant. Moreover, the models trained after eliminating irrelevant features perform better than the baseline most of the time, suggesting feature selection can not only help with interpretability, but also improve accuracy.

2. Preliminaries

2.1 ML Algorithms

All ML models are trained using the **scikit-learn** library (Buitinck et al., 2013). For input features X with d attributes, a ML model produces predictions $\widehat{Y}$. In supervised learning, the goal is to fit a model that minimizes the difference between $\widehat{Y}$ and the actual outcome Y corresponding to X. To quantify this difference, I use the *mean squared error* function, which is defined as $\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2$ where n is the number of samples. Also, we assume the outcome is continuous.

2.1.1 Ridge Regression (RR)

In RR, a prediction $\widehat{y_i}$ is a linear combination of the input features x_i and a set of learned parameters θ. In other words, $\widehat{y_i} = \theta_1 x_{i,1} + \theta_2 x_{i,2} + \ldots + \theta_d x_{i,d}$. θ is learned by minimizing the objective function $L(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 + \alpha \sum_{i=1}^{d} {\theta_i}^2$ where $\alpha \sum_{i=1}^{d} {\theta_i}^2$ corresponds to the L2 regularization penalty and α is a hyperparameter that will be tuned using a validation set (Hoerl and Kennard, 1970; Tikhonov, 1943). To find θ that minimizes $L(\theta)$, we use *gradient descent (GD)* (Cauchy et al., 1847) which starts with an initial guess $\theta^{(0)}$ and updates $\theta^{(t+1)}$ to be $\theta^{(t)} - \beta \times \nabla L(\theta^{(t)})$ where $\beta > 0$ is some small constant until either a maximum number of iterations is reached or the change in θ is negligible.

2.1.2 Decision Trees (DT)

From a starting dataset, DT (Breiman et al., 1984) constructs a tree by recursively splitting the data into smaller datasets at each level until either a maximum depth or some other base cases are reached. I measure the quality of a split using the default criterion in **scikit-learn** which is the *squared error*. Specifically, to determine how to best split a dataset $[D]$ into two datasets $[D_1]$ and $[D_2]$, for each feature j, the algorithm considers every value v for j in $[D]$. Then, $[D_1]$ consists of samples in $[D]$ which value for j is less than or equal to v and $[D_2] = [D] \setminus [D_1]$. The squared error $\mathcal{H}([D_1])$ is defined as $\sum (y_i - \bar{y})^2$ for all y_i corresponding to the samples in $[D_1]$ and $\bar{y}$ is the mean of these y_i. The algorithm finds the pair (j, v) that minimizes the weighted error $\frac{m_1\mathcal{H}([D_1]) + m_2\mathcal{H}([D_2])}{m_1+m_2}$ where m_1 and m_2 are the number of samples in $[D_1]$ and $[D_2]$, respectively. To obtain the prediction for the input features x_i, we traverse the tree until we reach a leaf node. The prediction for a leaf node is the mean of the outcomes for all training samples in that node.

2.1.3 Random Forests (RF)

In RF (Breiman, 2001), we randomly sample a starting dataset with replacement to obtain several new datasets. For each of these datasets, we train a DT but only consider a random subset of the features at each split. The prediction for the input features x_i is the mean of the predictions of all the trees. In **scikit-learn**, the number of trees in a RF is called **n_estimators.**

2.1.4 Neural Networks (NN)

A graph $\mathcal{G} = (V, E)$ is a directed acyclic graph (DAG) if there is at most one edge between every pair of vertices, E contains only directed edges, and $\mathcal{G}$ has no directed cycles. A NN (Rosenblatt, 1958) is a DAG such that the vertices are partitioned into sets $\mathcal{L}_1, \ldots, \mathcal{L}_k$ which are called *layers* and for every edge $u \to v$, $u \in \mathcal{L}_i$ and $v \in \mathcal{L}_{i+1}$ for some $1 \leq i < k$. The first layer $\mathcal{L}_1$ has d vertices, each of which corresponds to the value of an attribute in the input features. As the outcome is continuous, the last layer $\mathcal{L}_k$ has only one vertex for the prediction. The $\mathcal{L}_2, \ldots, \mathcal{L}_{k-1}$ layers are called *hidden layers* as their sizes can be tuned. Each edge $u \to v$ has a weight $\theta_{u,v}$. For any node v, its value is the result of a non-linear transformation of the linear combination between the values of its parents and the corresponding weights of edges coming into v. For the non-linear transformation, I use the *rectified linear unit (ReLU)* function which is defined as $f(x) = \max(x, 0)$. The set of all weights θ is learned using *stochastic gradient descent (SGD)* (Robbins and Monro, 1951). SGD is similar to GD, except that in every iteration, for each sample i, we update $\theta^{(t+1)}$ to be $\theta^{(t)} - \beta \times \nabla L_i(\theta^{(t)})$ where $\nabla L_i(\theta^{(t)})$ denotes the gradient for sample i.

