

## Final Project Code

```
1 # -----
2 # Vincent Johnson
3 # Joseph Kang
4 # Catherine Sung
5 # -----
6
7 # -----
8 # Set working directory
9 # -----
10
11 setwd("C:/Users/Public/Stats 385/Final Project")
12
13 # -----
14 # Load necessary libraries
15 # -----
16 library(car)
17 library(caret)
18 library(cluster)
19 library(corrplot)
20 library(dplyr)
21 library(e1071)
22 library(factoextra)
23 library(gbm)
24 library(ggplot2)
25 library(glmnet)
26 library(gridExtra)
27 library(ISLR2)
28 library(kableExtra)
29 library(knitr)
30 library(leaps)
31 library(pls)
32 library(randomForest)
33 library(tibble)
34 library(tidyverse)
35
36
37 # -----
38 # Read and Merge Data
39 # -----
40 math <- read.csv("math.csv")
41 por <- read.csv("portuguese.csv")
42
43 join_cols <- c("school", "sex", "age", "address", "famsize", "Pstatus",
44               "Medu", "Fedu", "Mjob", "Fjob", "reason", "nursery", "internet")
45 merged_data <- merge(math, por, by = join_cols, suffixes = c(".math", ".por"))
46
47 # -----
48 # Clean and Format Data
49 # -----
50 clean_data <- merged_data %>%
51   rename(mathp1 = G1.math,
52          mathp2 = G2.math,
53          mathfinal = G3.math,
54          porp1 = G1.por,
55          porp2 = G2.por,
56          porfinal = G3.por) %>%
```

```

57   select(starts_with("G"), everything()) %>%
58   mutate_if(is.character, as.factor)
59
60 clean_data <- clean_data %>%
61   select(-ends_with(".math"))
62 names(clean_data) <- gsub("\\.por$", "", names(clean_data))
63 colnames(clean_data)
64 #we have exactly 36 columns.
65
66 # -----
67 # Missing Data Check
68 # -----
69 cat("\nMissing data summary:\n")
70 print(sapply(clean_data, function(x) sum(is.na(x))))
71 ##No Missing data!
72
73 # -----
74 # Exploratory Data Analysis (EDA)
75 # -----
76 ##Histograms
77
78 hist_math <- ggplot(clean_data, aes(x = mathfinal)) +
79   geom_histogram(binwidth = 1, fill = "skyblue", color = "black") +
80   theme_minimal() +
81   labs(title = "Final Math Grades", x = "Grade", y = "Count")
82
83 hist_por <- ggplot(clean_data, aes(x = porfinal)) +
84   geom_histogram(binwidth = 1, fill = "salmon", color = "black") +
85   theme_minimal() +
86   labs(title = "Final Portuguese Grades", x = "Grade", y = "Count")
87
88 grid.arrange(hist_math, hist_por, ncol = 2)
89
90 ##Boxplots
91 box_math <- ggplot(clean_data, aes(x = sex, y = mathfinal)) +
92   geom_boxplot(fill = "lightblue") +
93   theme_minimal() +
94   labs(title = "Math Grades by Sex", x = "Sex", y = "Math Final Grade")
95
96 box_por <- ggplot(clean_data, aes(x = sex, y = porfinal)) +
97   geom_boxplot(fill = "lightcoral") +
98   theme_minimal() +
99   labs(title = "Portuguese Grades by Sex", x = "Sex", y = "Portuguese Final Grade")
100
101 grid.arrange(box_math, box_por, ncol = 2)
102
103
104 ##QQ Plots
105 par(mfrow = c(1, 2))
106 qqnorm(clean_data$mathfinal, main = "QQ Plot - Math")
107 qqline(clean_data$mathfinal)
108 qqnorm(clean_data$porfinal, main = "QQ Plot - Portuguese")
109 qqline(clean_data$porfinal)
110 par(mfrow = c(1, 1)) # Reset
111
112 ##Coorelation Heat Map
113
114 # Only numeric variables
115 numeric_data <- clean_data %>% select_if(is.numeric)

