Identifying Key Predictors of Smoking Abstinence through Lasso Regression and Best Subset Modeling

Michael Stewart

Abstract

This project, in collaboration with Dr. George Papandonatos from Brown University's Biostatistics Department, explores factors impacting smoking cessation in adults with major depressive disorder (MDD). Individuals with MDD often smoke more heavily, face greater nicotine dependence, and experience more severe withdrawal, making cessation efforts particularly challenging. We evaluated the smoking cessation drug varenicline alongside behavioral therapies tailored to address these specific challenges. Data came from a randomized, placebo-controlled, 2x2 factorial trial comparing behavioral activation for smoking cessation (BASC) to standard treatment (ST), and varenicline to placebo, in a sample of 300 adults with current or past MDD. Earlier findings indicated that BASC did not significantly outperform ST.

In our analysis, we examined baseline characteristics as potential moderators of treatment effects on end-of-treatment abstinence, applying Lasso regression models to uncover key predictors. Results showed that participants on placebo were significantly less likely to quit smoking compared to those on varenicline. Higher tobacco dependence scores (FTCD) also predicted lower abstinence rates, highlighting the challenge of quitting for individuals with stronger dependencies. Non-Hispanic White participants were marginally more likely to achieve abstinence, and the Nicotine Metabolism Ratio (NMR) had a modest association with outcomes.

For model evaluation, we used AUC to assess the discrimination of the predictions. Overall, these findings suggest that quitting interventions for MDD populations may benefit from an integrated approach combining pharmacologic and behavioral support.

1. Introduction

Quitting smoking presents unique challenges for individuals with major depressive disorder (MDD), who often face higher nicotine dependence and more intense withdrawal symptoms than the general population. These additional barriers make it crucial to understand which baseline factors might predict or influence treatment success. Demographic characteristics, depression severity, and tobacco dependence levels are some variables that could shape treatment outcomes, but their roles in smoking cessation for MDD populations are not fully clear.

To explore this, we used Lasso regression, which applies variable selection and shrinkage, and is well-suited to handling datasets with many predictors. By applying a penalty to reduce overfitting, Lasso identifies a comprehensive set of potential predictors without making the model overly complex.This study aims to develop a robust predictive model using Lasso regularization, multiple imputation for missing data, and an emphasis on exploring potential interaction terms. Bootstrapping was used to calculate confidence intervals for the model's estimates, and odds ratios were interpreted to provide insights into the effects of predictors on the outcome variable, abstinence.

2 Data Description and Preprocessing

2.1 Data Source and Study Design

The trial enrolled 300 adult smokers with current or past MDD, who were randomly assigned to one of four treatment groups: varenicline with behavioral activation for smoking cessation (BASC), varenicline with standard treatment (ST), placebo with BASC, or placebo with ST. BASC is a behavioral approach that aims to increase engagement in rewarding, non-smoking-related activities to counteract depressive symptoms, while ST focuses on general smoking cessation counseling without targeted behavioral activation. This design allowed for comparisons of the effects of varenicline versus placebo and BASC versus ST, both individually and in combination.

Table 1 presents a summary of the baseline variables for study participants, including demographic factors (age, sex, race/ethnicity), nicotine dependence indicators (such as the FTCD score and Nicotine Metabolism Ratio [NMR]), and psychological variables (including the Beck Depression Depression Inventory [BDI] score and readiness to quit).

Table 1: Participant Characteristics by Treatment Arm and Overall Sample, with Missing Data

	Treatment Groups				
Characteristic	**Overall**, N = 300	**BASC + Placebo**, N = 68	**BASC + Varenicline**, N = 83	**ST + Placebo**, N = 68	**ST + Varenicline**, N = 81
Age, Mean (SD)	50 (13)	51 (14)	50 (13)	50 (11)	49 (13)
Sex, n (%)					
Female	165 (55%)	38 (56%)	44 (53%)	39 (57%)	44 (54%)
Male	135 (45%)	30 (44%)	39 (47%)	29 (43%)	37 (46%)
Non-Hispanic White, n (%)	105 (35%)	24 (35%)	34 (41%)	22 (32%)	25 (31%)
Black, n (%)	157 (52%)	37 (54%)	37 (45%)	40 (59%)	43 (53%)
Hispanic, n (%)	18 (6.0%)	5 (7.4%)	4 (4.8%)	4 (5.9%)	5 (6.2%)
Income, n (%)					
$20,000–35,000	68 (23%)	16 (24%)	17 (20%)	14 (21%)	21 (26%)
$35,001–50,000	46 (15%)	8 (12%)	13 (16%)	14 (21%)	11 (14%)
$50,001–75,000	38 (13%)	12 (18%)	12 (14%)	8 (12%)	6 (7.4%)
Less than $20,000	110 (37%)	25 (37%)	30 (36%)	26 (38%)	29 (36%)
More than $75,000	35 (12%)	6 (8.8%)	10 (12%)	6 (8.8%)	13 (16%)
Unknown	3 (1.0%)	1 (1.5%)	1 (1.2%)	0 (0%)	1 (1.2%)
Education Level, n (%)					
College graduate	91 (30%)	19 (28%)	29 (35%)	17 (25%)	26 (32%)
Grade school	1 (0.3%)	1 (1.5%)	0 (0%)	0 (0%)	0 (0%)
High school graduate or GED	76 (25%)	23 (34%)	15 (18%)	11 (16%)	27 (33%)
Some college/technical school	116 (39%)	22 (32%)	32 (39%)	38 (56%)	24 (30%)
Some high school	16 (5.3%)	3 (4.4%)	7 (8.4%)	2 (2.9%)	4 (4.9%)
FTCD Score, Mean (SD)	5 (2)	5 (2)	5 (2)	5 (2)	5 (2)
Unknown	1	0	0	1	0
FTCD Score (5 mins), n (%)	138 (46%)	32 (47%)	33 (40%)	35 (51%)	38 (47%)
BDI Score, Mean (SD)	19 (11)	19 (12)	18 (11)	18 (11)	20 (12)
Cigarettes per day, Mean (SD)	15 (8)	16 (9)	16 (9)	15 (7)	14 (7)
Craving Total, Mean (SD)	7 (4)	7 (4)	7 (4)	7 (4)	7 (3)
Unknown	18	1	3	8	6
Hedonic Sum (Negative), Mean (SD)	23 (20)	23 (20)	23 (19)	21 (20)	23 (19)
Hedonic Sum (Positive), Mean (SD)	25 (19)	28 (22)	22 (17)	27 (20)	25 (19)
Shaps Score, Mean (SD)	2 (3)	2 (3)	2 (3)	3 (3)	2 (3)
Unknown	3	2	0	1	0
Other Diagnoses, n (%)	133 (44%)	35 (51%)	30 (36%)	28 (41%)	40 (49%)
Antidepressant Medication, n (%)	82 (27%)	28 (41%)	24 (29%)	15 (22%)	15 (19%)
Current Major Depression Episode, n (%)	147 (49%)	32 (47%)	40 (48%)	31 (46%)	44 (54%)
Nicotine Metabolism Ratio, Mean (SD)	0.36 (0.23)	0.34 (0.18)	0.38 (0.25)	0.37 (0.27)	0.36 (0.21)
Unknown	21	7	3	2	9
Only Menthol, n (%)	178 (60%)	40 (59%)	48 (59%)	43 (64%)	47 (58%)
Unknown	2	0	1	1	0
Readiness to Quit, Mean (SD)	7 (1)	7 (1)	7 (1)	7 (1)	7 (1)
Unknown	17	4	5	4	4

[1] Mean (SD) for continuous variables; n (%) for categorical variables.

2.2 Handling Missing Data

To address missing data, we first examined the extent and patterns of missingness across baseline variables. As summarized in **Table 1**, certain variables exhibited higher rates of missing data, including income, FTCD score (a measure of nicotine dependence), craving total, anhedonia score, Nicotine Metabolism Ratio (NMR), exclusive menthol use, and readiness to quit.

To address missing data, we used multiple imputation, generating five imputed datasets. We chose multiple imputation over complete case analysis to address missing data. A complete case analysis would have removed approximately 20% of the sample, reducing statistical power and potentially introducing bias if the missingness was not completely random (MCAR).