2.2 Model-agnostic Inference for Feature Importance

Gan et al. (2022) proposed a fast and model-agnostic method for feature importance inference called *minipatch leave-one-covariate-out inference (MLOCOI)*. This method was leveraged from the idea of *minipatch learning* (Toghani and Allen, 2021; Yao et al., 2021). In minipatch learning, we randomly sample several subsets of both input features and outcomes which are called *minipatches*. We use the minipatches to train different models. For continuous outcomes, the final prediction is the mean of the predictions of all models. Given a training dataset $(\mathbf{X}, \mathbf{Y})$ independently sampled from a probability distribution $p(X, Y)$ with n samples and d attributes, let $(\mathbf{X}_{n'}, \mathbf{Y}_{d'})$ denote any minipatch of $(\mathbf{X}, \mathbf{Y})$ with n' samples and d' attributes. Assume n' and d' are fixed. As the number of minipatches approaches ∞, we obtain the expected model

$$H(x; \mathbf{X}; \mathbf{Y}) = \frac{1}{\binom{n}{n'}\binom{d}{d'}} \sum_{(\mathbf{X}_{n'}, \mathbf{Y}_{d'})} h(x_{n'}; \mathbf{X}_{n'}; \mathbf{Y}_{d'}),$$

where $H(x; \mathbf{X}; \mathbf{Y})$ is the prediction of the expected model trained on $(\mathbf{X}, \mathbf{Y})$ for input feature x and $h(x_{n'}; \mathbf{X}_{n'}; \mathbf{Y}_{d'})$ is the prediction of a base model trained on $(\mathbf{X}_{n'}, \mathbf{Y}_{d'})$ for the corresponding subset $x_{n'}$ of x (Gan et al., 2022). Gan et al. (2022) define the *feature importance score* (FIS) for any feature j on $(\mathbf{X}, \mathbf{Y})$ as

$$\Delta_j^*(\mathbf{X}, \mathbf{Y}) = \mathbb{E}_{(x,y)}[\text{Loss}(y, H(x_{\setminus j}; \mathbf{X}_{\setminus j}; \mathbf{Y})) - \text{Loss}(y, H(x, \mathbf{X}, \mathbf{Y}))],$$

where $x_{\setminus j}$ denotes the vector x without the column for j. To compute the confidence interval (CI) for the FIS of any feature j, MLOCOI first uses minipatch learning to obtain several base learners. For each $(x_i, y_i) \in (\mathbf{X}, \mathbf{Y})$, the *leave-one-out (LOO)* prediction $H_{-i}(x_i)$ is the mean of the predictions of all base learners that are not trained on (x_i, y_i) and the *leave-one-out and leave-one-covariate-out (LOO + LOCO)* prediction $H_{-i}^{-j}(x_i)$ is the mean of the predictions of all base learners that are not trained on (x_i, y_i) and feature j. The

LOO feature occlusion score for (x_i, y_i) is defined as

$$\Delta_j(x_i, y_i) = \text{Loss}(y_i, H_{-i}^{-j}(x_i)) - \text{Loss}(y_i, H_{-i}(x_i)).$$

The sample mean $\bar{\Delta}_j$ is the mean of the LOO feature occlusion score for all (x_i, y_i). The sample standard deviation σ_j can be written as $\sqrt{\frac{\sum_{i=1}^{n}(\Delta_j(x_i,y_i)-\bar{\Delta}_j)^2}{n-1}}$. With $\bar{\Delta}_j$ and σ_j, we can derive any CI for the FIS of feature j. Gan et al. (2022) give theoretical guarantees that under some mild assumptions, the CI obtained from MLOCOI will cover the FIS Δ_j^* when the number of samples tends to ∞. Moreover, MLOCOI is fast as minipatch learning allows us to not have to refit models when leaving out feature j (Gan et al., 2022).

