Tumor Penetration Rates for Prostate Capsule

November 12, 2020

Executive Summary: With the common prostate cancer diagnostic among men, the goal of this analysis is to find the variables that could predict the chances of a tumor has penetrated of prostatic capsule. A dataset was provided by the Ohio State University Comprehensive Cancer Center to analyze how demographic and screening results could cause the likelihood of penetration. A tool was built using multiple logistic regressions to predict the chances of a patient getting diagnosed with prostate cancer using three variables results of digital rectal exam, Prostatic Specific Antigen value, and total Gleason score. This model was proved to be the best model with an AUC of 0.82, an accuracy of 0.77, Sensitivity of 0.75, and Specificity of 0.79. Some interesting findings includes a 269% increase in chances of having prostate cancer when a patient has a unilobar nodule from the left on their results of digital rectal exam compared to a patient with no nodule. Furthermore, patients who have bilobar nodule results on their digital rectal exam will have higher chance of having prostate cancer compared to patients with no nodule, 260% more to be exact. In addition, patients could gain 133% more likelihood of being diagnosed with prostate cancer when they have a higher Gleason score. Knowing these three variables (results of digital rectal exam, Prostatic Specific Antigen value, and total Gleason score) are significant factors of the likelihood of penetration, health care professionals would be more cautious in treating each patient. In addition, patients could detect their prostate cancer stages earlier and seek help accordingly.

Introduction: Prostate cancer is one of the most common cancers among men. The survival rate is vary depending on the stages of the cancer. Prostate cancer has a high probability of being treated if detected at early stages. There are many important characteristics that can predict whether a tumor has penetrated the prostatic capsule. With the results of penetration rate, doctors and nurses would prioritize the patients with the high-risk level. On the other hand, if the patients have low rate in penetration, health care workers could find the best treatments for them to slow down prostate cancer growth rate or even treat the patients. In other words, hospitals can plan their budgets and supplies to focus more on high risk patients and minimize supplies to low risk patients. Knowing the penetration rates does not only benefit health professionals, but it would also assist male patients to detect if their tumors have penetrated of prostatic capsule. This way, patients would seek professional help at an earlier stage to increase their survival rate. Many questions have proposed in favor of these issues including 1) What variables cause the penetration rate to increase or decrease? 2) What is the likelihood of penetration rate? 3) How accurate is the model proposed in this analysis? 4) How can we implement the model? and 5) How can we improve the current model? This report will answer all of the questions mentioned above by going through all crucial parts of the data analysis such as data exploratory analysis process to explore the relationship of each variable, model development using multiple logistic regression to build the models, and diagnostics process to examine the quality of the model. This model will serve the purpose of predicting penetration rate for prostate cancer in men to minimize human errors and provide an effective tool for health care workers.

Methods: A subset of data was provided from the Ohio State University Comprehensive Cancer Center to find variables to predict if a tumor has penetrated the prostatic capsule. This dataset contains a total of 380 subjects, out of which, 153 subjects had a tumor that penetrated the capsule. There are three observations contain missing values will be removed. After eliminating missing values to build more quality models, the dataset now has a total of 377 observations. These data contains one identification column with numbers of each subject, two demographic columns including age and race of the subjects, and four health measurements such as the results of digital rectal exam, detection of capsular involvement, Prostatic Specific Antigen value, and total Gleason score. Health measurement variables will be explained in detail in the exploratory data analysis section. The goal of this analysis is to find the average tumor penetration of prostatic capsule rate. To accomplish this goal, we will first analyze the relationship between each predictor and target variable, select models using bootstrapping method, consider interaction terms to be in the final model, check for the quality of the model, and conclude with an interpretation of the model outputs. All analysis will be done in R Studio with R version 3.6.2.

