

Penetration Rates for Prostate Cancer

November 12, 2020

Executive Summary:

Introduction:

Methods:

Exploratory Data Analysis:

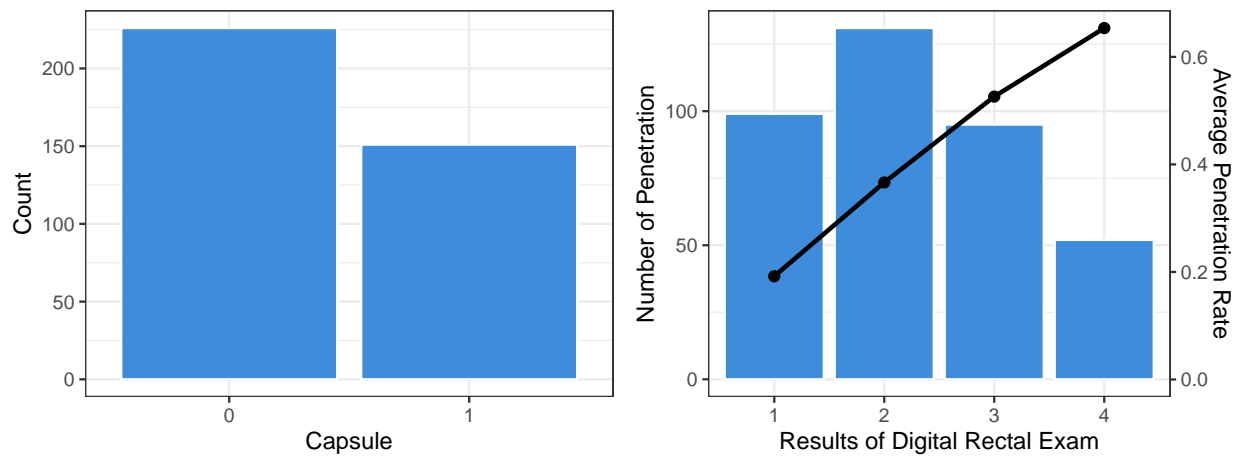


Figure 1

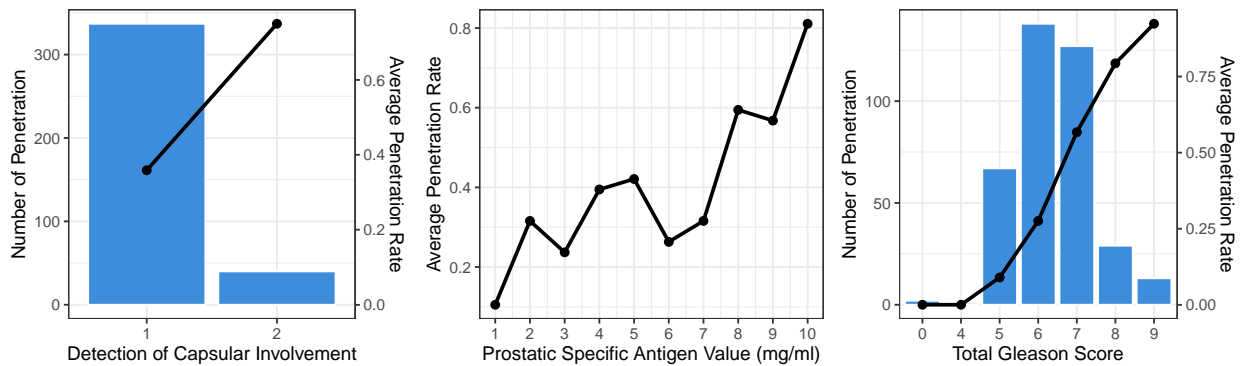


Figure 2

Model Fitting/Inferences:

term	estimate	std.error	statistic	p.value	OR	2.5 %	97.5 %
(Intercept)	-7.254	1.296	-5.595	0.000	0.001	0.000	0.008
dpros2	0.822	0.461	1.785	0.074	2.276	0.944	5.835
dpros3	1.306	0.483	2.700	0.007	3.689	1.466	9.892
dpros4	1.280	0.577	2.220	0.026	3.598	1.176	11.452
psa	0.041	0.015	2.760	0.006	1.042	1.015	1.075
gleason	0.846	0.200	4.231	0.000	2.331	1.599	3.514