Additionally, due to the unique value of one participant who only completed grade school, this case was removed from the analysis as it was causing instability in the Best Subset model.

After analyzing each imputed dataset separately, we applied Rubin's Rules to pool the results to get reliable, stable estimates of coefficients and standard errors.

2.3 Encoding Variables and Transformations

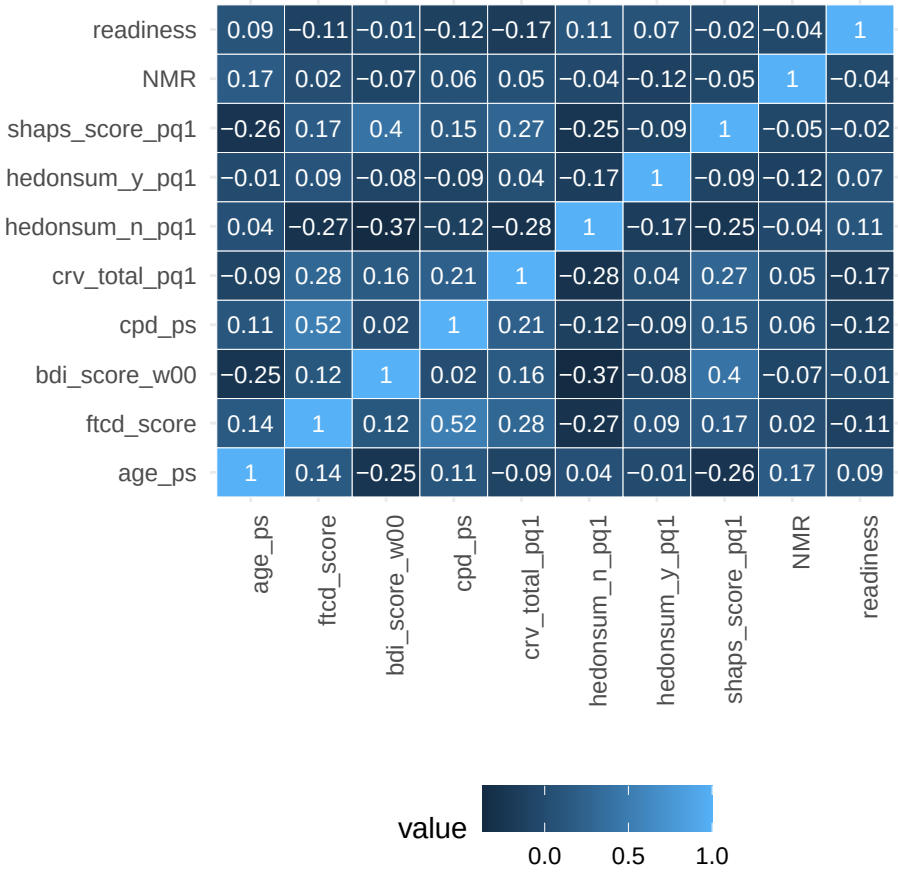
To ensure better model performance and interpretability, the categorical variable for education was restructured by combining the "high school or less" levels into a single category, creating a more balanced distribution across groups.

3. Exploratory Analysis

The exploratory data analysis phase involved several analyses to understand variable relationships and assess their significance as predictors of smoking cessation.

The correlation heatmap (**Figure 1**) offered a visual representation of relationships between continuous variables. We wanted to explore if any variables had some collinearity. This is important because multi-collinearity can impact the stability of variable selection in Lasso and Best Subset modeling. The highest Pearson correlation was observed between FTCD score and cigarettes per day (cpd_ps), with a correlation of 0.51, indicating a moderate positive association. Another notable correlation was between bdi_score_w00 (a measure of baseline depression) and shaps_score_pq1 (anhedonia score), with a correlation of 0.41, suggesting a moderate association between depression and anhedonia. Overall, multicollinearity was low enough to support the inclusion of all predictors in the Lasso model without significant issues.

Figure 1: Correlation Heatmap

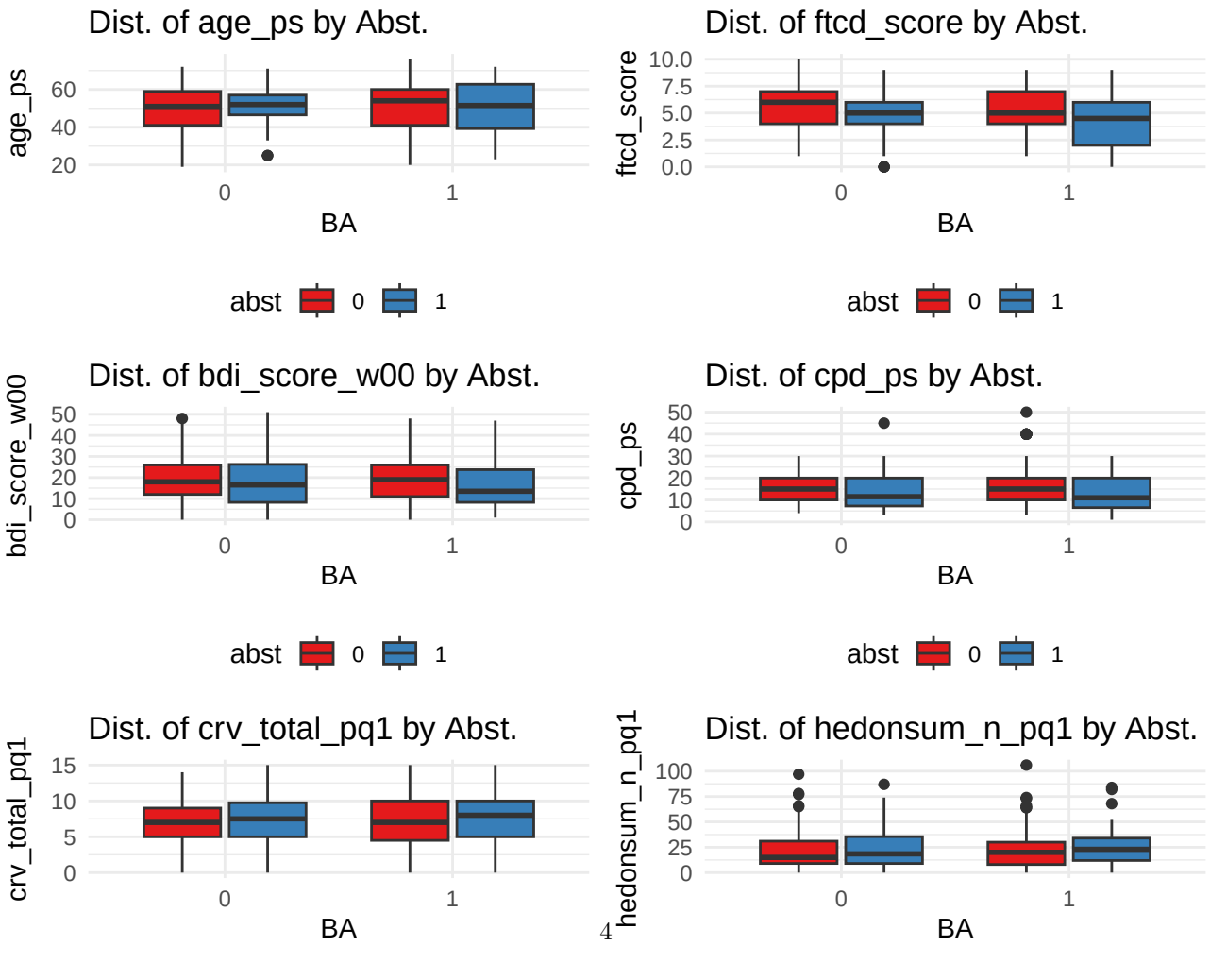


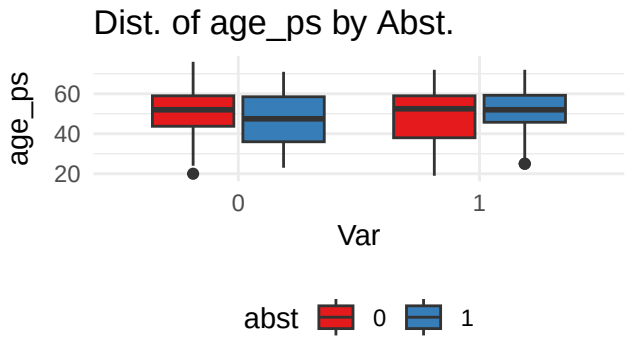
We conducted a chi-square test (**Table 2**) to examine associations between categorical variables, including treatment arms, race/ethnicity, education levels, and income brackets. Significant associations emerged, such as between education and income levels ($\chi^2 = 88.52$, $p < 0.001$) and between race/ethnicity and menthol cigarette use ($\chi^2 = 52.89$, $p < 0.001$).