Exploratory Data Analysis: Before selecting variables for our model development process, it is important to access the relationship of each independent variable with the target variable. Table A2 shows descriptive statistics of each independent variable. As seen in this table, the maximum value of Prostatic Specific Antigen is relatively high compared to the median and mean values suggesting there might be outliers. In addition, there are disproportion weights of each factor in race of the subjects and detection of capsular involvement. This indicates that it might be biased to predict the outcome if we use these two variables in the models. The rest of the variables seem proportionally balanced and no skewness were detected among them. In terms of the tumor penetration of prostatic capsule, the response variable has a balance proportion of 40% of tumor has penetrated the prostatic capsule out of 377 subjects. Figure 1.1 show a histogram of the chances of the tumor penetrated of prostatic capsule (1 = penetration, 0 = no penetration).
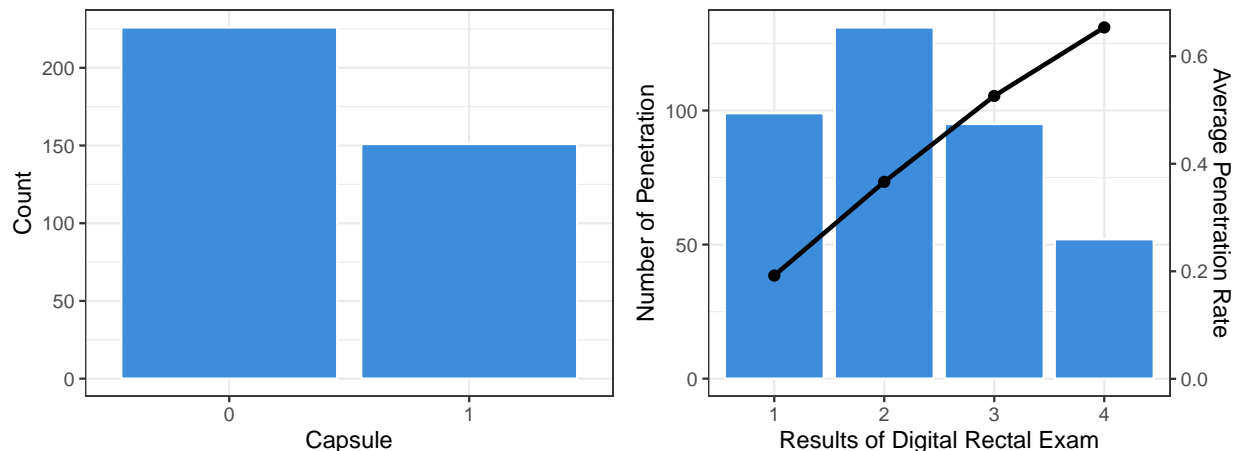

Figure 1: Histogram of 1 (Left): Tumor penetration of prostatic capsule (1 = penetration) and 2 (Right): Results of Digital Rectal Exam (1 = no nodule, 2 = unilobar nodule left, 3 = unilobar nodule right, 4 = bilobar nodule) with average penetration rate in black.

The digital rectal exam is a simple exam to examine the prostate to determine the size of the prostate and any abnormalities. Clearly, different abnormalities would have different chances of

penetration of prostatic capsule. Figure 1.2 shows four categories of the results of digital rectal exam including no nodule, unilobar nodule (left), unilobar nodule (right), and bilobar nodule noted using number from 1 to 4, respectively. Here, this variable is identified as categorical variable with four levels instead of a continuous variable. As seen in plot, these four categories obviously have different average penetration rates. For example, a subject with no nodule would be less likely to penetrate compared to a person with bilobar nodule. It makes sense because if there is no nodule detected in the prostate area, then there is less chances of the tumor will penetrate the prostatic capsule. In addition, Figure 1.2 also depicts the number of subjects in each category. Unilobar nodule (left) is the most common and bilobar nodule is the least common among prostate cancer. However, the proportion of each factor for digital rectal exam are not too different. Thus, this variable would provide significant impact for the model.

