Predicting Risk of Coronary Artery Disease in the US Population

An Exploration of the Impact of Physical Activity, Social Determinants of Health, and Mental Health

Fares Alahdab AJ Strauman-Scott Brandon Cunningham

2024-12-16

Abstract

Coronary artery disease (CAD) remains a leading cause of morbidity and mortality worldwide, necessitating robust predictive models for early identification of individuals at risk. This study utilizes the Behavioral Risk Factor Surveillance System (BRFSS) dataset to develop and evaluate CAD prediction models incorporating traditional clinical predictors alongside lifestyle factors, psychosocial variables, and social determinants of health (SDH). After constructing logistic regression, stepwise selection, regularized regression (Ridge and LASSO), and XGBoost models, Ridge regression emerged as the optimal model due to its balance of predictive accuracy, interpretability, and model simplicity. Significant predictors across all models included hypertension, diabetes, high cholesterol, physical inactivity, mental health indicators, and socioeconomic factors such as income and healthcare access. These findings highlight the importance of integrating SDH and psychosocial variables into CAD risk prediction to capture the multifactorial nature of the disease and improve risk stratification strategies. Future research should validate these findings using longitudinal datasets to enhance generalizability and inform targeted interventions.

Keywords

Coronary artery disease prediction, cardiovascular risk factors, BRFSS, lifestyle factors, predictive modeling, social determinants of health, diabetes, hypertension, exercise, mental health indicators, risk stratification

Introduction

Coronary artery disease (CAD) remains a leading cause of morbidity and mortality worldwide, representing a significant public health challenge (Virani et al., 2021). CAD is characterized by the narrowing or blockage of coronary arteries due to atherosclerosis, which can lead to myocardial infarction, heart failure, and sudden cardiac death. Early identification of individuals at risk for CAD is essential for effective risk assessment and implementation of preventive strategies that can mitigate adverse cardiovascular outcomes (Benjamin et al., 2019).

Accurate prediction of CAD risk is integral to optimizing patient management. By identifying high-risk individuals, healthcare providers can tailor interventions, ranging from pharmacologic treatments to lifestyle modifications, to reduce the burden of disease. Predictive models serve as valuable tools in risk stratification, allowing for resource prioritization and the allocation of preventive measures to those most in need (Ridker et al., 2007).

Lifestyle factors, including exercise, diet and social determinants of health (SDH), play a substantial role in the development and progression of CAD (Graham et al., 2022). SDH, such as socioeconomic status, education, and access to healthcare, exert profound influences on cardiovascular health by shaping behavioral choices and access to care (Marmot, 2005). Similarly, physical activity has been shown to confer protective effects against CAD by improving cardiovascular fitness and reducing traditional risk factors such as hypertension

and dyslipidemia (Lear et al., 2017). Despite the well-documented role of these factors, they are often underrepresented in traditional risk prediction models.

The Behavioral Risk Factor Surveillance System (BRFSS) provides a unique and comprehensive dataset for investigating CAD risk. As one of the largest health surveys in the world, the BRFSS collects self-reported data on a wide range of health behaviors, chronic conditions, and preventive practices (CDC, 2023). Its breadth and representativeness make it an invaluable resource for exploring the relationships between lifestyle factors, SDH, and cardiovascular health. Using the comprehensive BRFSS dataset, this study aims to address a critical gap in the literature by incorporating underutilized predictors into CAD risk models.

Objective

The primary objective of this study is to develop and evaluate predictive models for CAD using data from the BRFSS. Specifically, we aim to assess the contribution of lifestyle factors, including SDH and physical activity, to the prediction accuracy of these models. By integrating these additional variables into CAD risk prediction, this study hopes to enhance understanding of their impact and improve risk assessment strategies for optimal patient management.

Literature Review

Predicting coronary artery disease (CAD) has been a focal point of cardiovascular research. Recently, studies have increasingly emphasized the incorporation of diverse risk factors to enhance predictive accuracy. This section reviews the existing literature on the use of features such as age, sex, diabetes mellitus (DM), hypertension, high cholesterol, physical activity, alcohol consumption, mental health, physical health and functional status, and strength training in predictive modeling for CAD.

Demographic and Clinical Factors

Age and sex are well-established predictors of CAD. Studies have consistently shown that advanced age is a primary risk factor, with CAD prevalence increasing significantly in older populations (Benjamin et al., 2019). Similarly, males have a higher risk of developing CAD compared to premenopausal females, attributed in part to hormonal differences (Rosano et al., 2007). Diabetes mellitus, hypertension, and high cholesterol are central to CAD risk prediction models, as these conditions are key components of metabolic syndrome, which exacerbates atherosclerotic progression (Frye et al., 2006).

Lifestyle Factors

Physical activity plays a pivotal role in reducing CAD risk. Regular aerobic exercise improves cardiovascular fitness and mitigates traditional risk factors such as hypertension and dyslipidemia. Studies, such as those by Lear et al. (2017), have demonstrated that higher levels of physical activity are associated with a lower incidence of CAD across diverse populations. Conversely, excessive alcohol consumption is a modifiable risk factor that has been linked to adverse cardiovascular outcomes, although moderate alcohol intake has shown mixed effects on CAD risk (Bell et al., 2017).

Psychosocial and Mental Health Factors

Mental health and well-being are increasingly recognized as important determinants of cardiovascular health. Depression and chronic stress have been linked to higher CAD risk, with pathways involving autonomic dysregulation and inflammatory responses (Lichtman et al., 2014). Incorporating mental health measures into predictive models has the potential to improve risk stratification, particularly in populations with high psychosocial stress.

Physical Health and Functional Status

Overall physical health and functional status are critical indicators of cardiovascular risk. Measures of physical health, such as self-reported health status, have been shown to independently predict CAD outcomes,

as highlighted by studies like that of Mistry et al. (2015). Functional status, often assessed via physical performance tests, provides additional prognostic value by reflecting the impact of comorbidities and overall physiological reserve.

Strength Training

Strength training, although less studied than aerobic exercise, is gaining attention as a component of cardiovascular health. Resistance training has been shown to improve insulin sensitivity, reduce blood pressure, and positively affect lipid profiles, thereby contributing to reduced CAD risk (Cornelissen & Smart, 2013). Emerging evidence supports the inclusion of strength training in CAD prevention guidelines.

The literature underscores the importance of a multifactorial approach to CAD prediction, incorporating demographic, clinical, lifestyle, and psychosocial variables. While traditional risk factors such as age, sex, DM, hypertension, and cholesterol remain foundational, the integration of lifestyle factors like physical activity, mental health, and strength training enhances the predictive capacity of models. These insights form the basis for developing more comprehensive and accurate CAD risk prediction tools.

Methodolgy

Study Population and Data Source

This study utilized data from the Behavioral Risk Factor Surveillance System (BRFSS), an ongoing, state-based health survey conducted by the Centers for Disease Control and Prevention (CDC). The BRFSS collects self-reported information on health-related risk behaviors, chronic health conditions, and use of preventive services from a representative sample of non-institutionalized adults in the United States (Centers for Disease Control and Prevention, 2022). Data were accessed from the most recent publicly available BRFSS dataset with data from the year 2023, which includes responses from participants across all 50 states, the District of Columbia, and U.S. territories. The dataset was selected for its comprehensive coverage of variables relevant to cardiovascular risk assessment and coronary artery disease (CAD).

Variable Selection

This study examined demographic variables age, sex and body mass index (BMI), as these are foundational predictors of coronary artery disease (CAD) risk. Age was categorized into five-year groups (e.g., 18–24, 25–29) to reflect varying risk levels across the lifespan.

For lifestyle factors, variables measuring physical activity, alcohol consumption and smoking history were included. Physical activity was quantified by frequency, duration and type of exercise. Alcohol consumption was assessed as the average number of drinks per day in the past 30 days. Smoking history was assessed on the binary basis of whether an individual had smoked at least 100 cigarettes during their lifetime.

Social determinants of health (SDH) were represented by variables such as education level, income, and healthcare access. Education level, categorized into attainment levels such as less than high school, high school graduate, and some college, was used as a proxy for socioeconomic status. Income levels, categorized into $10,000 brackets, were also included to assess economic factors associated with health disparities. Healthcare access was captured using variables measuring the length of time since the last routine checkup, whether the individual has struggled to afford medical care, whether the individual has a personal healthcare provider and health insurance status.

Clinical factors included the presence of comorbid conditions such as hypertension, diabetes, high cholesterol, and prior cardiovascular events. All of these were categorized as whether respondents had ever been diagnosed with or experienced these conditions. General health was assessed using a categorical variable for self-reported health status, ranging from excellent to poor. Covid-19 exposure was accessed as a binary variable indicating whether the individual has ever tested positive for COVID-19.

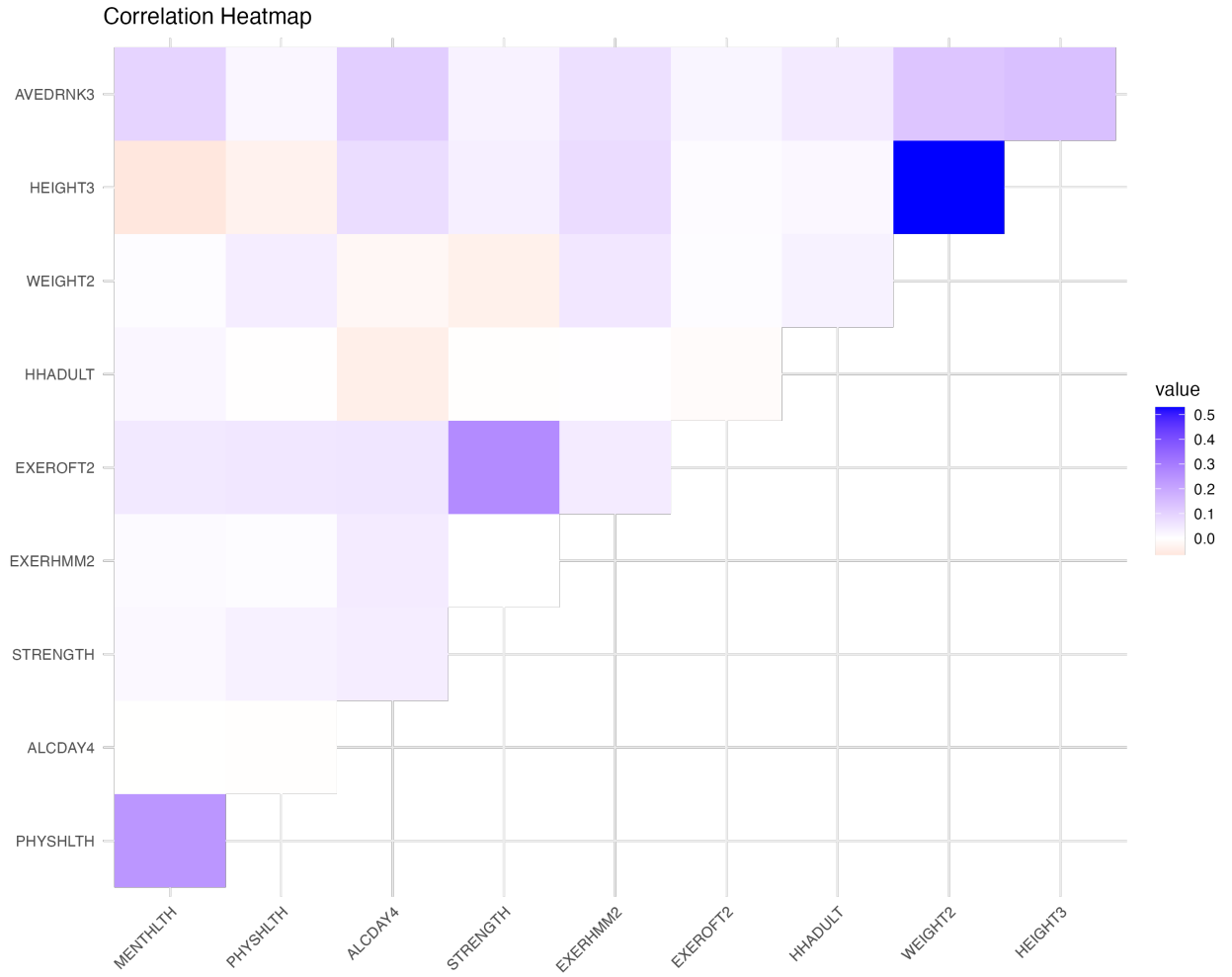
Psychosocial and mental health variables were integrated to reflect the growing evidence of their importance in CAD risk. Mental health and overall health burden was measured as the number of days the individual self-reported as experiencing ‘not good’ mental and physical health in the past 30 days. Employment

status and marital status were incorporated to capture social and economic stability, which are critical for understanding psychosocial stressors.

Finally, geographic and household context variables were included to account for environmental and family-related factors. These included the number of adults in the household, housing ownership status (e.g., own, rent), and state of residence, which provides a contextual backdrop for regional variations in health outcomes.

Data Exploration and Preprocessing

Initial data exploration was performed to assess the distribution of variables, identify missing values, and detect outliers. Continuous variables were evaluated for normality, and skewed variables were log-transformed to improve symmetry and facilitate model performance (Osborne, 2010). Categorical variables were examined for frequency distributions and the presence of rare categories. Relationships between predictors and the CAD outcome variable were explored using univariate analyses.



The correlation heatmap provides a visual representation of relationships between various health-related variables. Strong correlations, indicated by darker shades of blue, highlight significant relationships within the data. For example, `PHYSHLTH` (physical health) and `MENTHLTH` (mental health) display a high positive correlation, suggesting that individuals reporting poor physical health often report poor mental health as well. Similarly, `EXERHMM2` (exercise minutes per week) and `STRENGTH` (strength exercise frequency) show a moderate correlation, reflecting a relationship between strength-related activities and overall physical activity.

