STAT 230A Final Project
Utilizing Linear Models for Predicting Student Grades

Niki Chen
Department of Statistics
University of California, Berkeley
zixuanc2@berkeley.edu

Jinxuan Fan
Department of Statistics
University of California, Berkeley
jinxuan_fan@berkeley.edu

Abstract

In recent years, machine learning algorithms have become increasingly popular in educational research to predict student academic performance. This report presents an investigation into the use of linear models to predict student academic performance, utilizing five different linear models: linear regression, elastic net regression, ridge regression, lasso regression, and principle component analysis (PCA). The dataset used consists of 32 predictor variables and a target variable representing the final grade for 395 students, obtained from the UCI Machine Learning Repository. The data was split into training and testing sets, and descriptive statistics and data visualization were employed to explore the data. The categorical variables were encoded from their original data type of character. The R-squared value was calculated for each model, and other statistical machine learning techniques were also utilized for variable selection and exploring the significance of the predictors, such as F-tests. Finally, the models were compared based on the root mean squared error (RSME). The findings suggest that linear models, particularly elastic net regression, can be effective in predicting student grades, while principle component analysis may not be the best fit for this specific dataset. The findings of this study have implications for educators and administrators looking to identify predictors of student success and inform targeted interventions to improve academic outcomes.

1 Introduction

The assessment and evaluation of student academic performance is a critical aspect of the education system, providing valuable information for teachers, students, and educational institutions. The ability to accurately predict student grades can help teachers identify areas for improvement, make informed decisions about students' academic trajectory, and inform targeted interventions to improve academic outcomes. With the increasing amount of data generated in the education sector, there is a growing need for effective methods to analyze and make predictions based on this data. Linear models, offer a powerful tool for analyzing such data and can be particularly useful for predicting student grades.

The application of linear models in education research has become increasingly popular in recent years, providing valuable insights into the factors that affect student performance and informing interventions to improve academic outcomes. However, the effectiveness of linear models in predicting student grades can be impacted by various factors, including the quality and completeness of the data, the choice of predictor variables, and the type of linear model used. Therefore, it is important to explore the use of different linear models and compare their performance in predicting student grades.

In this paper, we aim to explore the use of linear models for predicting student grades and compare the performance of different linear models, including linear regression, elastic net regression, ridge regression, lasso regression and principle component analysis (PCA). Specifically, we will examine a dataset of 395 students and evaluate the effectiveness of different linear models for predicting student grades. Our study will provide valuable insights into the factors that impact student performance and the challenges and benefits associated with using linear models in education research. Our findings can inform educators and administrators looking to identify predictors of student success and inform targeted interventions to improve academic outcomes.

2 Final Regression Model: Elastic Net Regression Model

The final regression model chosen for predicting student grades is the elastic net regression model. This model is a combination of both ridge and Lasso regression models, and it helps overcome some of their limitations. The elastic net regression model works by adding a penalty term to the objective function of the linear regression model. The penalty term includes two parameters, alpha and lambda, which control the amount of shrinkage and sparsity in the model, respectively. The alpha parameter determines the balance between the ridge and Lasso penalties, and it ranges from 0 (ridge) to 1 (Lasso). The lambda parameter controls the strength of the penalty.

To build the elastic net model, the data set was split into training and testing sets using an 80/20 split. All categorical variables were encoded from their original data type of character, and the remaining 32 predictor variables were used to predict the final math grades. The model was then fitted using the glmnet function in R, with alpha ranging from 0 to 1 in increments of 0.1. Cross-validation was used to determine the optimal value of lambda that minimized the mean squared error (MSE). The R-squared and root mean squared error (RMSE) were used to evaluate the performance of the model.

The final elastic net model included the following predictors: studytime, failures, activities, romantic, famrel, Walc, health, absences, G1, and G2. These variables were identified as having the most significant impact on predicting student grades. The coefficients of the model were obtained using the "glmnet" package, and the selected predictors were found to have both positive and negative coefficients, indicating their contribution to either increasing or decreasing the final grade.

Based on the results (Figure 1 and Figure 2), the optimal alpha was found to be 0.9, which maximized the R-squared and minimized the RMSE.