Table 1: Summary regression of final model

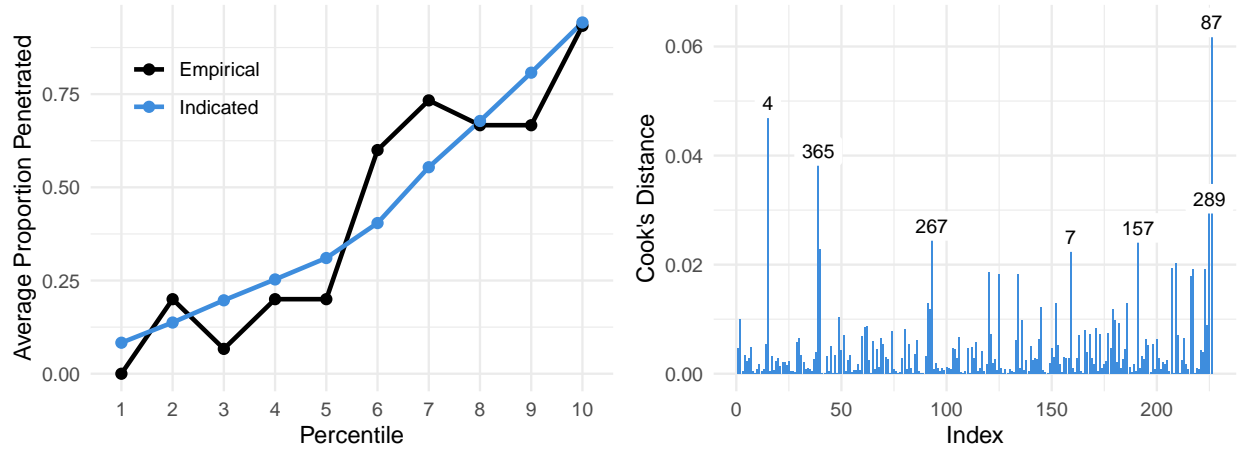


Figure 3

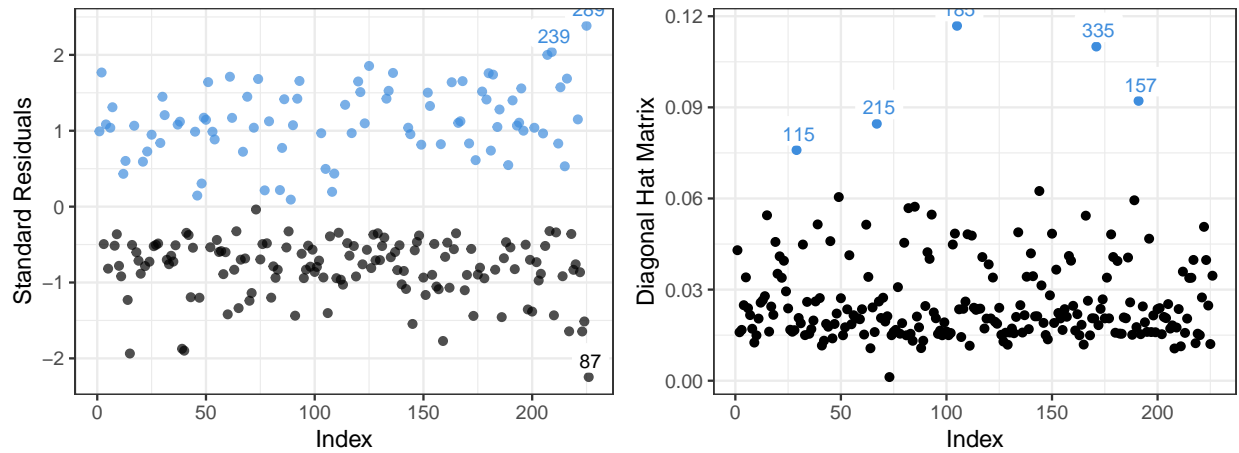


Figure 4

Conclusion:

Appendix A: Supplemental Tables

Table 2: Summary Statistics for all numerical independent features

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
capsule	377	0.401	0.491	0	0	1	1
age	377	66.066	6.425	47	62	71	79
psa	377	15.256	19.869	0.300	5.000	16.800	139.700
gleason	377	6.379	1.092	0	6	7	9
psa_bins	377	5.472	2.869	1	3	8	10

	random_partition	nrow_train	nrow_test	features	AIC	auc
1	0.900	339	38	dpros,psa,gleason,dcaps	298.077	0.822
2	0.800	301	76	dpros,psa,gleason,dcaps	278.940	0.820
3	0.600	226	151	dpros,psa,gleason	241.799	0.819
4	0.700	263	114	dpros,psa,gleason	260.832	0.818
5	0.600	226	151	dpros,psa,gleason,dcaps	240.567	0.818
6	0.700	263	114	dpros,psa,gleason,dcaps	258.675	0.815
7	0.900	339	38	dpros,psa,gleason	297.050	0.812
8	0.800	301	76	dpros,psa,gleason	277.654	0.809
9	0.600	226	151	psa,gleason,dcaps	251.167	0.805
10	0.900	339	38	dpros,gleason	301.355	0.805

Table 3

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-9.7465	3.8714	-2.52	0.0118
dpros2	1.7503	4.1518	0.42	0.6733
dpros3	2.9024	4.2654	0.68	0.4962
dpros4	4.8459	4.7046	1.03	0.3030
psa	0.2090	0.1227	1.70	0.0886
gleason	1.0748	0.6161	1.74	0.0811
dpros2:psa	-0.0986	0.0638	-1.55	0.1221
dpros3:psa	-0.1225	0.0801	-1.53	0.1260
dpros4:psa	-0.0601	0.0699	-0.86	0.3902
dpros2:gleason	0.0468	0.6584	0.07	0.9434
dpros3:gleason	-0.0354	0.6803	-0.05	0.9585
dpros4:gleason	-0.4452	0.7431	-0.60	0.5491
psa:gleason	-0.0130	0.0168	-0.77	0.4396

Table 4

	capsule	dpros	psa	gleason	rownames
1	0	2	51.20	7	4
2	0	4	31.90	7	7
3	0	4	17.10	9	87
4	1	4	45.30	6	115
5	0	4	25.20	7	121
6	1	1	44.40	6	157
7	1	1	85.40	7	185
8	1	3	17.70	5	196
9	1	4	53.90	6	215
10	1	1	6.70	6	239
11	0	4	25.10	7	254
12	1	4	18.70	5	267
13	1	1	8.90	6	272
14	1	1	6.80	5	289
15	0	3	18.10	8	307
16	1	2	58.00	6	335
17	0	2	48.00	7	365

Table 5

Appendix B: R Code

```

1 ##### Load packages #####
2 library(MASS)
3 library(magrittr)
4 library(tidyverse)
5 library(ggcorrplot)
6 library(car)
7 library(broom)
8
9
10 ##### Functions #####
11
12 #' Explore variables function, returning plots of variables count and average
    proportion
13 #'
14 #' @param df dataframe
15 #' @param xvar independent variable
16 #' @param count return histogram of count, default is TRUE
17 #' @param x_axis x axis name
18 #'
19 #' @return
20 #' @export
21 #'
22 #' @examples
23 ExploreVariable <- function(df, xvar, count = TRUE, x_axis){
24
25   group_df <- df %>%
26     group_by({{xvar}}) %>%
27     summarise(avg_prop_capsule = mean(capsule),
28               count = n(),
29               .groups = "drop")
30
31   ratio <- max(group_df$count) / max(group_df$avg_prop_capsule)
32
33   if(count == TRUE){
34     p <- ggplot(group_df, aes(x = factor({{xvar}}), group = 1)) +
35       geom_bar(aes(y = count), stat = "identity", fill = "#3e8ddd", col = "white") +
36       geom_point(aes(y = avg_prop_capsule * ratio), size = 2, color = "black") +
37       geom_line(aes(y = avg_prop_capsule * ratio), size = 1, color = "black") +
38       scale_y_continuous(sec.axis = sec_axis(~./ratio, name = "Average Penetration
    Rate"))
39
40     y_lab <- "Number of Penetration"
41
42   } else {
43     p <- ggplot(group_df, aes(x = {{xvar}}, group = 1)) +
44       geom_point(aes(y = avg_prop_capsule), size = 2, color = "black") +
45       geom_line(aes(y = avg_prop_capsule), size = 1, color = "black")
46
47     y_lab <- "Average Penetration Rate"
48   }
49   p +
50     labs(y = y_lab, x = x_axis) +
51     theme_bw()
52 }
53