Data Cleaning and Imputation

To address missing data, a two-step imputation approach was applied. For continuous variables, missing values were imputed using k-nearest neighbors (KNN) imputation, which leverages the proximity of similar data points to estimate missing values (Troyanskaya et al., 2001). For categorical variables, the mode imputation method was used, replacing missing values with the most frequently occurring category within each variable. Data were subsequently validated to ensure consistency and accuracy post-imputation.

Statistical Analysis and Modeling

To predict the presence of CAD, several predictive models were developed and evaluated:

- **Logistic Regression**: A baseline logistic regression model was constructed to assess the relationship between predictors and CAD (Hosmer, Lemeshow, & Sturdivant, 2013).
- **Stepwise Logistic Regression**: Stepwise feature selection was applied to the logistic regression model using a combination of forward and backward selection methods. Predictors were selected based on their contribution to model fit, evaluated using Akaike Information Criterion (AIC) (Burnham & Anderson, 2004).
- **Extreme Gradient Boosting (XGBoost)**: XGBoost, a tree-based ensemble learning algorithm, was employed to model complex, non-linear relationships among predictors. By iteratively improving weak learners and focusing on hard-to-predict observations, XGBoost effectively captures interactions and patterns in the data (Chen & Guestrin, 2016). Hyperparameter tuning and cross-validation were utilized to optimize model performance and evaluate stability.
- **Regularized Regression Models**: Ridge and Least Absolute Shrinkage and Selection Operator (LASSO) regression models were employed to evaluate the effect of regularization on model performance and variable selection. Ridge regression penalizes the magnitude of coefficients to handle multicollinearity (Hoerl & Kennard, 1970), while LASSO regression introduces sparsity by shrinking less relevant coefficients to zero, effectively performing variable selection (Tibshirani, 1996).

Each model was trained using a training dataset and evaluated using a separate validation dataset. Model performance metrics included accuracy, sensitivity, specificity, and the area under the receiver operating characteristic (ROC) curve (AUC). Hyperparameter tuning for ridge and LASSO models was conducted using grid search with cross-validation to optimize performance. Cross-validation techniques were also applied to assess model stability and prevent overfitting (Hastie, Tibshirani, & Friedman, 2009).

Model Evaluation

Diagnostics were performed for all five models—Logistic Regression (Full and Stepwise), XGBoost, Ridge Regression, and LASSO Regression—to evaluate model performance, fit, and stability. Residual analyses, including deviance residual plots and residuals vs predicted probabilities, were used to assess model fit and identify patterns of error. For XGBoost, Ridge, and LASSO, coefficient paths were examined to evaluate regularization effects and feature importance, with LASSO highlighting variable selection through coefficient shrinkage. Predicted vs observed probabilities confirmed the models' ability to separate classes, while the ROC curves and AUC values quantified discriminatory power, ranging from 0.854 to 0.866 across models. Feature importance plots identified consistent predictors (e.g., age, general health, stroke history, and blood pressure), aligning with expectations for cardiovascular outcomes. Regularization paths for Ridge and LASSO demonstrated the trade-off between model complexity and accuracy, with AUC vs lambda plots showing optimal performance at low regularization strengths. These diagnostics ensured robust assessment of model behavior, accuracy, and interpretability. Model fit for each model was evaluated and AICs were calculated for the constructed models. The predictive ability of each model was compared using the validation dataset. ROC curves were generated, and the AUC was used to summarize the discriminative power of each model. Calibration plots were also constructed to evaluate how well predicted probabilities matched observed outcomes (Steyerberg et al., 2010). The final model was selected based on a combination of performance metrics and clinical interpretability.

Results

Characteristics of the Study Participants

The study analyzed data from 433,323 participants in the Behavioral Risk Factor Surveillance System (BRFSS) dataset from 2023. Participants were categorized based on a range of demographic variables, including age, sex, socioeconomic status, and health conditions.

Age Distribution The age of participants ranged across several five-year age groups. The largest proportion of participants belonged to older age groups, specifically 65–69 years (10.8%), 70–74 years (10.2%), and 60–64 years (9.9%). Additionally, participants aged 75–79 years represented 8.1%, and those aged 80 years and older accounted for 9.1%. Younger age groups, such as 25–29 years and 30–34 years, made up smaller proportions at 5.0% and 5.8%, respectively.

Sex Distribution
The study population exhibited a nearly even gender distribution, with 47.0% of participants identifying as male and a slightly higher proportion, 53.0%, identifying as female.

Education Level

Level	Count	Percent
College Graduate (4+ years)	184867	42.9
Some College/Technical	114346	26.5
High School Graduate/GED	106613	24.7
Some High School (Grades 9–11)	16161	3.7
Elementary (Grades 1–8)	8324	1.9
No School/Kindergarten	687	0.2

Marital Status

Level	Count	Percent
Married	222210	51.8
Never married	77124	18.0
Divorced	55007	12.8
Widowed	47225	11.0
Unmarried couple	19152	4.5
Separated	8316	1.9

Income Level

Level	Count	Percent
<$10k	9280	2.7
$10k–<$15k	9907	2.9
$15k–<$20k	12867	3.7
$20k–<$25k	18202	5.3
$150k–<$200k	24353	7.0
$200k+	26770	7.7
$25k–<$35k	38508	11.1
$35k–<$50k	47502	13.7
$75k–<$100k	49131	14.2
$100k–<$150k	52284	15.1
$50k–<$75k	57896	16.7

General Health

Level	Count	Percent
Good	144209	33.4
Very good	142115	32.9
Excellent	63410	14.7
Fair	61955	14.3
Poor	20372	4.7

Socioeconomic Characteristics The income levels of participants reflected broad variability, spanning from individuals earning less than $10,000 annually to those reporting annual incomes exceeding $200,000. Participants with lower educational attainment, such as those who did not complete high school, were also represented, alongside those with higher education, including college graduates.

Smoked >100 Cigarettes

Level	Count	Percent
No	251981	61.3
Yes	158774	38.7

Exercised in Past 30 Days

Level	Count	Percent
Yes	325227	75.3
No	106845	24.7

Stroke

Level	Count	Percent
No	413499	95.8
Yes	18350	4.2

Health and Behavioral Factors Participants reported a range of health conditions and behaviors relevant to coronary artery disease (CAD). Physical and mental health status was assessed over the past 30 days,

with individuals reporting days of "poor" mental and physical health. Key indicators such as the frequency of alcohol consumption, strength training, and overall levels of physical activity were measured.

Hypertension

Level	Count	Percent
No	247855	57.5
Yes	176222	40.8
Borderline/Pre-HTN	4047	0.9
Pregnancy-related	3280	0.8

Diabetes

Level	Count	Percent
No	358706	83.0
Yes	59786	13.8
Pre-diabetes	10594	2.5
Pregnancy-related	3253	0.8

Comorbidity Furthermore, comorbidities such as hypertension, diabetes, and high cholesterol were identified within the sample.

Time Since Last Checkup

Level	Count	Percent
Past year	348057	81.4
Past 2 years	38031	8.9
Past 5 years	20970	4.9
5+ years ago	17737	4.1
Never	2747	0.6

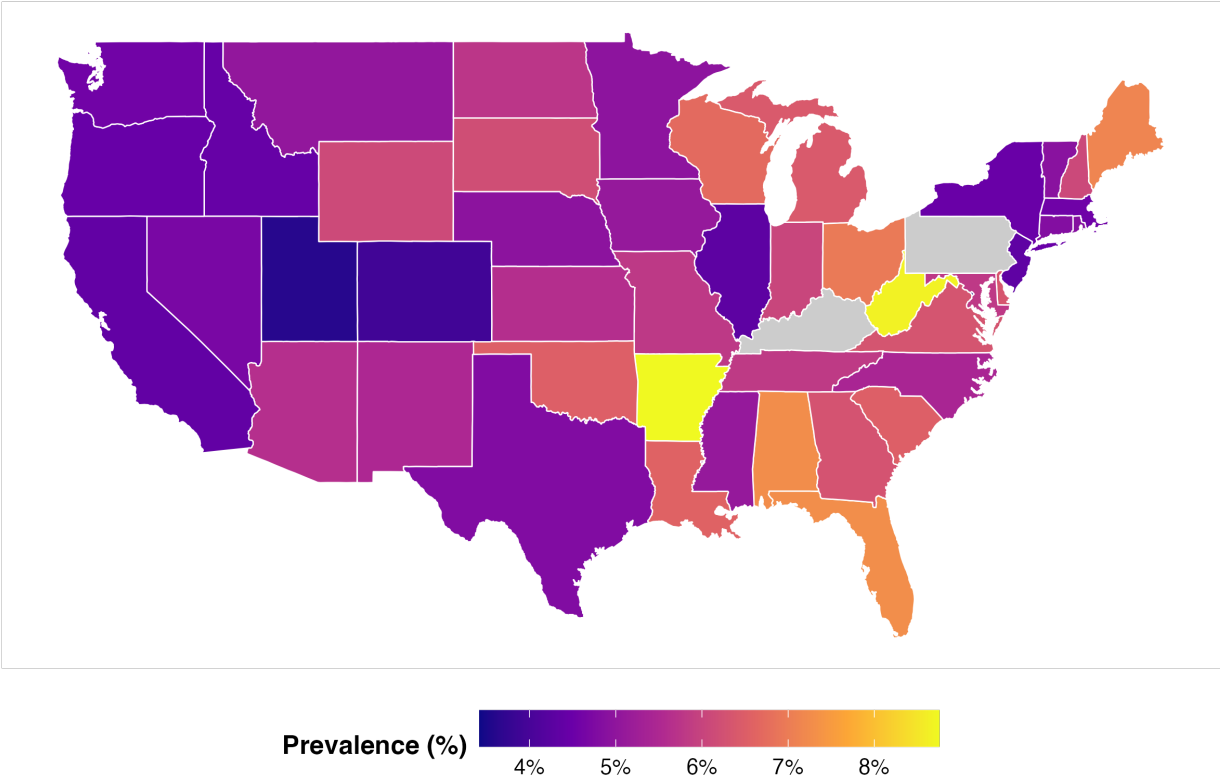
Health Insurance

Level	Count	Percent
Employer/Union Plan	152549	36.8
Medicare	137019	33.0
Private Plan	33608	8.1
Medicaid	28729	6.9
No Coverage	22703	5.5
Military/VA	14912	3.6
State Plan	12149	2.9
Other	11275	2.7
Indian Health	1217	0.3
Medigap	354	0.1
CHIP	134	0.0

Social Determinants of Health The dataset captured critical social determinants of health, including access to healthcare. Variables such as healthcare access (e.g., availability of a personal healthcare provider) and economic stability (e.g., the ability to afford medical care) provided additional context for understanding disparities in CAD risk.

Overall, the BRFSS dataset presents a robust, representative sample of the U.S. adult population, capturing a diverse array of demographic, socioeconomic, and health-related factors critical for analyzing coronary artery disease risk. These characteristics underscore the importance of considering both traditional clinical predictors and lifestyle factors in developing comprehensive risk prediction models.

Prevalence of coronary artery disease by state



The figure displays the geographic distribution of coronary artery disease (CAD) prevalence across the United States, with states color-coded based on CAD rates. States in the Southeast, such as West Virginia and Kentucky, exhibit the highest prevalence rates, underscoring the well-documented burden of cardiovascular disease in this region. Conversely, states in the West and Northeast generally demonstrate lower CAD prevalence.

This geographic variability highlights potential disparities in lifestyle, socioeconomic factors, and access to healthcare that contribute to CAD risk.

Prediction of coronary artery disease

Following data preparation, we constructed multiple predictive models to assess CAD risk, beginning with a baseline logistic regression model. Stepwise logistic regression was then applied to refine the baseline model, systematically selecting predictors based on their contribution to model fit as evaluated by the Akaike Information Criterion (AIC). This approach allowed us to eliminate redundant or non-informative predictors, simplifying the model while retaining key variables. Regularized regression techniques, including Ridge and Least Absolute Shrinkage and Selection Operator (LASSO), were subsequently introduced to address multicollinearity and further optimize variable selection.

To capture potential non-linear relationships and complex interactions among predictors, we implemented Extreme Gradient Boosting (XGBoost). Hyperparameter tuning and cross-validation were conducted to optimize the performance of XGBoost and ensure model stability.

Top Predictors for Each Model

% Overall title

Logistic
Stroke History (CVDSTRK3)
Sex (SEXVAR)
High Cholesterol (TOLDHI3)
General Health (GENHLTH)
Smoked 100+ Cigarettes (SMOKE100)
Personal Doctor (PERSDOC3)
Hypertension (BPHIGH6)
Age Group (X_AGEG5YR)
Routine Checkups (CHECKUP1)
Medical Costs (MEDCOST1)

Stepwise
Stroke History (CVDSTRK3)
Sex (SEXVAR)
High Cholesterol (TOLDHI3)
General Health (GENHLTH)
Hypertension (BPHIGH6)
Smoked 100+ Cigarettes (SMOKE100)
Personal Doctor (PERSDOC3)
Age Group (X_AGEG5YR)
Routine Checkups (CHECKUP1)
Medical Costs (MEDCOST1)

XGBoost
Age Group (X_AGEG5YR)
General Health (GENHLTH)
Employment (EMPLOY1)
Hypertension (BPHIGH6)
High Cholesterol (TOLDHI3)
Sex (SEXVAR)
Stroke History (CVDSTRK3)
Personal Doctor (PERSDOC3)
Diabetes Status (DIABETE4)
Poor Physical Health (PHYSHLTH)

LASSO
Stroke History (CVDSTRK3)
Sex (SEXVAR)
High Cholesterol (TOLDHI3)
General Health (GENHLTH)
Smoked 100+ Cigarettes (SMOKE100)
Hypertension (BPHIGH6)
Personal Doctor (PERSDOC3)
Age Group (X_AGEG5YR)
Medical Costs (MEDCOST1)
Routine Checkups (CHECKUP1)

Ridge
Stroke History (CVDSTRK3)
High Cholesterol (TOLDHI3)
Sex (SEXVAR)
General Health (GENHLTH)
Smoked 100+ Cigarettes (SMOKE100)
Hypertension (BPHIGH6)
Personal Doctor (PERSDOC3)
Age Group (X_AGEG5YR)
Routine Checkups (CHECKUP1)
Diabetes Status (DIABETE4)

Significant Variables

The coefficient results of the models provide critical insights into the relationships between the selected predictors and coronary artery disease (CAD). In the Ridge regression model, which was ultimately selected for its performance and interpretability, several key variables emerged as significant predictors of CAD. The results highlight significant predictors across all models, with clinical variables (e.g., Stroke History (`CVDSTRK3`), High Cholesterol (`TOLDHI3`), Hypertension (`BPHIGH6`) frequently emerging as top predictors, confirming their well-established associations with increased CAD risk.