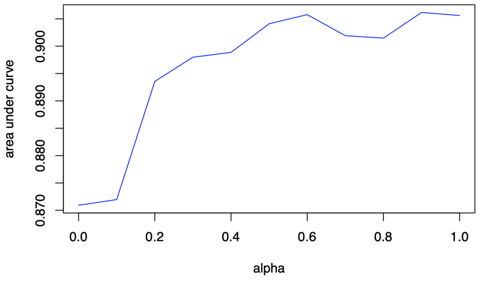

Figure 1: alpha vs Rsquared

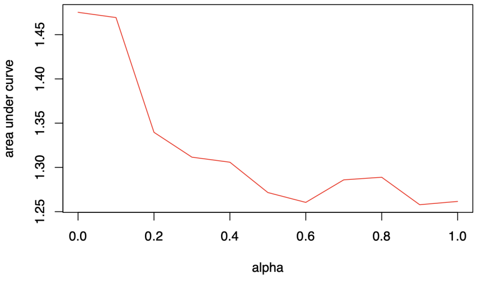

Figure 2: alpha vs RMSE

We then rebuilt the model using this optimal alpha value and obtained the final model (Figure 3).

The final model showed an R-squared value of 0.9061616 and an RMSE of 1.257866, indicating that it had a good fit to the data and was able to accurately predict student grades. The model also identified the most important predictors of student grades, including study time, number of past class failures, participation in extracurricular activities, romantic relationships, family relationships, weekend alcohol consumption, overall health, number of absences, and previous period grades (G1 and G2).

Overall, the elastic net regression model provided valuable insights into the factors that contribute to student success and can inform targeted interventions to improve academic outcomes. It is a powerful tool for educators and administrators looking to identify predictors of student success and make informed decisions to enhance academic outcomes.

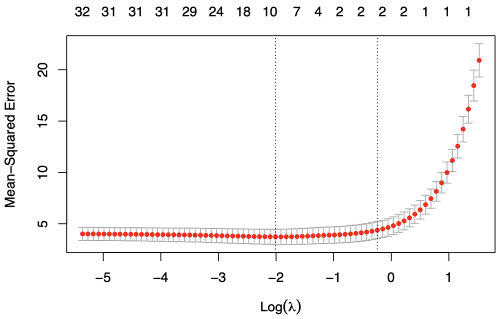

Figure 3: Final Elastic Net model with alpha = 0.9

3 Conclusions from Final Model and Limitations of the Analysis

In this study, we aimed to explore the use of linear models, including linear regression, lasso regression, ridge regression, and elastic net regression, for predicting student grades based on a dataset of 395 students. Our results suggest that the elastic net regression model with an alpha value of 0.9 was the most effective in predicting student grades, with an R-squared value of 0.9061616 and an RMSE of 1.257866.

The predictors selected by the elastic net model include study time, the number of past class failures, participation in extracurricular activities, having a romantic relationship, family relationship, weekend alcohol consumption, overall health status, absences, and the grades of the first two periods. These variables indicate that factors that have a significant impact on academic performance. Our results suggest that increasing students' study time, providing support for students with past failures, encouraging participation in extracurricular activities, and promoting healthy behaviors outside of school can help improve academic performance.

However, our study has several limitations that should be taken into account when interpreting the results. Firstly, the dataset used in this study is from a single school, and the results may not be generalizable to other schools or populations. Secondly, the data is limited to a single subject, mathematics, and the predictors may differ for other subjects. Additionally, the study is limited by the available data and the choice of linear models used. Other machine learning techniques, such as tree-based models or neural networks, may provide better accuracy in predicting student grades. Lastly, there may be unmeasured variables, such as home environment, that are not included in the dataset but may impact academic outcomes.

4 Main Findings and Conclusion in the research

In conclusion, our investigation into the use of linear models for predicting student academic performance using a dataset of 395 students has provided valuable insights into the factors that impact student success. Our findings suggest that linear models, particularly elastic net regression, can be effective in predicting student grades. The predictors identified by the elastic net regression model include study time, past class failures, extracurricular activities, having a romantic relationship, family relationship, weekend alcohol consumption, overall health status, absences, and grades of the first two periods. These findings indicate that factors beyond academic performance, such as lifestyle and behavior, can have a significant impact on student success.