3. Data

As mentioned, I use the data from Cortez and Silva (2008) which consists of two datasets corresponding to the Mathematics (**M** with 395 samples) and the Portugese (**P** with 649 samples) courses. Both datasets have the same 33 attributes which are divided into 4 groups as described in Table 1. The first group consists of binary variables which are converted to 0 or 1—the bolded value is mapped to 1. The second group contains ordinal variables and are kept intact. The third group includes nominal variables. For each of these variables, I create several 0/1 dummy variables corresponding to membership in each of the categories before dropping the original variable. For instance, for **Fjob**, prior to dropping this column, I construct 5 dummy 0/1 variables **Fjob_teacher**, **Fjob_health**, **Fjob_civil**, **Fjob_athome**, and **Fjob_other** whose value is 1 if and only if the value of **Fjob** is in the corresponding category. Variables in the last group are considered continuous and standardized to have 0 mean and 1 standard deviation. The outcome of interest is the final grade **G3**. Both datasets are split 50/20/30 for training, validating, and testing, respectively. Note that the standardization of continuous variables is performed using the mean and standard deviation of the training set.

4. Training and Validation Of Models

For all procedures, I choose the random state to be 42.

4.1 Baseline Models

For the naive predictor (NP), I output the second grade period **G2**. For RR, I tune the L2 regularization hyperparameter α by choosing a value from the set

$$\{0, 0.05, 0.1, 0.5, 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25\}$$

that minimizes the error on the validation set. For DT and RF, the maximum tree depth **max_depth** is tuned by choosing a depth in range $[3, d]$ where d is the number of attributes. Additionally, for RF, I set **n_estimators** to be 500. For NN, as a NN with two hidden layers is sufficient for approximating almost any function (Lippmann, 1987), I consider two two-hidden-layer NNs in which the size of the first layer is chosen as 50 so that it has more

Attribute	Domain
sex	student's sex: male (**M**) or female (F)
school	student's school: Gabriel Pereira (**GP**) or Mousinho da Silveira (MS)
address	student's home address type: urban (**U**) or rural (R)
Pstatus	parents' cohabitation status: together (**T**) or apart (A)
famsize	family size: > 3 (**GT3**) or ≤ 3 (LE3)
schoolsup	extra school enrollment: **yes** or no
famsup	family educational support: **yes** or no
paid	extra paid classes related to the course subject: **yes** or no
activities	extra-curricular activities: **yes** or no
nursery	attended nursery school: **yes** or no
internet	access to the Internet at home: **yes** or no
higher	desire to pursue higher education: **yes** or no
romantic	currently in a romantic relationship: **yes** or no
Fedu	father's education: none (0), up to 4th grade (1), 5-9th grade (2), 10-12th grade (3), or higher (4)
Medu	mother's education: none (0), up to 4th grade (1), 5-9th grade (2), 10-12th grade (3), or higher (4)
famrel	quality of family relationships: from very bad (1) to excellent (5)
freetime	free time after school: from very rare (1) to very frequent (5)
goout	going out with friends: from very rare (1) to very frequent (5)
Walc	weekend alcohol consumption: from very low (1) to very high (5)
Dalc	workday alcohol consumption: from very low (1) to very high (5)
health	current health status: from very bad (1) to very good (5)
traveltime	travel time to school: < 15 min (1), 15-30 min (2), 30 min to 1 hour (3), or > 1 hour (4)
studytime	weekly study time: < 2 hours (1), 2-5 hours (2), 5-10 hours (3), or > 10 hours (4)
failures	number of previous course failures: n if $1 \leq n < 3$, else 4
absences	number of school absences: 0 to 93
Fjob	father's job: teacher, health care related, civil services, at home, or other
Mjob	mother's job: teacher, health care related, civil services, at home, or other
reason	reason to choose student's school: close to home, school reputation, course preference, or other
guardian	student's guardian: mother, father, or other
age	student's age: 15 to 22
G1	first period grade: 0 to 20
G2	second period grade: 0 to 20
G3	final grade: 0 to 20

