Predicting Academic Performance using Demographics

Amos Roche, Mahvash Jebeli, N'yoma Diamond

December 16, 2021

Abstract

As education has become more ubiquitous over the years, the diversity of students has exploded in numerous aspects. This being the case, one of the most prevalent concerns within education is that of educational equity; i.e. how do we ensure that students are receiving the tools, resources, and attention necessary to perform on the same level as their peers? One of the most pressing problems in search of a solution with regards to educational equity is that of identifying which students need support in the first place. Our goal in this paper will be to analyze the use of various statistical methods to develop models for the prediction of grades as a means of identifying in-need students.

1 Introduction

Can we predict the future performance of a student based on the reality of one's present? Can we identify the ones more in need of help to prosper? Is our education system fair to students and provides them with the best-suited support? As statisticians, we believe we can predict the future of a system with measured data points, even if the system is an academic success. Further, because we think our current educational system is not fair and needs modifications, in this project, we will propose an evaluation of a dataset from demographic information of students and their academic performance. By analyzing potential academic performance as a result of demographic data, it could be possible to pre-emptively identify students who may perform poorly and provide them extra resources to perform on par with their peers. We will also be able to see which variables and factors are affecting the performance of the students which can help us identify which students need intervention. The proposed dataset is particularly interesting and relevant to examine as it has a real-world impact.

Previous studies on the relationship between the demographic data of the students and their academic performance has shown that the age, gender, and high school GPA significantly impact the academic performance of them later in college[1], which aligns with related research emphasizing the importance of the demographic situation of students during their high school careers[2, 3, 4]. This topic of research shows the importance of the investigations related to predicting the student performance to later help the ones more in need, as it has previously been evaluated for medical students in Japan[5].

The Student Performance Data Set[6] is a dataset containing demographic information for students in two Portuguese schools. This data was collected for the purpose of predicting the academic performance of a student in terms of grades. The dataset is split into two subsets: One for Mathematics classes, and one for Portuguese Language classes. The data was recorded for 395 students. Recorded data include information such as alcohol consumption, occupation of parents, health, leisure, previous academic history, family size, and other potentially meaningful metrics. Some other recorded information was the number of absences, freetime, and whether or not the students participated in extracurricular activities. For each data subset (Math and Language), the first period grade, second period grade, and final grades were recorded. In our analysis we will only make use of the final period grades in order to simplify our analysis.

To reach our purpose of identifying those in need of additional assistance, we will utilize a suite of models including various Linear Regression methods and Random Forest ensemble methods. Additionally, we will explore the usage of Principal Component Analysis, Varimax Orthogonal Rotation, and Partial Least Squares in analysis of the data itself and toward consideration of other potential directions for this work.

2 Pre-Processing and Component Analysis

Pre-processing and factor/component analysis is a vital step in data science problems because it is rare that our chosen dataset is ready for modeling/analysis. Pre-processing helps us scale our variables, so columns with large values are not given a stronger weight than columns with smaller variables. Re-encoding variables may be a necessary step in pre-processing, because the numbers represented in the column may refer to a label rather than a continuous numerical variable.

Principal Component Analysis (PCA) and factor analysis is useful when there are a lot of predictors in our data. Converting the data into a lower dimension of components may increase model interpretability while minimizing information loss of the data. These components in the data may explain most of the variability in the data as if we used most or all of the predictors. The student performance dataset consisted of over 40 features each for the Math and Language datasets. PCA and factor analysis was conducted on the student performance dataset to compact the dataset of 40+ features into a dataset consisting of a much lower number of dimensions.

The methods that were selected for Pre-Processing and Component analysis are as follows:

- Pre-Processing
  - Variable Re-encoding
  - Variable Standardization
  - Outlier Detection using Z-Scores
- Correlation Feature Selection
- Principal Component Analysis
- Varimax Orthogonal Rotation
- Partial Least Squares Regression

2.1 Pre-Processing

The first step was re-encoding variables using one-hot encoding. There were two variables named "Medu" and "Fedu" which are variables that contain information about the education level of the Mother and Father respectively[6]. These two variables needed to be re-encoded because these two columns ranged from 0-4, where 0 represents no education, 1 represents primary education (4th Grade), 2 represents 5th-9th Grade, 3 represents secondary education (High School), and 4 represents Higher Education studies[6]. Since these values are not numerical variables, rather these variables contain values which map to a specific category, it will be difficult for machine learning to interpret these values. The variables were re-encoded by adding 3 columns for "Medu" and "Fedu" called "Medu_1", "Medu_2", "Medu_3", "Fedu_1", "Fedu_2", "Fedu_3". For example, if Medu_1 and Fedu_1 contained a value of 1, that means both of the parents completed education until 9th grade. If all of the three columns contain a value of zero, this indicates that the parent did not receive any education.

There were also many binary data columns that needed to be re-encoded because they contained two linguistic terms as values. For example, the address column contained two values "R" for Rural and "U" for urban [6]. The address column was re-encoded, where Rural is labeled as 0 and Urban is labeled as 1. This same procedure was done for "school", "sex", "famsize", "Pstatus", "schoolsup", "famsup", "paid", "activities", "nursery", and "romantic".

The next step in the pre-processing process was standardizing the variables. Standardizing variables means scaling the variables to have a mean of 0 and a standard deviation of 1. This allows the variables to be on the same scale so columns with much larger values are not given more weight. Variable Standardization is important for principal component/factor analysis because component analysis tries to make linear combinations of the variables, and if a column has much larger values than another column, the coefficient for the linear combination for the large valued column may be very high, so that is why we want all the variables on the same scale. The data for the independent variables were standardized using Scikit-learn's StandardScaler function.

The last step in the pre-processing stage was trying to remove outliers in the data. This was done by calculating the Z-Scores for the data points using Scipy's Z-Score function. Data observations that

contained column values with Z-Scores greater than 3 were removed. 166 out of 395 data observations were removed for the Math dataset and 256 out of 649 data observations were removed for the Language dataset

2.2 Correlation Feature Selection

As stated before, the Math and Language datasets consisted of over 40+ features. Correlation feature selection was conducted to try to remove some of the correlated variables before conducting component analysis. Figure 1 represents the two correlation heatmap matrices for the Math and Language datasets respectively. If we take a look at the two figures, many of the features for these two datasets are already uncorrelated.