The findings of this study have important implications for educators and administrators, as they can be used to develop targeted interventions to improve academic outcomes. Encouraging study habits, supporting students with past failures, promoting extracurricular activities, and promoting healthy behaviors outside of school can help improve academic performance. Moreover, our investigation has provided a foundation for future research into the use of machine learning techniques for predicting student academic performance.

5 Alternative Models and Diagnostic Analysis

In this section, in addition to the elastic net regression model discussed earlier, we also constructed other alternative models to predict student grades using all available predictors.

5.1 Linear Regression

The data was split into 80% training data and 20% testing data, and the linear regression model was constructed using the training data. The model was then evaluated using the testing data, and various diagnostics were employed to explore the model's accuracy.

The linear regression model produced a reasonable R-squared value of 0.839 and an RMSE of 1.583396. The maximum prediction error among the 77 testing data points was 4.15 out of a 20 point scale, and the mean prediction error was 1.24. These values suggest that the linear regression model is acceptable for predicting student grades.

We also performed additional diagnostics to explore the significance of the predictors in the linear regression model. F-tests were used to evaluate the significance of each predictor, and we found that having a high quality of family relationships, having more absences, achieving high grades in G1 and G2, having a larger age, and participating in extra activities were significant predictors of student grades.

Overall, our investigation into the linear regression model for predicting student grades provides valuable insights into the factors that impact academic performance. While the elastic net regression model was found to be the most effective in predicting student grades, the linear regression model can also be a useful tool for educators and administrators looking to identify predictors of student success and inform targeted interventions to improve academic outcomes.

5.2 Ridge Regression

In addition to the linear regression model, we also explored the use of ridge regression to predict student grades based on the same dataset. Ridge regression is a technique that can be used to address the issue of multicollinearity, which can occur when predictor variables are highly correlated with one another.

Firstly, the data was split into training and testing sets with 80% training data and 20% testing data. Then, we used the glmnet package to construct the ridge regression model with alpha = 0, and plotted the coefficients against the L1 Norm. Figure 4 shows that as L1 Norm increases, the coefficients shrink towards zero, but none of them reach exactly zero, meaning all the variables are included in the model.

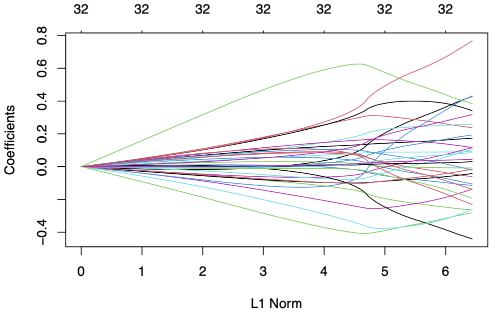

Figure 4: Coefficients v.s. L1 Norm (Ridge)

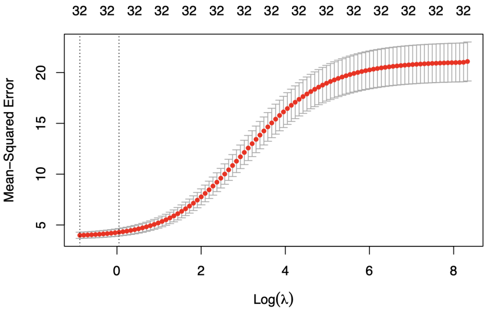

Figure 5: MSE v.s. Log(lambda) (Ridge)

Next, we used k-fold cross-validation to identify the lambda value that produced the lowest test mean squared error (MSE). The plot of cross-validation errors for each lambda value (Figure 5) shows that the test MSE decreases as log(lambda) increases, then starts to increase after a certain point, indicating overfitting. The lambda.min value with the lowest cross-validation error was chosen as the optimal lambda value, and the coefficients were calculated for that lambda value.

To evaluate the performance of the model, we used the best model of lambda to make predictions on the testing data. The R-squared value was 0.8709272, higher than that for linear regression, which was 0.839. This suggests that ridge regression is a better model for predicting student grades than linear regression. The RMSE was 1.475236, also lower than that for linear regression.

Overall, our results suggest that ridge regression is a useful tool for predicting student grades and can address issues of multicollinearity. However, there are limitations to this analysis, including the fact that the dataset is limited to a single subject, and the predictors may differ for other subjects. Additionally, other machine learning techniques, such as tree-based models or neural networks, may provide better accuracy in predicting student grades.