Table 1: Attributes in datasets

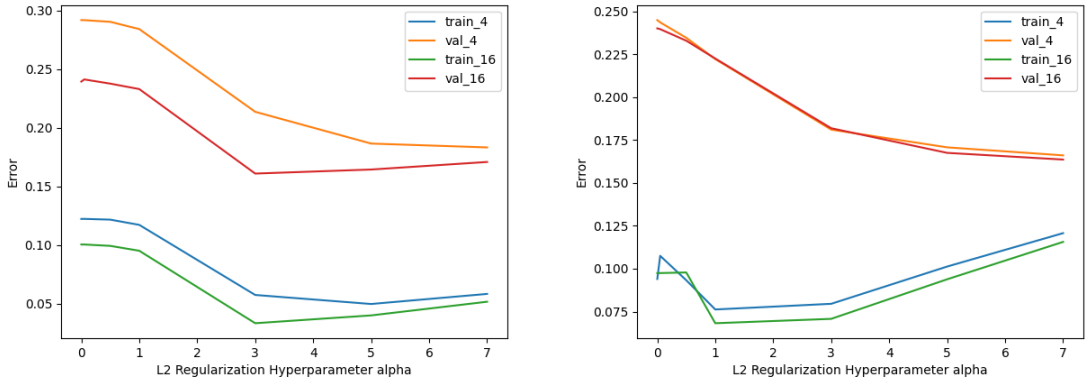

Figure 1: Training and validation errors of NNs with the second hidden layer of size 4 and 16 for **M** (left) and **P** (right).

RR	**alpha** = **7**, **25**
DT	**max_depth** = **5**, **3**
RF	**max_depth** = **3**, **4**
NN	**alpha** = **3**, **7**; **hidden_layer_sizes** = (**50, 16**), (**50, 16**)

Table 2: Parameters used in the final baseline models for **M** and **P**.

nodes than the number of input features. The size of the second layer is either 4 or 16. I set the **solver** to be `sgd` which refers to SGD and the maximum number of iterations **max_iter** to be 10000. Then, I tune the L2 regularization α by choosing a value from the set $\{0, 0.05, 0.5, 1, 3, 5, 7\}$ that minimizes the error on the validation set. Figure 1 shows the training and validation errors of both types of NNs across different values of α. The values for the tuned parameters in the final models are described in Table 2. For the parameters not mentioned, I use the default values in **scikit-learn**.

4.2 MLOCOI

For both **M** and **P**, I train 15 RF with **n_estimators** = 50 and **max_depth** = 3 as base learners on minipatches of 30 samples and 5 features. The 95% CI for the FIS of some features in both **M** and **P** are listed in Table 3. The results confirm with the findings in Cortez and Silva (2008) that the majority of the input features are irrelevant and that previous performances (**G1** and **G2**) are good predictors for the outcome.

4.3 Reduced Models

After obtaining the 95% CI for the FIS of the features, I only keep features with the 95% CI $[x, y]$ where $x, y > 0$. Specifically, for **M**, I keep the **sex**, **famsize**, **Pstatus**, **studytime**, **schoolsup**, **nursery**, **higher**, **romantic**, **famrel**, **goout**, **health**, **absences**, **G1**, **G2**,

G1	[4.370, 6.205]	[2.007, 2.626]
G2	[8.027, 10.528]	[1.903, 2.564]
absences	[2.437, 4.370]	[-0.747, -0.461]
sex	[0.387, 1.155]	[-0.248, 0.082]

Table 3: The 95% CI for the FIS of some features in both **M** and **P**.

RR	**alpha** = **3**, **25**
DT	**max_depth** = **3**, **3**
RF	**max_depth** = **3**, **3**
NN	**alpha** = **5**, **5**; **hidden_layer_sizes** = (**20, 16**), (**25, 16**)

Table 4: Parameters used in the final reduced models for **M** and **P**.

Mjob_other, **Mjob_health**, **guard_mom**, **guard_fat**, and **guard_other** features. For **P**, I keep the **age**, **address**, **Pstatus**, **Fedu**, **paid**, **activities**, **romantic**, **famrel**, **Dalc**, **Walc**, **health**, **G1**, **G2**, **Mjob_athome**, **Fjob_athome**, **Fjob_teacher**, **guard_mom**, **guard_fat**, and **guard_other** features. Then, I use the same process as described in subsection 4.1 to train the models. The only modification is that the sizes of the first hidden layer in the NNs for **M** and **P** are set to 20 and 25 to reflect the reduced number of features. The values for the tuned parameters in the final reduced models are shown in Table 4.

5. Results

Table 5 shows the test errors for the baseline **(b)** and reduced **(r)** models. For the baseline models, except for RF, other models do not differ much from the NP in terms of test errors. The fact that RF performs best on both datasets is consistent with the findings in Cortez and Silva (2008). Moreover, the reduced models achieve lower test errors compared to their baseline counterparts most of the time which also confirms the conclusion drawn in Cortez and Silva (2008) that the majority of input features are irrelevant. Furthermore, this also indicates that feature selection can be of great importance for not only interpretability but also accuracy.

