Environmental and Behavioral Factors that Influence Student Performance

Christopher M. Kollbaum

Northwest Missouri State University, Maryville MO 64468, USA
s556968@nwmissouri.edu

Abstract. Education of our youth is such an important part of our future. Teachers are always looking to find new ways to teach our children and looking for different factors that affect the way a student learns. What are the different components of their lives that cause students to perform better or worse? In this study, I took data found on the UC Irvine Machine Learning Repository where 395 students answered survey questions about different aspects of their lives, and, then had their performance measured in mathematics. Different factors were chosen such as the study times a student spent in a week, the overall health of the student, and the mother's level of education to predict how the student would perform on a comprehensive math exam. Different regression algorithms were used to predict the students' scores. Four such algorithms were run on the data including linear regression, lasso regression, random forest regression, and a neural network. The accuracy of these models turned out to be .708, .674, .832, and .653, respectively, on the test set of data. Our results showed that using a random forest regression model, one can predict performance on summative math exam with a relatively high level of precision.

Keywords: data analytics · machine learning · education · student performance · random forest regression

1 Introduction

In this project, I will use data sets on these different factors and student performance along with machine learning techniques to develop a model that can predict a student's test scores based on these factors. I have been in education my entire career and improving student learning has always interested me. I plan on finding my data source on the UC Irvine Machine Learning Repository website. The steps in completing this study include finding a data set, exploring and cleaning the data, making a decision about the most influential factors that affect student learning, testing and training a model using the data set and the selected factors, analyzing the results, creating a story using visualizations with possible implications for teachers and students, and making conclusions and suggestions for further study (there is always more to study as education is always changing). Some key components when completing this project include the different factors

both regarding student behavior and student environment, as well as, test scores. Student behavior factors could include sleep, gaming time, study time, and others. Student environment factors could include economic level, parent education level, and others. There are limitations that could affect the project. This data is sometimes confidential, so finding appropriate data may be a limitation that we have. Another limitations is that there may not be data on all the influential factors that could affect performance. More limitations regarding this project will be shared later in the document.

1.1 Goals of this Project

The goal of this project is to use different student and environmental factors to train and test a model that will predict student performance. Different regression models will be applied to the data in an attempt to predict student scores and find which model achieves best when working with a particular data set.

2 Methodology

2.1 Data Collection

The data being used is static data retrieved from the UC Irvine Machine Learning Repository. This data includes different characteristics and factors that could possibly affect a student's performance, as well as, test scores for the students. Other data was considered, when searching for the most appropriate data set to obtain our goal. Several data sets on Kaggle were taken into account. The problem with most of these data sets was that the data was synthetic or made up for machine learning purposes. While great to get a feel and practice machine learning, I wanted to chose a data set that was a real-life study of students. Data was downloaded as a csv file from the repository. There was no data scraping that was used to retrieve the data. Many attributes will be used to build our machine learning model. These include such topics as parental education, study time, a student's extra-curricular activities, the gender of the student, whether he or she had internet access at home, the health of the student, and other attributes. The target attribute is the performance on a particular comprehensive math exam taken by selected students. There are no other extraction details that are relevant to this project. Data cleaning including getting rid of unwanted data attributes and converting categorical data to numerical data will be discussed in coming subsections.

2.2 Preprocessing

Cleaning the data for this project is an essential step in getting the machine learning process to work efficiently and accurately. The data was originally downloaded as a csv file where the attributes were listed in one single column separated by semicolons. The first part of the process was to separate these attributes into

individual columns using the text to column function in microsoft excel. After this initial step, the altered csv file was opened and read in a Jupyter .ipynb notebook where pandas and numpy were used to clean the data. There were 33 different attributes in the original data set. These 33 attributes were paired down to what I thought were the most important attributes by using the pandas drop function.

data.drop(['address', 'school', 'age', 'nursery', 'famsize', 'higher', 'failures', 'Mjob', 'romantic', 'Fjob', 'reason', 'goout', 'guardian', 'famsup', 'Dalc', 'Walc', 'G2'], axis = 1, inplace=True)

The other problem that was encountered was the fact that some of the attributes chosen were categorical in nature. The numpy "where" function was used to label these categorical values so now they are integers and can be used in the machine learning process. I wanted to keep the original column in my data frame for reference, so I created a new columns with these numerical values. For example,

data['NewInternetCol'] = np.where(data['internet']=='no',0,1)

Another consideration that must be made is that of the data containing missing or null values. This was check using data.isnull().any(). What was nice about this data set is that it did not contain any missing or null values so no cleaning had to be done when it came to this consideration. It is also worth noting that there were no duplicate rows in the data, either. After all the cleaning was performed, the new data frame ended with 16 attributes and 395 records in the cleaned data.