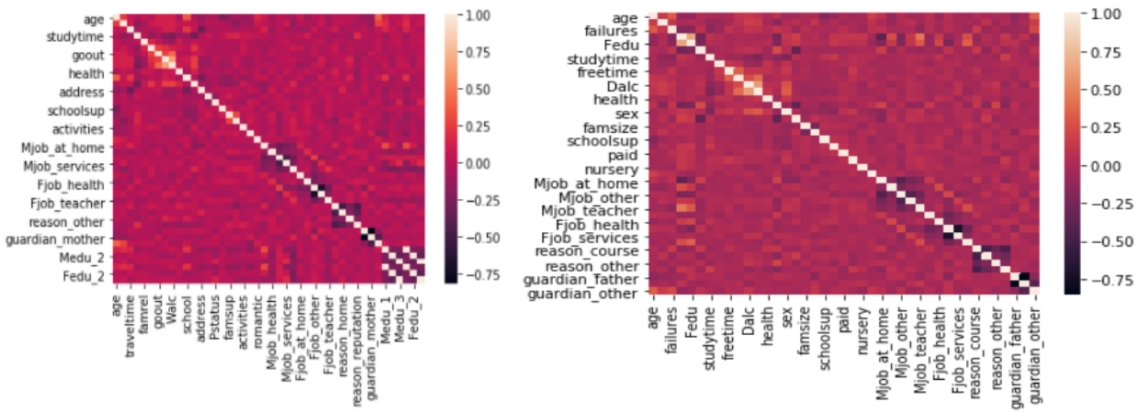


Figure 1: The left graph represents the correlations between the independent variables for the Math dataset. The right graph represents the correlations between the independent variables for the Language dataset. As we can see in the graphs, many of the features are uncorrelated

The implementation for the correlation feature selection was removing features with a correlation above 0.5 or below -0.5, since no features had correlations above in the 0.7+ range, except the correlations with two same features (the correlation score of two same features is 1). 5 columns were removed for the math dataset, while 3 were removed for the language dataset.

2.3 Principal Component Analysis

Principal component analysis was first conducted by fitting the standardized data without removing the outliers. Since the Math and Language datasets only have 395 and 649 observations respectively, removing outliers for the Math data would remove 42% of the data observations and if we removed the outliers for the Language data, that would remove 39.4% of the data observations. Losing data to that magnitude can hurt the modeling process especially when we do not have too many data points to begin with.

After using Scikit-learn's PCA function[1], the two plots from Figure 2 were generated. If we look at the left graph from Figure 2 (Math Dataset) and Figure 3 (Language Dataset), we can see each principal component does not capture too much variability in the data. It takes over 20 components to capture 80% of the datasets variability. Usually in PCA, we want to have a low number of components because we want each component to carry more variability. We can also see in the right graph in Figures 2 and 3, there are over 20 components with eigenvalues greater than 1. A big reason why it takes a lot of components to capture the data's variability is because PCA excels when many of the independent variables are correlated and in the Math/Language datasets, most of the features are extremely uncorrelated as their respective correlation scores indicate little to no correlation. We can see in Figure 4, the combined percentage of total variability for both of the datasets for the first three Principal components is only around 17-18%.

[1]See https://scikit-learn.org/stable/modules/decomposition.html#pca

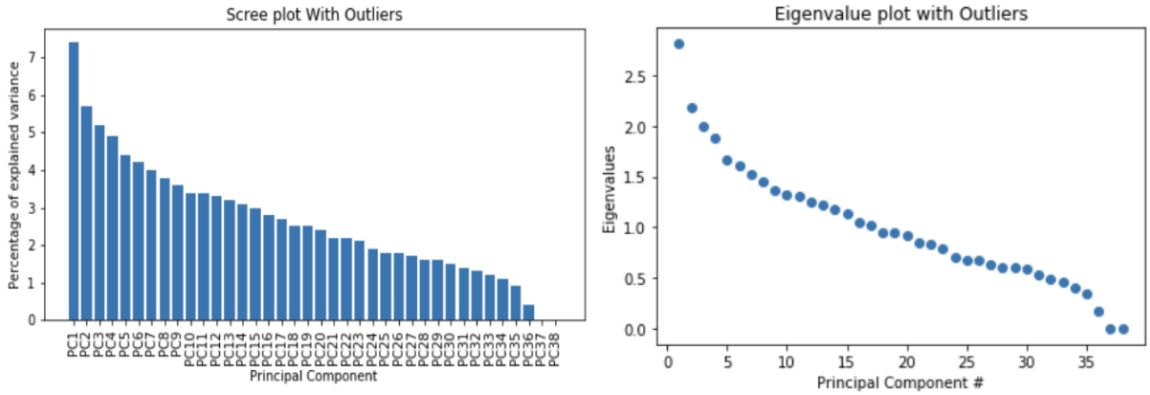


Figure 2: The left graph represents the Scree plot for the Math Data which represents the variability of each Principal Component for the PCA data that was fit with outliers included. The right graph represents the eigenvalue plot for the PCA data which shows us the eigenvalue for each principal component.

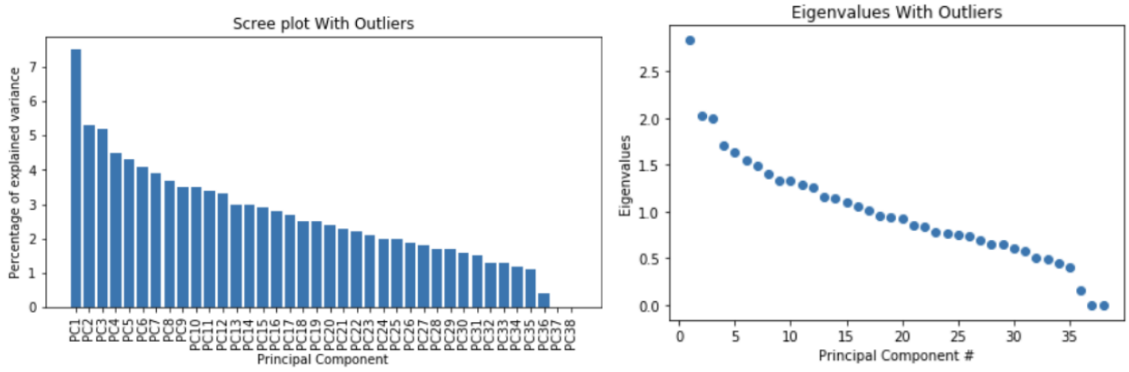


Figure 3: The left graph represents the Scree plot for the Language Data which represents the variability of each Principal Component for the PCA data that was fit with outliers included. The right graph represents the eigenvalue plot for the PCA data which shows us the eigenvalue for each principal component.