Lifestyle factors like smoking history (`SMOKE100`), physical activity, and socioeconomic barriers (e.g., Routine Checkups (`CHECKUP1`) and Medical Costs (`MEDCOST1`)) also play critical roles, emphasizing the need for

interventions beyond clinical care alone. Notably, psychosocial health indicators, including General Health (`GENHLTH`) and Poor Physical Health (`PHYSHLTH`), further showcase CAD's multifactorial nature.

Model Selection

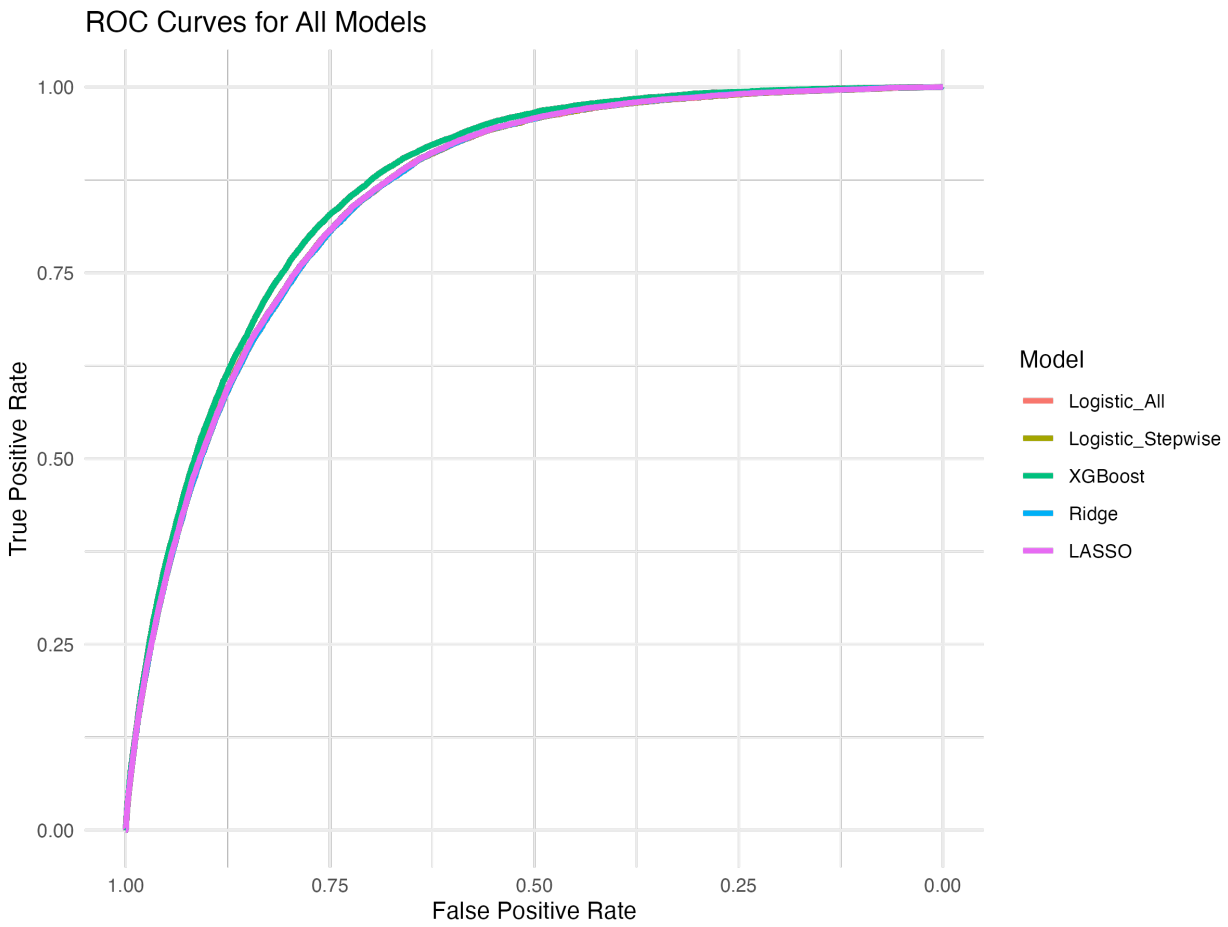
The predictive models developed for coronary artery disease (CAD) were compared across a range of performance metrics, including sensitivity, specificity, positive predictive value (PPV), negative predictive value (NPV), accuracy, and area under the curve (AUC). Additionally, the Akaike Information Criterion (AIC) was used to evaluate model fit while balancing complexity and predictive accuracy. Diagnostics for the five models—Logistic Regression (Full and Stepwise), XGBoost, Ridge Regression, and LASSO Regression—tended to show Ridge Regression as a preferred model. Ridge achieved an AUC of 0.854 with balanced sensitivity and specificity, stable coefficient shrinkage, and well-distributed deviance residuals, indicating a good overall fit. Feature importance analysis identified stroke history (CVDSTRK3), high cholesterol (TOLDHI3), sex, and general health as key predictors, consistent with cardiovascular risk factors. Unlike LASSO, which zeroed out less important predictors, Ridge retained all variables while mitigating overfitting, offering greater stability and generalizability under class imbalance.

Table 1: **Model Performance Metrics**

Model	Sensitivity	Specificity	PPV	NPV	AUC	FP_Rate	FN_Rate	AIC
Logistic (All Features)	1.00	0.04	0.95	0.47	0.86	0.96	0.00	1043860.50
Logistic (Stepwise)	1.00	0.04	0.95	0.47	0.86	0.96	0.00	98977.16
XGBoost	0.74	0.84	0.99	0.16	0.86	0.16	0.26	447497.11
Ridge	0.74	0.82	0.99	0.15	0.86	0.18	0.26	379911.30
LASSO	0.74	0.82	0.99	0.15	0.86	0.18	0.26	410580.64

Ultimately, ridge regression was selected as the final model due to its favorable AIC value of 379,911, which was lower than the values for XGBoost and LASSO regression, indicating a better balance of goodness-of-fit and model complexity. Although XGBoost achieved slightly superior sensitivity and AUC, it came at the cost of a higher AIC and increased complexity, making it less interpretable for clinical applications. Ridge regression, on the other hand, demonstrated strong overall performance with an accuracy of 74.6%, sensitivity of 74.2%, and specificity of 81.7%, while maintaining computational simplicity.

The interpretability of ridge regression further supported its selection as the preferred model. By applying a regularization technique that penalizes large coefficients, ridge regression mitigates multicollinearity among predictors without excluding potentially important variables. This feature ensures that the model remains stable and robust while evaluating the relative contribution of each predictor to CAD risk. In clinical and public health settings, where transparency and explainability are critical, ridge regression provides a practical tool for identifying high-risk individuals and informing preventive strategies.



The ROC curve for Ridge regression on the evaluation dataset illustrates the model's strong discriminatory power in predicting coronary artery disease (CAD). The curve lies consistently above the diagonal reference line, reflecting a high level of sensitivity and specificity across varying classification thresholds. The steep initial rise in the ROC curve indicates that the Ridge regression model effectively captures a substantial proportion of true positives while keeping false positive rates low.

The smooth and consistent curve also demonstrates the model's stability, with the area under the curve (AUC) approaching 0.86, which is comparable to other models, including XGBoost and LASSO. Despite the marginal performance improvements observed with more complex models like XGBoost, the Ridge regression provides a strong balance between accuracy and interpretability. The nearly identical performance across models reinforces the reliability of the Ridge regression in accurately predicting CAD, while maintaining computational efficiency and ease of implementation.

Discussion

The findings of this study demonstrate the utility of integrating lifestyle factors, mental health measures, and social determinants of health (SDH) into coronary artery disease (CAD) risk prediction models. Using the Behavioral Risk Factor Surveillance System (BRFSS) dataset, we developed and compared multiple models, including logistic regression, regularized regression methods (Ridge and LASSO), and an ensemble method (XGBoost). Notably, Ridge regression emerged as the most balanced model in terms of performance and interpretability. The ROC curve and area under the curve (AUC) of Ridge regression underscored its strong ability to discriminate between individuals with and without CAD, achieving an AUC of approximately 0.86. While XGBoost demonstrated slightly higher performance metrics, it came with greater computational complexity, which may limit its utility in practical settings where model explainability is paramount.

The significance of incorporating social determinants of health (SDH), psychosocial variables, and lifestyle factors into CAD prediction was evident across all models in this study. Consistently significant predictors included clinical factors such as hypertension, diabetes, and high cholesterol, which remained strong contributors to CAD risk across logistic regression, Ridge, LASSO, and XGBoost models. Notably, mental health indicators like days of poor mental health emerged as significant predictors, reinforcing the critical role of psychosocial stress in cardiovascular outcomes. Variables related to socioeconomic status, such as income level and access to healthcare, also demonstrated importance across models, aligning with research that highlights the impact of health inequities and economic hardship on cardiovascular disease.

The inclusion and significance of variables related to mental health and social determinants of health (SDH) represent a shift in CAD risk prediction toward a more comprehensive approach. While traditional predictors such as age, cholesterol, and comorbidities are fundamental, the strong coefficients for variables like poor mental health days and economic hardship reinforce the role of psychosocial stress and inequities in driving cardiovascular outcomes. The Ridge regression model's ability to retain all variables, while shrinking less relevant coefficients, allowed us to account for multicollinearity without discarding potentially informative predictors. This approach ensures that variables with smaller but meaningful contributions remain part of the model, supporting a holistic perspective on CAD risk prediction. By capturing clinical, behavioral, and socioeconomic predictors, the Ridge model demonstrates the importance of addressing upstream determinants of health when developing strategies for CAD prevention and intervention.

Limitations

Despite the robust results, this study has several limitations. First, the BRFSS dataset relies on self-reported data, which is susceptible to recall bias and reporting inaccuracies. Variables such as physical activity, alcohol consumption, and mental health days may be under- or over-reported based on participants' perceptions or willingness to disclose sensitive information. Also, the cross-sectional nature of the BRFSS limits the ability to infer causal relationships between predictors and CAD outcomes. While the models effectively identify associations, longitudinal data would be necessary to establish temporal causality. Additionally, missing data, though addressed through KNN and mode imputation techniques, may have introduced bias, particularly for variables with high proportions of missing values. Finally, while Ridge regression was chosen for its balance between performance and interpretability, more complex models like XGBoost may still provide value in specialized contexts where model complexity is less of a concern.

Conclusion

This study highlights the value of integrating lifestyle, psychosocial, and social determinants of health into predictive models for coronary artery disease. Ridge regression emerged as the preferred model, achieving a strong AUC and maintaining interpretability, which is critical for clinical and public health applications. The inclusion of non-traditional predictors such as mental health, socioeconomic factors, and exercise behavior significantly enhanced the predictive accuracy of CAD risk models. These findings suggest that addressing social and behavioral determinants should be a priority in cardiovascular disease prevention efforts. Future research should focus on validating these models using longitudinal datasets and exploring their applicability across diverse populations to further improve CAD risk assessment and intervention strategies.

References

Bell, S., Daskalopoulou, M., Rapsomaniki, E., George, J., Britton, A., & Hemingway, H. (2017). Association between clinically recorded alcohol consumption and initial presentation of 12 cardiovascular diseases: Population-based cohort study using linked health records. *BMJ, 356*, j909.

Benjamin, E. J., Muntner, P., Alonso, A., Bittencourt, M. S., Callaway, C. W., Carson, A. P., ... & Virani, S. S. (2019). Heart disease and stroke statistics—2019 update: A report from the American Heart Association. *Circulation, 139*(10), e56–e528. https://doi.org/10.1161/CIR.0000000000000659

Burnham, K. P., & Anderson, D. R. (2004). *Model selection and multimodel inference: A practical information-theoretic approach.* Springer Science & Business Media.

Centers for Disease Control and Prevention (CDC). (2022). *Behavioral Risk Factor Surveillance System.* Retrieved from https://www.cdc.gov/brfss/index.html

Centers for Disease Control and Prevention (CDC). (2023). *Behavioral Risk Factor Surveillance System (BRFSS).* Retrieved from https://www.cdc.gov/brfss/index.html

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM. https://doi.org/10.1145/2939672.2939785

Cornelissen, V. A., & Smart, N. A. (2013). Exercise training for blood pressure: A systematic review and meta-analysis. *Journal of the American Heart Association, 2*(1), e004473.

Frye, R. L., August, P., Brooks, M. M., Hardison, R. M., Kelsey, S. F., MacGregor, J. M., ... & Wilson, G. (2006). A randomized trial of therapies for type 2 diabetes and coronary artery disease. *New England Journal of Medicine, 356*(15), 1587–1602.

Graham, G. N., Ostrowski, M., & Sabina, A. (2022). Social determinants of health and health disparities: COVID-19 exposures and outcomes among African Americans in the United States. *Journal of Racial and Ethnic Health Disparities, 9*(1), 288-298. https://doi.org/10.1007/s40615-020-00856-y

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction.* Springer Science & Business Media.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics, 12*(1), 55-67.

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression* (3rd ed.). Wiley.