Detection of capsular involvement can describe if the tumor is involving or extending beyond the prostate capsule. The tumor is involved when there is capsular involvement present. Noticeably, the patients who have capsular involvement present can be riskier than patients who do not have capsular involvement. Figure 2.1 shows the count and proportion of penetration for the detection of capsular involvement. This variable has two levels (1 = no, 2 = yes) to suggest if a subject has capsular involvement or not. As seen in the figure, patients with capsule involvement would be much more likely to have tumor penetrated of prostatic capsule compared to patients that do not have capsule involvement. However, the count for each level is not proportional. Majority of the subjects have level 1 with no capsular involvement. This variable could be a good variable for our model since it shows a strong signal. However, caution about the disproportional needs to be considered before finalizing the model.
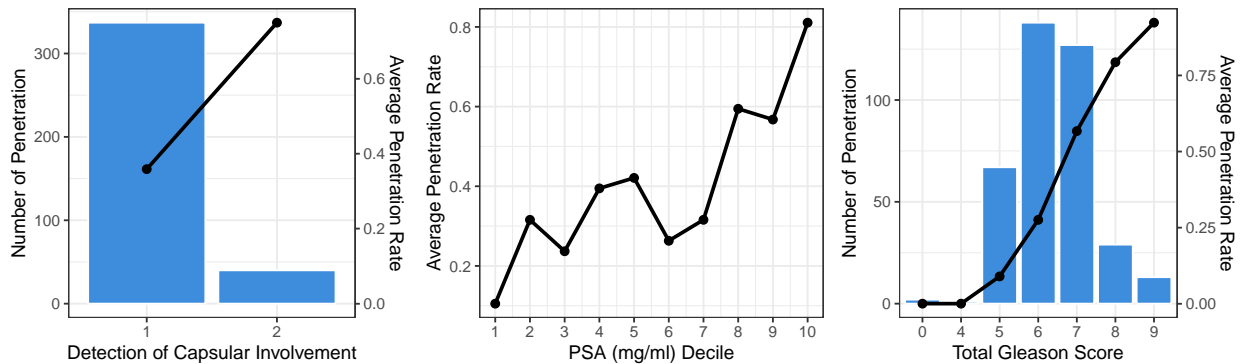

Figure 2: Plots of each variable with average penetration rate in black of 1 (Left): Detection of capsular involvement along with the frequency of each factor, 2 (Center): Prostatic Specific Antigen value with mg/ml unit by decile, and 3 (Right): Total Gleason score along with its histogram.

Prostatic Specific Antigen (PSA) can be found in the prostate gland cells. This screening test measures the amount of prostate-specific antigen in patients' blood using milligram per milliliter (mg/ml) of blood. Higher PSA level could lead to a higher possibility of diagnosing with prostate cancer. Figure 2.2 shows a plot of the relationship between PSA and the average penetration of prostatic capsule rate. Here, the data point is being sorted from lowest to highest, divided into 10 equal buckets, and assigned a decide for visualization purposes. As one can see, as the decile number increases by one unit, the chances of penetration rate increases, on average. Since PSA in higher buckets are generally the largest, they have higher risk of getting prostate cancer than PSA

in lower buckets. Therefore, this proves that PSA could provide a significant relationship for the likelihood of tumor penetrated of prostatic capsule.

Last but not least, total Gleason score can measure the abnormality of the cells to see how the cells are arranged using a scale from 1 to 10. Higher scores lead to a higher possibility of penetration in prostatic capsule. On the other hand, cancer cells that looks similar to normal cells can be considered as low risk and have lower scores. Figure 2.3 illustrates a histogram of total Gleason score overlayed by the average risk of having cancer. Total Gleason score seems to have a normal distribution with less subjects having low or high scores and most subjects having median score of 6. In addition, the average penetration rate is increasing as the total Gleason score is higher. This mean that if a patient's cancer cells look similar to normal cells, they have less chances of being diagnosed with prostate cancer. Nevertheless, if the Gleason scores are high, the patients would have high risk of being diagnosed with prostate cancer. Hence, this variable can also provide excellent signal for the response variable. Other variables did not mention above have little to no relationship with tumor penetration including age and race of the subjects.

After accessing the relationship of predictors and response variable, it is important to verify the correlation between predictors. Fortunately, all correlation between predictors are less than 40%. Total Gleason score and PSA values have the highest correlation score of 38.60% which is considered as low. This suggests that all variables can be in the same models without raising the multicollinearity issues. However, to make sure, VIF scores will confirm if there are multicollinearity issues after the modeling process. Therefore, we will go ahead and build our model using four variables including results of digital rectal exam ("dpros"), detection of capsule involvement ("dcaps"), PSA value ("psa"), and total Gleason score ("gleason").