5.3 Lasso Regression

In addition to linear regression and ridge regression, we also applied Lasso regression to predict student grades based on the dataset. Lasso regression is a linear model that incorporates L1 regularization to perform variable selection, where it can reduce the influence of irrelevant predictors by shrinking their coefficients to zero.

We first split the dataset into training and testing sets, then performed Lasso regression using the glmnet package with an alpha value of 1. We then used 10-fold cross-validation to identify the optimal lambda value, which produces the lowest test mean squared error. The two plots show the coefficients versus the L1 Norm values (Figure 6) and the cross-validation mean squared error versus log(lambda) values (Figure 7), respectively. The coefficients plot helps to visualize the effect of regularization on the model by showing the size of the coefficients for each predictor as L1 Norm varies. The second plot shows the optimal lambda value selected by cross-validation, which indicates the amount of regularization required to achieve the best model fit.

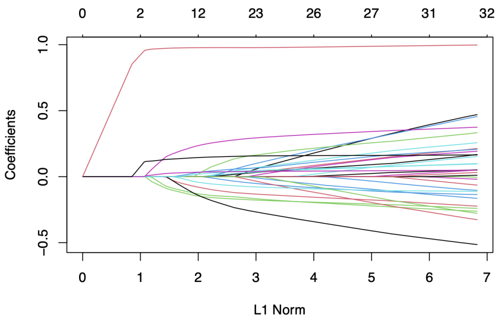

Figure 6: Coefficients v.s. L1 Norm (Lasso)

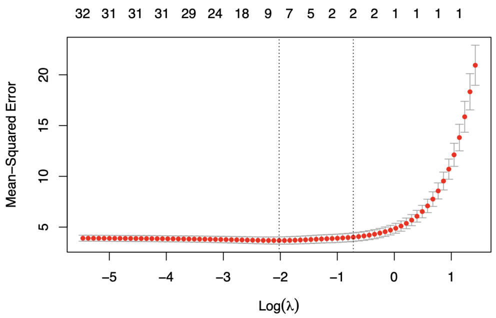

Figure 7: MSE v.s. Log(lambda) (Lasso)

The coefficients of the Lasso regression model indicate that factors such as study time, the number of past class failures, participation in extracurricular activities, having a romantic relationship, family relationship, weekend alcohol consumption, overall health status, absences, and the grades of the first two periods have a significant impact on academic performance.

The R-squared value obtained from the Lasso regression model is 0.9056086, which is higher than the R-squared values obtained from both linear and ridge regression models. The RMSE value for the Lasso regression model is 1.261567, which is also lower than that of the linear regression model.

Overall, the Lasso regression model outperformed both linear and ridge regression models in predicting student grades, and the variable selection capability of Lasso regression can help identify the most important predictors of academic performance. However, this model may have limitations in cases where all predictors are relevant or when the sample size is too small.

5.4 Principle Component Analysis (PCA)

In addition to linear regression, ridge regression, and Lasso regression, we also employed principle component analysis (PCA) to predict student grades based on the dataset of 395 students. The goal of PCA is to identify a smaller number of underlying variables, called principal components, that explain the majority of the variation in the dataset.

After splitting the data into training and testing sets, we utilized the "pls" package in R to perform PCA. We used cross-validation to identify the optimal number of principal components to include in the model.

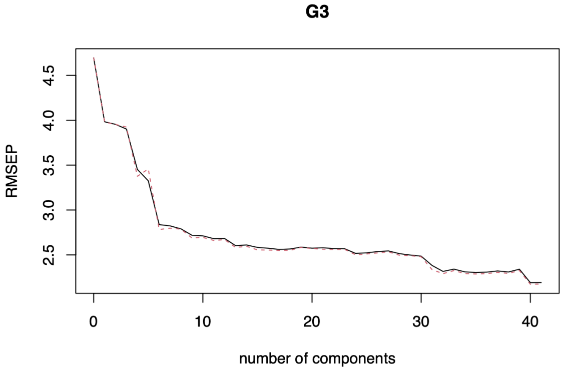

Figure 8: validationplot of PCA model

The validation plot (Figure 8) provides a visualization of the cross-validation process, showing the RMSE on the y-axis and the number of components on the x-axis. The plot displays a decreasing trend in RMSE as the number of components increases, indicating that including more components leads to a better fit of the model. However, the plot also shows that the improvement in RMSE decreases as the number of components increases, suggesting that including too many components may result in overfitting the model.