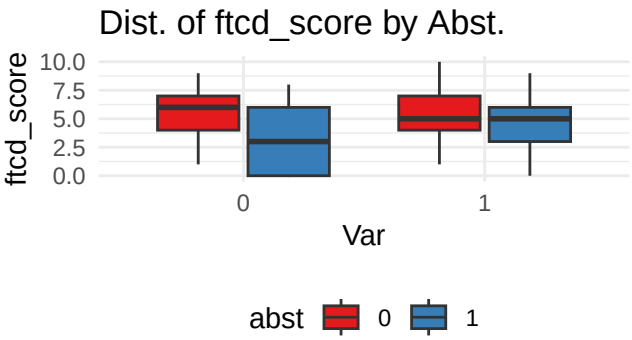
Although the Lasso and Best Subset models independently selected variables based on predictive power, these associations provided valuable context for interpreting results. For example, if demographic or socioeconomic factors, such as race or income, appeared as important predictors in our models, knowing that these factors are associated with other variables (like menthol cigarette use) helped us understand why they might influence smoking cessation outcomes. This context allows us to interpret the results with an understanding of the broader social or behavioral patterns that may impact abstinence success.

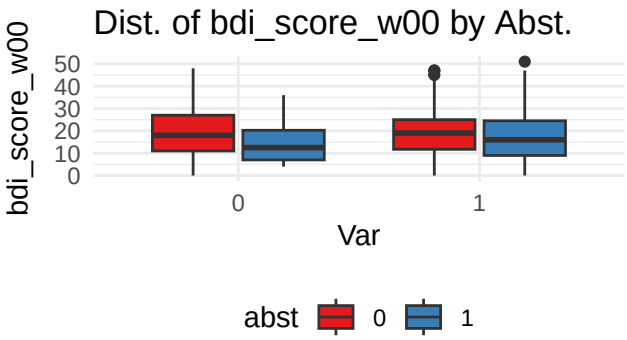
Table 2: Significant Chi-Square Test Results

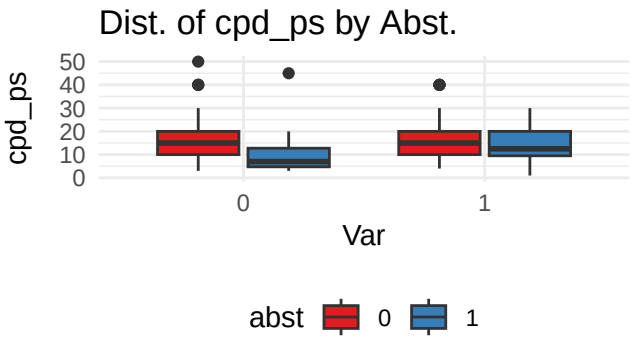
Variable 1	Variable 2	X-squared	df	p-value
Black	NHW	174.14	1	< 2e-16
edu	inc	82.89	8	1.28e-14
Black	Only.Menthol	69.77	1	< 2e-16
NHW	Only.Menthol	53.88	1	2.13e-13
inc	Only.Menthol	36.66	4	2.12e-07
Black	inc	35.59	4	3.51e-07
Black	edu	33.06	2	6.62e-08
edu	Only.Menthol	26.26	2	1.98e-06
mde_curr	otherdiag	24.59	1	7.09e-07
inc	NHW	24.42	4	6.57e-05
abst	Var	21.86	1	2.94e-06
edu	NHW	18.36	2	1.03e-04
inc	mde_curr	15.24	4	4.22e-03
Black	Hisp	11.35	1	7.56e-04
Hisp	NHW	8.74	1	3.11e-03
antidepmed	BA	7.02	1	8.06e-03
Hisp	Only.Menthol	6.78	1	9.22e-03
edu	ftcd.5.mins	6.41	2	4.05e-02
NHW	sex_ps	6.22	1	1.26e-02
abst	NHW	5.73	1	1.67e-02
Black	sex_ps	4.52	1	3.35e-02
antidepmed	NHW	4.49	1	3.41e-02