Model Fitting/Inferences: After four variables were discovered to be significant, logistic regressions will be used in model development process. Bootstrapping method is a common method for variables selection using random partitions. Here, the models are generated using a combination of each variable using different random partition of training and testing data. An average AUC and AIC of each model was computed after to compare. Table A3 demonstrates the top 10 models with the highest AUC scores. Models with three variables including results of digital rectal exam, PSA values, and total Gleason scores seems to be the best model with overall AUC above 80% and low AIC scores compared to other models. Therefore, the model development process will continue using the three variables mentioned previously with a train set of 70% and hold-out test set of 30%.

Before finalizing the model, all combinations of interaction terms were added to the model with three variables including results of digital rectal exam, PSA value, and total Gleason score. The results of this model are shown in Table A4. The coefficients of all three variables seem to increase in the model with interaction terms. This proves that the model with interaction terms' coefficient has steeper slopes than the model without interaction terms (see Table 1). However, p-values for all interaction terms are quite high suggesting that interaction terms are not needed in the model. To further prove this argument, a stepwise model selection was run using AIC as a validation metric. The best model appears to be the model with the original three variables: results of digital rectal exam, PSA value, and total Gleason score. As seen in Table 1, the final model's coefficients are positive consistent with the exploratory analysis. In addition, all variables in this model have

low p-values of less than 0.05 and odd ratios' confident intervals do not contain 1 suggesting all variables are significant. Lastly, this model has a low AIC of 248.56, high AUC of 0.82, and high accuracy of 0.77 consistent with high probability of producing precise predictions.

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 the final model with three variables: results of digital rectal exam (dpros), PSA value (psa), and total Gleason score (gleason). The columns represent the feature names, coefficient estimated values, standard errors, test statistics, significant p-values, odd ratios, and 95 percent lower and upper confident interval of odd ratios, respectively from left to right.

To access how good the model fit, predictions were generated using the hold-out test set. Figure 3.1 depicts a lift chart of empirical and indicated likelihood of penetration of prostate capsule. Here, the predicted proportion were sorted from smallest to largest, divided into 10 equal buckets, and assigned a decile number from 1 to 10. The lift chart illustrates that the average predicted proportion of each bucket are similar to the average empirical values. This implies that the model is predicting precisely.
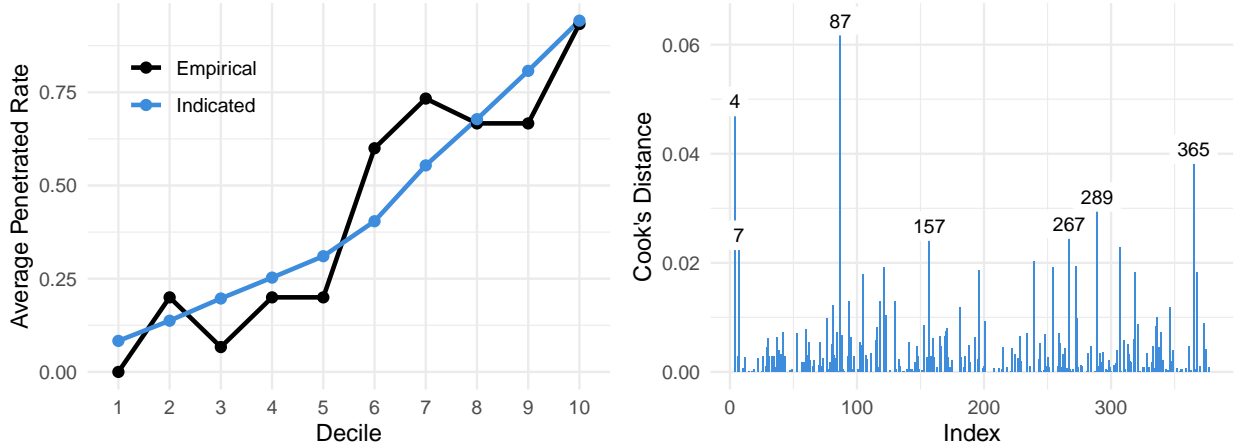
Figure 3: 1 (Left): Lift chart of empirical (black) and indicated (blue) penetration rate sorted from smallest to largest and assigned by decile, and 2 (Right): Cook's Distance plot with labeled outlier observations in blue.