If we look at Figure 5, we can see the results of conducting PCA on the math dataset. The results are very similar to when the PCA was fit on the data without removing outliers. Each component does not capture too much of the data's total variability and there are also many components with eigenvalues greater than 1. The results of the PCA fit on the language dataset with outliers removed were very similar as the math dataset.

2.4 Varimax Orthogonal Rotation

Since the results of the PCA consisted of getting many components with low variability and many components with eigenvalues greater than 1, we wanted to observe how the Varimax orthogonal rotation of the datasets would affect the component/factor analysis. Orthogonal rotation is used when we assume the independent variables are uncorrelated. This helps minimize the complexity of the factor loadings to make a simpler structure to interpret [7]. Rotating the data can also help maximize the variance of the principal components. If we look at Figure 6, we can say the varimax rotation helps reduce the amount of eigenvalues greater than 1. When we conducted PCA, there were over 20 components with eigenvalues greater than 1, now we have around 10 components with eigenvalues greater than 1. This allows for better interpretability, because if many principal components have eigenvalues greater than 1, it makes it difficult to see the importance of each principal component. If

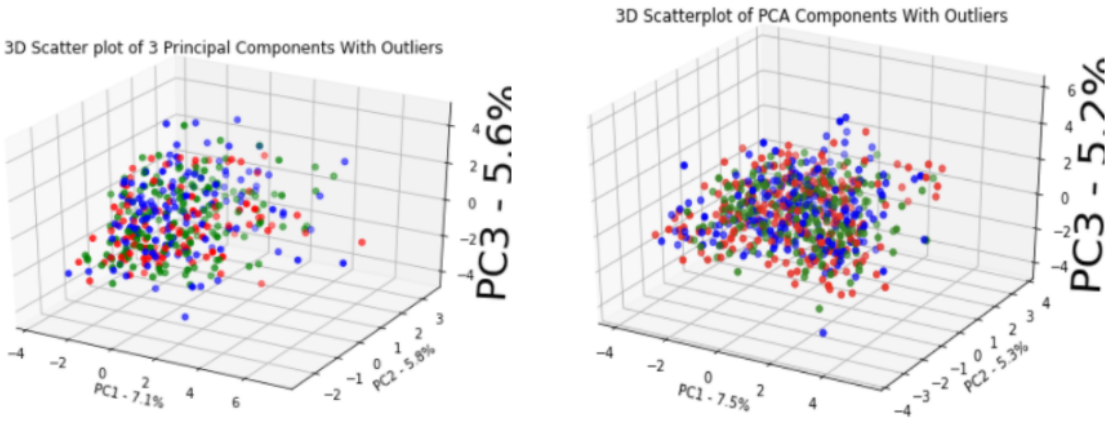


Figure 4: This figure represents the 3-D scatterplot of the first 3 PC's plotted against each other. The left graph represents the scatterplot for the Math dataset. The right graph represents the scatterplot for the language dataset.

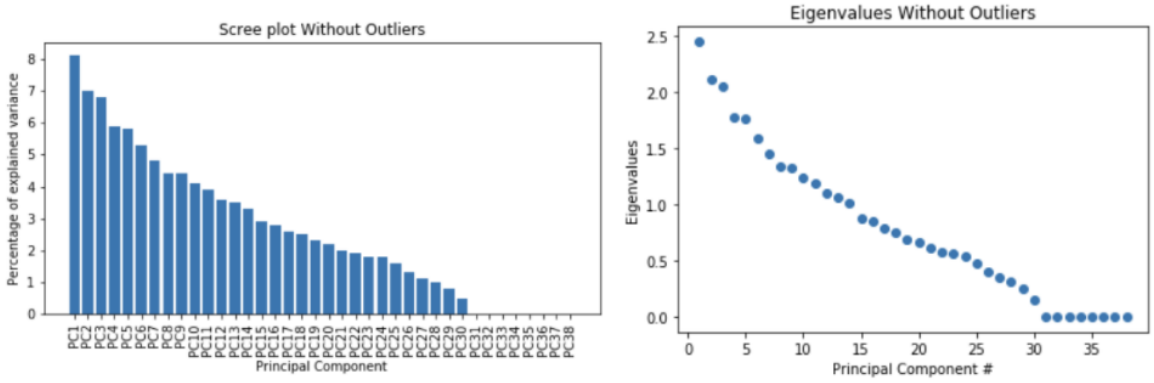


Figure 5: The figure represents fitting PCA on the Math dataset when outliers were removed. The left graph represents the variability for each principal component. The right graph represents the eigenvalue for each principal component.

we look at the left graph of Figure 6, we can see the data is much more in individual clusters after we rotate the data rather than being all in one large cluster when PCA was conducted. Figure 8 represents the varimax rotation for the Math dataset, the varimax results were very similar for the language dataset as the number of components with eigenvalues greater than 1 was also around 10 components. We can see in Figure 7, that the varimax rotated data for the language helped us improve our predictive power by producing a test RMSE of 2.33, which was the lowest test RMSE for all of the models conducting on all of the possible datasets.