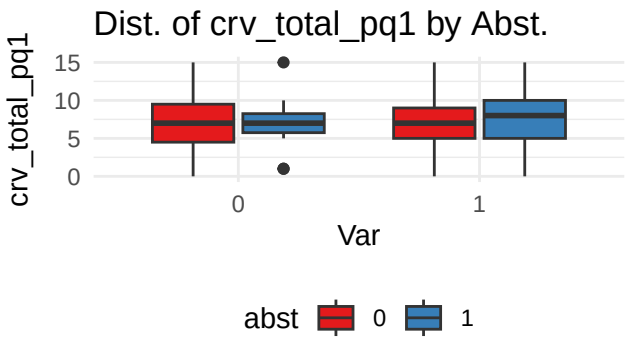


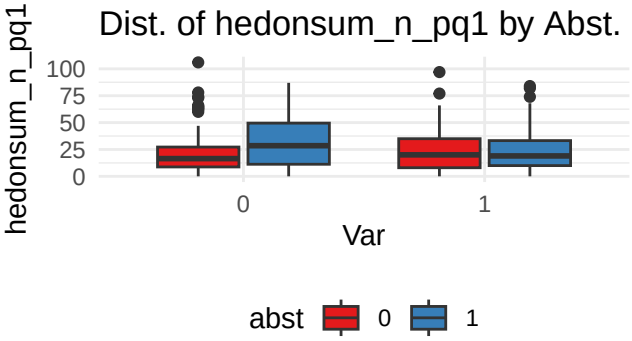

Dist. of age_ps by Abst.
age_ps
60
40
20
0
1
Var
abst
0
1

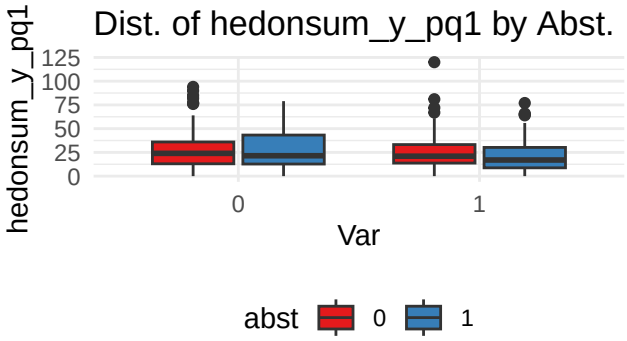

Dist. of ftcd_score by Abst.
ftcd_score
10.0
7.5
5.0
2.5
0.0
0
1
Var
abst
0
1

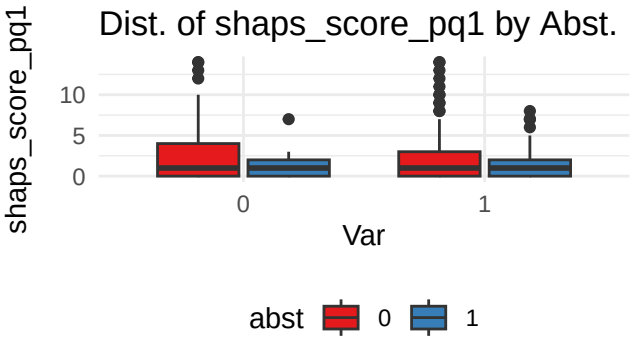

Dist. of bdi_score_w00 by Abst.
bdi_score_w00
50
40
30
20
10
0
0
1
Var
abst
0
1

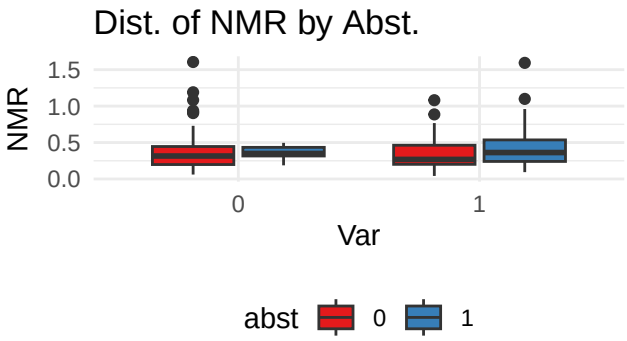

Dist. of cpd_ps by Abst.
cpd_ps
50
40
30
20
10
0
0
1
Var
abst
0
1

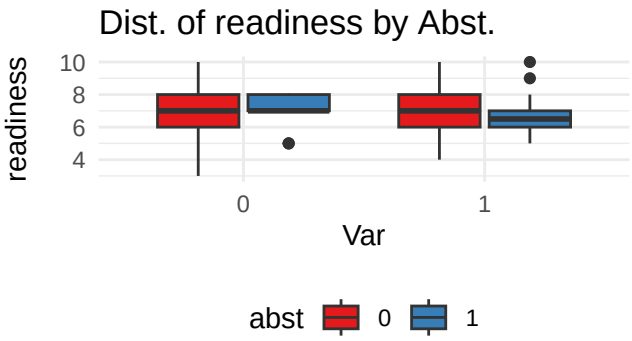

Dist. of crv_total_pq1 by Abst.
crv_total_pq1
15
10
5
0
0
1
Var
abst
0
1


Dist. of hedonsum_n_pq1 by Abst.
hedonsum_n_pq1
100
75
50
25
0
0
1
Var
abst
0
1


Dist. of hedonsum_y_pq1 by Abst.
hedonsum_y_pq1
125
100
75
50
25
0
0
1
Var
abst
0
1

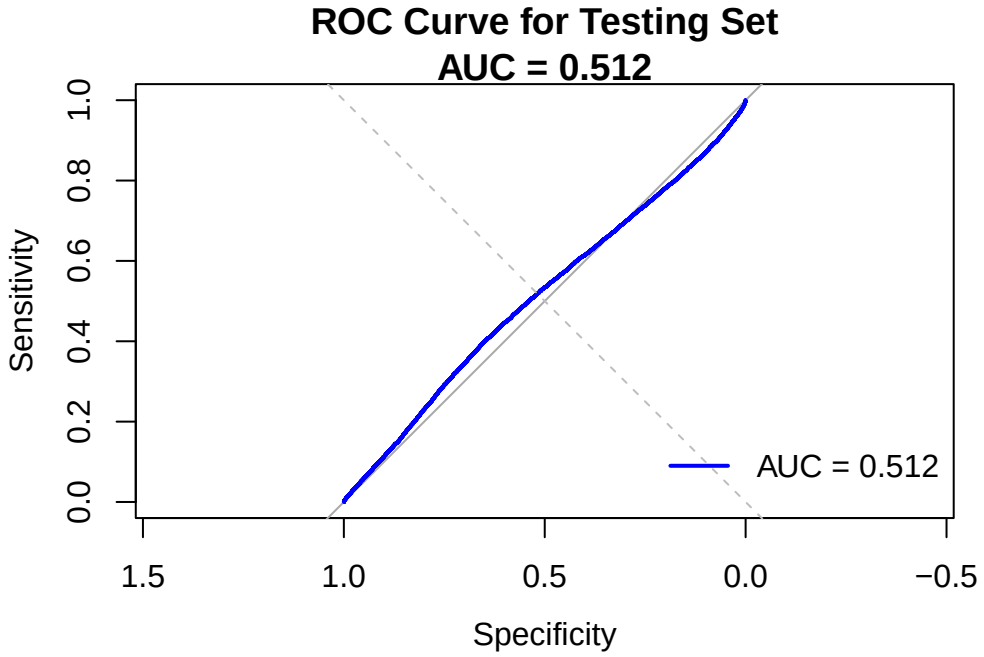

Dist. of shaps_score_pq1 by Abst.
shaps_score_pq1
10
5
0
0
1
Var
abst
0
1


Dist. of NMR by Abst.
NMR
1.5
1.0
0.5
0.0
0
1
Var
abst
0
1


Dist. of readiness by Abst.
readiness
10
8
6
4
0
1
Var
abst
0
1

Based on the boxplot visualizations, several factors were identified for inclusion as interaction terms in the model. The boxplots displayed how various continuous variables, such as age, BDI score, readiness, and others, varied by the binary outcome of abstinence (abst). The inclusion of interaction terms like readiness:BA, ftcd_score:Var, cpd_ps:Var, hedonsum_n_pq1:Var, and hedonsum_y_pq1:Var was informed by these visual patterns. These interactions highlighted that the relationship between certain predictors and abstinence varied depending on other variables. For example, the readiness and BA score showed distinct distributions across the abstinence groups, prompting the inclusion of an interaction term between these two variables. Similarly, the variables ftcd_score, cpd_ps, and hedonsum_n_pq1 demonstrated a significant difference between abstainers and non-abstainers, suggesting that their combined effects might be worth exploring in the model. The inclusion of Black:Only.Menthol was based on a Chi-square test result that suggested a significant interaction between race and menthol usage in relation to abstinence, making it an important variable to include. This approach helped to capture more complex relationships between variables that could be missed by considering them independently, ultimately improving the robustness of the analysis.

```
=== AUC Summary ===
Training Set AUC: Mean = 0.8794 | SD = 0.0559
Testing Set AUC:  Mean = 0.6953 | SD = 0.062
```



4. Methods

For our main analyses, we applied Lasso. This model was chosen to identify key predictors of abstinence and provide robust variable selection through different methods.

4.1 Lasso Regression

To identify key predictors of smoking abstinence, we applied Lasso regression because of its ability to perform variable selection by shrinking the coefficients of less important predictors toward zero. We began by splitting each imputed dataset into an 70/30 train-test split. We fit a Lasso model for each imputed dataset, performing 5-fold cross-validation to select the optimal penalty parameter, .We hose the smallest value as our optimal parameter. This process was repeated across all five imputed datasets, resulting in five sets of coefficient estimates.

5. Results

The results from the analysis provide several insights into the predictors of the outcome. The intercept had a high odds ratio (OR) of 6.19 (95% CI: 0.01 - 27804492.51), suggesting substantial baseline variation in the outcome when

all predictors are at their reference levels. However, the wide confidence interval highlights the uncertainty of this estimate. Predictors such as "NHW1" (OR = 3.30; 95% CI: 0.36 - 892.01) and "ftcd.5.mins1" (OR = 1.85; 95% CI: 0.40 - 24.15) showed higher odds of the outcome, though their wide confidence intervals suggest limited precision in the estimates. Interaction terms like "Black1:Only.Menthol1" (OR = 1.98; 95% CI: 0.06 - 387.37) also indicated an increased likelihood of the outcome but with considerable uncertainty. These predictors might suggest potential effects, but the variability in confidence intervals makes it difficult to draw strong conclusions.

On the other hand, several predictors were associated with lower odds of the outcome. For example, "eduSome College/Technical School" (OR = 0.599; 95% CI: 0.07 - 3.25) and "ftcd_score" (OR = 0.569; 95% CI: 0.12 - 1.01) were linked to a decreased likelihood of the outcome compared to their reference groups. Notably, "BA1" had an OR of 0.015 (95% CI: 0.00 - 1.00), suggesting a strong negative association with the outcome. However, the wide confidence intervals for many variables indicate uncertainty in these estimates, potentially due to small sample sizes, multicollinearity, or other model limitations. This highlights the need for further refinement and additional data to improve the stability of the findings. Overall, while certain predictors appear meaningful, the uncertainty limits their interpretability without additional validation.

The ROC curves for the Lasso model across the five imputed datasets indicate a consistent performance, with each curve closely following a similar trajectory. The mean AUC of 0.721 and a standard deviation of 0.009 suggest moderate predictive accuracy and low variability between imputations.

These results indicate that the Lasso model has stable sensitivity and specificity across the different datasets, with its predictive ability being relatively consistent regardless of data imputation. The small standard deviation also reinforces the reliability of the AUC estimate across multiple imputations, providing confidence in the model's predictive power for the chosen variables.