Examining the quality of the model is one of the most important steps in a data analysis. First, multicollinearity would not be an issue in this analysis since VIF scores are smaller than 2 for all the variables in the model. Secondly, Figure 3.2 shows the Cook's distance of each observation point. A few observations did not meet the cutoff point of 0.02 including rows 4, 7, 87, 121, 157, 196, 239, 254, 267, 272, 289, 307, and 365. These points influence all fitted values. In other words, these subjects with row number mentioned previously have huge impacts on the chances of getting

penetrated of prostate capsule.

The studentized residuals are also very essential to validate while accessing the quality of the model. Figure 4.1 portrays a residuals plot of the model. The plot is color coded by two section, one for underestimate points and one for overestimate subjects. The number of overestimated values and overestimated values are approximately the same which constructed a mean value for residuals of approximately 0, or -0.07 to be exact. Furthermore, there are no patterns to be detected in the residuals consistent with constant variance. Most residuals are between -2 and 2 which is ideal for all residual values. However, there are a few points that evidently have large residuals including rows 87, 239, and 289. This indicates that these three subjects are outliers.
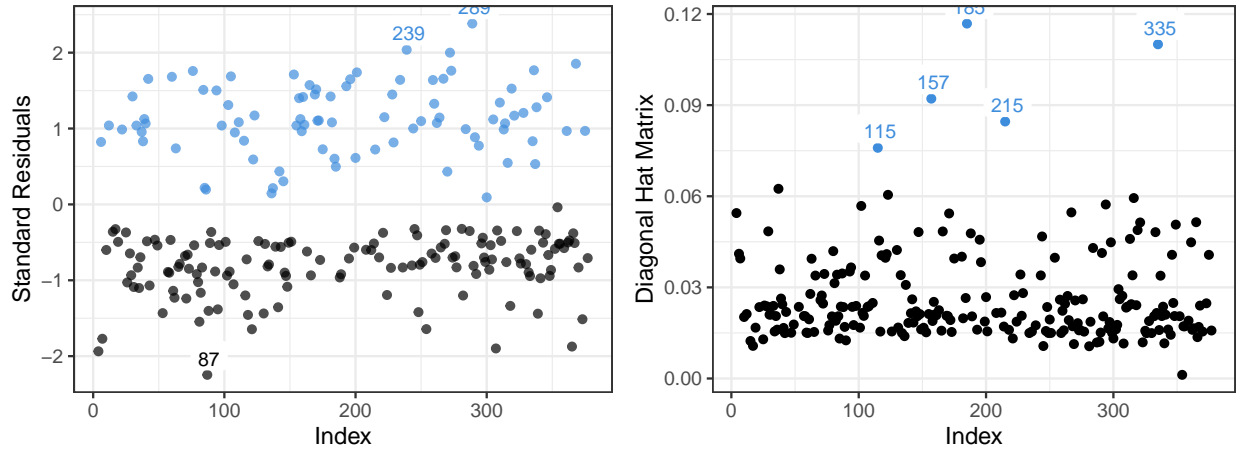

Figure 4: 1 (Left): Studentized residuals plot with blue color are positive residuals (overestimate values) and black are negative residuals (underestimate values), and 2 (Right): Diagonal hat matrix plot with influential points colored in blue.

Last of all, large hat values show potential outlying observations with respect to each predictor. As seen in Figure 4, rows 115, 215, 185, and 335 have higher hat values than others. These subjects potentially have high/low values consisted in one or more of their predictors. To further access the outliers and influential points, Table A5 was constructed to show the details of each observation that were detected as outliers or influential points. Most subjects noticeably have really high PSA scores compared to the mean PSA scores of 15.26. However, all subjects were decided to be kept in the model since there are no valid explanations to eliminate them from the model.