2.5 Partial Least Squares Regression

The last stage of the component/factor analysis was conducting a partial least squares regression to see the predictive power of PLS for the math and language datasets. If we look at the left graph for Figure 8, the graph shows us the number of components vs the cross-validated MSEs. As we can see in the graph 1-2 components, gives us the lowest cross-validated MSEs for the Math dataset. The right graph in Figure 8 shows us the resulting test MSE after fitting the PLS regression with two principal components. The resulting test RMSE was 4.79 for the two component PLS regression fitted with the standardized Math dataset. The left graph in Figure 10 shows us that 3 components is the optimal amount for the PLS regression on the language dataset. The right graph in 9, shows us that

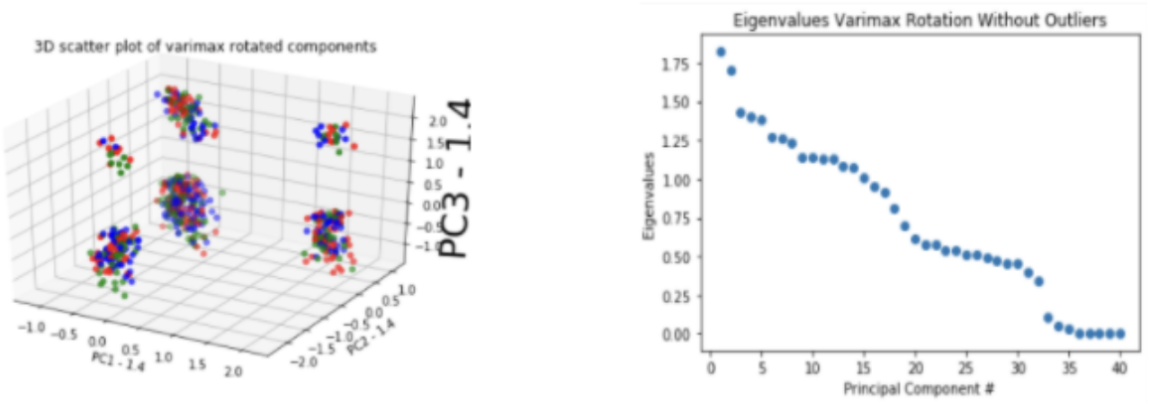


Figure 6: The left graph represents the 3-D scatterplot of the first three principal components of the Math dataset. The right graph represents the eigenvalue plot which tells us the eigenvalue corresponding to each principal component.

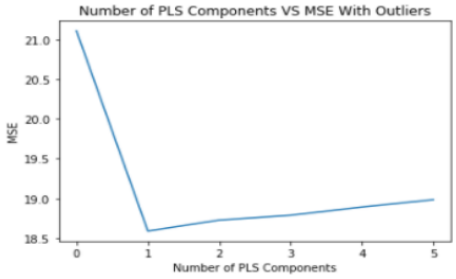
```
from sklearn.metrics import mean_squared_error
rf = RandomForestRegressor(**search.best_params_)
rf.fit(X_train, y_train)

# evaluating the test RMSE
display(np.sqrt(mean_squared_error(y_test, rf.predict(X_test))))
```

```
2.333091220647903
```

Figure 7: This figure displays code for fitting the random forest with the Varimax rotated data for the Language dataset. The resulting test MSE was 2.33.

the resulting test RMSE was 2.376 for the 3 component PLS regression on the language data. The RMSE for the language dataset was much higher than the math dataset which may suggest using PLS regression with the language dataset may give better predictive power.




```
pls2 = PLSRegression(n_components=2)
pls2.fit(X_train[selected_columns], y_train)
Y_pred = pls2.predict(X_test[selected_columns])
np.sqrt(mean_squared_error(y_test, Y_pred))
```

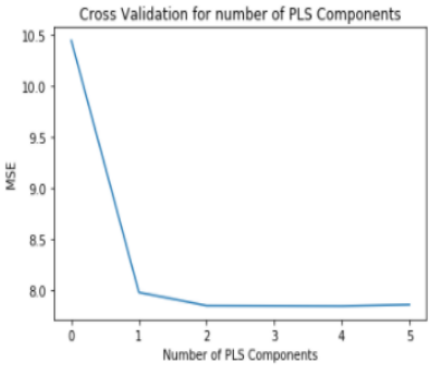
```
4.79107829849631
```

Figure 8: The left graph represents the cross-validated MSE corresponding to the number of PLS components for the Math dataset. The right graph represents fitting the PLSRegression for the Math Dataset with 2 components and calculated a test RMSE of 4.79.

3 Linear Regressions

Linear regressions are some of the simplest and most commonly used statistical analysis and prediction techniques. By fitting linear relationships between a predictors and a response variable, we receive a highly explainable model describing numerically the connections between a predictor and a desired response variable. In our case, we would be fitting final grades as our response variable with respect to demographic data or the principal components of the dataset.

Given all the different options available for linear regressions, we decided to select a suite of feature




```
pls2 = PLSRegression(n_components=3)
pls2.fit(X_train, y_train)
Y_pred = pls2.predict(X_test)
np.sqrt(mean_squared_error(y_test, Y_pred))
```

```
2.3758997387848138
```

Figure 9: The left graph represents the cross-validated MSE corresponding to the number of PLS components for the Language dataset. The right graph represents fitting the PLSRegression for the Language Dataset with 3 components and calculated a test RMSE of 2.376.

selection and regularization techniques in order to cast a wide net in finding an optimal model. We settled on the following suite of techniques to use in our model development and analysis:

- Forward Stepwise Selection
- Backward Stepwise Selection
- Recursive Feature Elimination[2]
- Ridge Regression
- Lasso Regression

All models were evaluated using cross-validation in order to identify the most optimal hyperparameters and feature subsets. Leave-one-out cross-validation was used for Ridge and Lasso regressions, while 20-fold cross-validation was used for Forward and Backward Stepwise Selection, as well as Recursive Feature Elimination due to computational intensiveness. The training set used to fit all models was 80 percent of each of our datasets, so 20 percent was withheld for evaluation during testing.

Table 1 displays which and how many features were selected for models trained using each of these different methods. Table 2 displays the number of occurrences for each feature or principal component within the models trained for each of our datasets. Looking at these tables we can make some interesting cursory observations: One immediate observation that can be made is that Recursive Elimination selected significantly fewer features compared to other techniques for mathematics dataset with both the raw data as well as the PCA version of the dataset. By contrast, with the raw language dataset Recursive Elimination used almost all of the features (37 out of the 40 available features). Unfortunately due to a lack of knowledge as to how Recursive Elimination works "under-the-hood" it is unclear as to why this behavior occurred. Another point of interest for the models fitted on the raw data is the fact that every model made use of the "failures" feature. In addition to being one of only two numeric predictors we allowed the models to use, but it is the only predictor that is not a demographic; the "failures" feature refers to the number of classes a student has failed in the past[citation here]. This feature is likely directly related to academic performance and future performance prediction as the number of times a student has failed classes can easily be assumed to relate to their likelihood of future academic success. Another feature of the raw data used in every single model was the "health" feature, which refers to the health status of students at the time of data collection on a scale of 1 (being "very bad") to 5 ("very good")[citation here]. This is similarly unsurprising as it can be assumed a student in poor health will underperform relative to their peers compared to a student in good health. Other features of the raw data which saw frequent use are the "studytime", "sex", "schoolsup", and "Mjob_health" features, which were present in seven out of eight models trained to eliminate features; and the "Walc", "famsize", "romantic", "Mjob_services", "Fjob_services" features, which were present