6. Ablation Study

To simulate distribution shift, I add random noise to the validation and test data. The random noise follows a normal distribution with mean 0 and standard deviation 0.1. Then, I train models using the same procedure as described in subsection 4.1 and subsection 4.3. Table 6 gives the test errors for the baseline and reduced models. RF still performs best compared to other algorithms. Moreover, as expected, the test errors increase most of the time.

	M (b)	M (r)	P (b)	P (r)
NP	0.243	0.243	0.141	0.141
RR	0.262	0.244	0.140	**0.131**
DT	0.208	0.183	0.164	0.164
RF	**0.158**	**0.176**	**0.122**	0.132
NN	0.236	0.196	0.148	**0.131**

Table 5: Test errors for the final models.

	M (b)	M (r)	P (b)	P (r)
NP	0.269	0.256	0.156	0.156
RR	0.279	0.257	0.149	0.140
DT	0.261	0.223	0.150	0.149
RF	**0.208**	**0.200**	**0.127**	**0.135**
NN	0.249	0.209	0.156	0.143

Table 6: Test errors for the final models when random noise is added to the validation and test data.

7. Discussion and Conclusion

This analysis seeks to construct more interpretable ML models for predicting academic performance using a fast and model-agnostic method for feature selection proposed in Gan et al. (2022). The results indicate that this procedure produces not only more interpretable but also more accurate models.

That said, my analysis has several limitations. First, the assumption that all rows of data are independent and identically distributed is violated. Notice that both datasets consist of information for students from two secondary schools: Gabriel Pereira or Mousinho da Silveira. Apparently, rows of data for students from the same school are not independent. For instance, the resources available at a school might make it easier for students to participate in extra-curricular activities compared to the other. In addition, for MLOCOI, due to limited computational power, I can only train 15 base learners on relatively small minipatches of 30 samples and 5 features. For future work, one might consider training more base learners on larger minipatches so that the obtained CI is more likely to cover the FIS. Moreover, for simplicity, I treat grade (G1, G2, and G3) which is a numeric value on a scale from 0 to 20 as a continuous variable. It would be interesting to see how the analysis would change if the problem at hand were considered a classification task.

References

Joao Pedro Azevedo, Amer Hasan, Diana Goldemberg, Syedah Aroob Iqbal, and Koen Geven. Simulating the Potential Impacts of COVID-19 School Closures on Schooling and Learning Outcomes : A Set of Global Estimates. `https://openknowledge.worldbank.org/handle/10986/33945`, Jun 2020. [Online; accessed 12-November-2022].

Joao Pedro Azevedo, Marcela Gutierrez, Rafael de Hoyos, and Jaime Saavedra. The Unequal Impacts of COVID-19 on Student Learning. In Fernando M. Reimers, editor, *Primary and Secondary Education During Covid-19*, pages 421–459. Springer, Gewerbestrasse 11, 6330 Cham, Switzerland, 2022.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Leo Breiman, Jerome H Friedman, Richard A Olshen, and Charles J Stone. *Classification and Regression Trees*. Routledge, 1984.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

Augustin Cauchy et al. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536–538, 1847.

Paulo Cortez and Alice Maria Gonçalves Silva. Using Data Mining to Predict Secondary School Student Performance. 2008.

Luqin Gan, Lili Zheng, and Genevera I Allen. Inference for interpretable machine learning: Fast, model-agnostic confidence intervals for feature importance. *arXiv preprint arXiv:2206.02088*, 2022.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

R. Lippmann. An Introduction to Computing with Neural Nets. *IEEE ASSP Magazine*, 4 (2):4–22, 1987. doi: 10.1109/MASSP.1987.1165576.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

Andrey Nikolayevich Tikhonov. On the stability of inverse problems. In *Dokl. Akad. Nauk SSSR*, volume 39, pages 195–198, 1943.

Mohammad Taha Toghani and Genevera I Allen. Mp-boost: Minipatch boosting via adaptive feature and observation sampling. In *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 75–78. IEEE, 2021.

Tianyi Yao, Daniel LeJeune, Hamid Javadi, Richard G Baraniuk, and Genevera I Allen. Minipatch learning as implicit ridge-like regularization. In *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 65–68. IEEE, 2021.