```



```

110         features = paste(xvars, collapse = ","),
111         AIC = mean(all_aic),
112         auc = mean(all_auc))
113     index <- index + 1
114 }
115 all_iteration %<>% bind_rows()
116
117 return(all_iteration)
118 }
119
120 SafelyBinaryFit <- safely(BinaryFit)
121
122
123 #' Function to loop all the models through each variable
124 #'
125 #' @param rp
126 #' @param nrounds
127 #' @param list_of_xvars
128 #' @param number_of_xvars
129 #'
130 #' @return
131 #' @export
132 #'
133 #' @examples
134 LoopAllVars <- function(rp, nrounds, list_of_xvars, number_of_xvars){
135   # group variables
136   vars <- combn(list_of_xvars, number_of_xvars)
137
138   # how many group do we have?
139   no_vars <- dim(vars)[[2]]
140
141   # initiate print list and index
142   print_list <- list()
143   k <- 1
144
145   for (i in 1:no_vars) {
146     iteration_log <- SafelyBinaryFit(rp, xvars = vars[,i], nrounds)
147     print_list[[k]] <- iteration_log
148     k <- k + 1
149   }
150   # map all separate log to bind them together into a dataframe
151   iteration_log <- map(print_list, "result") %>% bind_rows()
152
153   return(iteration_log)
154 }
155
156
157 ##### Load data #####
158
159 ## Load in prostate data set
160 prostate <- read.table("data/prostate.txt", header = T)
161
162
163 ##### Clean data & Data Pre-processing #####
164
165 ## check for NA

```

```

166 sum(is.na(prostate)) # missing data: 3 in race
167
168 ## omit NA
169 prostate %>%
170   na.omit()
171
172 ## change categorical variables to class factor
173 prostate %>%
174   modify_at(c("dcaps", "dpros", "race"), as.factor)
175
176 ## add bins for psa
177 prostate %>%
178   mutate(psa_bins = ntile(psa, 10))
179
180
181 ##### Data Exploratory Analysis #####
182
183 ## Target Variable — capsule — histogram
184 ggplot(prostate, aes(factor(capsule))) +
185   geom_histogram(stat = "count", fill = "#3e8ddd", col = "white") +
186   theme_bw() +
187   labs(x = "Capsule",
188        y = "Count")
189
190 ## Predictors
191 ExploreVariable(prostate, age, count = FALSE, x_axis = "Age")
192 ExploreVariable(prostate, race, x_axis = "Race")
193 ExploreVariable(prostate, dpros, x_axis = "Results of Digital Rectal Exam")
194 ExploreVariable(prostate, dcaps, x_axis = "Detection of Capsular Involvement")
195 ExploreVariable(prostate, psa_bins, count = FALSE,
196                 x_axis = "Prostatic Specific Antigen Value (mg/ml)") +
197   scale_x_continuous(breaks = seq(1, 10, 1))
198 ExploreVariable(prostate, gleason, x_axis = "Total Gleason Score")
199
200 # after the variable exploratory, we can see that
201 # — age don't have a relationship with capsule
202 # — there is little to no relationship with capsule, and unbalance factors
203 # — dpros has a strong effect of getting penetration by different levels
204 # — dcaps also has strong effect of getting penetration
205 # — psa has a strong positive relationship with capsule
206 # — gleason also has a strong positive relationship with capsule
207 # we can continue the analysis using 4 variables: dpros, dcaps, psa, and gleason
208
209
210 ##### Check Correlation for Continuous variables #####
211 corr_table <- prostate %>%
212   select(age, psa, gleason) %>%
213   cor()
214
215 ggcorrplot(corr_table, hc.order = TRUE,
216            outline.col = "white",
217            colors = c("blue", "white", "red"))
218
219 # no high correlation between continuous predictors
220 # all variables can be in the same model
221