```

```

116 cor_matrix <- cor(numeric_data, use = "complete.obs")
117
118 corrrplot(cor_matrix, method = "color", type = "upper",
119           title = "Correlation Matrix (Math & Portuguese)", mar = c(0,0,1,0))
120
121 summary_table <- tibble(
122   Statistic = c("Min", "1st Quartile", "Median", "Mean", "3rd Quartile", "Max", "SD"),
123   Math = c(
124     min(clean_data$mathfinal, na.rm = TRUE),
125     quantile(clean_data$mathfinal, 0.25, na.rm = TRUE),
126     median(clean_data$mathfinal, na.rm = TRUE),
127     mean(clean_data$mathfinal, na.rm = TRUE),
128     quantile(clean_data$mathfinal, 0.75, na.rm = TRUE),
129     max(clean_data$mathfinal, na.rm = TRUE),
130     sd(clean_data$mathfinal, na.rm = TRUE)
131   ),
132   Portuguese = c(
133     min(clean_data$porfinal, na.rm = TRUE),
134     quantile(clean_data$porfinal, 0.25, na.rm = TRUE),
135     median(clean_data$porfinal, na.rm = TRUE),
136     mean(clean_data$porfinal, na.rm = TRUE),
137     quantile(clean_data$porfinal, 0.75, na.rm = TRUE),
138     max(clean_data$porfinal, na.rm = TRUE),
139     sd(clean_data$porfinal, na.rm = TRUE)
140   )
141 )
142
143 print(summary_table)
144
145 kable(summary_table, format = "latex", booktabs = TRUE,
146       caption = "Model Performance Summary for Math and Portuguese")
147
148 # Statistic      Math Portuguese
149 # 1 Min          0           0
150 # 2 1st Quartile 8           11
151 # 3 Median       11          13
152 # 4 Mean         10.4        12.5
153 # 5 3rd Quartile 14          14
154 # 6 Max          20          19
155 # 7 SD           4.69        2.95
156
157
158 # -----
159 # Subset and split Data
160 # -----
161 math_data <- clean_data %>% mutate_if(is.character, as.factor) %>%
162   select(-contains("por"))
163
164 por_data <- clean_data %>% mutate_if(is.character, as.factor) %>%
165   select(-contains("math"))
166
167 set.seed(38520251)
168 train_index <- createDataPartition(math_data$mathfinal, p = 0.7, list = FALSE)
169 train <- math_data[train_index, ]
170 test <- math_data[-train_index, ]
171
172 set.seed(38520251)
173 train_indepx <- createDataPartition(por_data$porfinal, p = 0.7, list = FALSE)
174 trainp <- por_data[train_indepx, ]

```

```

175 testp <- por_data[-train_indepx, ]
176
177
178
179 # -----
180 # Linear Models + Assumption Checks
181 # -----
182
183 # ----- Linear Model -----
184 mlr_model <- lm(mathfinal ~ ., data = train)
185 summary(mlr_model)
186
187 plr_model <- lm(porfinal ~ ., data = trainp)
188 summary(plr_model)
189
190 # Assumption Checks - Math
191 par(mfrow = c(2, 2)) # Diagnostic plots
192 plot(mlr_model)      # Includes: Residuals vs Fitted, QQ plot, Scale-Location,
                      # Residuals vs Leverage
193
194 # Assumption Checks - Portuguese
195 par(mfrow = c(2, 2))
196 plot(plr_model)
197
198
199 # Convert VIF outputs to data frames with variable names
200 vif_math <- as.data.frame(vif(mlr_model)) %>%
201   rownames_to_column(var = "Variable") %>%
202   rename_with(~ paste0("Math_", .), -Variable)
203
204 vif_por <- as.data.frame(vif(plr_model)) %>%
205   rownames_to_column(var = "Variable") %>%
206   rename_with(~ paste0("Portuguese_", .), -Variable)
207
208 # Merge the two VIF tables by variable name
209 vif_comparison <- full_join(vif_math, vif_por, by = "Variable")
210
211 # Print the comparison table
212 print(vif_comparison)
213
214 kable(vif_comparison, format = "latex", booktabs = TRUE,
215       caption = "Side-by-Side VIF Comparison for Math and Portuguese Linear Models")
216 # Variable Math_GVIF Math_Df Math_GVIF^(1/(2*Df)) Portuguese_GVIF Portuguese_Df
217 # 1 guardian 1.480193 2 1.103010 1.571115 2
218 # 2 goout 1.608989 1 1.268459 1.681249 1
219 # 3 school 1.641374 1 1.281161 1.614344 1
220 # 4 sex 1.480817 1 1.216888 1.496284 1
221 # 5 age 1.683393 1 1.297456 1.633526 1
222 # 6 address 1.434334 1 1.197637 1.520429 1
223 # 7 famsize 1.155246 1 1.074824 1.248482 1