The ROC curves for the Best Subset model across the five imputed datasets indicate consistent performance, with an average AUC of 0.728 and a low standard deviation (0.004). This suggests that the model's predictive accuracy and discrimination is stable and reliable across multiple imputations.

6. Conclusion and Limitations

Our analysis identified treatment type and nicotine dependence (ftcd_score) as key predictors of smoking cessation in adults with MDD. Both Lasso and Best Subset models consistently highlighted these variables, aligning with the correlation heatmap, which showed their association with cessation outcomes. However, chi-square tests emphasized demographic factors like race, education, and menthol use rather than treatment type, suggesting these characteristics are related to smoking behavior but may have limited predictive power for cessation success.

This project has some limitations. Lasso's regularization can exclude weaker predictors, potentially missing subtle effects, while Best Subset risks overfitting by testing all combinations. The calibration plots, particularly for the Best Subset model, also revealed significant deviations that suggest overfitting. With a relatively high number of predictors and a limited sample size, the risk of overfitting was heightened, as small datasets often struggle with generalizability. These limitations suggest that future work might benefit from external validation sets or incorporating non-linear models to improve stability and generalizability.

References

Hitsman, B., Papandonatos, G. D., Gollan, J. K., Huffman, M. D., Niaura, R., Mohr, D. C., Veluz-Wilkins, A. K., Lubitz, S. F., Hole, A., Leone, F. T., Khan, S. S., Fox, E. N., Bauer, A. M., Wileyto, E. P., Bastian, J., & Schnoll, R. A. (2023). Efficacy and safety of combination behavioral activation for smoking cessation and varenicline for treating tobacco dependence among individuals with current or past major depressive disorder: A 2 x 2 factorial, randomized, placebo-controlled trial. Addiction (Abingdon, England), 118(9), 1710–1725. https://doi.org/10.1111/add.16209