1. Download data set from UC Irvine Machine Learning Repository
2. Convert csv to have a column for each attribute
3. Load data into a .ipynb notebook in Jupyter Notebooks
4. Check for null or missing values
5. Check for duplicates
6. Select most important attributes and drop unwanted attributes
7. Convert categorical attributes to integer values for machine learning
8. Save and store altered dataset

Data Attributes One of the most important aspects of this project is the selection of the attributes to feed into the machine learning model. As said 16 attributes were selected for this study. The attributes sex, Pstatus, Medu, Fedu, traveltime, studytime, schoolsup, paid, activities, internet, famrel, freetime, health, G1 and absenses are independent variables with G3 performance being the dependent variable. The selected attributes can be seen with detailed definitions in table 1.

Table 1. Data Attributes

Attribute	Definition
Sex	Gender of student
Pstatus	Cohabitation status of parents
Medu	Mother's education level
Fedu	Father's education level
paid	Paid for extra classes in math
schoolsup	extra educational support
traveltime	Travel from home to school
studytime	Weekly studytime of student in hours
activities	If the student is in extra-curricular activities
internet	If the student has at-home internet access
famrel	Level of student's relationship with family
freetime	Number of hours student has free after school
health	Level of student's health
absences	Number of absences during the school year
G1	Math Class Grade
G3	Performance on given math test

2.3 Exploratory Data Analysis

What is Exploratory Data Analysis? Why is it important? Exploratory data analysis is getting to know the data before we get to work on building our machine learning model. This includes using statistics and visualizations to spot patterns, identify outliers and anomalies, test hypotheses, and check assumptions we have about our data. This is important as we can find any possible errors or interesting features in the data. This, also, includes cleaning our data (getting rid of missing values, etc). With EDA, we will look at the different distributions of the attributes to find out the spreads, the centers of our data columns, minimums, maximums, and other statistics relevant to the variables. We will look for relationships amongst our variables and just get a better understanding of our data. This will allow us to be more efficient and accurate in building our model.

Exploratory Data Analysis Process The first part of the process is to get to know what data you are working with. I started by figuring out what the attributes were and how many I was working with, typical values of these attributes, and data types that were present in the data. I checked the head and tail of the data using data.head() and data.tail() in a Jupyter notebook with Python pandas library. This allowed me to see all my columns and start the process of selecting which attributes I was going to using in my model. Columns like address and if the student was in a relationship didn't seem as important as study time and if the student took a prep course, so these columns were eventually dropped. I was also interested in the shape of the data, so I used data.shape and found I had 33 columns and 395 records. This also helped me decide to limit

my model to 16 attributes to avoid confusion and overfitting. I used data.dtypes to figure out what types of data I had and realized that I would have to convert some of my data from strings to integers to make them usable for my eventual model. Finally, I used data.describe to show some basic statistics of the columns and found no unusual min or max values, as well as, no unusual means or standard deviations, telling me I was good to keep the attributes I selected as they were originally laid out.

The next step in my EDA journey was to go ahead and clean the data. The unwanted columns were dropped using data.drop () in Python. Some of my selected variables were categorical in nature so I used the numpy library in Python to convert these variables into integer dummy variables that can be used in my model. The np.where function was perfect to make these conversions. I checked for null/missing values using data.isnull() and found that I had no null or missing values that I had to drop or fill in.

When all the necessary cleaning was completed, I wanted to find out about any relationships that may not be obvious by just looking at the data, but may exist. After importing the Seaborn library in Python, I decided to create a heat map using sns.heatmap (multivariate data analysis) to find out about any correlations that may exist between the variables. I saw that most of the correlations between the variables are low, which is good for running a linear regression model. Using sns.displot, histograms (univariate data analysis) were created to get a better idea of the distributions I was working with and, again, to get a feel for the typical values of the variables.[6] Examples of these visualizations can be found in figures 1 and 2.