```

224	# 8	Pstatus	1.189486	1	1.090636	1.220292	1
			1.104668				
225	# 9	Medu	3.225022	1	1.795835	3.197912	1
			1.788271				
226	# 10	Fedu	2.256879	1	1.502291	2.371832	1
			1.540075				
227	# 11	Mjob	4.711666	4	1.213799	4.768852	4
			1.215631				
228	# 12	Fjob	2.917813	4	1.143226	3.076145	4
			1.150803				
229	# 13	reason	1.812125	3	1.104158	1.901376	3
			1.113041				
230	# 14	nursery	1.247515	1	1.116922	1.200685	1
			1.095758				
231	# 15	internet	1.282259	1	1.132369	1.337381	1
			1.156452				
232	# 16	mathp1	5.567227	1	2.359497	NA	NA
			NA				
233	# 17	mathp2	5.443060	1	2.333037	NA	NA
			NA				
234	# 18	traveltime	1.374866	1	1.172547	1.388562	1
			1.178373				
235	# 19	studytime	1.453808	1	1.205740	1.601372	1
			1.265453				
236	# 20	failures	1.475491	1	1.214698	1.495908	1
			1.223073				
237	# 21	schoolsup	1.395075	1	1.181133	1.401559	1
			1.183875				
238	# 22	famsup	1.250634	1	1.118317	1.199915	1
			1.095406				
239	# 23	paid	1.233115	1	1.110457	1.251845	1
			1.118859				
240	# 24	activities	1.209743	1	1.099883	1.303474	1
			1.141698				
241	# 25	higher	1.367377	1	1.169349	1.432669	1
			1.196942				
242	# 26	romantic	1.278935	1	1.130900	1.262127	1
			1.123444				
243	# 27	famrel	1.220360	1	1.104699	1.226586	1
			1.107513				
244	# 28	freetime	1.448505	1	1.203539	1.338110	1
			1.156767				
245	# 29	Dalc	1.941432	1	1.393353	2.008371	1
			1.417170				
246	# 30	Walc	2.239369	1	1.496452	2.403011	1
			1.550165				
247	# 31	health	1.295174	1	1.138057	1.351867	1
			1.162698				
248	# 32	absences	1.260738	1	1.122826	1.312955	1
			1.145842				
249	# 33	porp1	NA	NA	NA	5.218640	1
			2.284434				
250	# 34	porp2	NA	NA	NA	5.504888	1
			2.346250				
251							
252							
253	#	----- Additional Residual Analysis - Math -----					
254							
255		par(mfrow = c(1, 2))					

```

256 residuals_math <- resid(mlr_model)
257 shapiro.test(residuals_math) # Normality test for Math residuals
258 plot(fitted(mlr_model), residuals_math,
259      main = "Residuals vs Fitted - Math",
260      xlab = "Fitted values", ylab = "Residuals")
261 abline(h = 0, col = "red") # Red line at 0 for reference
262
263 # ----- Additional Residual Analysis - Portuguese -----
264 residuals_por <- resid(plr_model)
265 shapiro.test(residuals_por) # Normality test for Portuguese residuals
266 plot(fitted(plr_model), residuals_por,
267      main = "Residuals vs Fitted - Portuguese",
268      xlab = "Fitted values", ylab = "Residuals")
269 abline(h = 0, col = "red") # Red line at 0 for reference
270
271 # Reset layout to 1 plot per row (optional, for future plotting)
272 par(mfrow = c(1, 1))
273
274
275
276
277 # Reset plot layout
278 par(mfrow = c(1,1))
279
280 # -----
281 # Stepwise Model Selection (AIC, BIC, Forward, Backward)
282 # -----
283 step_aic <- step(mlr_model, direction = "both", trace = 0)
284 summary(step_aic)
285 step_aicp <- step(plr_model, direction = "both", trace = 0)
286 summary(step_aicp)
287
288 step_bic <- step(mlr_model, direction = "both", k = log(nrow(train)), trace = 0)
289 summary(step_bic)
290 step_bicp <- step(plr_model, direction = "both", k = log(nrow(train)), trace = 0)
291 summary(step_bicp)
292
293 forward <- step(lm(mathfinal ~ 1, data = train), scope = formula(mlr_model), direction
294               = "forward")
295 summary(forward)
296 forwardp <- step(lm(porfinal ~ 1, data = trainp), scope = formula(plr_model),
297               direction = "forward")
298 summary(forwardp)
299
300 backward <- step(mlr_model, direction = "backward")
301 summary(backward)
302 backwardp <- step(plr_model, direction = "backward")
303 summary(backwardp)
304
305 ##Comparisons
306
307 # Helper to extract model summary info
308 get_model_info <- function(model) {
309   data.frame(
310     Variables = paste(names(coef(model))[-1], collapse = ", "),
311     R2 = summary(model)$r.squared,
312     Adj_R2 = summary(model)$adj.r.squared,
313     AIC = AIC(model),