```



```

222
223 ##### Model Development #####
224
225 ## parameter for bootstrapping
226
227 xvars <- c("dpros", "psa", "gleason", "dcaps")
228
229 rp_par <- c(0.6, 0.7, 0.8, 0.9)      # random partition parameters
230
231 nrounds <- 10                        # number of rounds per model
232
233 ## Bootstrapping
234
235 # initiate print list
236 print_list <- list()
237
238 # loop
239 for (n_features in 1:4){
240   results <- LoopAllVars(rp = rp_par,
241                           nrounds = nrounds,
242                           list_of_xvars = xvars,
243                           number_of_xvars = n_features)
244   print_list[[n_features]] <- results
245 }
246
247 # bind iteration log into a dataframe
248 iteration_log <- print_list %>% bind_rows()
249
250 # after running the iteration logs, models with variables dpros, psa, and gleason
251 # has the high AUC and low AIC, using train set of 60% prop and test set of 40%
252
253
254 ##### Select Model after bootstrap #####
255
256 ## Split data to train/test set using random partition with proportion .7/.3
257 set.seed(1234)
258
259 n <- floor(.6 * nrow(prostate))
260
261 train_ind <- sample(seq_len(nrow(prostate)), size = n)
262
263 train <- prostate[train_ind, ]
264
265 test <- prostate[-train_ind, ]
266
267
268 ## start with initial fit using 4 variables
269 fit1 <- glm(capsule ~ dpros + psa + gleason,
270             family = binomial(link = logit),
271             data = train)
272
273
274 ## Checking for Interaction term
275
276 fit2 <- glm(capsule ~ .*,
277             family = "binomial",

```

```

278         data = train %>% dplyr::select(dpros, psa, gleason, capsule))
279
280 test$preds2 <- predict(fit2, newdata = test, type = "response")
281
282 test_roc <- pROC::roc(response = test$capsule, predictor = test$preds2)
283 test_auc <- as.data.frame(pROC::auc(test_roc))
284 aic <- fit2$aic
285
286 summary(fit2)
287
288 stepAIC(fit2, direction = c("both"))
289
290 # although auc for this model is high and AIC is low,
291 # each coefficient in the model is not statistically significant
292 # so we're not considering this model to be the final model
293 # also using model selection function stepAIC,
294 # interaction terms are not needed in the model
295
296 ##### Final Model #####
297
298 fit_final <- fit1
299
300 ## generate predictions using test set
301 test$preds <- predict(fit_final, newdata = test, type = "response")
302
303 ## create percentile to generate lift chart
304 test$percentile <- ntile(test$preds, 10)
305
306 lift_plot_df <- test %>%
307   group_by(percentile) %>%
308   summarise("Empirical" = mean(capsule),
309             "Indicated" = mean(preds)) %>%
310   gather("key", "value", -percentile)
311
312 ## Lift chart to visualize the quality of the model
313 ggplot(lift_plot_df, aes(x = factor(percentile), y = value, color = key, group = key
314   )) +
315   geom_line(size = 1) +
316   geom_point(size = 2) +
317   theme_minimal() +
318   labs(x = "Percentile",
319        y = "Average Proportion Penetrated") +
320   theme(legend.title = element_blank()) +
321   scale_color_manual(values = c("black", "#3e8ddd"))
322
323 ## compute AIC and AUC of final model
324 aic <- fit_final$aic
325 test_roc <- pROC::roc(response = test$capsule, predictor = test$preds)
326 test_auc <- as.data.frame(pROC::auc(test_roc))
327
328 tidy <- broom::tidy(fit_final) %>%
329   mutate(OR = exp(fit_final$coefficients)) %>%
330   cbind(exp(confint(fit_final))) %>%
331   modify_if(is.numeric, round, 4)
332
333 ## wald test