Lear, S. A., Hu, W., Rangarajan, S., Gasevic, D., Leong, D., Iqbal, R., ... & Yusuf, S. (2017). The effect of physical activity on mortality and cardiovascular disease in 130,000 people from 17 high-income, middle-income, and low-income countries: The PURE study. *The Lancet, 390*(10113), 2643–2654. https://doi.org/10.1016/S0140-6736(17)31634-3

Lichtman, J. H., Froelicher, E. S., Blumenthal, J. A., Carney, R. M., Doering, L. V., Frasure-Smith, N., ... & Wulsin, L. (2014). Depression as a risk factor for poor prognosis among patients with acute coronary syndrome: Systematic review and recommendations: A scientific statement from the American Heart Association. *Circulation, 129*(12), 1350–1369.

Marmot, M. (2005). Social determinants of health inequalities. *The Lancet, 365*(9464), 1099–1104. https://doi.org/10.1016/S0140-6736(05)74234-3

Mistry, R., Rosansky, J., McGuire, K., McDermott, C., Jarvik, L., & Pavel, M. (2015). Self-rated health predicts coronary artery disease risk. *American Journal of Preventive Medicine, 48*(3), 278–284.

Osborne, J. W. (2010). Improving your data transformations: Applying the Box-Cox transformation. *Practical Assessment, Research, and Evaluation, 15*(12), 1–9.

Ridker, P. M., Buring, J. E., Rifai, N., & Cook, N. R. (2007). Development and validation of improved algorithms for the assessment of global cardiovascular risk in women: The Reynolds Risk Score. *JAMA, 297*(6), 611–619. https://doi.org/10.1001/jama.297.6.611

Rosano, G. M. C., Vitale, C., Marazzi, G., & Volterrani, M. (2007). Menopause and cardiovascular disease: The evidence. *Climacteric, 10*(1), 19–24.

Steyerberg, E. W., Vickers, A. J., Cook, N. R., Gerds, T., Gonen, M., Obuchowski, N., ... & Altman, D. G. (2010). Assessing the performance of prediction models: A framework for traditional and novel measures. *Epidemiology, 21*(1), 128-138.

Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society: Series B (Methodological), 58*(1), 267–288.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., ... & Altman, R. B. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics, 17*(6), 520–525.

Virani, S. S., Alonso, A., Aparicio, H. J., Benjamin, E. J., Bittencourt, M. S., Callaway, C. W., ... & Tsao, C. W. (2021). Heart disease and stroke statistics—2021 update: A report from the American Heart Association. *Circulation, 143*(8), e254–e743. https://doi.org/10.1161/CIR.0000000000000950