```

```

313     BIC = BIC(model)
314   )
315 }
316
317 # Create comparison table
318 model_comparison <- tibble::tibble(
319   Method = c("AIC", "BIC", "Forward", "Backward"),
320
321   Math_Vars = c(
322     get_model_info(step_aic)$Variables,
323     get_model_info(step_bic)$Variables,
324     get_model_info(forward)$Variables,
325     get_model_info(backward)$Variables
326   ),
327   Math_R2 = c(
328     get_model_info(step_aic)$R2,
329     get_model_info(step_bic)$R2,
330     get_model_info(forward)$R2,
331     get_model_info(backward)$R2
332   ),
333   Math_AdjR2 = c(
334     get_model_info(step_aic)$Adj_R2,
335     get_model_info(step_bic)$Adj_R2,
336     get_model_info(forward)$Adj_R2,
337     get_model_info(backward)$Adj_R2
338   ),
339   Math_AIC = c(
340     get_model_info(step_aic)$AIC,
341     get_model_info(step_bic)$AIC,
342     get_model_info(forward)$AIC,
343     get_model_info(backward)$AIC
344   ),
345   Math_BIC = c(
346     get_model_info(step_aic)$BIC,
347     get_model_info(step_bic)$BIC,
348     get_model_info(forward)$BIC,
349     get_model_info(backward)$BIC
350   ),
351
352   Port_Vars = c(
353     get_model_info(step_aicp)$Variables,
354     get_model_info(step_bicp)$Variables,
355     get_model_info(forwardp)$Variables,
356     get_model_info(backwardp)$Variables
357   ),
358   Port_R2 = c(
359     get_model_info(step_aicp)$R2,
360     get_model_info(step_bicp)$R2,
361     get_model_info(forwardp)$R2,
362     get_model_info(backwardp)$R2
363   ),
364   Port_AdjR2 = c(
365     get_model_info(step_aicp)$Adj_R2,
366     get_model_info(step_bicp)$Adj_R2,
367     get_model_info(forwardp)$Adj_R2,
368     get_model_info(backwardp)$Adj_R2
369   ),
370   Port_AIC = c(
371     get_model_info(step_aicp)$AIC,