Code Appendix

```
knitr::opts_chunk$set(warning = FALSE,
                      message = FALSE,
                      echo = FALSE,
                      fig.align = "center")

library(tidyverse)
library(knitr)
library(kableExtra)
library(mice)
library(corrplot)
library(gtsummary)
library(RColorBrewer)
library(reshape2)
library(gridExtra)
library(glmnet)
library(caret)
library(LOLearn)
library(pROC)

project2 <- read.csv("~/Downloads/project2.csv")


# define treatment categories
project2 <- project2 %>%
  mutate(
    treatment = case_when(
      Var == 1 & BA == 1 ~ "BASC + Varenicline",
      Var == 0 & BA == 1 ~ "BASC + Placebo",
      Var == 1 & BA == 0 ~ "ST + Varenicline",
      Var == 0 & BA == 0 ~ "ST + Placebo"
    )
  )

# recode income, education, and sex levels
project2 <- project2 %>%
  mutate(
    income = case_when(
      inc == 1 ~ "Less than $20,000",
      inc == 2 ~ "$20,000-35,000",
      inc == 3 ~ "$35,001-50,000",
      inc == 4 ~ "$50,001-75,000",
      inc == 5 ~ "More than $75,000",
      TRUE ~ "Unknown"
    ),
    education = case_when(
      edu == 1 ~ "Grade school",
      edu == 2 ~ "Some high school",
      edu == 3 ~ "High school graduate or GED",
      edu == 4 ~ "Some college/technical school",
      edu == 5 ~ "College graduate",
      TRUE ~ "Unknown"
    ),
    sex = case_when(
      sex_ps == 1 ~ "Male",
      sex_ps == 2 ~ "Female",
```

```
      TRUE ~ "Unknown")
  )

# summary table with baseline variables
baseline_table <- project2 %>%
  select(
    treatment, age_ps, sex, NHW, Black, Hisp, income, education,
    ftcd_score, ftcd.5.mins, bdi_score_w00, cpd_ps, crv_total_pq1,
    hedonsum_n_pq1, hedonsum_y_pq1, shaps_score_pq1, otherdiag,
    antidepmed, mde_curr, NMR, Only.Menthol, readiness
  ) %>%
  tbl_summary(
    by = treatment,
    label = list(
      age_ps ~ "Age",
      sex ~ "Sex",
      NHW ~ "Non-Hispanic White",
      Black ~ "Black",
      Hisp ~ "Hispanic",
      income ~ "Income",
      education ~ "Education Level",
      ftcd_score ~ "FTCD Score",
      ftcd.5.mins ~ "FTCD Score (5 mins)",
      bdi_score_w00 ~ "BDI Score",
      cpd_ps ~ "Cigarettes per day",
      crv_total_pq1 ~ "Craving Total",
      hedonsum_n_pq1 ~ "Hedonic Sum (Negative)",
      hedonsum_y_pq1 ~ "Hedonic Sum (Positive)",
      shaps_score_pq1 ~ "Shaps Score",
      otherdiag ~ "Other Diagnoses",
      antidepmed ~ "Antidepressant Medication",
      mde_curr ~ "Current Major Depression Episode",
      NMR ~ "Nicotine Metabolism Ratio",
      Only.Menthol ~ "Only Menthol",
      readiness ~ "Readiness to Quit"
    ),
    statistic = c(all_continuous() ~ "{mean} ({sd})",
                  all_categorical() ~ "{n} ({p}%)"),
    type = list(readiness ~ "continuous"),
    missing = "ifany"
  ) %>%
  add_overall() %>%
  add_stat_label() %>%
  modify_spanning_header(
    all_stat_cols() ~ "**Treatment Groups**"
  ) %>%
  modify_footnote(
    all_stat_cols() ~ "Mean (SD) for continuous variables; n (%) for categorical variables."
  ) %>%
  as_kable_extra(
    booktabs = TRUE,
    caption = "Participant Characteristics by Treatment Arm and Overall Sample, with Missing Data"
  ) %>%
  kable_styling(
    latex_options = c("hold_position", "scale_down", "striped"),
    full_width = FALSE,
```

```
    position = "center"
  )

baseline_table


project2 <- project2 %>%
  mutate(
    edu = case_when(
      edu %in% c(1, 2, 3) ~ "High School or Less",  # Combine Grade School, Some HS, and HS Grad
      edu == 4 ~ "Some College/Technical School",
      edu == 5 ~ "College Graduate",
      TRUE ~ "Unknown"
    )
  )

# continuous and categorical variables
cont_vars <- c("age_ps", "ftcd_score", "bdi_score_w00", "cpd_ps",
                     "crv_total_pq1", "hedonsum_n_pq1", "hedonsum_y_pq1",
                     "shaps_score_pq1", "NMR", "readiness")
categorical_vars <- c("abst", "Var", "BA", "inc", "edu", "sex_ps", "NHW", "Black",
                      "Hisp", "ftcd.5.mins", "otherdiag",
                      "antidepmed", "mde_curr", "Only.Menthol")

# make categorical variables factors
project2 <- project2 %>%
  mutate(across(all_of(categorical_vars), as.factor))



# continuous variables and make it a matrix
continuous_vars <- project2 %>%
  select(-id) %>%
  select_if(is.numeric) %>%
  as.matrix()  # Convert to matrix

# correlation matrix
cor_matrix <- cor(continuous_vars, use = "complete.obs")
cor_melted <- melt(cor_matrix)

# correlation heatmap
ggplot(cor_melted, aes(Var1, Var2, fill = value)) +
  geom_tile(color = "white") +
  geom_text(aes(label = round(value, 2)), color = "white", size = 3) +  # Add correlation values as text
  labs(title = "Figure 1: Correlation Heatmap", x = "", y = "") +
  theme_minimal() +
  theme(
    legend.position = "bottom",
    axis.text.x = element_text(angle = 90, hjust = 1),
    plot.title = element_text(hjust = 0.5)
  )



# chi-square test funciton
chi_square <- function(var1, var2, data) {
```

```
  contingency_table <- table(data[[var1]], data[[var2]])
  test_result <- chisq.test(contingency_table)
  data.frame(
    Variable1 = var1,
    Variable2 = var2,
    X_squared = test_result$statistic,
    df = test_result$parameter,
    p_value = test_result$p.value
  )
}

chi_square_results <- expand.grid(var1 = categorical_vars,
                                  var2 = categorical_vars,
                                  stringsAsFactors = FALSE) %>%
  filter(var1 < var2) %>%  # Exclude self-pairs
  rowwise() %>%
  do(chi_square(.$var1, .$var2, data = project2)) %>%  # run for each pair
  bind_rows() %>%
  ungroup()


signif_results <- chi_square_results %>%
  filter(p_value < 0.05) %>%         # keep only signif results
  rowwise() %>%
  mutate(
    pair_key = paste(sort(c(Variable1, Variable2)), collapse = "_")
  ) %>%
  distinct(pair_key, .keep_all = TRUE) %>% # no self pairs or repeats
  select(-pair_key) %>%
  ungroup() %>%
  arrange(desc(X_squared)) %>%
  mutate(
    X_squared = round(X_squared, 2),
    p_value = format.pval(p_value, digits = 3, scientific = TRUE)
  )

kable(signif_results, format = "latex", booktabs = TRUE, align = "lcccc",
      col.names = c("Variable 1", "Variable 2", "X-squared", "df", "p-value"),
      caption = "Significant Chi-Square Test Results") %>%
  kable_styling(latex_options = c("hold_position", "striped"))


library(patchwork)

# Initialize an empty list to store plots for BA################
plot_list <- list()

# Names of the continuous variables
cont_names <- c("age_ps", "ftcd_score", "bdi_score_w00", "cpd_ps",
                "crv_total_pq1", "hedonsum_n_pq1", "hedonsum_y_pq1",
                "shaps_score_pq1", "NMR", "readiness")

# Loop through variables and store plots in the list
for (var in cont_names) {
  p <- ggplot(project2, aes_string(x = "BA", y = var, fill = "abst")) +
    geom_boxplot() +
```

```
    labs(title = paste("Dist. of", var, "by Abst.")) +
    theme_minimal() +
    theme(legend.position = "bottom") +
    scale_fill_brewer(palette = "Set1")

  plot_list[[var]] <- p # Add the plot to the list
}

# Display plots in a grid layout with 2 columns
combined_plot <- wrap_plots(plot_list, ncol = 2)
print(combined_plot)

# Initialize an empty list to store plots for Var ################
plot_list2 <- list()

# Loop through variables and store plots in the list
for (var in cont_names) {
  p2 <- ggplot(project2, aes_string(x = "Var", y = var, fill = "abst")) +
    geom_boxplot() +
    labs(title = paste("Dist. of", var, "by Abst.")) +
    theme_minimal() +
    theme(legend.position = "bottom") +
    scale_fill_brewer(palette = "Set1")

  plot_list2[[var]] <- p2 # Add the plot to the list
}

# Display plots in a grid layout with 2 columns
combined_plot2 <- wrap_plots(plot_list2, ncol = 2)
print(combined_plot2)

load("bootstrap_lasso_results.RData")
project2 <- project2 %>%
  select(-income, -education, -sex, -treatment)  # Remove unwanted columns

# Define the proportion for training set
train_index <- createDataPartition(project2$abst, p = 0.7, list = FALSE)
train_data <- project2[train_index, ]
test_data  <- project2[-train_index, ]

# Identify factor variables in train_data
factor_vars <- setdiff(names(train_data)[sapply(train_data, is.factor)], "abst")


# Data Preparation & Imputation



# Set the number of imputations and bootstraps
M <- 5     # Number of multiple imputations
B <- 200   # Number of bootstrap iterations per imputed dataset


# Define the interaction formula
interaction_formula <- as.formula(paste(
  "abst ~ .",
```

```
  "+ readiness:BA",
  "+ ftcd_score:Var",
  "+ hedonsum_n_pq1:Var",
  "+ cpd_ps:Var",
  "+ Black:Only.Menthol"
))

# Set a seed for reproducibility
set.seed(1)