[2]See Seehttps://scikit-learn.org/stable/modules/feature_selection.html#rfe

in six out of eight models trained to eliminate features. Another interesting observation to consider is the fact that many of the models trained on the principal components made use of components which describe little variation in the data. Namely, every single model for the Math dataset using principal components made use of components 17 and 29, while completely ignoring more notable components such as components 4 and 7. Similarly, every single model for the Language dataset using principal components made use of components 21, 24, 25, 29, 31, 32, and 35, while all simultaneously ignoring component 8 and others. It is unclear as to why this might be the case, as typical assumptions about PCA would suggest that earlier components would describe a greater proportion of variation in the data and thus be better for prediction, however it would appear this was not the case in practice.

	Dataset	Method	Selected Features	Count
Raw	**Math**	Forward Selection	age, failures, Medu, studytime, freetime, goout, Walc, health, sex, famsize, schoolsup, famsup, activities, romantic, Mjob_health, Mjob_services, Fjob_other, Fjob_services, reason_home, reason_othe	20
		Backward Selection	age, failures, studytime, sex, famsize, schoolsup, romantic, Mjob_health, Mjob_services, Fjob_health, Fjob_other, Fjob_services, Fjob_teacher, reason_course, reason_home, reason_other, reason_reputation, guardian_father, guardian_mother, guardian_other	20
		Recursive Elimination	failures, Mjob_at_home, Mjob_health, Fjob_teacher	4
		Lasso	age, failures, Medu, traveltime, studytime, freetime, goout, Walc, health, school, sex, address, famsize, schoolsup, famsup, paid, romantic, Mjob_health, Mjob_services, Mjob_teacher, Fjob_other, reason_cours	22
		Ridge	All	40
	Language	Forward Selection	failures, Medu, studytime, famrel, freetime, Walc, health, school, sex, address, famsize, schoolsup, activities, nursery, romantic, Mjob_health, Mjob_services, Mjob_teacher, Fjob_services, guardian_fathe	20
		Backward Selection	failures, studytime, Walc, health, school, sex, schoolsup, Mjob_at_home, Mjob_other, Mjob_services, Fjob_at_home, Fjob_health, Fjob_other, Fjob_services, Fjob_teacher, reason_course, reason_other, guardian_father, guardian_mother, guardian_othe	20
		Recursive Elimination	age, failures, Medu, studytime, famrel, freetime, goout, Dalc, Walc, health, school, sex, address, famsize, Pstatus, schoolsup, paid, activities, nursery, romantic, Mjob_at_home, Mjob_health, Mjob_other, Mjob_services, Mjob_teacher, Fjob_at_home, Fjob_health, Fjob_other, Fjob_services, Fjob_teacher, reason_course, reason_home, reason_other, reason_reputation, guardian_father, guardian_mother, guardian_other	37
		Lasso	age, failures, Medu, studytime, famrel, freetime, goout, Dalc, Walc, health, school, sex, address, famsize, schoolsup, activities, nursery, romantic, Mjob_at_home, Mjob_health, Mjob_other, Mjob_teacher, Fjob_services, Fjob_teacher, reason_other, reason_reputation, guardian_father	27
		Ridge	All	40

	Dataset	Method	Selected Features	Count
PCA	**Math**	Forward Selection	PC1, PC2, PC5, PC6, PC8, PC9, PC10, PC11, PC12, PC13, PC14, PC15, PC17, PC24, PC25, PC28, PC29, PC30, PC35, PC36	20
		Backward Selection	PC1, PC2, PC6, PC8, PC9, PC10, PC12, PC13, PC15, PC16, PC17, PC24, PC28, PC29, PC30, PC31, PC32, PC33, PC36, PC38	20
		Recursive Elimination	PC1, PC17, PC29, PC36, PC37, PC38, PC39, PC40	8
		Lasso	PC1, PC2, PC3, PC6, PC8, PC9, PC12, PC13, PC17, PC24, PC29	11
		Ridge	All	40
	Language	Forward Selection	PC1, PC3, PC7, PC13, PC15, PC16, PC17, PC21, PC23, PC24, PC25, PC26, PC27, PC29, PC31, PC32, PC34, PC35, PC37	19
		Backward Selection	PC1, PC3, PC7, PC10, PC13, PC16, PC17, PC21, PC23, PC24, PC25, PC26, PC27, PC29, PC31, PC32, PC34, PC35, PC37	19
		Recursive Elimination	PC1, PC3, PC7, PC13, PC21, PC24, PC25, PC29, PC31, PC32, PC35, PC37, PC38	13
		Lasso	PC1, PC2, PC3, PC4, PC5, PC6, PC7, PC9, PC10, PC11, PC12, PC13, PC14, PC15, PC16, PC17, PC20, PC21, PC23, PC24, PC25, PC26, PC27, PC28, PC29, PC31, PC32, PC34, PC35	29
		Ridge	All	38

Table 1: Features selected using different linear regression development techniques. Features were eliminated from Lasso regression if their coefficient equaled zero. Note that the principal components for the Math and Language datasets are not the same, as PCA was fitted separately for each of the datasets and the Math dataset had 40 principal components, while the language dataset had 38.