```

```

372     get_model_info(step_bicp)$AIC,
373     get_model_info(forwardp)$AIC,
374     get_model_info(backwardp)$AIC
375   ),
376   Port_BIC = c(
377     get_model_info(step_aicp)$BIC,
378     get_model_info(step_bicp)$BIC,
379     get_model_info(forwardp)$BIC,
380     get_model_info(backwardp)$BIC
381   )
382 )
383
384 print(model_comparison)
385
386
387 # Save LaTeX table as a string
388 latex_table <- kable(model_comparison, format = "latex", booktabs = TRUE,
389                       caption = "Stepwise Model Comparison for Math and Portuguese")
390
391 # Write to .tex file
392 writeLines(latex_table, "model_comparison_table.tex")
393
394 # -----
395 # Lasso Regression - Math
396 # -----
397 x <- model.matrix(mathfinal ~ ., train)[, -1]
398 y <- train$mathfinal
399 lasso_mod <- cv.glmnet(x, y, alpha = 1)
400 best_lambda <- lasso_mod$lambda.min
401 lasso_final <- glmnet(x, y, alpha = 1, lambda = best_lambda)
402 coef(lasso_final)
403
404 # -----
405 # Lasso Regression - Portuguese
406 # -----
407 xp <- model.matrix(porfinal ~ ., trainp)[, -1]
408 yp <- trainp$porfinal
409 lasso_modp <- cv.glmnet(xp, yp, alpha = 1)
410 best_lambdap <- lasso_modp$lambda.min
411 lasso_finalp <- glmnet(xp, yp, alpha = 1, lambda = best_lambdap)
412 coef(lasso_finalp)
413
414 # Get non-zero coefficients
415 coef_math <- coef(lasso_final)
416 coef_por <- coef(lasso_finalp)
417
418 # Convert to data frames
419 coef_math_df <- as.data.frame(as.matrix(coef_math))
420 coef_math_df <- tibble::rownames_to_column(coef_math_df, var = "Variable")
421 colnames(coef_math_df)[2] <- "Math_Coefficient"
422
423 coef_por_df <- as.data.frame(as.matrix(coef_por))
424 coef_por_df <- tibble::rownames_to_column(coef_por_df, var = "Variable")
425 colnames(coef_por_df)[2] <- "Portuguese_Coefficient"
426
427 # Merge by variable name
428 lasso_compare <- full_join(coef_math_df, coef_por_df, by = "Variable")
429
430

```



```

431 # Keep only coefficients that are non-zero in at least one model
432 lasso_compare_filtered <- lasso_compare %>%
433   filter(Math_Coefficient != 0 | Portuguese_Coefficient != 0)
434
435
436
437
438 kable(lasso_compare_filtered, format = "latex", booktabs = TRUE,
439       caption = "Lasso Coefficients for Math and Portuguese Final Grades") %>%
440   kable_styling(latex_options = c("striped", "hold_position"))
441
442
443
444 # -----
445 # Bagging and Boosting Math
446 # -----
447 set.seed(38520251)
448 bag_mod <- randomForest(mathfinal ~ ., data = train, mtry = ncol(train) - 1)
449 boost_mod <- gbm(mathfinal ~ ., data = train, distribution = "gaussian", n.trees =
450   1000, interaction.depth = 4)
451
452 # -----
453 # Bagging and Boosting Portuguese
454 # -----
455 set.seed(38520251)
456 bag_modp <- randomForest(porfinal ~ ., data = trainp, mtry = ncol(trainp) - 1)
457 boost_modp <- gbm(porfinal ~ ., data = trainp, distribution = "gaussian", n.trees =
458   1000, interaction.depth = 4)
459
460 # Math Model Predictions
461 pred_bag <- predict(bag_mod, newdata = test)
462 pred_boost <- predict(boost_mod, newdata = test, n.trees = 1000)
463
464 # Portuguese Model Predictions
465 pred_bagp <- predict(bag_modp, newdata = testp)
466 pred_boostp <- predict(boost_modp, newdata = testp, n.trees = 1000)
467
468 # RMSE Calculation Function
469 rmse <- function(actual, predicted) {
470   sqrt(mean((actual - predicted)^2))
471 }
472
473 # RMSE for Math Models
474 rmse_bag_math <- rmse(test$mathfinal, pred_bag)
475 rmse_boost_math <- rmse(test$mathfinal, pred_boost)
476
477 # RMSE for Portuguese Models
478 rmse_bag_por <- rmse(testp$porfinal, pred_bagp)
479 rmse_boost_por <- rmse(testp$porfinal, pred_boostp)
480
481 # Collecting RMSE into a summary table
482 bagging_boosting_summary <- data.frame(
483   Model = c("Bagging (Math)", "Boosting (Math)", "Bagging (Portuguese)", "Boosting (
484     Portuguese)"),
485   RMSE = c(rmse_bag_math, rmse_boost_math, rmse_bag_por, rmse_boost_por)
486 )
487
488 # Print the summary table
489 print(bagging_boosting_summary)