We found that using all 41 principal components explained 100% of the variation in the data, while using only 23 principal components explained 81.35% of the variation. However, when we compared the performance of the model using 23 components versus all 41 components, we found that the model with 41 components performed better, with an R-squared of 0.851307 and an RMSE of 1.583396, compared to an R-squared of 0.7394109 and an RMSE of 2.09615 for the model with 23 components.

These results suggest that the problem may not be well-suited for PCA, as there is not significant multicollinearity between the features. Therefore, the use of PCA may not be the most appropriate method for predicting student grades in this particular dataset.

Overall, while PCA did not perform as well as other linear regression models for predicting student grades in this dataset, the use of PCA and cross-validation techniques can provide valuable insights into the underlying structure of the data and the optimal number of components to include in the model.

6 Comparison of Different Linear Models

In this study, we compared the performance of five different linear models for predicting student grades: linear regression, elastic net regression, ridge regression, lasso regression, and principle component analysis (PCA). The results of the comparison are summarized in the following Table 1.

Based on the table, it can be seen that elastic net regression achieved the best performance among all the models, with the highest R-squared value of 0.9061616 and the lowest RMSE of 1.257866. This indicates that elastic net regression was able to capture the relationships between the input features and the target variable more accurately than the other models.

Model	R-squared	RMSE
Linear Regression	0.839	1.583396
Elastic Net Regression	0.9061616	1.257866
Ridge Regression	0.8709272	1.475236
Lasso Regression	0.9056086	1.261567
PCA	0.851307	1.583396

Table 1: Comparison of different linear models for predicting student grades

The results also show that both ridge regression and lasso regression performed better than linear regression and PCA in terms of R-squared and RMSE. This indicates that regularization techniques are effective in reducing the overfitting problem and improving the model performance. On the other hand, the PCA model is not appropriate for this problem because the variables do not exhibit strong multicollinearity. As a result, it is not able to explain a sufficient amount of variance in the response variable.

In conclusion, the Elastic Net Regression model is the best model for predicting student grades in this study. The model can be used to identify important predictor variables and to make accurate predictions of student grades. The results of this study can be useful for educators and policy makers in developing effective strategies to improve student performance in secondary education.

References

[1] Cortez, P., Silva, A. (2008). Using data mining to predict secondary school student performance. In A. Brito J. Teixeira Eds., Proceedings of 5th Future Business Technology Conference (FUBUTEC 2008), pp. 1-5. Porto, Portugal: EUROSIS. ISBN 978-9077381-39-7. Available at: `http://www3.dsi.uminho.pt/pcortez/student.pdf`

[2] UCI Machine Learning Repository. (n.d.). Student Performance Data Set. Retrieved from http://archive.ics.uci.edu/ml/datasets/Student+Performance

[3] Kuhn, M., Johnson, K. (2013). Applied predictive modeling. New York: Springer.

[4] Friedman, J., Hastie, T., Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software, 33(1), 1-22.

[5] Geladi, P., Kowalski, B. R. (1986). Partial least-squares regression: A tutorial. Analytica Chimica Acta, 185, 1-17.

[6] Hastie, T., Tibshirani, R., Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). New York: Springer.

Appendix

Complete Table of Attributes for the Student Performance Data Set

The table below lists all attributes of the Student Performance Data Set used in this study, including the target attribute G3.

Attribute	Description	Type
school	Student's school	Nominal Binary
sex	Student's sex	Nominal Binary
age	Student's age (numeric: from 15 to 22)	Numeric
address	Student's home address type	Nominal Binary
famsize	Family size	Nominal Binary
Pstatus	Parent's cohabitation status	Nominal Binary
Medu	Mother's education	Ordinal Numeric
Fedu	Father's education	Ordinal Numeric
Mjob	Mother's job	Nominal
Fjob	Father's job	Nominal
reason	Reason to choose this school	Nominal
guardian	Student's guardian	Nominal
traveltime	Home to school travel time	Ordinal Numeric
studytime	Weekly study time	Ordinal Numeric
failures	Number of past class failures	Ordinal Numeric
schoolsup	Extra educational support	Nominal Binary
famsup	Family educational support	Nominal Binary
paid	Extra paid classes within the course subject (Math or Portuguese)	Nominal Binary
activities	Extra-curricular activities	Nominal Binary
nursery	Attended nursery school	Nominal Binary
higher	Wants to take higher education	Nominal Binary
internet	Internet access at home	Nominal Binary
romantic	With a romantic relationship	Nominal Binary
famrel	Quality of family relationships	Ordinal Numeric
freetime	Free time after school	Ordinal Numeric
goout	Going out with friends	Ordinal Numeric
Dalc	Workday alcohol consumption	Ordinal Numeric
Walc	Weekend alcohol consumption	Ordinal Numeric
health	Current health status	Ordinal Numeric
absences	Number of school absences	Numeric
G1	First period grade	Numeric
G2	Second period grade	Numeric
G3	Final grade (output target)	Numeric