After completing model development and validation, the model is ready for interpretation. All interpretation values used in this section are from Table 1. The results of digital exam have great impact on penetration. For instant, a patient with a unilobar nodule (left) would be 2.28 times or 128% riskier than a patient without a nodule. A person with a unilobar nodule (right) would be 3.69 times or 269% more likely to have their tumor penetrated with prostate capsule. Lastly, a person with bilobar nodule would be more likely to be diagnosed with prostate cancer 3.60 times higher compared to patients with no nodule. In addition, if a patient's PSA score increases by 1 mg/ml of blood, on average, the patient will have 4% increase in the risk of having a prostate cancer. Last but not least, having a low Gleason score would decrease the chances of having cancer since an increase of 1-unit in Gleason score would cause the patient to be 2.33 times riskier.

Conclusion: Generally, there are protocols for doctors and nurses to follow when it comes to important diagnosis like prostate cancer. Manually, it is quite difficult to avoid human errors when providing a diagnosis to a patient. With this model, health professionals can be able to minimize predicting errors and generate the likelihood of a male patients getting prostate cancer more efficiently. The three main factors that change the chance of a tumor has penetrated of prostate capsule are the results of digital rectal exam, PSA score, and total Gleason score. Penetration rates are likely to increase as the results of digital rectal exam are in a critical nodule, PSA values are higher, and total Gleason scores are larger. On the other hand, having no nodule in the results of digital rectal exam, low PSA scores, and low Gleason scores would cause the likelihood of penetration to be very low. This leads to lower risk of having prostate cancer. With an AUC of 0.82 and accuracy of 0.77, we can consider this model to be the best model that can provide the likelihood of penetration rate in male patients. There are many limitations for this model. First, the sample size of this dataset is relatively small. The subset of this small sample size could cause the results to be not as accurate as the full dataset. If we have access to the full dataset from the Ohio State University Comprehensive Cancer Center, then our results might be different. In addition, there were only six predators provided to use in the model. If there are more variables, the model could provide better results. For example, we can consider genetic as a potential a factor that cause the risk of having prostate cancer in the family of the patient to improve the current model.

Appendix A: Supplemental Tables

Table 2: Summary statistics for independent features

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
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

	random_partition	nrow_train	nrow_test	features	AIC	auc
1	0.600	226	151	dpros,psa,gleason,dcaps	239.163	0.816
2	0.700	263	114	dpros,psa,gleason	257.498	0.812
3	0.600	226	151	dpros,gleason	245.001	0.810
4	0.800	301	76	dpros,psa,gleason	276.720	0.810
5	0.800	301	76	dpros,psa,gleason,dcaps	276.583	0.809
6	0.900	339	38	dpros,psa,gleason	296.099	0.809
7	0.700	263	114	dpros,psa,gleason,dcaps	256.692	0.809
8	0.900	339	38	dpros,psa,gleason,dcaps	295.629	0.807
9	0.700	263	114	dpros,gleason	262.166	0.805
10	0.800	301	76	dpros,gleason	282.050	0.804

Table 3: Iteration log of model bootstrapping process. Showing the top 10 models that have the highest AUC scores. The columns represent random partition for train set, number of rows in the training data, number of rows in the testing data, predictors, AIC scores, and AUC scores, respectively from left to right.

	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: Summary regression of the final model with all combination of interaction terms. The columns represent the variables, coefficient estimated values, standard errors, test statistics, and significant p-values, respectively from left to right.

	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: Table of all potential outlier and influential values. The columns represent the tumor penetration of prostatic capsule, results of digital rectal exam, PSA scores, total Gleason scores, and row number of each subject, respectively from left to right.