```

```

487 kable(bagging_boosting_summary, format = "latex", booktabs = TRUE,
488       caption = "Model Comparison: Bagging and Boosting (Math and Portuguese)")
489
490
491 # -----
492 # Principal Component Regression (PCR) Math
493 # -----
494 pcr_mod <- pcr(mathfinal ~ ., data = train, scale = TRUE, validation = "CV")
495 summary(pcr_mod)
496 validationplot(pcr_mod, val.type = "MSEP")
497
498 # -----
499 # Principal Component Regression (PCR) Portuguese
500 # -----
501 pcr_modp <- pcr(porfinal ~ ., data = trainp, scale = TRUE, validation = "CV")
502 summary(pcr_modp)
503 validationplot(pcr_modp, val.type = "MSEP")
504
505 # -----
506 # Partial Least Squares Regression (PLS) Math
507 # -----
508 pls_mod <- plsr(mathfinal ~ ., data = train, scale = TRUE, validation = "CV")
509 summary(pls_mod)
510 validationplot(pls_mod, val.type = "MSEP")
511
512 # -----
513 # Partial Least Squares Regression (PLS) Portuguese
514 # -----
515 pls_modp <- plsr(porfinal ~ ., data = trainp, scale = TRUE, validation = "CV")
516 summary(pls_modp)
517 validationplot(pls_modp, val.type = "MSEP")
518
519
520 # Optimal number of components for PCR and PLS (Math)
521 opt_comp_pcr_math <- which.min(MSEP(pcr_mod)$val[1, , ])
522 opt_comp_pls_math <- which.min(MSEP(pls_mod)$val[1, , ])
523
524 # Optimal number of components for PCR and PLS (Portuguese)
525 opt_comp_pcr_por <- which.min(MSEP(pcr_modp)$val[1, , ])
526 opt_comp_pls_por <- which.min(MSEP(pls_modp)$val[1, , ])
527
528 # Predict using Optimal Components
529
530 # Math
531 pred_pcr <- predict(pcr_mod, newdata = test, ncomp = opt_comp_pcr_math)
532 pred_pls <- predict(pls_mod, newdata = test, ncomp = opt_comp_pls_math)
533
534 # Portuguese
535 pred_pcrp <- predict(pcr_modp, newdata = testp, ncomp = opt_comp_pcr_por)
536 pred_plsp <- predict(pls_modp, newdata = testp, ncomp = opt_comp_pls_por)
537
538 # RMSE
539 rmse_pcr_math <- rmse(test$mathfinal, pred_pcr)
540 rmse_pcr_por <- rmse(testp$porfinal, pred_pcrp)
541 rmse_pls_math <- rmse(test$mathfinal, pred_pls)
542 rmse_pls_por <- rmse(testp$porfinal, pred_plsp)
543
544 # Summary table
545

```

```

546 pcr_pls_summary <- data.frame(
547   Model = c("PCR (Math)", "PCR (Portuguese)", "PLS (Math)", "PLS (Portuguese)"),
548   Optimal_Components = c(opt_comp_pcr_math, opt_comp_pcr_por, opt_comp_pls_math, opt_
     comp_pls_por),
549   RMSE = c(rmse_pcr_math, rmse_pcr_por, rmse_pls_math, rmse_pls_por)
550 )
551
552 kable(pcr_pls_summary, format = "latex", booktabs = TRUE,
553       caption = "PCR and PLS Model Comparison Using Optimal Number of Components")
554
555 # -----
556 # Random Forest - Math
557 # -----
558 rf_mod <- randomForest(mathfinal ~ ., data = train, importance = TRUE)
559 print(rf_mod)
560 varImpPlot(rf_mod)
561
562 # -----
563 # Random Forest - Portuguese
564 # -----
565 rf_modp <- randomForest(porfinal ~ ., data = trainp, importance = TRUE)
566 print(rf_modp)
567 varImpPlot(rf_modp)
568
569 # -----
570 # Support Vector Machine (SVM) math
571 # -----
572 svm_mod <- svm(mathfinal ~ ., data = train)
573 pred_svm <- predict(svm_mod, newdata = test)
574
575 # -----
576 # Support Vector Machine (SVM) Portugese
577 # -----
578 svm_modp <- svm(porfinal ~ ., data = trainp)
579 pred_svmp <- predict(svm_modp, newdata = testp)
580
581 # Define RMSE function (if not already)
582 rmse <- function(actual, predicted) {
583   sqrt(mean((actual - predicted)^2))
584 }
585
586
587 # Predictions for Random Forest
588 pred_rf <- predict(rf_mod, newdata = test)
589 pred_rfp <- predict(rf_modp, newdata = testp)
590
591 # RMSE for Random Forest
592 rmse_rf_math <- rmse(test$mathfinal, pred_rf)
593 rmse_rf_por <- rmse(testp$porfinal, pred_rfp)
594
595 # RMSE for SVM
596 rmse_svm_math <- rmse(test$mathfinal, pred_svm)
597 rmse_svm_por <- rmse(testp$porfinal, pred_svmp)
598
599 rf_svm_summary <- data.frame(
600   Model = c("Random Forest (Math)", "Random Forest (Portuguese)",
601             "SVM (Math)", "SVM (Portuguese)"),
602   RMSE = c(rmse_rf_math, rmse_rf_por, rmse_svm_math, rmse_svm_por)
603 )