Appendex A: Code

```
set.seed(123)
# libraries
library(tidyverse)
# library(haven)
library(sf)
library(tableone)
library(ggplot2)
library(gridExtra)
library(forcats)
library(dplyr)
library(Hmisc)
library(writexl)
library(GGally)
library(reshape2)
library(VIM)
library(caret)
library(glmnet)
library(MASS)
library(missForest)
library(pROC)
library(xgboost)
library(nnet)
library(maps)
library(knitr)
library(kableExtra)

brfss_df <- read.csv("data/BRFSS/raw-csv-files/Filtered_LLCP2023.csv")

dim(brfss_df)

# colnames(brfss_df)

# Starting with choosing the relevant variables and performing data exploration:
selected_vars <- c("X_AGEG5YR", "SEXVAR", "INCOME3", "EDUCA", "SDLONELY", "EMTSUPRT",
                   "MENTHLTH", "SDHFOOD1", "SMOKE100", "PERSDOC3", "BPHIGH6", "PHYSHLTH",
                   "EXERANY2", "DIABETE4", "ALCDAY4", "GENHLTH", "CVDCRHD4", "CVDSTRK3", "TOLDHI3",
                   "STRENGTH", "EXERHMM2", "EXEROFT2", "EXRACT22", "CHECKUP1", "MEDCOST1",
                   "PRIMINS1", "HHADULT", "MARITAL", "RENTHOM1", "EMPLOY1", "WEIGHT2", "HEIGHT3",
                   "AVEDRNK3", "COVIDPO1", "X_STATE")

# Subset
# brfss_small <- brfss_df[, selected_vars]

# dim(brfss_small)
# temp save #1
# write.csv(brfss_small, file = "data/brfss_small.csv", fileEncoding = "UTF-8")
```

```r
brfss_small <- read.csv("data/brfss_small.csv", fileEncoding = "UTF-8")

# str(brfss_small)
# head(brfss_small)
# summary(brfss_small)

# missing values
sapply(brfss_small, function(x) sum(is.na(x)))

## CODEBOOK

# This information was taken and summarized from the main codebook of the BRFSS dataset: https://www.cd
#
# X_AGEG5YR: Categorical: age groups in 5-year chunks.
# Five-year age groups (e.g., 18-24, 25-29, etc.).
#
# SEXVAR: Categorical.
# Binary (1 = Male, 2 = Female).
#
# INCOME3: Categorical.
# Income level categories (e.g., less than 10,000, 10,000-14,999, 15,000-19,999, etc.).
#
# EDUCA: Categorical.
# Education level categories (e.g., less than high school, high school graduate, some college, etc.).
#
# NUMADULT: Numeric: Number of Adults in Household.
# Number of people living in the household.
#
# SDLONELY: Categorical: How often do you feel lonely?
# Categorical (1 = always, 2 = usually, 3 = sometimes, ... etc).
#
# EMTSUPRT: Categorical: How often get emotional support needed.
# Categorical (1 = always, 2 = usually, 3 = sometimes, ... etc).
#
# MENTHLTH: Numeric: Number of Days Mental Health Not Good.
# Number of days mental health was poor in the past 30 days (0-30).
#
# SDHFOOD1: Categorical: How often did the food that you bought not last, and you didn't have money to g
# Categorical (1 = always, 2 = usually, 3 = sometimes, ... etc).
#
# SMOKE100: Categorical: Smoked at Least 100 Cigarettes.
# Binary (1 = Smoked at least 100 cigarettes in lifetime, 2 = Did not).
#
# PERSDOC3: Categorical: Have Personal Health Care Provider?
# Binary (1 = yes, only 1, 2 = more than one, 3 = no).
#
# BPHIGH6: Categorical: Ever Told Blood Pressure High.
# Binary (1 or 2 = yes, 3 = No, 4 = told borderline or pre-HTN).
#
# PHYSHLTH: Numeric: Number of Days Physical Health Not Good.
# Number of days physical health was poor in the past 30 days (0-30).
#
# EXERANY2: Categorical: Exercise in Past 30 Days.
```

```
# Binary (1 = Participated in physical activity in the past month, 2 = Did not).
#
# DIABETE4: Categorical: (Ever told) you had diabetes.
# Multiple categories (1 or 2 = yes, 3 = No, 4 = yes for prediabetes).
#
# ALCDAY4: Numeric: Days in past 30 had alcoholic beverage.
# 101 - 199: Days per week (Notes: 1\_ \_ = Days per week).
# 201 - 299: Days in past 30 days (Notes: 2\_ \_ = Days in past 30).
# Average number of alcoholic drinks consumed per day over the past 30 days.
#
# GENHLTH: Categorical: General Health.
# Self-reported general health (1 = Excellent, 2 = Very good, 3 = Good, 4 = Fair, 5 = Poor).
#
# CVDCRHD4: Categorical: Ever Diagnosed with Angina or Coronary Heart Disease.
# Binary (1 = yes, 2 = No).
#
# CVDSTRK3: Categorical: Ever Diagnosed with a Stroke.
# Binary (1 = yes, 2 = No).
#
# TOLDHI3: Categorical: Ever Told Cholesterol Is High.
# Binary (1 = yes, 2 = No).
#
# STRENGTH: Numeric: Strength Activity Frequency per Week.
# 0 - 98999: Strength Activity times per week (3 implied decimal places).
#
# EXERHMM2: Numeric: Minutes or Hours Walking, Running, Jogging, or Swimming.
# 1 - 759: Hours and Minutes.
# 800 - 959: Hours and Minutes.
#
# EXEROFT2: Numeric: How Many Times Walking, Running, Jogging, or Swimming.
# 101 - 199: times per week.
# 201 - 299: times per month.
#
# EXRACT22: Categorical: Other Type of Physical Activity Giving Most Exercise During Past Month.
# Type of second physical activity (e.g., walking, running).
#
# CHECKUP1: Categorical: Length of time since last routine checkup.
# Last time visited a doctor for a routine checkup (e.g., within the past year, within the past 2 years
# ( 1 = Within past year (anytime less than 12 months ago), 2 = Within past 2 years (1 year but less tha
#
# MEDCOST1: Categorical: Could Not Afford To See Doctor.
# Binary (1 = yes, 2 = no).
#
# PRIMINS1: Categorical: What is Current Primary Source of Health Insurance?
# Binary (1 = A plan purchased through an employer or union (including plans purchased through another p
# etc).
#
# HHADULT: Numeric: Number of Adults in Household.
# Number of adults in the household.
#
# MARITAL: Categorical: Marital Status.
# Marital status (e.g., 1 = married, 2 = divorced, 3 = widowed, 4 = separated, 5 = never married, 6 = un
#
# RENTHOM1: Categorical: Own or Rent Home.
```

```
# Housing ownership (1 = Own, 2 = Rent, 3 = Other arrangement).
#
# EMPLOY1: Categorical: Employment Status.
# Employment status (e.g., 1 = employed, 2 = self-0employed, 3 = out of work for 1 year or more, 4 out
#
# WEIGHT2: Numeric: Reported Weight in Pounds.
# 50 - 0776: Weight (pounds) (Notes: 0 \_ \_ \_ = weight in pounds).
# 9023 - 9352: Weight (kilograms) (Notes: The initial ´9´ indicates this was a metric value.)
#
# HEIGHT3: Numeric: Reported Height in Feet and Inches.
# 200 - 711: Height (ft/inches) (Notes: 0 \_ / \_ \_ = feet / inches).
# 9061 - 9998: Height (meters/centimeters) (Notes: The initial ´9 ´ indicates this was a metric value.)
#
# AVEDRNK3: Numeric: Avg alcoholic drinks per day in past 30.
# Average number of alcoholic drinks consumed per day in the past 30 days.
#
# COVIDPO1: Categorical: Have you ever tested positive for COVID-19?
# Binary (1 = yes, 2 = no).
#
# X_STATE: Categorical.
# State of residence (numerical codes for each U.S. state).
#brfss_small <- brfss_small %>% select(-X)

# grouping vars
categorical_vars <- c("X_AGEG5YR", "SEXVAR", "INCOME3", "EDUCA", "SDLONELY", "EMTSUPRT", "SDHFOOD1",
                      "SMOKE100", "PERSDOC3", "BPHIGH6", "EXERANY2", "DIABETE4", "GENHLTH",
                      "CVDCRHD4", "CVDSTRK3", "TOLDHI3", "EXRACT22", "CHECKUP1", "MEDCOST1",
                      "PRIMINS1", "MARITAL", "RENTHOM1", "EMPLOY1", "COVIDPO1", "X_STATE")
continuous_vars <- c("MENTHLTH", "PHYSHLTH", "ALCDAY4", "STRENGTH", "EXERHMM2", "EXEROFT2",
                     "HHADULT", "WEIGHT2", "HEIGHT3", "AVEDRNK3")


# unifying how missing data is coded (ie, merging the "no response", "don't know", "refused to respond"
missing_values <- list(
  X_AGEG5YR = c("14"),
  SEXVAR = c(""),
  INCOME3 = c("77", "99", ""),
  EDUCA = c("9", ""),
  SDLONELY = c("7", "9", ""),
  EMTSUPRT = c("7", "9", ""),
  SDHFOOD1 = c("7", "9", ""),
  SMOKE100 = c("7", "9", ""),
  PERSDOC3 = c("7", "9", ""),
  BPHIGH6 = c("7", "9", ""),
  EXERANY2 = c("7", "9", ""),
  DIABETE4 = c("7", "9", ""),
  GENHLTH = c("7", "9", ""),
  CVDCRHD4 = c("7", "9", ""),
  CVDSTRK3 = c("7", "9", ""),
  TOLDHI3 = c("7", "9", ""),
  EXRACT22 = c("77", "99", ""),
  CHECKUP1 = c("7", "9", "6", ""),
  MEDCOST1 = c("7", "9", ""),
  PRIMINS1 = c("77", "99", "12", ""),
```

```
  MARITAL = c("9", ""),
  RENTHOM1 = c("7", "9", ""),
  EMPLOY1 = c("9", ""),
  COVIDPO1 = c("7", "9", ""),
  X_STATE = c("3", "7", "14", "21", "42", "43", "52", ""),
  MENTHLTH = c("", "77", "99"),
  PHYSHLTH = c("", "77", "99"),
  ALCDAY4 = c("777", "999", ""),
  STRENGTH = c("777", "999", ""),
  EXERHMM2 = c("999", "777", ""),
  EXEROFT2 = c("777", "999", ""),
  HHADULT = c("", "77", "99"),
  WEIGHT2 = c("", "7777", "9999"),
  HEIGHT3 = c("", "7777", "9999"),
  AVEDRNK3 = c("77", "99", ""))


# cat vars
clean_categorical <- function(data, variable, missing_vals) {
  data[[variable]] <- ifelse(data[[variable]] %in% missing_vals, "", data[[variable]])
  data[[variable]] <- as.factor(data[[variable]])
  return(data)}

# cont vars
clean_continuous <- function(data, variable, missing_vals) {
  data[[variable]] <- ifelse(data[[variable]] %in% missing_vals, NA_real_, as.numeric(data[[variable]]))
  return(data)}

# apply
for (var in names(missing_values)) {
  if (var %in% categorical_vars) {
    brfss_small <- clean_categorical(brfss_small, var, missing_values[[var]])
  } else if (var %in% continuous_vars) {
    brfss_small <- clean_continuous(brfss_small, var, missing_values[[var]])}}


#unified coding of missingness
brfss_small[brfss_small == "NA"] <- ""
brfss_small[brfss_small == ""] <- NA
#brfss_small[is.na(brfss_small)] <- ""


# correcting the coding of the Alcohol variable and the height and weight, and primary insurance, among
# ALCDAY4
brfss_small <- brfss_small %>%
  mutate(ALCDAY4 = case_when(
    ALCDAY4 >= 101 & ALCDAY4 <= 199 ~ (ALCDAY4 %/% 100) * 7,
    ALCDAY4 >= 201 & ALCDAY4 <= 299 ~ ALCDAY4 %% 100,
    ALCDAY4 == 888 ~ 0,
    TRUE ~ NA_real_))

# STRENGTH
brfss_small <- brfss_small %>%
  mutate(STRENGTH = case_when(
```

```
    STRENGTH >= 101 & STRENGTH <= 199 ~ (STRENGTH %/% 100) * 4,
    STRENGTH >= 201 & STRENGTH <= 299 ~ STRENGTH %% 100,
    STRENGTH == 888 ~ 0,
    TRUE ~ NA_real_))

# EXERHMM2
brfss_small <- brfss_small %>%
  mutate(EXERHMM2 = case_when(
    EXERHMM2 >= 1 & EXERHMM2 <= 759 ~ (EXERHMM2 %/% 100) * 60 + (EXERHMM2 %% 100),
    EXERHMM2 >= 800 & EXERHMM2 <= 959 ~ (EXERHMM2 %/% 100) * 60,
    EXERHMM2 == 777 ~ NA_real_,
    TRUE ~ NA_real_))

# EXEROFT2
brfss_small <- brfss_small %>%
  mutate(EXEROFT2 = case_when(
    EXEROFT2 >= 101 & EXEROFT2 <= 199 ~ (EXEROFT2 %/% 100) * 4,
    EXEROFT2 >= 201 & EXEROFT2 <= 299 ~ EXEROFT2 %% 100,
    TRUE ~ NA_real_))

# WEIGHT2 (& converting both to lbs)
brfss_small <- brfss_small %>%
  mutate(WEIGHT2 = case_when(
    WEIGHT2 >= 50 & WEIGHT2 <= 776 ~ WEIGHT2,
    WEIGHT2 >= 9023 & WEIGHT2 <= 9352 ~ (WEIGHT2 %% 9000) * 2.20462,
    TRUE ~ NA_real_))

# HEIGHT3 (& converting to inches)
brfss_small <- brfss_small %>%
  mutate(HEIGHT3 = case_when(
    HEIGHT3 >= 200 & HEIGHT3 <= 711 ~ ((HEIGHT3 %/% 100) * 12) + (HEIGHT3 %% 100),
    HEIGHT3 >= 9061 & HEIGHT3 <= 9998 ~ (HEIGHT3 %% 9000) / 2.54,
    TRUE ~ NA_real_))

# AVEDRNK3
brfss_small <- brfss_small %>%
  mutate(AVEDRNK3 = ifelse(AVEDRNK3 == 88, 0, AVEDRNK3))

# MENTHLTH  & PHYSHLTH
brfss_small <- brfss_small %>%
  mutate(
    MENTHLTH = ifelse(MENTHLTH == 88, 0, MENTHLTH),
    PHYSHLTH = ifelse(PHYSHLTH == 88, 0, PHYSHLTH))

# Creating **Table 1** to summarize the characteristics of the study participants:

# labels for the variables themselves
variable_labels <- c(
  X_AGEG5YR = "Age Groups (5-Year)",
  SEXVAR = "Sex",
  INCOME3 = "Income Level",
  EDUCA = "Education Level",
  SDLONELY = "Loneliness Frequency",
  EMTSUPRT = "Emotional Support Frequency",
```

```
  MENTHLTH = "Mental Health (Past 30 Days)",
  SDHFOOD1 = "Food Insecurity Frequency",
  SMOKE100 = "Ever Smoked 100+ Cigarettes",
  PERSDOC3 = "Has Personal Health Care Provider",
  BPHIGH6 = "Hypertension",
  PHYSHLTH = "Physical Health (Past 30 Days)",
  EXERANY2 = "Exercised in Past 30 Days",
  DIABETE4 = "Diabetes",
  ALCDAY4 = "Alcohol Consumption (Days in Past 30)",
  GENHLTH = "General Health",
  CVDCRHD4 = "Coronary Heart Disease",
  CVDSTRK3 = "Stroke",
  TOLDHI3 = "High Cholesterol",
  STRENGTH = "Strength Activity Frequency (Per Week)",
  EXERHMM2 = "Minutes/Hours of Physical Activity",
  EXEROFT2 = "Frequency of Physical Activity",
  EXRACT22 = "Type of Physical Activity",
  CHECKUP1 = "Time Since Last Routine Checkup",
  MEDCOST1 = "Could Not Afford Doctor",
  PRIMINS1 = "Health Insurance",
  HHADULT = "Number of Adults in Household",
  MARITAL = "Marital Status",
  RENTHOM1 = "Housing Ownership",
  EMPLOY1 = "Employment Status",
  WEIGHT2 = "Weight (lbs)",
  HEIGHT3 = "Height (ft/in)",
  AVEDRNK3 = "Average Alcoholic Drinks per Day",
  COVIDPO1 = "Ever Tested Positive for COVID-19",
  X_STATE = "State of Residence")
for (var in names(variable_labels)) {
  label(brfss_small[[var]]) <- variable_labels[[var]]}

# adding labels for the categorical variable values:
brfss_small <- brfss_small %>%
  mutate(
    X_AGEG5YR = fct_recode(factor(X_AGEG5YR),
      "18-24" = "1",
      "25-29" = "2",
      "30-34" = "3",
      "35-39" = "4",
      "40-44" = "5",
      "45-49" = "6",
      "50-54" = "7",
      "55-59" = "8",
      "60-64" = "9",
      "65-69" = "10",
      "70-74" = "11",
      "75-79" = "12",
      "80+" = "13"),
    SEXVAR = fct_recode(factor(SEXVAR),
      "Male" = "1", "Female" = "2"),
    INCOME3 = fct_recode(factor(INCOME3),
      "<$10k" = "1",
      "$10k-<$15k" = "2",
```

```
  "$15k-<$20k" = "3",
  "$20k-<$25k" = "4",
  "$25k-<$35k" = "5",
  "$35k-<$50k" = "6",
  "$50k-<$75k" = "7",
  "$75k-<$100k" = "8",
  "$100k-<$150k" = "9",
  "$150k-<$200k" = "10",
  "$200k+" = "11"),
EDUCA = fct_recode(factor(EDUCA),
  "No School/Kindergarten" = "1",
  "Elementary (Grades 1-8)" = "2",
  "Some High School (Grades 9-11)" = "3",
  "High School Graduate/GED" = "4",
  "Some College/Technical" = "5",
  "College Graduate (4+ years)" = "6"),
SDLONELY = fct_recode(factor(SDLONELY),
  "Always" = "1", "Usually" = "2", "Sometimes" = "3",
  "Rarely" = "4", "Never" = "5"),
EMTSUPRT = fct_recode(factor(EMTSUPRT),