Raw Feature	Math Models	Language Models	TOTAL	Principal Component	Math Models	Language Models
age	3	2	5	PC1	4	4
failures	4	4	8	PC2	3	1
Medu	2	3	5	PC3	1	4
Fedu	0	0	0	PC4	0	1
traveltime	1	0	1	PC5	1	1
studytime	3	4	7	PC6	3	1
famrel	0	3	3	PC7	0	4
freetime	2	3	5	PC8	3	0
goout	2	2	4	PC9	3	1
Dalc	0	2	2	PC10	2	2
Walc	2	4	6	PC11	1	1
health	4	4	8	PC12	3	1
school	1	4	5	PC13	3	4
sex	3	4	7	PC14	1	1
address	1	3	4	PC15	2	2
famsize	3	3	6	PC16	1	3
Pstatus	0	1	1	PC17	4	3
schoolsup	3	4	7	PC18	0	0
famsup	2	0	2	PC19	0	0
paid	1	1	2	PC20	0	1
activities	1	3	4	PC21	0	4
nursery	0	3	3	PC22	0	0
romantic	3	3	6	PC23	0	3
Mjob_at_home	1	3	4	PC24	3	4
Mjob_health	4	3	7	PC25	1	4
Mjob_other	0	3	3	PC26	0	3
Mjob_services	3	3	6	PC27	0	3
Mjob_teacher	1	3	4	PC28	2	1
Fjob_at_home	0	2	2	PC29	4	4
Fjob_health	1	2	3	PC30	2	0
Fjob_other	3	2	5	PC31	1	4
Fjob_services	2	4	6	PC32	1	4
Fjob_teacher	2	3	5	PC33	1	0
reason_course	2	2	4	PC34	0	3
reason_home	2	1	3	PC35	1	4
reason_other	2	3	5	PC36	3	0
reason_reputation	1	2	3	PC37	1	3
guardian_father	1	4	5	PC38	2	1
guardian_mother	1	2	3	PC39	1	0
guardian_other	1	2	3	PC40	1	0

Table 2: Number of models containing each feature or principal component, excluding Ridge regressions. Ridge regression was ignored as it used all features in the dataset and would not be meaningful to include in this summary. Features were eliminated from Lasso regression if their coefficient equaled zero. Note that the principal components for the Math and Language datasets are not the same, as the PCA was fitted separately for each of the datasets and the Math dataset had 40 principal components, while the language dataset had 38. As such totals have not been calculated for principal components.

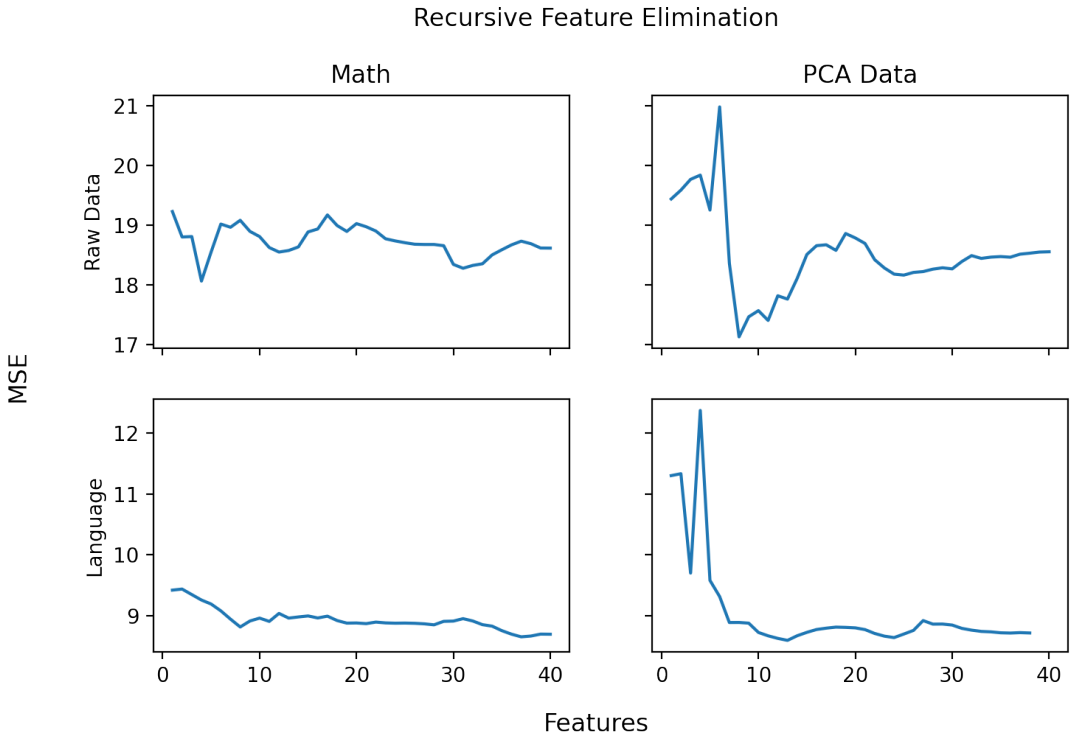


Figure 10: Cross-validation error versus feature count for Recursive Feature Elimination models with different datasets. Error measured using Mean Squared Error.

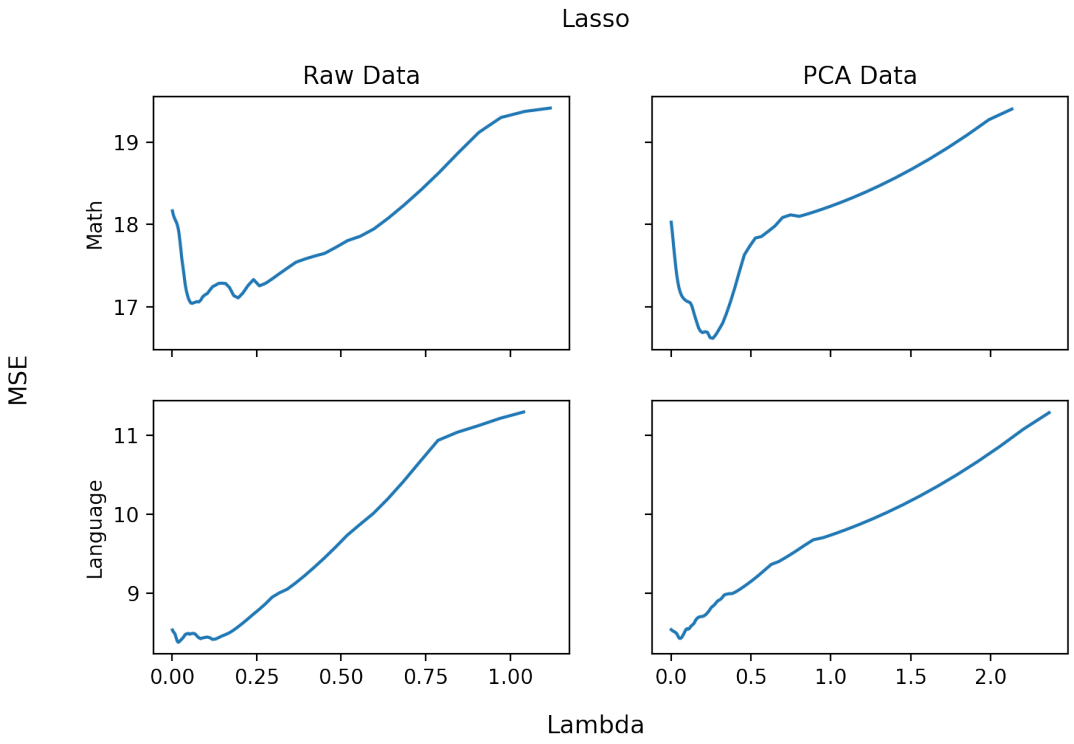


Figure 11: Cross-validation error versus lambda for Lasso regression with different datasets. Error measured using Mean Squared Error.