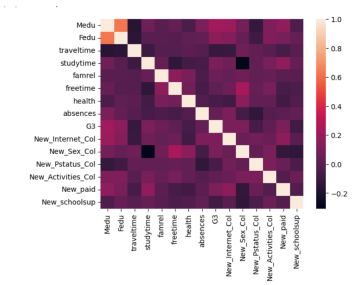

Fig. 1. Heat Map of Features

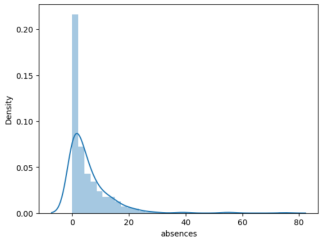

Fig. 2. Distribution of Absences

Results of EDA and what was learned After performing the analysis with these commands in Python, I was able to find that I needed to pair down my attributes to a select few. Again, I ended with 16 attributes from the original 33. I found no reason to drop any records because of missing or duplicate values. I ended with the original 395 records. I also was able to identify some variables that I had to convert to integers. For example, I converted the gender of each

student from female and male to 0 and 1, respectively, to make them usable in the project. Creating histograms allowed me to see my distributions. For example, I found that the absences variable is skewed right with typical values close to zero. Lastly, I found that the correlations amongst my variables was rather low in almost all cases making this data a prime candidate for a linear regression model or other models. This EDA process made me very confident that my data is ready to continue the project and solve the problem of predicting performance based on a student's behavioral and environmental factors.

2.4 Machine Learning and Regression

Mechanism and Pipeline To start the model building process, I loaded a new .ipynb notebook in Jupyter Notebooks. In the new notebook, I created multiple pipelines for each of the different regression model I created. Each pipeline included running the data through the standard scalar in sklearn to manipulate the data so the mean of each variable was zero with a standard deviation of one followed by running the data through the different machine learning algorithms.

pipelinelinreg=Pipeline([('scalar1', StandardScaler()),('Linreg', LinearRegression())])

After the pipelines were created, a list was made containing the different pipelines and a for loop was used to fit each pipeline and create the models.

```
for pipe in pipelines:
    pipe.fit(X_train, y_train)
```

After each algorithm was fit, a dictionary was created containing the different algorithms and another for loop was used to display the algorithm with its accuracy score.

```
pip_dict = {0: 'Linear Regression', 1: 'Lasso Regression', 2: 'Random Forest Regressor}

for i,model in enumerate(pipelines):
    print("{} train accuracy {}".format(pip_dict[i], model.score(X_train,y_train)))
```

Machine Learning Algorithms Different regression algorithms were chosen to see which model would perform best. Linear regression was chosen first to see if the data could be fit to a regression line. I tried a lasso regression which forces some variables to have a coefficient of zero if they are not important or are redundant to the model. Random forest (uses decision trees which perform binary splits on predictor variables)[7] and neural net regression (makes decisions based on brain science and simulated brain networks) were also considered.

Training and Testing Process The data was split using the SKLearn model selection library and the test, train, split function. The data was split so that 80 percent of the data went to the training set, and 20 percent of the data was used

for the testing set. This gave me 316 records in the training set and 79 records in the testing set. A random state was set so that when the split was performed, the data was shuffled identically each time.

Implementation and Evaluation Implementation of this project involved several steps. A problem and goal was established. Data was found and downloaded that would help solve the problem and meet the goal. Exploratory data analysis was performed. In the EDA step, I became familiar with the data attributes, their data types, and the size and shape of the data. Data cleaning was performed. Unwanted features were dropped and dummy variables were inserted for the categorical features. Lastly, in the EDA step, I explored the relationships and distributions of the features using heat maps and histograms. The next step in the implementation was the building of the predictive model. Again, the data was split into 80 percent training data and 20 percent testing data. The different regression algorithms were selected. Pipelines were created that scaled the data in a standard fashion and then ran it through each algorithm. Finally, the mean absolute error and the accuracy of the algorithm were calculated to make a decision on which model performed the best on the different attributes. [4]A list of the code used can be found at https://github.com/kollbaumc/KollbaumCapstone.

3 Limitations

Our data was limited to just 395 students in a small area. While predictions could be made for these students, these features in different cultures or areas of the world may have different affects on these predictions. It would be difficult to find a different data set that would measure results of students worldwide. Another hangup that was encountered was that the models ran poorly when not including previous student performance in a math course. When not including this metric, all models had an accuracy of around .2 with mean absolute errors above 4. Lastly, this data had included 33 features that definitely could affect how a student performs on a comprehensive test, but there are many other factors that also could be included in the study. John Hattie, who is one of the foremost experts in the field, has come up with hundreds of different factors affecting learning including using calculators, using games in instruction, and ability grouping just to name a few.[2][3][5] So our data may be limited when it comes to the big picture of instruction and all of these factors.