```

```

604 kable(rf_svm_summary, format = "latex", booktabs = TRUE,
605       caption = "Model Comparison: Random Forest and SVM for Math and Portuguese")
606 # -----
607 # Model Evaluation - Math
608 # -----
609 pred_lm <- predict(mlr_model, newdata = test)
610 pred_rf <- predict(rf_mod, newdata = test)
611 pred_pcr <- predict(pcr_mod, newdata = test, ncomp = 5)
612 pred_pls <- predict(pls_mod, newdata = test, ncomp = 5)
613 pred_bag <- predict(bag_mod, newdata = test)
614 pred_boost <- predict(boost_mod, newdata = test, n.trees = 1000)
615
616 rmse <- function(actual, predicted) {
617   sqrt(mean((actual - predicted)^2))
618 }
619 r2 <- function(actual, predicted) {
620   cor(actual, predicted)^2
621 }
622
623 # RMSE & R for Math
624 math_metrics <- data.frame(
625   Model = names(math_rmse),
626   RMSE = round(math_rmse, 3),
627   R2 = round(c(
628     r2(test$mathfinal, pred_lm),
629     r2(test$mathfinal, predict(step_aic, newdata = test)),
630     r2(test$mathfinal, predict(step_bic, newdata = test)),
631     r2(test$mathfinal, predict(forward, newdata = test)),
632     r2(test$mathfinal, predict(backward, newdata = test)),
633     r2(test$mathfinal, predict(lasso_final, s = best_lambda, newx = model.matrix(test$
634       mathfinal ~ ., test)[, -1])),
635     r2(test$mathfinal, pred_pcr),
636     r2(test$mathfinal, pred_pls),
637     r2(test$mathfinal, pred_bag),
638     r2(test$mathfinal, pred_boost),
639     r2(test$mathfinal, pred_rf),
640     r2(test$mathfinal, pred_svm)
641   ), 3)
642 )
643
644 # RMSE & R for Portuguese
645 portugese_metrics <- data.frame(
646   Model = names(portugese_rmse),
647   RMSE = round(portugese_rmse, 3),
648   R2 = round(c(
649     r2(testp$porfinal, pred_lmp),
650     r2(testp$porfinal, predict(step_aicp, newdata = testp)),
651     r2(testp$porfinal, predict(step_bicp, newdata = testp)),
652     r2(testp$porfinal, predict(forwardp, newdata = testp)),
653     r2(testp$porfinal, predict(backwardp, newdata = testp)),
654     r2(testp$porfinal, predict(lasso_finalp, s = best_lambdap, newx = model.matrix(
655       testp$porfinal ~ ., testp)[, -1])),
656     r2(testp$porfinal, pred_pcrp),
657     r2(testp$porfinal, pred_plsp),
658     r2(testp$porfinal, pred_bagp),
659     r2(testp$porfinal, pred_boostp),
660     r2(testp$porfinal, pred_rfp),
661     r2(testp$porfinal, pred_svm)

```

```

661   ), 3)
662 )
663
664 ##Combine into final table
665
666 final_metrics <- data.frame(
667   Model = math_metrics$Model,
668   RMSE_Math = math_metrics$RMSE,
669   R2_Math = math_metrics$R2,
670   RMSE_Portuguese = portugese_metrics$RMSE,
671   R2_Portuguese = portugese_metrics$R2
672 )
673
674 # Latex
675 kable(final_metrics, format = "latex", booktabs = TRUE,
676       caption = "Model Performance Comparison: RMSE and R2 for Math and Portuguese
                   Final Grades")

```