Note: Nominal binary variables have two possible values, while nominal variables have more than two possible values without any meaningful ordering. Ordinal variables have more than two possible values with a meaningful ordering. Numeric variables are continuous or discrete numeric values.

R code

```
library(tidyverse)
library(grid)
library(knitr)
library(dplyr)
library(ggplot2)
library(gridExtra)
library(readr)

#Import data:

math_grade2 <- read.csv("student-mat.csv")
```

```
head(math_grade2)
dim(math_grade2)

#The data set has 395 observations and 33 columns.

#Missing values:

nrow(math_grade2[!complete.cases(math_grade2), ])

#There is no missing value.

#We are going to predict the students final math grades based
   using the remaining 32 predictors.

colnames(math_grade2)

#Check the data types of all columns

sapply(math_grade2, class)

#The variables having "character" as their data types are
   school, sex, address, famsize, Pstatus, Mjob, Fjob, reason,
    guardian, schoolsup, famsup, paid, activities, nursery,
   higher, internet, romantic

#Encode all categorical variables from their original data type
    of character.

math_grade2school=factor(mathgrade2school = factor(math_
   grade2school)
math_grade2sex=factor(mathgrade2sex = factor(math_grade2sex)
math_grade2address=factor(mathgrade2address = factor(math_
   grade2address)
math_grade2famsize=factor(mathgrade2famsize = factor(math_
   grade2famsize)
math_grade2Pstatus=factor(mathgrade2Pstatus = factor(math_
   grade2Pstatus)
math_grade2Mjob=factor(mathgrade2Mjob = factor(math_grade2Mjob)
math_grade2Fjob=factor(mathgrade2Fjob = factor(math_grade2Fjob)
math_grade2reason=factor(mathgrade2reason = factor(math_
   grade2reason)
math_grade2guardian=factor(mathgrade2guardian = factor(math_
   grade2guardian)
math_grade2schoolsup=factor(mathgrade2schoolsup = factor(math_
   grade2schoolsup)
math_grade2famsup=factor(mathgrade2famsup = factor(math_
   grade2famsup)
math_grade2paid=factor(mathgrade2paid = factor(math_grade2paid)
math_grade2activities=factor(mathgrade2activities = factor(math
   _grade2activities)
math_grade2nursery=factor(mathgrade2nursery = factor(math_
   grade2nursery)
math_grade2higher=factor(mathgrade2higher = factor(math_
   grade2higher)
math_grade2internet=factor(mathgrade2internet = factor(math_
   grade2internet)
math_grade2romantic=factor(mathgrade2romantic = factor(math_
   grade2romantic)
```

```
#Check data types

unique(sapply(math_grade2, class))

#Data plot for the final grade to be predicted:
#G1 - first period grade (numeric: from 0 to 20)
#G2 - second period grade (numeric: from 0 to 20)
#G3 - final grade (numeric: from 0 to 20, output target)

hist(math_grade2$G1)
hist(math_grade2$G2)
hist(math_grade2$G3)

#The three distributions are normal as expected. However, there
     are nearly 40 students gained 0. This might because they
   did not attend the exam.

#Plot gender (Female students are a little more than male
   students):

ggplot(data = math_grade2, aes(x = sex)) + geom_bar()

#Plot address (most students live in urban):

ggplot(data = math_grade2, aes(x = address)) + geom_bar()

#Plot mother job:

ggplot(math_grade2, aes(x=Mjob)) + geom_bar()

#Plot father job:

ggplot(math_grade2, aes(x=Fjob)) + geom_bar()


# Elastic net

set.seed(1)
rsq_list = c()
rmse_list = c()
# create a list to record alpha
al = c()
for (d in 0:10){
  # setting alpha to be 0.1, 0.2 ... to 1.0
  al_pha = d/10
  al = append(al,al_pha)