4 Results

The random forest regression performed the best when working with this data followed by the linear regression model. It was strange that my lasso regression model was outperformed by the linear regression model as this is usually not the case, but was in this case. As we can see from the following table, the random forest model, by far, outperformed the others in both accuracy and

mean absolute error. The other models had similar performance with accuracy scores all around 60 to 70 percent and mean absolute errors around 2. The random forest had both a significantly lower mean absolute error (.69 and 1.44, respectively) and higher accuracy on both the training and test sets (.956 and .832, respectively). Bar charts were created to better illustrate the fact that the random forest algorithm performed far better than the other models with the random forest having a significantly higher bar for accuracy and significantly lower bar for mean absolute error.

Table 2. Performance of Machine Learning Models

Model	MAE(Training)	Accuracy(Training)	MAE(Testing)	Accuracy(Testing)
Linear Regression	1.92	.656	2.05	.708
Lasso Regression	1.91	.593	1.96	.674
Random Forest	.69	.956	1.44	.832
Neural Net	1.92	.663	2.21	.653

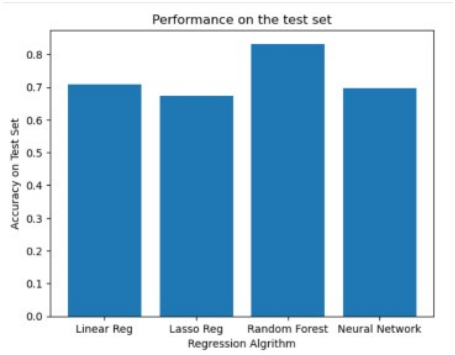

Fig. 3. Accuracy of Models

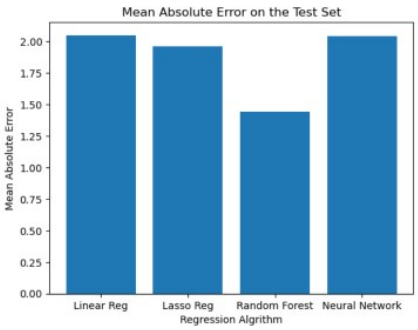

Fig. 4. MAE of Models

5 Conclusions and Further Study

There are many conclusions and inferences that we can make after analyzing the data using the different models and creating visualizations. Scatter plots were created that showed how well our random forest model predicted the student's scores in figures 5, 6, and 7. The actual score is represented by the x-axis and the predicted score is represented by the y-axis. A red line was added to show what the perfect predictions would be. As we can see from the figures, our model did a good job of predicting the actual scores. It did, however, struggle a bit when

the actual scores were zero. With all the different attributes being considered, the model, while predicting poor scores, didn't (for the most part) predict scores of zero. Also of note, there don't really seem to be very many outliers as far as the scatter plots go with the exception of one outlier in the test set where the prediction of the student was much lower than the actual score. With only 79 instances in the test set, this is probably the reason the accuracy on this set was a little lower than it was on the training set.