4 Random Forest Regressions

To investigate the complexity in the regression between the predictors and the final grade, we have utilized Random Forest regression model for both training datasets (80% of the data) of math and language in addition to what we have investigated so far. In short, we have modeled the data using the cross validation to find the best set of parameters with 5-fold and 50 iterations. Then after visualizing the first tree of the model provided, see Figure 12, we pruned the forest to have maximum depth of 3 and visualized the first tree again, see Figure 13 and Figure 14. We have seen a more complex model for the language data compared to math data, see Figure 12. It is noteworthy to mention that we used Random Search instead of Grid Search since the latter is too computationally intensive; we did not perform an exhaustive optimization. Even though utilizing PCA data did not change RMSE a noticeable amount (see 3) it made the trees simpler, and reduced the calculation costs significantly, resulting in more infereable models.

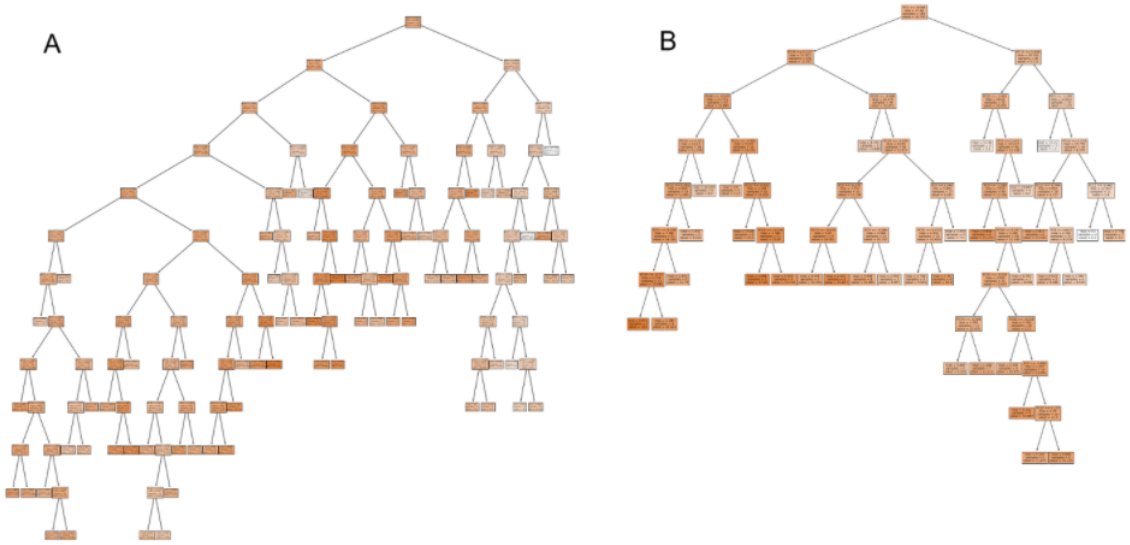


Figure 12: Unpruned first trees of the random forest performed show more complexity in models obtained on language data. A) Language data, B) Math data.

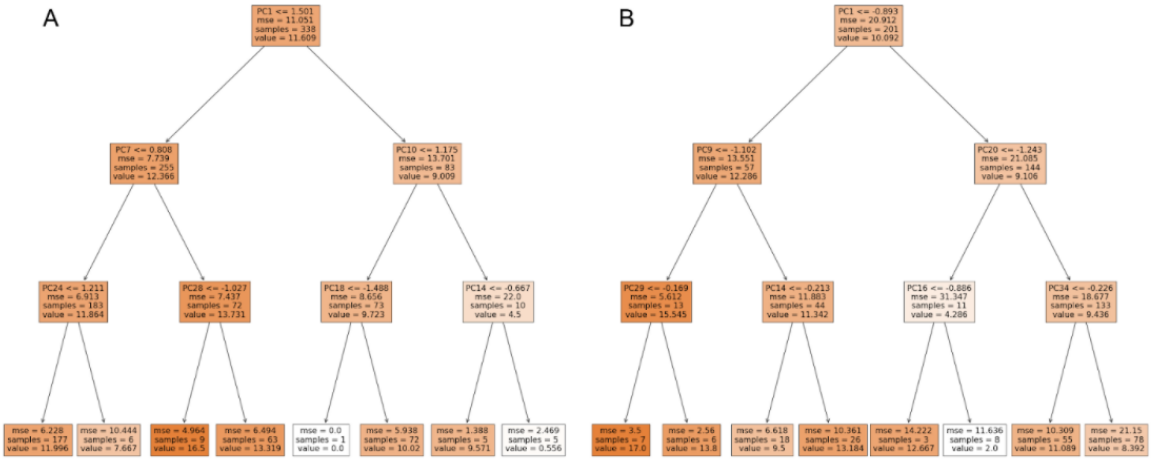


Figure 13: Pruned trees to max depth of 3. A) Language data, B) Math data

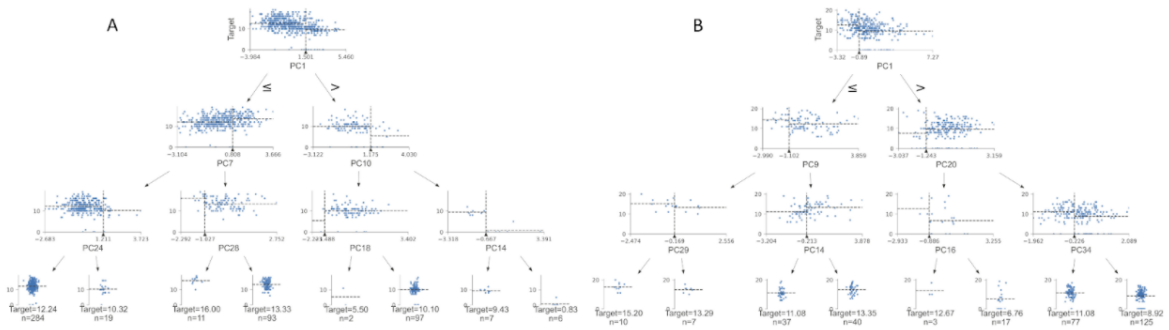


Figure 14: Pruned trees, with depiction of decision points with the data visualized in scatter plots.