Appendix B: R Code

```r
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
```

```r
#' Simple binomial regression to run bootstrap for variable selection
#'
#' @param rp random partition
#' @param xvars independent variable/s
#' @param nrounds number of times running the model
#'
#' @return
#' @export
#'
#' @examples
BinaryFit <- function(rp, xvars, nrounds){

  # select varaibles
  df  <- prostate %>% dplyr::select(capsule, all_of(xvars))

  all_auc <- c()
  all_aic <- c()
  all_iteration <- list()
  index <- 1

  for(each_rp in rp){
    for (i in 1:nrounds){
      # split train and test randomly based on proportion
      n <- floor(each_rp * nrow(df))

      train_ind <- sample(seq_len(nrow(df)), size = n)

      train <- df[train_ind, ]

      test <- df[-train_ind, ]

      # build model
      fit <- glm(capsule ~ ., data = train, family = "binomial")

      # predict
      preds <- predict(fit, newdata = test, type = "response")

      test$preds <- preds

      # Validate AUC
      test_roc <- pROC::roc(response = test$capsule, predictor = test$preds)

      auc <- pROC::auc(test_roc)[[1]]

      all_auc <- c(all_auc, auc)
      all_aic <- c(all_aic, fit$aic)
    }
    # iteration log
    all_iteration[[index]] <- data.frame(iteration_date = Sys.time(),
                                         model_type = "glm",
                                         distibution = "binomial",
                                         random_partition = each_rp,
                                         nrow_train = nrow(train),
                                         nrow_test = nrow(test),
                                         nrounds = nrounds,
                                         n_features = length(xvars),
```

```r
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
```

```r
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
```

```r
#### Model Development ####

## parameter for bootstrapping

xvars <- c("dpros", "psa", "gleason", "dcaps")

rp_par <- c(0.6, 0.7, 0.8, 0.9)          # random partition parameters

nrounds <- 10                            # number of rounds per model

## Bootstrapping

# initiate print list
print_list <- list()

# loop
for (n_features in 1:4){
  results <- LoopAllVars(rp = rp_par,
                         nrounds = nrounds,
                         list_of_xvars = xvars,
                         number_of_xvars = n_features)
  print_list[[n_features]] <- results
}

# bind iteration log into a dataframe
iteration_log <- print_list %>% bind_rows()

# after running the iteration logs, models with variables dpros, psa, and gleason
# has the high AUC and low AIC, using train set of 60% prop and test set of 40%


#### Select Model after bootstrap ####

## Split data to train/test set using random partition with proportion .7/.3
set.seed(1234)

n <- floor(.6 * nrow(prostate))

train_ind <- sample(seq_len(nrow(prostate)), size = n)

train <- prostate[train_ind, ]

test <- prostate[-train_ind, ]


## start with initial fit using 4 variables
fit1 <- glm(capsule ~ dpros + psa + gleason,
            family = binomial(link = logit),
            data = train)


## Checking for Interaction term

fit2 <- glm(capsule ~ .*.,
            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   geom_line(size = 1) +
315   geom_point(size = 2) +
316   theme_minimal() +
317   labs(x = "Percentile",
318        y = "Average Proportion Penetrated") +
319   theme(legend.title = element_blank()) +
320   scale_color_manual(values = c("black", "#3e8ddd"))

321

322 ## compute AIC and AUC of final model
323 aic <- fit_final$aic
324 test_roc <- pROC::roc(response = test$capsule, predictor = test$preds)
325 test_auc <- as.data.frame(pROC::auc(test_roc))

326

327 tidy <- broom::tidy(fit_final) %>%
328   mutate(OR = exp(fit_final$coefficients)) %>%
329   cbind(exp(confint(fit_final))) %>%
330   modify_if(is.numeric, round, 4)

331

332 ## wald test
```

```r
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 proportial 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     .cooksd > 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 = .cooksd)) +
377   geom_bar(stat = "identity", fill = "#3e8ddd", col = "white") +
378   labs(y = "Cook's Distance") +
379   theme_minimal() +
380   geom_label(data = diag %>% filter(.cooksd > 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() +
```

```r
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 multicolinearity
416
417
418 #### Find influential point in data ####
419
420 inf_points <- (diag %>%
421                  filter(.std.resid > 2 | .std.resid < -2 | .cooksd > cutoff | .hat >
       .07))$.rownames %>%
422    as.numeric()
423
424 prostate[inf_points, ] %>%
425    as.data.frame() %>%
426    dplyr::select(capsule, dpros, psa, gleason) %>%
427    mutate(rownames = inf_points) %>%
428    arrange(rownames)
```

Listing 1: Appendix of Code