```

```

333 aod::wald.test(b = coef(fit_final), Sigma = vcov(fit_final), Terms = 4)
334
335
336 ## Generate confusion matrix and accuracy
337 test %>%
338   mutate(preds = case_when(preds < 0.5 ~ 0.5, TRUE ~ 1.1))
339
340 confusion_matrix <- ftable(test$capsule, test$preds)
341
342 accuracy <- sum(diag(confusion_matrix))/nrow(test)*100
343
344 accuracy
345
346
347 ##### Diagnostics #####
348
349 ##### Outliers test #####
350 ## Bonferonni p-value for most extreme obs
351 outlierTest(fit_final)
352
353
354 ## Influential Observations
355 influencePlot(fit_final,
356               id.method="identify",
357               main="Influence Plot",
358               sub="Circle size is proportional to Cook's Distance" )
359
360 diag <- augment(fit_final) %>%
361   mutate(Index = 1:nrow(.))
362
363 cutoff <- 4/((nrow(train) - length(fit_final$coefficients) - 2))
364
365 diag %>%
366   mutate(high_cooksd = case_when(
367     .cooks > cutoff ~ 1, TRUE ~ 0),
368     col_stdresid = case_when(
369       .std.resid > 0 ~ 1,
370       .std.resid < 0 ~ 0),
371     high_hat = case_when(
372       .hat > .07 ~ 1,
373       TRUE ~ 0))
374
375 ## cook's distant plot
376 ggplot(diag, aes(x = Index, y = .cooks)) +
377   geom_bar(stat = "identity", fill = "#3e8ddd", col = "white") +
378   labs(y = "Cook's Distance") +
379   theme_minimal() +
380   geom_label(data = diag %>% filter(.cooks > cutoff + .005),
381             aes(label = .rownames), label.size = NA, size = 3)
382
383
384 ## standard residual plot
385 ggplot(diag, aes(x = Index, y = .std.resid, color = factor(col_stdresid))) +
386   geom_point(alpha = 0.7) +
387   labs(y = "Standard Residuals") +
388   theme_bw() +

```

```

389   scale_color_manual(values = c("black", "#3e8ddd")) +
390   theme(legend.position = "none") +
391   geom_label(data = diag %>% filter(.std.resid > 2 | .std.resid < -2),
392             aes(label = .rownames), label.size = NA, size = 3, hjust=0.45, vjust
              =-.15)
393
394 ## diagonal hat matrix plot
395 ggplot(diag, aes(x = Index, y = .hat, color = factor(high_hat))) +
396   geom_point() +
397   labs(y = "Diagonal Hat Matrix") +
398   theme_bw() +
399   scale_color_manual(values = c("black", "#3e8ddd")) +
400   theme(legend.position = "none") +
401   geom_label(data = diag %>% filter(.hat > .07),
402             aes(label = .rownames), label.size = NA, size = 3, hjust=0.45, vjust
              =-.15)
403
404 ## student residual distribution
405 sresid <- studres(fit_final)
406 hist(sresid, freq=FALSE,
407       main="Distribution of Studentized Residuals")
408 xfit<-seq(min(sresid),max(sresid),length=40)
409 yfit<-dnorm(xfit)
410 lines(xfit, yfit, col = "red")
411 # looks normal
412
413 ##### Collinearity #####
414 vif(fit_final) # variance inflation factors
415 # all VIF are low = no multicollinearity
416
417
418 ##### Find influential point in data #####
419
420 inf_points <- (diag %>%
421               filter(.std.resid > 2 | .std.resid < -2 | .cooksd > cutoff | .hat >
422                     .07))$.rownames %>%
423   as.numeric()
424
425 prostate[inf_points, ] %>%
426   as.data.frame() %>%
427   dplyr::select(capsule, dpros, psa, gleason) %>%
428   mutate(rownames = inf_points) %>%
429   arrange(rownames)

```

Listing 1: Appendix of Code