  "Always" = "1", "Usually" = "2", "Sometimes" = "3",
  "Rarely" = "4", "Never" = "5"),
SDHFOOD1 = fct_recode(factor(SDHFOOD1),
  "Always" = "1", "Usually" = "2", "Sometimes" = "3",
  "Rarely" = "4", "Never" = "5"),
SMOKE100 = fct_recode(factor(SMOKE100),
  "Yes" = "1", "No" = "2"),
PERSDOC3 = fct_recode(factor(PERSDOC3),
  "Yes, only one" = "1", "More than one" = "2", "No" = "3"),

BPHIGH6 = fct_recode(factor(BPHIGH6),
  "Yes" = "1",
  "Pregnancy-related" = "2",
  "No" = "3",
  "Borderline/Pre-HTN" = "4"),
EXERANY2 = fct_recode(factor(EXERANY2),
  "Yes" = "1",
  "No" = "2"),
DIABETE4 = fct_recode(factor(DIABETE4),
  "Yes" = "1",
  "Pregnancy-related" = "2",
  "No" = "3",
  "Pre-diabetes" = "4"),
GENHLTH = fct_recode(factor(GENHLTH),
  "Excellent" = "1",
  "Very good" = "2",
  "Good" = "3",
  "Fair" = "4",
  "Poor" = "5"),
CVDCRHD4 = fct_recode(factor(CVDCRHD4),
  "Yes" = "1",
  "No" = "2"),
CVDSTRK3 = fct_recode(factor(CVDSTRK3),
  "Yes" = "1",
```

```
  "No" = "2"),
TOLDHI3 = fct_recode(factor(TOLDHI3),
  "Yes" = "1",
  "No" = "2"),
EXRACT22 = fct_recode(factor(EXRACT22),
  "Walking" = "1",
  "Running" = "2",
  "Gardening" = "3",
  "Bicycling" = "4",
  "Aerobics" = "5",
  "Calisthenics" = "6",
  "Elliptical" = "7",
  "Household activities" = "8",
  "Weight lifting" = "9",
  "Yoga, Pilates, or Tai Chi" = "10",
  "Other" = "11",
  "No activity" = "88"),
CHECKUP1 = fct_recode(factor(CHECKUP1),
  "Past year" = "1",
  "Past 2 years" = "2",
  "Past 5 years" = "3",
  "5+ years ago" = "4",
  "Never" = "8"),
MEDCOST1 = fct_recode(factor(MEDCOST1),
  "Yes" = "1",
  "No" = "2"),
PRIMINS1 = fct_recode(factor(PRIMINS1),
  "Employer/Union Plan" = "1",
  "Private Plan" = "2",
  "Medicare" = "3",
  "Medigap" = "4",
  "Medicaid" = "5",
  "CHIP" = "6",
  "Military/VA" = "7",
  "Indian Health" = "8",
  "State Plan" = "9",
  "Other" = "10",
  "No Coverage" = "88"),
MARITAL = fct_recode(factor(MARITAL),
  "Married" = "1",
  "Divorced" = "2",
  "Widowed" = "3",
  "Separated" = "4",
  "Never married" = "5",
  "Unmarried couple" = "6"),
RENTHOM1 = fct_recode(factor(RENTHOM1),
  "Own" = "1",
  "Rent" = "2",
  "Other" = "3"),
EMPLOY1 = fct_recode(factor(EMPLOY1),
  "Employed" = "1",
  "Self-employed" = "2",
  "Unemployed 1+ years" = "3",
  "Unemployed <1 year" = "4",
```

```
      "Homemaker" = "5",
      "Student" = "6",
      "Retired" = "7",
      "Unable to work" = "8"),
    COVIDPO1 = fct_recode(factor(COVIDPO1),
      "Yes" = "1",
      "No" = "2"),
    X_STATE = fct_recode(factor(X_STATE),
      "Alabama" = "1", "Alaska" = "2", "Arizona" = "4", "Arkansas" = "5",
      "California" = "6", "Colorado" = "8", "Connecticut" = "9", "Delaware" = "10",
      "District of Columbia" = "11", "Florida" = "12", "Georgia" = "13", "Hawaii" = "15",
      "Idaho" = "16", "Illinois" = "17", "Indiana" = "18", "Iowa" = "19", "Kansas" = "20",
      "Louisiana" = "22", "Maine" = "23", "Maryland" = "24", "Massachusetts" = "25",
      "Michigan" = "26", "Minnesota" = "27", "Mississippi" = "28", "Missouri" = "29",
      "Montana" = "30", "Nebraska" = "31", "Nevada" = "32", "New Hampshire" = "33",
      "New Jersey" = "34", "New Mexico" = "35", "New York" = "36", "North Carolina" = "37",
      "North Dakota" = "38", "Ohio" = "39", "Oklahoma" = "40", "Oregon" = "41",
      "Rhode Island" = "44", "South Carolina" = "45", "South Dakota" = "46",
      "Tennessee" = "47", "Texas" = "48", "Utah" = "49", "Vermont" = "50", "Virginia" = "51",
      "Washington" = "53", "West Virginia" = "54", "Wisconsin" = "55", "Wyoming" = "56",
      "Guam" = "66", "Puerto Rico" = "72", "Virgin Islands" = "78"))

# var types
brfss_small <- brfss_small %>%
  mutate(
    X_AGEG5YR = as.factor(X_AGEG5YR),
    SEXVAR = as.factor(SEXVAR),
    INCOME3 = as.factor(INCOME3),
    EDUCA = as.factor(EDUCA),
    SDLONELY = as.factor(SDLONELY),
    EMTSUPRT = as.factor(EMTSUPRT),
    MENTHLTH = as.numeric(MENTHLTH),
    SDHFOOD1 = as.factor(SDHFOOD1),
    SMOKE100 = as.factor(SMOKE100),
    PERSDOC3 = as.factor(PERSDOC3),
    BPHIGH6 = as.factor(BPHIGH6),
    PHYSHLTH = as.numeric(PHYSHLTH),
    EXERANY2 = as.factor(EXERANY2),
    DIABETE4 = as.factor(DIABETE4),
    ALCDAY4 = as.numeric(ALCDAY4),
    GENHLTH = as.factor(GENHLTH),
    CVDCRHD4 = as.factor(CVDCRHD4),
    CVDSTRK3 = as.factor(CVDSTRK3),
    TOLDHI3 = as.factor(TOLDHI3),
    STRENGTH = as.numeric(STRENGTH),
    EXERHMM2 = as.numeric(EXERHMM2),
    EXEROFT2 = as.numeric(EXEROFT2),
    EXRACT22 = as.factor(EXRACT22),
    CHECKUP1 = as.factor(CHECKUP1),
    MEDCOST1 = as.factor(MEDCOST1),
    PRIMINS1 = as.factor(PRIMINS1),
    HHADULT = as.numeric(HHADULT),
    MARITAL = as.factor(MARITAL),
    RENTHOM1 = as.factor(RENTHOM1),
```

```
    EMPLOY1 = as.factor(EMPLOY1),
    WEIGHT2 = as.numeric(WEIGHT2),
    HEIGHT3 = as.numeric(HEIGHT3),
    AVEDRNK3 = as.numeric(AVEDRNK3),
    COVIDPO1 = as.factor(COVIDPO1),
    X_STATE = as.factor(X_STATE))



# temp save #2
# write.csv(brfss_small, file = "data/brfss_small2_labelled_cleaned.csv", fileEncoding = "UTF-8")
# brfss_small <- read.csv("data/brfss_small2_labelled_cleaned.csv", fileEncoding = "UTF-8")

# building a table 1 to summarize the characteristics of the study participants
brfss_table1 <- CreateTableOne(
  vars = c(categorical_vars, continuous_vars),
  data = brfss_small,
  factorVars = categorical_vars)

table1_df <- as.data.frame(print(brfss_table1, showAllLevels = TRUE))
table1_df <- cbind(
  Variable = sapply(rownames(table1_df), function(x) {
    var_name <- sub("\\.\\.*.*$", "", x)
    if (var_name %in% names(variable_labels)) {
      variable_labels[[var_name]]
    } else {x}}), table1_df)
# save to an excel file
write_xlsx(table1_df, "Table1_BRFSS_Cleaned_Labels.xlsx")

#### DATA EXPLORATION

# continuous variables are numeric
brfss_small[continuous_vars] <- lapply(brfss_small[continuous_vars], function(x) as.numeric(as.characte

# categorical variables are factors
brfss_small[categorical_vars] <- lapply(brfss_small[categorical_vars], function(x) as.factor(as.charact

# function to plot variables using labels
plot_variable <- function(data, var) {
  var_label <- variable_labels[[var]] %||% var
  rotate_text <- function(labels) {
    any(sapply(labels, function(label) nchar(label) > 5))
  }
  if (is.factor(data[[var]])) {
    # Cat vars
    p <- ggplot(data, aes(x = .data[[var]])) +
      geom_bar(fill = "skyblue", color = "black", alpha = 0.7) +
      scale_x_discrete(drop = FALSE) +
      labs(title = paste("Frequency of", var_label), x = var_label, y = "Count") +
      theme_minimal() +
      theme(axis.text.x = element_text(
        angle = ifelse(rotate_text(levels(data[[var]])), 45, 0),
        hjust = ifelse(rotate_text(levels(data[[var]])), 1, 0.5)
      ))
```

```
  } else if (is.numeric(data[[var]])) {
    # Cont vars
    bin_width <- (max(data[[var]], na.rm = TRUE) - min(data[[var]], na.rm = TRUE)) / 30
    p <- ggplot(data, aes(x = .data[[var]])) +
      geom_histogram(binwidth = bin_width, fill = "orange", color = "black", alpha = 0.7) +
      labs(title = paste("Distribution of", var_label), x = var_label, y = "Frequency") +
      theme_minimal()
  } else {
    message(paste("Variable", var, "is neither numeric nor categorical. Skipping."))
    return(NULL)}
  return(p)}


variables_to_plot <- names(brfss_small)
variables_to_plot <- setdiff(variables_to_plot, "X")


plots <- lapply(variables_to_plot, function(var) plot_variable(brfss_small, var))
plots <- Filter(Negate(is.null), plots)
#plot_grid <- marrangeGrob(grobs = plots, ncol = 3, nrow = 12)
#ggsave("plots/brfss_updated_summary_plots_with_labels.png", plot = plot_grid, width = 20, height = 20,

# for better viewing:
plots_per_grid <- 4
plot_chunks <- split(plots, ceiling(seq_along(plots) / plots_per_grid))

for (i in seq_along(plot_chunks)) {
  plot_grid <- marrangeGrob(
    grobs = plot_chunks[[i]],
    ncol = 2, nrow = 2)
  ggsave(
    filename = sprintf("plots/brfss_summary_plots_grid_%02d.png", i),
    plot = plot_grid,
    width = 16, height = 12, units = "in")}

###
# temp save
# write.csv(brfss_small, file = "data/brfss_small_temp_viz.csv", fileEncoding = "UTF-8")
brfss_small <- read.csv("data/brfss_small_temp_viz.csv", fileEncoding = "UTF-8")
############################################################
#######- good visualizations to include in the paper itself

## (1) US Map and coronary artery disease

# Ensure CVDCRHD4 is numeric: 1 = Yes, 2 = No
brfss_small$CVDCRHD4 <- ifelse(brfss_small$CVDCRHD4 == "Yes", 1,
                               ifelse(brfss_small$CVDCRHD4 == "No", 0, NA))

# Load US map data
us_map <- map_data("state")

# Calculate prevalence by state
# Ensure dplyr functions are used explicitly
state_prevalence <- brfss_small %>%
```

```
  dplyr::group_by(X_STATE) %>%
  dplyr::summarize(prevalence = mean(CVDCRHD4 == 1, na.rm = TRUE)) %>%
  dplyr::mutate(X_STATE = tolower(X_STATE))


# Merge prevalence data with map data
us_map <- us_map %>%
  left_join(state_prevalence, by = c("region" = "X_STATE"))

# Plotting the map
usmap1 <- ggplot(us_map, aes(x = long, y = lat, group = group, fill = prevalence)) +
  geom_polygon(color = "white", size = 0.3) +
  coord_fixed(1.3) +
  scale_fill_viridis_c(
    option = "plasma",
    na.value = "grey80",
    name = "Prevalence (%)",
    breaks = seq(0.04, 0.08, by = 0.01),
    labels = scales::percent_format(accuracy = 1)) +
  labs(
    title = "",
    caption = "Source: BRFSS Dataset") +
  theme_minimal() +
  theme(
    axis.text = element_blank(),
    axis.title = element_blank(),
    axis.ticks = element_blank(),
    panel.grid = element_blank(),
    panel.background = element_rect(fill = "white", color = NA),
    legend.position = "bottom",
    legend.title = element_text(size = 12, face = "bold"),
    legend.text = element_text(size = 10),
    legend.key.height = unit(0.6, "cm"),
    legend.key.width = unit(1.5, "cm"),
    plot.title = element_text(size = 18, face = "bold", hjust = 0.5),
    plot.caption = element_text(size = 10, hjust = 1))

ggsave(
  filename = "plots/coronary_disease_prevalence_map.png",
  plot = usmap1,
  width = 8, height = 6, units = "in", dpi = 300)

brfss_small <- read.csv("data/brfss_small_temp_viz.csv", fileEncoding = "UTF-8")

# Bar plots for categorical variables
cat_bar_plots <- lapply(categorical_vars, function(var) {
  ggplot(brfss_small, aes_string(x = var)) +
    geom_bar(fill = "skyblue", color = "black", alpha = 0.7) +
    labs(title = paste("Frequency of", variable_labels[[var]] %||% var),
         x = variable_labels[[var]] %||% var, y = "Count") +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))})
```

```
# Density plots for continuous variables
cont_density_plots <- lapply(continuous_vars, function(var) {
  ggplot(brfss_small, aes_string(x = var)) +
    geom_density(fill = "orange", alpha = 0.7) +
    labs(title = paste("Density of", variable_labels[[var]] %||% var),
         x = variable_labels[[var]] %||% var, y = "Density") +
    theme_minimal()})


# Box plots for continuous variables
cont_box_plots <- lapply(continuous_vars, function(var) {
  ggplot(brfss_small, aes_string(y = var)) +
    geom_boxplot(fill = "skyblue", color = "black", alpha = 0.7) +
    labs(title = paste("Box Plot of", variable_labels[[var]] %||% var),
         y = variable_labels[[var]] %||% var, x = "") +
    theme_minimal()})


# Pair plots for continuous variables
pair_plot <- ggpairs(brfss_small[continuous_vars])


# Correlation heatmap for continuous variables
correlation_matrix <- cor(brfss_small[continuous_vars], use = "complete.obs")
correlation_matrix[lower.tri(correlation_matrix, diag = TRUE)] <- NA
correlation_melted <- melt(correlation_matrix, na.rm = TRUE)
correlation_heatmap <- ggplot(correlation_melted, aes(Var1, Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "red", high = "blue", mid = "white", midpoint = 0) +
  labs(title = "Correlation Heatmap",
       x = "Variables",
       y = "Variables") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.title.x = element_blank(),
        axis.title.y = element_blank())


# Save all visualizations to a folder
dir.create("plots", showWarnings = FALSE)
lapply(seq_along(cat_bar_plots), function(i) {
  ggsave(filename = paste0("plots/cat_bar_", categorical_vars[i], ".png"),
         plot = cat_bar_plots[[i]], width = 8, height = 6)})
lapply(seq_along(cont_density_plots), function(i) {
  ggsave(filename = paste0("plots/cont_density_", continuous_vars[i], ".png"),
         plot = cont_density_plots[[i]], width = 8, height = 6)})
lapply(seq_along(cont_box_plots), function(i) {
  ggsave(filename = paste0("plots/cont_box_", continuous_vars[i], ".png"),
         plot = cont_box_plots[[i]], width = 8, height = 6)})
ggsave("plots/pair_plot.png", plot = pair_plot, width = 12, height = 12)
ggsave("plots/correlation_heatmap.png", plot = correlation_heatmap, width = 10, height = 8)


## Data Preparation, Transformation, and Creating new variables
```

```
brfss_small <- brfss_small %>%
  mutate(
    weight_kg = WEIGHT2 * 0.453592, # lbs to kg
    height_m = HEIGHT3 * 0.0254,   # in to m
    BMI = weight_kg / (height_m^2))

ggplot(brfss_small, aes(x = BMI)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Distribution of BMI", x = "BMI", y = "Frequency") +
  theme_minimal()
ggsave("plots/BMI_distribution_histogram.png", width = 8, height = 6)

ggplot(brfss_small, aes(y = BMI)) +
  geom_boxplot(fill = "skyblue", color = "black", alpha = 0.7) +
  labs(title = "Box Plot of BMI", y = "BMI", x = "") +
  theme_minimal()
ggsave("plots/BMI_box_plot.png", width = 8, height = 6)

# temp save #3
# write.csv(brfss_small, file = "data/brfss_small3_labelled_cleaned.csv", fileEncoding = "UTF-8")
# brfss_small <- read.csv("data/brfss_small3_labelled_cleaned.csv", fileEncoding = "UTF-8")

#
# **Transformations**
#
# Looking at the box plots and density plots of the continuous variables, it seems that we can conclude


variables_to_transform <- c("ALCDAY4", "AVEDRNK3", "EXERHMM2", "EXEROFT2", "MENTHLTH", "PHYSHLTH", "STR