  r_model = glmnet(x = data.matrix(grade_trn[, 1:32]), y =
     grade_trn[,33], alpha = al_pha)
  cv.fit = cv.glmnet(x = data.matrix(math_grade2[, 1:32]), y =
     math_grade2[,33],
                     nfolds = 10, alpha = al_pha)

  best_lambda <- cv.fit$lambda.min
  y_predicted <- predict(r_model, s = best_lambda, newx = data.
     matrix(grade_tst[, 1:32]))
  tss = sum((grade_tst[,33] - mean(grade_tst[,33]))^2)
  sse = sum((y_predicted - grade_tst[,33])^2)
  #find R-Squared
  rsq <- 1 - sse/tss
  rsq_list = append(rsq_list, rsq)
  #find RMSE
  rmse = sqrt(mean((grade_tst$G3 - y_predicted)^2))
  rmse_list = append(rmse_list, rmse)
}

#plot alpha vs Rsquared for the coresponding alpha
plot(al, rsq_list,
     xlab = "alpha",
     ylab = "area␣under␣curve",
     type="l",
     col="blue",
)
#plot alpha vs rmse for the coresponding alpha
plot(al, rmse_list,
     xlab = "alpha",
     ylab = "area␣under␣curve",
     type="l",
     col="red",
)

which.max(rsq_list)
which.min(rmse_list)
# alpha = 0.9 maximize the R-squared, and minimized rmse
rsq_list[which.max(rsq_list)]
rmse_list[which.min(rmse_list)]

# rebuild model when alpha = 0.9
r_model = glmnet(x = data.matrix(grade_trn[, 1:32]), y = grade_
   trn[,33], alpha = 0.9)
cv.fit = cv.glmnet(x = data.matrix(math_grade2[, 1:32]), y =
   math_grade2[,33],
                   nfolds = 10, alpha = 0.9)
plot(cv.fit)
best_lambda <- cv.fit$lambda.min
best_model = glmnet(x = data.matrix(grade_trn[, 1:32]), y =
   grade_trn[,33], alpha = 0.9, lambda = best_lambda)
coef(best_model)

#The R-squared is higher than the one(0.8709272) we got in
   ridge regression.
#Up to now, the elastic net when alpha = 0.9 performs the best,
    having R-squared 0.9061616 and rmse 1.257866.
#The coefficients selected by the this model (elastic net when
   alpha = 0.9) are studytime, failures, activities, romantic,
    famrel, Walc, health, absences, G1, G2.

#Split data to training and tesing (set 80% training data and
   20% testing data), then we use the all predictors to
   predict G3 by constructing a linear regression:

library(caret)
set.seed(1)
trn_idx = createDataPartition(math_grade2$G3, p = 0.80, list =
   FALSE)
grade_trn = math_grade2[trn_idx, ]
grade_tst = math_grade2[-trn_idx,]
model1 = lm(G3~.,data = grade_trn)
```

```
summary(model1)

pred_test = predict(model1, grade_tst)
data_prediction = (data.frame((pred_test), (grade_tst$G3),(abs(
   pred_test - grade_tst$G3))))
colnames(data_prediction) <- c("Predicted␣G3","Real␣G3","
   Difference")
head(data_prediction,10)
max(data_prediction$Difference)
mean(data_prediction$Difference)
dim(data_prediction)

# calculate RMSE
sqrt(mean((grade_tst$G3 - pred_test)^2))


#We can see that the maximum prediction error among the 77
   testing data points is 4.15 of a 20 points scale; the mean
   of the prediction error is 1.24 which is quite acceptable.

#The R-squared is 0.839, indicating the linear regression model
    for predicting students final math grade is reasonable.
   RMSE is 1.583396

#Later we will also perform Lasso regression and principle
   components analysis to do variable selection. We will also
   do testings like F-tests to explore the significance of the
    predictors. Finally, we will compare models based on the
   RSME.

#According to the p-value, having a high quality of family
   relationships, having more absences (a little strange),
   getting high grades in G1 and G2 will generate positive