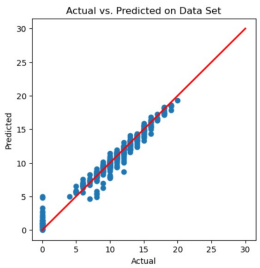

Fig. 5. Scatter Data Set

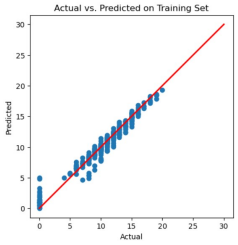

Fig. 6. Scatter Training Set

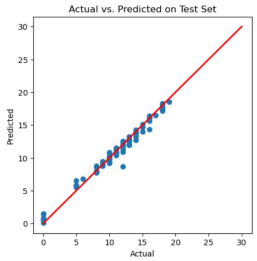

Fig. 7. Scatter Test Set

One of the most interesting and helpful visualizations was that of the following horizontal bar chart which shows the importance that each attribute has on the random forest model. As you can see in figure 8, 'G1', the prior performance of the student in their math course, is the most important attribute in the model. Absences the student had during the school year was the second most important attribute to the model and contributed greatly to the prediction. The model was run without these two attributes in doing a little exploratory analysis, but the accuracy of the model decreased significantly with accuracy scores around .2. In order to get decent predictions, it would seem that some sort of measure of prior performance or measure of intelligence is needed to get a adequate prediction. One surprising aspect that this chart showed was that the gender of the student really didn't matter when it came to predicting the student's test score. There has always been the myth out there that math is a boy's subject and that females are not as good at math. If that was truly the case, then gender would be a good indicator of performance, but it wasn't in this data. Even though it wasn't important in this data set, gender and its influence over math performance would be an interesting topic to look at in the future. We have to remember that this data only includes 395 students. In a larger data set gender may be influential in predicting performance. Another surprising observation that could be taken from the horizontal bar chart was that a mother's level of education had more relative importance to the model than the father's level of education. This would be another instance where more research and analysis could be done in the future to really see if this is truly the case and and why this may be true. Finally, it was also interesting that whether a particular student had aspirations of obtaining a higher level of education had practically no influence on the model. One would

think this, again, would be a good indicator of performance with students who had motivation to further their education performing better. There are so many factors that can have some affect on student performance that no one feature may make a huge difference when taking out prior class performance and school attendance. In fact, in one study, John Hattie lists over 200 features that could make a difference, and most are just things that can be done in the classroom.[5] There so many other factors outside the classroom, also, that could influence how well a student does. With hundreds of possible influences, it may be hard to create a good model with just 10-15 features. Overall, however, it does seem possible to get at least a decent idea of how a student will perform in a certain subject by using different aspects of their lives and running this data through different machine learning algorithms.

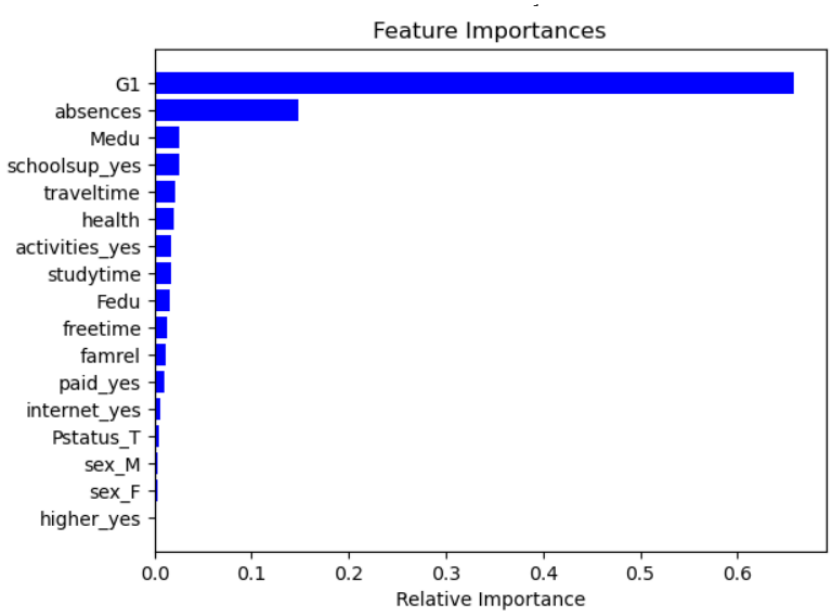

Fig. 8. Importance of Data Attributes

References

1. Acharya, B.R.: Factors affecting difficulties in learning mathematics by mathematics learners. International Journal of Elementary Education **6**(2), 8–15 (2017)
2. Hattie, J.: Influences on student learning. Inaugural lecture given on August **2**(1999), 21 (1999)
3. Hattie, J.: Teachers make a difference, what is the research evidence? (2003)

4. Kollbaum, C.: Kollbaum github captone (2023)
5. Learning, V.: Hattie ranking: 252 influences and effect sizes related to student achievement. Visible Learning. Retrieved September **13**, 2021 (2018)
6. Shin, T.: An extensive step by step guide to exploratory data analysis (2020)
7. Speiser, J.L., Miller, M.E., Tooze, J., Ip, E.: A comparison of random forest variable selection methods for classification prediction modeling. Expert systems with applications **134**, 93–101 (2019)
8. Yang, X.: Study on factors affecting learning strategies in reading comprehension. Journal of Language Teaching and Research **7**(3), 586 (2016)