original_vars <- brfss_small[variables_to_transform]

# (log requires positive values)
transformed_vars <- lapply(original_vars, function(x) ifelse(x > 0, log(x), NA))

# temp save #4
# write.csv(brfss_small, file = "data/brfss_small4_transformed.csv", fileEncoding = "UTF-8")
# brfss_small <- read.csv("data/brfss_small4_transformed.csv", fileEncoding = "UTF-8")

# Combine original and transformed data into a long format for plotting
plot_data <- bind_rows(
  as.data.frame(original_vars) %>% mutate(type = "Original"),
  as.data.frame(transformed_vars) %>% mutate(type = "Transformed"))
plot_data_long <- pivot_longer(
  plot_data,
  cols = -type,
  names_to = "variable",
  values_to = "value")

pre_post_transformation <- ggplot(plot_data_long, aes(x = value, fill = type)) +
  geom_density(alpha = 0.5) +
  facet_wrap(~ variable, scales = "free", ncol = 2) +
  labs(
    title = "Density Plots Before and After Log Transformation",
```

```
    x = "Value",
    y = "Density",
    fill = "Data Type") +
  theme_minimal() +
  theme(
    strip.text = element_text(size = 10, face = "bold"),
    legend.position = "bottom")
ggsave(filename = "plots/pre_post_transformation_density_plot.png",
  plot = pre_post_transformation,
  width = 12,
  height = 12,
  units = "in",
  dpi = 300)

## MISSING DATA IMPUTATION

# removing unnecessary index columns
brfss_small <- brfss_small %>%
  select(-X.1, -X)

# checking missing values
missing_summary <- sapply(brfss_small, function(x) sum(is.na(x)))
missing_percentage <- sapply(brfss_small, function(x) mean(is.na(x)) * 100)

missing_data_summary <- data.frame(
  Variable = names(brfss_small),
  Missing_Count = missing_summary,
  Missing_Percentage = missing_percentage)

missing_data_summary <- missing_data_summary %>%
  arrange(desc(Missing_Percentage))

print(missing_data_summary)

## --

# recoding the factor/categorical variables using numerical values rather than characters
brfss_small <- brfss_small %>%
  mutate(
    X_AGEG5YR = as.numeric(recode(X_AGEG5YR,
      "18-24" = "1", "25-29" = "2", "30-34" = "3", "35-39" = "4",
      "40-44" = "5", "45-49" = "6", "50-54" = "7", "55-59" = "8",
      "60-64" = "9", "65-69" = "10", "70-74" = "11", "75-79" = "12",
      "80+" = "13")),
    SEXVAR = as.numeric(recode(SEXVAR, "Male" = "1", "Female" = "2")),
    INCOME3 = as.numeric(recode(INCOME3,
      "<$10k" = "1", "$10k-<$15k" = "2", "$15k-<$20k" = "3",
      "$20k-<$25k" = "4", "$25k-<$35k" = "5", "$35k-<$50k" = "6",
      "$50k-<$75k" = "7", "$75k-<$100k" = "8", "$100k-<$150k" = "9",
      "$150k-<$200k" = "10", "$200k+" = "11")),
    EDUCA = as.numeric(recode(EDUCA,
      "No School/Kindergarten" = "1", "Elementary (Grades 1-8)" = "2",
      "Some High School (Grades 9-11)" = "3", "High School Graduate/GED" = "4",
      "Some College/Technical" = "5", "College Graduate (4+ years)" = "6")),
```

```
SDLONELY = as.numeric(recode(SDLONELY,
  "Always" = "1", "Usually" = "2", "Sometimes" = "3",
  "Rarely" = "4", "Never" = "5")),
EMTSUPRT = as.numeric(recode(EMTSUPRT,
  "Always" = "1", "Usually" = "2", "Sometimes" = "3",
  "Rarely" = "4", "Never" = "5")),
SDHFOOD1 = as.numeric(recode(SDHFOOD1,
  "Always" = "1", "Usually" = "2", "Sometimes" = "3",
  "Rarely" = "4", "Never" = "5")),
SMOKE100 = as.numeric(recode(SMOKE100, "Yes" = "1", "No" = "2")),
PERSDOC3 = as.numeric(recode(PERSDOC3,
  "Yes, only one" = "1", "More than one" = "2", "No" = "3")),
BPHIGH6 = as.numeric(recode(BPHIGH6,
  "Yes" = "1", "Pregnancy-related" = "2", "No" = "3", "Borderline/Pre-HTN" = "4")),
EXERANY2 = as.numeric(recode(EXERANY2, "Yes" = "1", "No" = "2")),
DIABETE4 = as.numeric(recode(DIABETE4,
  "Yes" = "1", "Pregnancy-related" = "2", "No" = "3", "Pre-diabetes" = "4")),
GENHLTH = as.numeric(recode(GENHLTH,
  "Excellent" = "1", "Very good" = "2", "Good" = "3",
  "Fair" = "4", "Poor" = "5")),
CVDCRHD4 = as.numeric(recode(CVDCRHD4, "Yes" = "1", "No" = "2")),
CVDSTRK3 = as.numeric(recode(CVDSTRK3, "Yes" = "1", "No" = "2")),
TOLDHI3 = as.numeric(recode(TOLDHI3, "Yes" = "1", "No" = "2")),
EXRACT22 = as.numeric(recode(EXRACT22,
  "Walking" = "1", "Running" = "2", "Gardening" = "3",
  "Bicycling" = "4", "Aerobics" = "5", "Calisthenics" = "6",
  "Elliptical" = "7", "Household activities" = "8",
  "Weight lifting" = "9", "Yoga, Pilates, or Tai Chi" = "10",
  "Other" = "11", "No activity" = "88")),
CHECKUP1 = as.numeric(recode(CHECKUP1,
  "Past year" = "1", "Past 2 years" = "2", "Past 5 years" = "3",
  "5+ years ago" = "4", "Never" = "8")),
MEDCOST1 = as.numeric(recode(MEDCOST1, "Yes" = "1", "No" = "2")),
PRIMINS1 = as.numeric(recode(PRIMINS1,
  "Employer/Union Plan" = "1", "Private Plan" = "2", "Medicare" = "3",
  "Medigap" = "4", "Medicaid" = "5", "CHIP" = "6", "Military/VA" = "7",
  "Indian Health" = "8", "State Plan" = "9", "Other" = "10", "No Coverage" = "88")),
MARITAL = as.numeric(recode(MARITAL,
  "Married" = "1", "Divorced" = "2", "Widowed" = "3",
  "Separated" = "4", "Never married" = "5", "Unmarried couple" = "6")),
RENTHOM1 = as.numeric(recode(RENTHOM1, "Own" = "1", "Rent" = "2", "Other" = "3")),
EMPLOY1 = as.numeric(recode(EMPLOY1,
  "Employed" = "1", "Self-employed" = "2", "Unemployed 1+ years" = "3",
  "Unemployed <1 year" = "4", "Homemaker" = "5", "Student" = "6",
  "Retired" = "7", "Unable to work" = "8")),
COVIDPO1 = as.numeric(recode(COVIDPO1, "Yes" = "1", "No" = "2")),
X_STATE = as.numeric(recode(X_STATE,
  "Alabama" = "1", "Alaska" = "2", "Arizona" = "4", "Arkansas" = "5",
  "California" = "6", "Colorado" = "8", "Connecticut" = "9",
  "Delaware" = "10", "District of Columbia" = "11", "Florida" = "12",
  "Georgia" = "13", "Hawaii" = "15", "Idaho" = "16", "Illinois" = "17",
  "Indiana" = "18", "Iowa" = "19", "Kansas" = "20", "Louisiana" = "22",
  "Maine" = "23", "Maryland" = "24", "Massachusetts" = "25", "Michigan" = "26",
  "Minnesota" = "27", "Mississippi" = "28", "Missouri" = "29", "Montana" = "30",
```

```
      "Nebraska" = "31", "Nevada" = "32", "New Hampshire" = "33", "New Jersey" = "34",
      "New Mexico" = "35", "New York" = "36", "North Carolina" = "37",
      "North Dakota" = "38", "Ohio" = "39", "Oklahoma" = "40", "Oregon" = "41",
      "Rhode Island" = "44", "South Carolina" = "45", "South Dakota" = "46",
      "Tennessee" = "47", "Texas" = "48", "Utah" = "49", "Vermont" = "50",
      "Virginia" = "51", "Washington" = "53", "West Virginia" = "54",
      "Wisconsin" = "55", "Wyoming" = "56", "Guam" = "66", "Puerto Rico" = "72",
      "Virgin Islands" = "78")))


# temp save #5
# write.csv(brfss_small, file = "data/brfss_small5_recoded.csv", fileEncoding = "UTF-8")
# brfss_small <- read.csv("data/brfss_small5_recoded.csv", fileEncoding = "UTF-8")

## GROUP NOTES:
# The missing values are:
#            Variable Missing_Count Missing_Percentage
# AVEDRNK3   AVEDRNK3        223588         51.5984612
# EMTSUPRT   EMTSUPRT        210276         48.5263879
# EXERHMM2   EXERHMM2        209788         48.4137699
# SDHFOOD1   SDHFOOD1        209505         48.3484606
# EXEROFT2   EXEROFT2        208801         48.1859952
# SDLONELY   SDLONELY        208751         48.1744565
# EXRACT22   EXRACT22        115638         26.6863287
# HHADULT     HHADULT         91472         21.1094265
# INCOME3     INCOME3         86623         19.9903998
# TOLDHI3     TOLDHI3         55084         12.7119954
# BMI             BMI         38431          8.8689038
# WEIGHT2     WEIGHT2         34142          7.8791110
# weight_kg weight_kg         34142          7.8791110
# COVIDPO1   COVIDPO1         34094          7.8680338
# ALCDAY4     ALCDAY4         29940          6.9093955
# SMOKE100   SMOKE100         22568          5.2081242
# HEIGHT3     HEIGHT3         22118          5.1042756
# height_m   height_m         22118          5.1042756
# PRIMINS1   PRIMINS1         18674          4.3094874
# STRENGTH   STRENGTH         14173          3.2707703
# PHYSHLTH   PHYSHLTH         10785          2.4889055
# MENTHLTH   MENTHLTH          8108          1.8711215
# X_AGEG5YR X_AGEG5YR          7779          1.7951967
# EMPLOY1     EMPLOY1          7681          1.7725807
# CHECKUP1   CHECKUP1          5781          1.3341087
# MARITAL     MARITAL          4289          0.9897928
# PERSDOC3   PERSDOC3          4234          0.9771002
# CVDCRHD4   CVDCRHD4          4231          0.9764079
# RENTHOM1   RENTHOM1          4090          0.9438687
# EDUCA         EDUCA          2325          0.5365513
# BPHIGH6     BPHIGH6          1919          0.4428567
# MEDCOST1   MEDCOST1          1538          0.3549315
# CVDSTRK3   CVDSTRK3          1474          0.3401620
# GENHLTH     GENHLTH          1262          0.2912377
# EXERANY2   EXERANY2          1251          0.2886992
# DIABETE4   DIABETE4           984          0.2270823
# X                 X             0          0.0000000
```

```
# SEXVAR       SEXVAR             0          0.0000000
# X_STATE     X_STATE             0          0.0000000
#
#
# Some social and personal well-being factors have a large proportion of missing values (AVEDRNK3, EMTSU
#
# On the other hands, there are other variables that have a smaller proportion of missing values (BMI, A
#
# We will also limit the data we are going to work with to the rows that do not have missing values witl


# Limiting the dataset to where the target variable is not missing
brfss_small <- brfss_small[!is.na(brfss_small$CVDCRHD4), ]

# removing the variables with a very high proportion of missing values, as well as removing weight and l
vars_to_remove <- c("AVEDRNK3", "EMTSUPRT", "EXERHMM2", "SDHFOOD1", "EXEROFT2", "SDLONELY", "WEIGHT2", "


brfss_small_filtered_subset <- brfss_small %>%
  dplyr::select(-all_of(vars_to_remove))

##---########################

### Imputation of continuous and categorical variables

# Continuous variables to be imputed using KNN
variables_to_impute_continuous <- c("BMI", "ALCDAY4", "STRENGTH", "PHYSHLTH", "MENTHLTH", "HHADULT")

# Categorical variables to be imputed using mode
variables_to_impute_categorical <- c(
  "X_AGEG5YR", "INCOME3", "EDUCA", "PERSDOC3", "BPHIGH6",
  "EXERANY2", "DIABETE4", "GENHLTH", "CVDSTRK3", "TOLDHI3",
  "EXRACT22", "CHECKUP1", "MEDCOST1", "PRIMINS1", "MARITAL",
  "RENTHOM1", "EMPLOY1", "COVIDPO1", "SMOKE100")

# Dataset copy for imputation
brfss_small_filtered_subset_knn <- brfss_small_filtered_subset

# KNN chunked imputation for continuous variables
knn_chunked_impute <- function(data, variables, chunk_size = 10000, k = 5) {
  rows <- nrow(data)
  indices <- split(seq_len(rows), ceiling(seq_len(rows) / chunk_size))
  imputed_chunks <- lapply(indices, function(idx) {
    chunk <- data[idx, variables, drop = FALSE]
    kNN(chunk, k = k, imp_var = FALSE)})
  do.call(rbind, imputed_chunks)}

# Applying KNN imputation
chunk_size <- 10000
imputed_data_continuous <- knn_chunked_impute(brfss_small_filtered_subset_knn, variables_to_impute_conti

# Assigning imputed continuous values
brfss_small_filtered_subset_knn[variables_to_impute_continuous] <- imputed_data_continuous
```

```
# Mode imputation for categorical variables
impute_mode <- function(x) {
  mode_value <- names(sort(table(x), decreasing = TRUE))[1]
  x[is.na(x)] <- mode_value
  return(x)}

brfss_small_filtered_subset_knn[variables_to_impute_categorical] <- lapply(
  brfss_small_filtered_subset_knn[variables_to_impute_categorical], impute_mode)

# Validate imputation
missing_values_summary <- colSums(is.na(brfss_small_filtered_subset_knn))
print(missing_values_summary)

# Save the final cleaned dataset
write.csv(brfss_small_filtered_subset_knn, "data/brfss_small_filtered_subset_final.csv", row.names = FA

##---#########################

colSums(is.na(brfss_small_filtered_subset_knn))

################################################################################

## BUILD MODELS

# code to load the data ready for model building - to save time
brfss_small_knn_impute_dropped <- read.csv("data/brfss_small_filtered_subset_final.csv")

# subsetting
brfss_small_filtered <- brfss_small_knn_impute_dropped %>%
  dplyr::select(-X)

# target
brfss_small_filtered$CVDCRHD4 <- ifelse(brfss_small_filtered$CVDCRHD4 == 1, 1,
                                        ifelse(brfss_small_filtered$CVDCRHD4 == 2, 0, NA))
brfss_small_filtered$CVDCRHD4 <- as.numeric(brfss_small_filtered$CVDCRHD4)
target_var <- "CVDCRHD4"

# predictors
predictors <- setdiff(colnames(brfss_small_filtered), target_var)

# sapply(brfss_small_filtered[categorical_vars], levels)

# splitting into training and testing
train_indices <- sample(1:nrow(brfss_small_filtered), size = 0.7 * nrow(brfss_small_filtered))
train_data <- brfss_small_filtered[train_indices, ]
test_data <- brfss_small_filtered[-train_indices, ]

train_data[[target_var]] <- factor(train_data[[target_var]], levels = c(0, 1))
test_data[[target_var]] <- factor(test_data[[target_var]], levels = c(0, 1))

# ---------------


### Model 1: Logistic regression using GLM and all features included
```

```
# - Evaluate the performance of a logistic regression model to predict the target variable `CVDCRHD4`
# - Uses 10-fold cross-validation
# - Fits a logistic regression modelwith a binomial family
# - Calculates RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error).


# 10-fold cv
train_control <- trainControl(method = "cv", number = 10)

formula <- as.formula(paste(target_var, "~", paste(predictors, collapse = " + ")))

# logistic regression model
logistic_model_cv <- train(
  formula,
  data = train_data,
  method = "glm",
  family = binomial,
  trControl = train_control)

logistic_model_cv

summary(logistic_model_cv)

final_logistic_model_cv <- logistic_model_cv$finalModel
coefficients_df <- as.data.frame(summary(final_logistic_model_cv)$coefficients)
coefficients_df <- coefficients_df[, c("Estimate", "Pr(>|z|)")]