   impact to final grades, while having larger age and having
   extra activities will negatively affect the final grade.

# Ridge regression:

set.seed(123)
library(glmnet)
r_model = glmnet(x = data.matrix(grade_trn[, 1:32]), y = grade_
   trn[,33], alpha = 0)
plot(r_model)

# use k-fold cross validation to identify the lambda value that
    produces the lowest test mean squared error (MSE)
cv.fit = cv.glmnet(x = data.matrix(math_grade2[, 1:32]), y =
   math_grade2[,33],
                  nfolds = 10, alpha = 0)

plot(cv.fit)

# the coefficients:
coef(cv.fit,s = "lambda.min")
# prediction
# pred.cv = predict(cv.fit, data.matrix(math_grade2[, 1:32]),
#                   type = "response", s = "lambda.min")

best_lambda <- cv.fit$lambda.min
```

```
# use the best model of lambda to make predictions on testing
   data
y_predicted <- predict(r_model, s = best_lambda, newx = data.
   matrix(grade_tst[, 1:32]))
tss = sum((grade_tst[,33] - mean(grade_tst[,33]))^2)
sse = sum((y_predicted - grade_tst[,33])^2)
#find R-Squared
rsq <- 1 - sse/tss
rsq
#find RMSE
sqrt(mean((grade_tst$G3 - y_predicted)^2))

#The R-squared is higher than the one we got in linear
   regression, which is 0.839.
#Since this is a ridge regression, none of the coefficients are
    opted out, we use all of them to predict G3.
#The RMSE is 1.475236, also is lower than that for linear
   regression.

# Lasso
set.seed(1)
library(glmnet)
r_model = glmnet(x = data.matrix(grade_trn[, 1:32]), y = grade_
   trn[,33], alpha = 1)
plot(r_model)

# use k-fold cross validation to identify the lambda value that
    produces the lowest test m
cv.fit = cv.glmnet(x = data.matrix(math_grade2[, 1:32]), y =
   math_grade2[,33],
                   nfolds = 10, alpha = 1)
plot(cv.fit)

# the coefficients:
coef(cv.fit,s = "lambda.min")

# prediction
# pred.cv = predict(cv.fit, data.matrix(math_grade2[, 1:32]),
#                   type = "response", s = "lambda.min")
best_lambda <- cv.fit$lambda.min
# use the best model of lambda to make predictions on testing
   data
y_predicted <- predict(r_model, s = best_lambda, newx = data.
   matrix(grade_tst[, 1:32]))
tss = sum((grade_tst[,33] - mean(grade_tst[,33]))^2)
sse = sum((y_predicted - grade_tst[,33])^2)
#find R-Squared
rsq <- 1 - sse/tss
rsq

#find RMSE
sqrt(mean((grade_tst$G3 - y_predicted)^2))

# principle component

set.seed(1)
library(pls)
```

```
pcr_model = pcr(G3~.,data = grade_trn, scale = TRUE, validation
    = "CV")
summary(pcr_model)
validationplot(pcr_model)

# make predictions: (if only use 23 comps, which explains
   81.35% of the variation)
pcr_pred = predict(pcr_model, grade_tst, ncomp = 23)
#find R-Squared
tss = sum((grade_tst[,33] - mean(grade_tst[,33]))^2)
sse = sum((pcr_pred - grade_tst[,33])^2)
rsq <- 1 - sse/tss
rsq
rmse = sqrt(mean((grade_tst$G3 - pcr_pred)^2))
rmse

# make predictions: (if use all(41) comps, which explains 100%
   of the variation)
pcr_pred = predict(pcr_model, grade_tst, ncomp = 41)
#find R-Squared
tss = sum((grade_tst[,33] - mean(grade_tst[,33]))^2)
sse = sum((pcr_pred - grade_tst[,33])^2)
rsq <- 1 - sse/tss
rsq
rmse = sqrt(mean((grade_tst$G3 - pcr_pred)^2))
rmse

#If we use 23 principal components, the R-squared is quite low,
    which is 0.7394109, the rmse is 2.09615.
#If we use all(41) principal components, the R-squared is
   0.851307 and the rmse is 1.583396.
#This indicate this problem is not proper to use PCA techniques
   , which means that the multi-colinearity between features
   are not obvious.
```