# Multiple Imputation on Training Data

# Perform Multiple Imputation on training data
# Ensure that 'abst' is not imputed
# pred_matrix_train <- make.predictorMatrix(train_data)
# pred_matrix_train[, "abst"] <- 0  # Do not use 'abst' to impute other variables
# pred_matrix_train["abst", ] <- 0  # Do not impute 'abst' itself
#
# imputed_train <- mice(
#   data            = train_data,
#   m               = M,         # Number of imputations
#   maxit           = 5,         # Number of iterations
#   predictorMatrix = pred_matrix_train,
#   seed            = 1
# )


# Multiple Imputation on Testing Data


# Perform Multiple Imputation on testing data
# Ensure that 'abst' is not imputed
# pred_matrix_test <- make.predictorMatrix(test_data)
# pred_matrix_test[, "abst"] <- 0  # Do not use 'abst' to impute other variables
# pred_matrix_test["abst", ] <- 0  # Do not impute 'abst' itself
#
# imputed_test <- mice(
#   data            = test_data,
#   m               = M,         # Number of imputations (same as training)
#   maxit           = 5,         # Number of iterations
#   predictorMatrix = pred_matrix_test,
#   seed            = 1
# )

#
# # 5. Align Factor Levels Between Imputed Datasets
#
#
# # Initialize lists to store aligned imputed datasets
# aligned_imputed_train <- list()
# aligned_imputed_test  <- list()
#
# # Loop over each imputation to align factor levels
# for (m_i in 1:M) {
#   cat("Aligning Factor Levels for Imputation:", m_i, "of", M, "\n")
```

```
#
#   # Complete the m_i-th imputed training dataset
#   train_imp <- complete(imputed_train, m_i)
#
#   # Align factor levels in training data
#   for (var in factor_vars) {
#     if(var %in% names(train_imp)){
#       train_imp[[var]] <- factor(train_imp[[var]], levels = levels(train_data[[var]]))
#     }
#   }
#
#   # Store the aligned training dataset
#   aligned_imputed_train[[m_i]] <- train_imp
#
#   # Complete the m_i-th imputed testing dataset
#   test_imp <- complete(imputed_test, m_i)
#
#   # Align factor levels in testing data
#   for (var in factor_vars) {
#     if(var %in% names(test_imp)){
#       test_imp[[var]] <- factor(test_imp[[var]], levels = levels(train_data[[var]]))
#     }
#   }
#
#   # Store the aligned testing dataset
#   aligned_imputed_test[[m_i]] <- test_imp
# }
#
#
#
# coef_list <- list()         # To store coefficient vectors from each bootstrap run
# auc_train_list <- c()       # To store AUCs for training (in-sample)
# auc_test_list  <- c()       # To store AUCs for testing (out-of-sample)
# preds_test_list <- list()   # To store prediction probabilities for testing set
#
# # Bootstrapping and LASSO Logistic Regression
#
#
# # Loop over each imputed dataset
# for (m_i in 1:M) {
#   cat("\nProcessing Imputation:", m_i, "of", M, "\n")
#
#   # Retrieve the m_i-th aligned imputed training and testing datasets
#   train_imp <- aligned_imputed_train[[m_i]]
#   test_imp  <- aligned_imputed_test[[m_i]]
#
#   # Loop over each bootstrap iteration
#   for (b_i in 1:B) {
#     # Print progress every 50 iterations
#     if(b_i %% 50 == 0){
#       cat("  Bootstrap Iteration:", b_i, "of", B, "\n")
#     }
#
#     # Bootstrap sampling from the imputed training set
#     boot_index <- sample(nrow(train_imp), size = nrow(train_imp), replace = TRUE)
#     boot_sample <- train_imp[boot_index, ]
```

```
#
#     # Create model matrix with interactions (excluding the intercept)
#     x_train <- model.matrix(interaction_formula, data = boot_sample)[, -1]  # Remove intercept
#     y_train <- boot_sample$abst
#
#     # Fit LASSO with cross-validation to find best lambda
#     cv_lasso <- cv.glmnet(
#       x_train,
#       y_train,
#       alpha        = 1,         # LASSO penalty
#       family       = "binomial",
#       type.measure = "auc"      # Optimize for AUC
#     )
#     best_lambda <- cv_lasso$lambda.min
#
#     # Fit final LASSO model with best lambda
#     lasso_fit <- glmnet(
#       x_train,
#       y_train,
#       alpha   = 1,
#       family  = "binomial",
#       lambda  = best_lambda
#     )
#
#     # Store coefficients (as a named vector)
#     coefs <- as.numeric(coef(lasso_fit))
#     names(coefs) <- rownames(coef(lasso_fit))
#     coef_list[[length(coef_list) + 1]] <- coefs
#
#     # Calculate AUC on training (in-sample)
#     preds_train_prob <- predict(lasso_fit, newx = x_train, type = "response")
#     roc_train <- roc(y_train, as.numeric(preds_train_prob), quiet = TRUE)
#     auc_train <- auc(roc_train)
#     auc_train_list <- c(auc_train_list, auc_train)
#
#     # Predict on imputed testing set
#     x_test <- model.matrix(interaction_formula, data = test_imp)[, -1]  # Remove intercept
#     y_test <- test_imp$abst
#
#     preds_test_prob <- predict(lasso_fit, newx = x_test, type = "response")
#
#     # Ensure that preds_test_prob is a vector
#     preds_test_prob <- as.vector(preds_test_prob)
#
#     # Store test predictions and true labels
#     preds_test_list[[length(preds_test_list) + 1]] <- preds_test_prob
#     auc_test <- auc(roc(y_test, as.numeric(preds_test_prob), quiet = TRUE))
#     auc_test_list <- c(auc_test_list, auc_test)
#   }
# }


# ###### Summarize AUCs and Plot ROC Curve


# Calculate Mean and Standard Deviation of Training AUCs
```

```r
mean_auc_train <- mean(auc_train_list)
sd_auc_train   <- sd(auc_train_list)

# Calculate Mean and Standard Deviation of Testing AUCs
mean_auc_test <- mean(auc_test_list)
sd_auc_test   <- sd(auc_test_list)

# Display AUC Summary
cat("\n=== AUC Summary ===\n")
cat("Training Set AUC: Mean =", round(mean_auc_train, 4), "| SD =", round(sd_auc_train, 4), "\n")
cat("Testing Set AUC:  Mean =", round(mean_auc_test, 4), "| SD =", round(sd_auc_test, 4), "\n")

# Aggregate all test predictions and true labels
all_test_preds <- unlist(preds_test_list)

# Repeat the test labels for each imputation and bootstrap iteration (M * B times)
all_test_labels <- rep(test_data$abst, times = M * B)


# Ensure lengths match
if(length(all_test_preds) != length(all_test_labels)){
  stop("Length mismatch between all_test_preds and all_test_labels.")
}

# Create ROC object for testing set
roc_test_all <- roc(all_test_labels, all_test_preds, quiet = TRUE)

# Plot ROC Curve for Testing Set
plot(
  roc_test_all,
  col  = "blue",
  lwd  = 2,
  main = paste("ROC Curve for Testing Set\nAUC =", round(auc(roc_test_all), 3))
)

# Add a diagonal line for reference
abline(a = 0, b = 1, lty = 2, col = "gray")

# Add legend
legend("bottomright", legend = paste("AUC =", round(auc(roc_test_all), 3)),
       col = "blue", lwd = 2, bty = "n")

# ------------------------------
# Coefficient Summary and Saving Results

# Get predictor names from column names
coef_matrix <- do.call(rbind, coef_list)  # Combine the list of coefficient vectors into a matrix

predictors <- colnames(coef_matrix)
library(knitr)
library(kableExtra)

# Calculate Mean, CI, OR, and OR_CI for each coefficient
coef_summary <- data.frame(
  Coefficient = predictors,
  Mean = apply(coef_matrix, 2, mean),
```

```
  CI_lower = apply(coef_matrix, 2, quantile, 0.025),
  CI_upper = apply(coef_matrix, 2, quantile, 0.975),
  stringsAsFactors = FALSE
)

# Calculate Odds Ratios and their Confidence Intervals
coef_summary$OR <- exp(coef_summary$Mean)
coef_summary$OR_CI_lower <- exp(coef_summary$CI_lower)
coef_summary$OR_CI_upper <- exp(coef_summary$CI_upper)

# Combine CIs into single strings
coef_summary$CI <- paste0(round(coef_summary$CI_lower, 2), " - ", round(coef_summary$CI_upper, 2))
coef_summary$OR_CI <- paste0(round(coef_summary$OR_CI_lower, 2), " - ", round(coef_summary$OR_CI_upper, 2))

# Select relevant columns
coef_summary_df <- coef_summary[, c("Coefficient", "Mean", "CI", "OR", "OR_CI")]

# Order the table by OR in descending order
coef_summary_df <- coef_summary_df[order(coef_summary_df$OR, decreasing = TRUE), ]

# Reset row names
rownames(coef_summary_df) <- NULL


table <- kable(coef_summary_df,
               caption = "Coefficient Summary with Odds Ratios (ORs) and 95% CI",
               align = 'c',
               col.names = c("Coefficient", "Mean", "95% CI", "OR", "OR 95% CI"),
               escape = FALSE) %>%
         kable_styling(bootstrap_options = c("striped", "condensed"),
                       full_width = FALSE)


# Save the results to an RData file for future use
save(coef_list, auc_train_list, auc_test_list, preds_test_list,
     file = "bootstrap_lasso_results.RData")



# Load necessary libraries

# library(ROCR)
# library(Metrics)
# library(knitr)
# library(kableExtra)
#
# # Assign the true labels for train and test
# train_data$true <- train_data$abst
# test_data$true <- test_data$abst
#
# # ------------------------------
# # 2. Predictions
# # ------------------------------
# train_data$pred <- preds_train_prob
# test_data$pred <- preds_test_prob
#
```

```