5 Results, Comparisons, and Shortcomings

Table 3 displays the results of our models in testing for the Math and Language datasets both with the raw and PCA versions of the data. For the Math data our best model was generated using Backward Selection on the PCA data, with a Root Mean Squared Error (RMSE) of 4.752. Of note is the fact that the next best model for Math grades used Forward Selection on the raw data and only had an difference in error of only 0.001. This indicates that it may not be necessary to go through the effort of computing principal components to achieve meaningful results. Additionally, the range in performance across the different models is only roughly 0.25, indicating that simpler or computationally cheap models may still be reasonable to use with only minimal performance degradation. All of this said, the mean RMSE for all models of the Math dataset hovered around roughly 4.833. Given that the grades are on a 20-point scale, this indicates a notably high error and therefore less-than-optimal predictive capabilities, suggesting that these models may not necessarily be the most optimal for these data, or that the relationship between demographics and grade outcomes is notably weak and therefore impractical to utilize.

		Math		**Language**	
		Raw	**PCA**	**Raw**	**PCA**
Linear Models	**Forward Selection**	4.753	4.791	2.422	2.416
	Backward Selection	4.830	4.752	2.433	2.370
	RFE	4.819	4.821	2.404	2.472
	Lasso	4.767	4.776	2.391	2.377
	Ridge	4.821	4.904	2.387	2.381
Ensemble Models	**Random Forest**	4.966	5.000	2.345	2.518

Table 3: Testing error of our models. Measured using Root Mean Squared Error.

For the Language dataset our best model was Random Forest trained on the raw data, with a RMSE of 2.345. Notably, the next best model for Language data was generated using Backwards Selection on the PCA data, with a RMSE of 2.370. This being the case, we can draw similar conclusions as with the Math dataset with regards to the use of PCA versus the raw data. Also similarly to the models generated for the Math dataset, the different models for the Language dataset all performed very similarly, within a range of 0.173, suggesting that we may use less complicated or computationally intensive models if and still achieve meaningful results. Looking at the results for the Language dataset more generally, the mean RMSE for all of the Language models hovered around roughly 2.410. This is a notably smaller error than we saw with the Math dataset and indicates reasonable predictive power. Further, this indicates that there is a much more notable meaningful relationship between demographics and grade outcomes for Language data.

Conducting PCA and factor analysis on the Math and Language datasets proved useful in helping us scale down from the very large list of predictors. However, the results of the PCA were not too surprising because many of our features were uncorrelated, so that is why it took upwards of

20 components to capture any more than 80% of the data's variability. Using varimax rotation on the data helped increase interpretability as the number of components with eigenvalues greater than 1 decreased from 20 components to around 10 components. Rotating the dataset allowed less components to capture more of the data's variability as compared to conducting PCA on the two datasets. Using the varimax rotated data on the language dataset with the Random Forest (hyper-parameters were found by 5-fold GridSearchCV) produced the best test RMSE for the Language data with a value of 2.33. This is an improvement of roughly 0.12 over the next best model. This being the case, the Varimax rotated data helped increase the predictive power and computational ease by decreasing the dimensionality of our data. With all of this data, it is apparent that the nature of our dataset has introduced complications into the problem making it more difficult to model than initially expected. Due to the large number of binary, categorical, and ordinal data and a notable absence of numerical data, it was deemed very difficult to identify meaningful relationships in our dataset.

6 Conclusion

In this paper we fit models for the prediction of academic outcomes and identified their potential concerns and shortcomings. For the Student Performance Data Set we found that both Linear Regression and Random Forest are seemingly ineffective for meaningful prediction on Math data, while Random Forest is the most effective on Language data. Further, we identified that Principal Component Analysis did not provide any tangible benefit over the use of the raw data. In future work, further analysis could be done into the use of Varimax rotation, as a cursory analysis suggested notable improvement which we did not have time to explore. Additionally, it would be beneficial to explore these same models on a higher quality/larger dataset which is more attuned to this task

References

[1] N. Alhajraf and A. Alasfour, "The impact of demographic and academic characteristics on academic performance," *International Business Research*, vol. 7, 03 2014.

[2] T. Thiele, A. Singleton, D. Pope, and D. Stanistreet, "Predicting students' academic performance based on school and socio-demographic characteristics," *Studies in Higher Education*, vol. 41, no. 8, pp. 1424–1446, 2016.

[3] A. Wada, D. Wagner, F. Qassab, M. Mohamed, M. Hamad, and S. Alshabatti, "The relationship between socio-demographic and lifestyle factors and academic performance," *Iranian journal of public health*, vol. 45, pp. 699–701, 05 2016.

[4] S. Masud, S. Mufarrih, N. Qureshi, S. Khan, and N. Khan, "Academic performance in adolescent students: The role of parenting styles and socio-demographic factors – a cross sectional study from peshawar, pakistan," *Frontiers in Psychology*, vol. 10, 10 2019.

[5] N. Nawa, M. Numasawa, M. Nakagawa, M. Sunaga, T. Fujiwara, Y. Tanaka, and A. Kinoshita, "Associations between demographic factors and the academic trajectories of medical students in japan," *Plos One*, vol. 15, no. 5, 2020.

[6] P. Cortez and A. Silva, "Using data mining to predict secondary school student performance," *EUROSIS*, 01 2008.

[7] H. F. Kaiser, "The varimax criterion for analytic rotation in factor analysis," *Psychometrika*, vol. 23, no. 3, p. 187–200, 1958.