# # ------------------------------
# # 3. Define a Function to Calculate Metrics
# # ------------------------------
# library(ROCR)
#
# calculate_metrics <- function(true, pred) {
#   # Convert true labels to factors if not already
#   true <- as.factor(true)
#
#   # Prediction object for ROCR
#   pred_obj <- ROCR::prediction(pred, true)
#
#   # Calculate AUC
#   auc <- ROCR::performance(pred_obj, measure = "auc")@y.values[[1]]
#
#   # Calculate Sensitivity and Specificity
#   roc_perf <- ROCR::performance(pred_obj, "sens", "spec")
#   sensitivity <- roc_perf@x.values[[1]][1]
#   specificity <- roc_perf@y.values[[1]][1]
#
#   # Calculate Accuracy
#   accuracy <- sum((pred > 0.5) == (true == 1)) / length(true)
#
#   # Calculate Brier Score
#   brier_score <- mean((pred - as.numeric(true))^2)
#
#   # Return metrics as a named vector
#   c(
#     Accuracy = accuracy,
#     Sensitivity = sensitivity,
#     Specificity = specificity,
#     AUC = auc,
#     Brier_Score = brier_score
#   )
# }
# # ------------------------------
# # 4. Calculate Metrics for Train and Test
# # ------------------------------
# metrics_train <- calculate_metrics(train_data$abst, train_data$pred)
# metrics_test <- calculate_metrics(test_data$abst, test_data$pred)
#
# # ------------------------------
# # 5. Create a Summary Table
# # ------------------------------
# # Combine metrics into a data frame
# evaluation_table <- data.frame(
#   Dataset = c("Train", "Test"),
#   Accuracy = c(metrics_train["Accuracy"], metrics_test["Accuracy"]),
#   Sensitivity = c(metrics_train["Sensitivity"], metrics_test["Sensitivity"]),
#   Specificity = c(metrics_train["Specificity"], metrics_test["Specificity"]),
#   AUC = c(metrics_train["AUC"], metrics_test["AUC"]),
#   Brier_Score = c(metrics_train["Brier_Score"], metrics_test["Brier_Score"])
# )
#
# # Display the table using kable
# library(knitr)
```

```
# kable(
#   evaluation_table,
#   format = "html",
#   table.attr = "class='table table-striped table-condensed'"
#)
# library(viridis)
#
# # store auc
# auc_values <- numeric(m)
#
# # plot
# plot(0, 0, type = "n", xlim = c(1, 0), ylim = c(0, 1),
#      xlab = "Specificity", ylab = "Sensitivity",
#      main = "ROC Curves Across Imputed Datasets - Lasso")
# abline(a = 1, b = -1, col = "grey", lty = 2)
#
# # loop to calculate and plot ROC curves
# for (i in 1:m) {
#   # Calculate ROC curve
#   roc_curve <- roc(Y_test, lasso_test_predictions[[i]])
#   auc_values[i] <- auc(roc_curve)  # Store AUC for this dataset
#   plot(roc_curve, col = viridis(m, option = "mako")[i], lwd = 2, add = TRUE)
# }
#
# # mean and standard deviation of AUC
# mean_auc <- mean(auc_values)
# sd_auc <- sd(auc_values)
#
# text(x = 1, y = 0.95, labels = paste("Mean AUC:", round(mean_auc, 3), "\nSD:",
#                                      round(sd_auc, 3)),
#      adj = 0, col = "black", cex = 0.8)
# legend("bottomright", legend = paste("Imputed Dataset", 1:m),
#        col = viridis(m, option = "mako"), lwd = 2)



# # prepare df for calibration data
# calibration_data <- do.call(rbind, lapply(1:m, function(i) {
#   data.frame(
#     predicted = as.vector(lasso_test_predictions[[i]]),
#     observed = as.numeric(Y_test) - 1  # Convert factor to numeric for binary outcomes
#   )
# }))
#
# # bin the predictions
# calibration_data <- calibration_data %>%
#   mutate(bin = cut(predicted, breaks = 5)) %>%
#   group_by(bin) %>%
#   summarize(
#     expected = mean(predicted),
#     observed = mean(observed),
#     se = sqrt((observed * (1 - observed)) / n())
#   )
#
# # create plot
# ggplot(calibration_data, aes(x = expected, y = observed)) +
```

```
#   geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
#   geom_smooth(method = "loess", color = "black", se = TRUE, fill = "gray") +
#   geom_smooth(method = "lm", color = "blue", linetype = "dotted", se = F) +
#   labs(
#     x = "Expected Proportion",
#     y = "Observed Proportion",
#     title = "Calibration Plot - Lasso"
#   ) +
#   theme_minimal() +
#     theme(legend.position = "bottom")


# # set up train-test split (80% train, 20% test)
# set.seed(1)  # for reproducibility
# trainIndex <- createDataPartition(imputed_data[[1]]$abst, p = 0.8, list = FALSE)
#
# # initialize lists for predictions, AUC values, and coefficient estimates
# best_subset_test_predictions <- list()
# best_subset_coef_estimates <- list()  # store coefficients for each imputed dataset
# Y_test <- NULL  # store observed test outcomes once
#
# # loop through each imputed dataset
# for (i in 1:m) {
#   completed_data <- complete(imputed_data, i)
#
#   # split data into train and test sets
#   train_data <- completed_data[trainIndex, ]
#   test_data <- completed_data[-trainIndex, ]
#
#   # prepare model matrix and outcome for training and testing
#   X_train <- model.matrix(formula, data = train_data)
#   Y_train <- train_data$abst
#   X_test <- model.matrix(formula, data = test_data)
#   Y_test <- test_data$abst
#
#   # fit best subset model with cross-validation
#   best_sub_model <- L0Learn.cvfit(X_train, Y_train, loss = "Logistic",
#                                   nFolds = 5, maxIter = 100)
#
#   # get optimal gamma and lambda
#   min_ind <- which(best_sub_model$cvMeans[[1]] == min(best_sub_model$cvMeans[[1]]),
#                    arr.ind = TRUE)
#   gamma_min <- best_sub_model$fit$gamma[[min_ind[2]]]
#   lambda_min <- best_sub_model$fit$lambda[[min_ind[2]]][min_ind[1]]
#
#   # extract coefficients at optimal parameters
#   coef_output <- as(coef(best_sub_model, gamma = gamma_min, lambda = lambda_min),
#                     "matrix")
#   rownames(coef_output) <- c("(Intercept)", colnames(X_train))
#   best_subset_coef_estimates[[i]] <- coef_output  # store coefficients
#
#   # make predictions on test set
#   predictions <- predict(best_sub_model, newx = X_test, gamma = gamma_min,
#                          lambda = lambda_min, type = "response")
#   best_subset_test_predictions[[i]] <- predictions
# }
```

```
#
# # combine coefficients across imputations
# best_subset_coef_matrix <- do.call(cbind, best_subset_coef_estimates)
#
# # pool results using rubin's rules
# variable_names <- rownames(best_subset_coef_estimates[[1]])
# best_subset_pooled_results <- data.frame(Variable = variable_names,
#                                          Mean = NA, SE = NA)
#
# for (var in variable_names) {
#   coefs <- best_subset_coef_matrix[var, ]
#   best_subset_pooled_results[best_subset_pooled_results$Variable == var, "Mean"] <- mean(coefs, na.rm = T
#
#   # calculate standard error using rubins rules
#   se_within <- sqrt(mean((coefs - mean(coefs, na.rm = TRUE))^2, na.rm = TRUE))
#   se_between <- var(coefs, na.rm = TRUE)
#   best_subset_pooled_results[best_subset_pooled_results$Variable == var, "SE"] <- sqrt(se_within + (1 + 1
# }
#
# # filter non-zero mean coefficients and sort by mean
# best_subset_selected_sorted <- best_subset_pooled_results %>%
#   filter(Mean != 0, Variable != "(Intercept)") %>%
#   arrange(desc(Mean))
#
# # display results in a table
# best_sub_table <- kable(best_subset_selected_sorted,
#                         caption = "Best Subset Model Selected Variables
#                         (Non-Zero Coefficients)") %>%
#   kable_styling(full_width = F, font_size = 12)
#
# best_sub_table


# best_subset_auc_values <- numeric(m)
#
#
# plot(0, 0, type = "n", xlim = c(1, 0), ylim = c(0, 1),
#      xlab = "1 - Specificity", ylab = "Sensitivity",
#      main = "ROC Curves Across Imputed Datasets - Best Subset")
# abline(a = 1, b = -1, col = "grey", lty = 2)
#
# # loop through each imputed dataset to calculate and plot ROC curves
# for (i in 1:m) {
#   # convert predictions to numeric form if needed
#   predictions_numeric <- as.numeric(best_subset_test_predictions[[i]])
#
#   # calculate ROC curve
#   roc_curve_best_subset <- roc(Y_test, predictions_numeric)
#   best_subset_auc_values[i] <- auc(roc_curve_best_subset)  # store AUC for dataset
#
#   # plot each ROC curve with "mako" colors
#   plot(roc_curve_best_subset, col = viridis(m, option = "mako")[i],
#        lwd = 2, add = TRUE)
# }
#
# # calculate mean and standard deviation of AUC across imputations
```

```
# mean_auc_best_subset <- mean(best_subset_auc_values)
# sd_auc_best_subset <- sd(best_subset_auc_values)
#
#
# text(x = 1, y = 0.95,
#      labels = paste("Mean AUC:", round(mean_auc_best_subset, 3), "\nSD:",
#                     round(sd_auc_best_subset, 3)),
#      adj = 0, col = "black", cex = 0.8)
#
# legend("bottomright", legend = paste("Imputed Dataset", 1:m),
#        col = viridis(m, option = "mako"), lwd = 2)


# # prepare df for calibration by combining predictions and observed values
# calibration_data_best_subset <- do.call(rbind, lapply(1:m, function(i) {
#     data.frame(
#         predicted = as.vector(best_subset_test_predictions[[i]]),
#         observed = as.numeric(Y_test) - 1
#     )
# }))
#
# # bin predictions for calibration
# calibration_data_best_subset <- calibration_data_best_subset %>%
#     mutate(bin = cut(predicted, breaks = 5)) %>%
#     group_by(bin) %>%
#     summarize(
#         expected = mean(predicted),
#         observed = mean(observed),
#         se = ifelse(n() > 1, sqrt((observed * (1 - observed)) / n()), NA)
#     )
#
# # Create calibration plot
# ggplot(calibration_data_best_subset, aes(x = expected, y = observed)) +
#   geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
#   geom_smooth(method = "loess", color = "black", se = TRUE, fill = "gray") +
#   geom_smooth(method = "lm", color = "blue", linetype = "dotted", se = FALSE) +
#   labs(
#     x = "Expected Proportion",
#     y = "Observed Proportion",
#     title = "Calibration Plot for Best Subset Model"
#   ) +
#   theme_minimal()
```