print(coefficients_df)

## MODEL RESULTS
# Significant predictors: X_AGE5YR, EDUCA, PERSDOC3: May reflect higher access to diagnostic care.
# GENHLTH: Poor self-rated health strongly predicts worse outcomes.
# Negative Associations:Being female (SEXVAR),  History of smoking (SMOKE100), High blood pressure (BPH

# Performance Metrics:
#  Accuracy: 94.5%, indicating a high proportion of correct classifications.
#     Kappa: 0.058, highlighting significant class imbalance or poor class discrimination.
#     AIC: 98,983, balancing model fit and complexity, useful for model comparison.
#
#     Traditional regression metrics like RMSE, R-squared, and MAE are less informative for classificati
#     Use AIC (98,810) for further model comparisons and selection.


# predict for the test data
predicted_prob <- predict(logistic_model_cv, newdata = test_data, type = "prob")[,2]

# threshold of 0.5
predicted_class <- ifelse(predicted_prob >= 0.5, 1, 0)

# confusion matrix
conf_mat <- confusionMatrix(
  data = factor(predicted_class, levels = c(0,1)),
  reference = factor(test_data[[target_var]], levels = c(0,1))
)
```

```
accuracy <- conf_mat$overall["Accuracy"]
sensitivity <- conf_mat$byClass["Sensitivity"]
specificity <- conf_mat$byClass["Specificity"]

two_class_accuracy <- (sensitivity + specificity) / 2

# ROC curve and AUC
roc_obj <- roc(test_data[[target_var]], predicted_prob)
auc_value <- auc(roc_obj)

conf_mat
cat("Accuracy:", accuracy, "\n")
cat("Sensitivity:", sensitivity, "\n")
cat("Specificity:", specificity, "\n")
cat("Balanced Accuracy:", two_class_accuracy, "\n")
cat("AUC:", auc_value)

# The dataset is highly imbalanced, with a small percentage of the population having CAD.
# Balanced Accuracy is used instead of accuracy to account for class imbalance.
#
# Balanced Accuracy: 0.519, slightly better than random chance.
# AUC-ROC: 0.855

# --------------------

# Model 1 dignostics

# Residual Diagnostics
# Deviance Residuals
deviance_residuals <- residuals(final_logistic_model_cv, type = "deviance")
plot(deviance_residuals,
     main = "Deviance Residuals",
     ylab = "Residuals", xlab = "Index", pch = 19, col = "blue")

# Residual Plot
fitted_values <- fitted(final_logistic_model_cv)
plot(fitted_values, deviance_residuals,
     main = "Residuals vs Fitted Values",
     xlab = "Fitted Values", ylab = "Residuals", pch = 19, col = "red")
abline(h = 0, lty = 2)

# Cook's Distance
cooks_dist <- cooks.distance(final_logistic_model_cv)
plot(cooks_dist,
     main = "Cook's Distance",
     ylab = "Cook's Distance", xlab = "Index", pch = 19, col = "purple")
abline(h = 4 / nrow(train_data), col = "red", lty = 2)

# Variance Inflation Factor (VIF) for Multicollinearity
library(car)
vif_values <- vif(final_logistic_model_cv)
print(vif_values)
cat("Predictors with VIF > 5 indicate potential multicollinearity.\n")
```

```
# The deviance residuals plot indicates a bimodal residual structure, with residuals generally within tl
#
# The Cook's Distance plot highlights that no single observation has excessive leverage on the model, as

##############################################################################

### Model 2: GLM using stepwise feature selection

# Use stepwise forward selection to identify the best logistic regression model. The "both" approach (fo
# Forward selection, with a maximum of 378 models, is computationally efficient while achieving the same

# code to load the data after imputation for model building - to save time
brfss_small_knn_impute_dropped <- read.csv("data/brfss_small_filtered_subset_final.csv")

# subsetting
brfss_small_filtered <- brfss_small_knn_impute_dropped %>%
  dplyr::select(-X)

# target
brfss_small_filtered$CVDCRHD4 <- ifelse(brfss_small_filtered$CVDCRHD4 == 1, 1,
                                        ifelse(brfss_small_filtered$CVDCRHD4 == 2, 0, NA))
brfss_small_filtered$CVDCRHD4 <- as.numeric(brfss_small_filtered$CVDCRHD4)
target_var <- "CVDCRHD4"

# predictors
predictors <- setdiff(colnames(brfss_small_filtered), target_var)

# sapply(brfss_small_filtered[categorical_vars], levels)

# splitting into training and testing
train_indices <- sample(1:nrow(brfss_small_filtered), size = 0.7 * nrow(brfss_small_filtered))
train_data <- brfss_small_filtered[train_indices, ]
test_data <- brfss_small_filtered[-train_indices, ]

train_data[[target_var]] <- factor(train_data[[target_var]], levels = c(0, 1))
test_data[[target_var]] <- factor(test_data[[target_var]], levels = c(0, 1))

# full model with all predictors
full_model <- glm(formula, data = train_data, family = binomial)

# null model with intercept only
null_model <- glm(CVDCRHD4 ~ 1, data = train_data, family = binomial)

# stepwise selection
stepwise_model <- stepAIC(null_model,
                          scope = list(lower = null_model, upper = full_model),
                          direction = "forward",
                          trace = FALSE)

summary(stepwise_model)

## MODEL RESULTS
```

```
# Positive Associations: Age, poorer self-rated health, having a personal doctor, education, mental heal
# Negative Associations: Being male, high cholesterol, high blood pressure, smoking history, alcohol co
# - Both models highlight age, general health status, and high cholesterol, ex, smoking history, and hig
# - Secondary predictors (e.g., education, mental health, physical health, strength training) show redu


# predict for the test data
predicted_prob_stepwise <- predict(stepwise_model, newdata = test_data, type = "response")

# threshold of 0.5
predicted_class_stepwise <- ifelse(predicted_prob_stepwise >= 0.5, 1, 0)

# confusion matrix
conf_mat_stepwise <- confusionMatrix(
  data = factor(predicted_class_stepwise, levels = c(0,1)),
  reference = factor(test_data[[target_var]], levels = c(0,1))
)

accuracy_stepwise <- conf_mat_stepwise$overall["Accuracy"]
sensitivity_stepwise <- conf_mat_stepwise$byClass["Sensitivity"]
specificity_stepwise <- conf_mat_stepwise$byClass["Specificity"]

balanced_accuracy_stepwise <- (sensitivity_stepwise + specificity_stepwise) / 2

# ROC curve and AUC
roc_obj_stepwise <- roc(test_data[[target_var]], predicted_prob_stepwise)
auc_value_stepwise <- auc(roc_obj_stepwise)


conf_mat_stepwise
cat("Accuracy:", accuracy_stepwise, "\n")
cat("Sensitivity:", sensitivity_stepwise, "\n")
cat("Specificity:", specificity_stepwise, "\n")
cat("Balanced Accuracy:", balanced_accuracy_stepwise, "\n")
cat("AUC:", auc_value_stepwise)

## MODEL PERFORMANCE
# Specificity: 3.87%, poor at identifying the minority class (presence of CAD).
# Balanced Accuracy: 51.8%, close to random chance due to class imbalance.
# AUC-ROC: 0.855, strong ability to distinguish between classes under varying thresholds.
#
# - The model ineffective because struggles with minority class classification.


# Model 2 diagnostics

# Deviance Residuals
deviance_residuals <- residuals(stepwise_model, type = "deviance")
plot(deviance_residuals,
     main = "Deviance Residuals",
     ylab = "Residuals", xlab = "Index",
     col = "blue", pch = 20)

# Residuals vs Fitted Values
```

```
fitted_values <- fitted(stepwise_model)
plot(fitted_values, deviance_residuals,
     main = "Residuals vs Fitted Values",
     xlab = "Fitted Values", ylab = "Residuals",
     col = "red", pch = 20)
abline(h = 0, col = "gray", lty = 2)

# Cook's Distance
cooks_dist <- cooks.distance(stepwise_model)
plot(cooks_dist,
     main = "Cook's Distance",
     xlab = "Index", ylab = "Cook's Distance",
     col = "purple", pch = 20)
abline(h = 4 / nrow(train_data), col = "red", lty = 2)

# VIF
library(car)
vif_values <- vif(stepwise_model)
print(vif_values)
cat("Predictors with VIF > 5 indicate potential multicollinearity.\n")


# 1. Deviance Residuals:
# The deviance residuals plot shows a significant number of residuals concentrated between -2 and 4. The
#
# 2. Residuals vs. Fitted Values:
# The residuals vs. fitted values plot shows two curved bands, which is a typical characteristic of logi
#
# 3. Cook's Distance:
# The Cook's distance plot identifies observations that have a disproportionate influence on the model.
#
# 4. Variance Inflation Factor (VIF):
# The VIF values for predictors show no evidence of extreme multicollinearity. All VIF values are below


##############################################################################

### Model 3: XGBoost
# Class Imbalance Strategy: Oversample the minority class to create a more balanced training dataset.


# code to load the data after imputation for model building - to save time
brfss_small_knn_impute_dropped <- read.csv("data/brfss_small_filtered_subset_final.csv")

# subsetting
brfss_small_filtered <- brfss_small_knn_impute_dropped %>%
  dplyr::select(-X)

# target
brfss_small_filtered$CVDCRHD4 <- ifelse(brfss_small_filtered$CVDCRHD4 == 1, 1,
                                        ifelse(brfss_small_filtered$CVDCRHD4 == 2, 0, NA))
brfss_small_filtered$CVDCRHD4 <- as.numeric(brfss_small_filtered$CVDCRHD4)
target_var <- "CVDCRHD4"
```

```
# predictors
predictors <- setdiff(colnames(brfss_small_filtered), target_var)

# sapply(brfss_small_filtered[categorical_vars], levels)

# splitting into training and testing
train_indices <- sample(1:nrow(brfss_small_filtered), size = 0.7 * nrow(brfss_small_filtered))
train_data <- brfss_small_filtered[train_indices, ]
test_data <- brfss_small_filtered[-train_indices, ]

train_data[[target_var]] <- factor(train_data[[target_var]], levels = c(0, 1))
test_data[[target_var]] <- factor(test_data[[target_var]], levels = c(0, 1))

XGB_train_data <- train_data
XGB_test_data <- test_data

XGB_train_data[[target_var]] <- factor(XGB_train_data[[target_var]], levels = c(0, 1), labels = c("class
XGB_test_data[[target_var]] <- factor(XGB_test_data[[target_var]], levels = c(0, 1), labels = c("class0

oversampled_train_data <- upSample(x = XGB_train_data[, predictors],
                                   y = XGB_train_data[[target_var]],
                                   yname = target_var)

# Set up cross-validation with class probability summaries
train_control <- trainControl(method = "cv",
                              number = 10,
                              classProbs = TRUE,
                              summaryFunction = twoClassSummary)

# Define a basic tuning grid for XGBoost (customize as needed)
xgb_grid <- expand.grid(
  nrounds = 100,
  max_depth = 3,
  eta = 0.3,
  gamma = 0,
  colsample_bytree = 1,
  min_child_weight = 1,
  subsample = 1
)

# Train the XGBoost model on the oversampled data
xgb_model <- train(
  as.formula(paste(target_var, "~", paste(predictors, collapse = " + "))),
  data = oversampled_train_data,
  method = "xgbTree",
  trControl = train_control,
  metric = "ROC",
  tuneGrid = xgb_grid
)

# View model results
print(xgb_model)

# AUC-ROC: 0.866, strong discriminatory power.
```

```
# Sensitivity: 73.9%, good at identifying true positives.
# Specificity: 84.3%, slightly favors true negatives.
# Balanced performance in identifying both classes.


predicted_prob <- predict(xgb_model, newdata = XGB_test_data, type = "prob")[, "class1"]
predicted_class <- ifelse(predicted_prob >= 0.5, "class1", "class0")


conf_mat <- confusionMatrix(factor(predicted_class, levels = c("class0", "class1")),
                            XGB_test_data[[target_var]])
print(conf_mat)

roc_obj <- roc(response = XGB_test_data[[target_var]], predictor = predicted_prob)
auc_value <- auc(roc_obj)
cat("AUC:", auc_value, "\n")

feature_importance <- varImp(xgb_model, scale = TRUE)
cat("Feature Importance:\n")
print(feature_importance)

## MODEL EVALUATION

# AUC: 0.866, strong discriminatory ability similar to logistic regression models.
# Sensitivity (73.9%) and Specificity (84.3%) achieves better balance, with higher balanced accuracy tha
#
# Top Predictors: Age, general self-reported health, and high blood pressure, high cholesterol, employme
# Modest Contributors: Diabetes, physical health, and having a personal doctor remain relevant but less


# Model 3 diagnostics

# Plot feature importance
xgb_importance <- varImp(xgb_model, scale = TRUE)
plot(xgb_importance, top = 15, main = "Top 15 Feature Importance")

# Create a data frame for predictions and actuals
diagnostics_df <- data.frame(
  Observed = ifelse(XGB_test_data[[target_var]] == "class1", 1, 0),
  Predicted_Prob = predicted_prob
)

# Plot observed vs predicted probabilities
ggplot(diagnostics_df, aes(x = Predicted_Prob, y = Observed)) +
  geom_point(alpha = 0.2) +
  geom_smooth(method = "loess", color = "blue") +
  labs(title = "Predicted vs Observed Probabilities",
       x = "Predicted Probability",
       y = "Observed Outcome") +
  theme_minimal()


# Compute residuals
diagnostics_df$residuals <- diagnostics_df$Observed - diagnostics_df$Predicted_Prob
```

```
# Plot residuals
ggplot(diagnostics_df, aes(x = Predicted_Prob, y = residuals)) +
  geom_point(alpha = 0.2, color = "red") +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title = "Residuals vs Predicted Probabilities",
       x = "Predicted Probability",
       y = "Residuals") +
  theme_minimal()

# Extract learning curve data
xgb_eval_log <- xgb_model$results

# Plot learning curve (AUC vs Iterations)
ggplot(xgb_eval_log, aes(x = nrounds, y = ROC)) +
  geom_line(color = "darkgreen") +
  geom_point(size = 1.5) +
  labs(title = "Learning Curve: AUC vs Iterations",
       x = "Number of Boosting Rounds",
       y = "AUC (ROC)") +
  theme_minimal()


# Bin predicted probabilities
diagnostics_df$bins <- cut(diagnostics_df$Predicted_Prob, breaks = seq(0, 1, by = 0.1), include.lowest =

# Summarize observed probabilities within each bin
calibration <- diagnostics_df %>%
  group_by(bins) %>%
  summarize(Mean_Predicted = mean(Predicted_Prob),
            Mean_Observed = mean(Observed))

# Plot calibration curve
ggplot(calibration, aes(x = Mean_Predicted, y = Mean_Observed)) +
  geom_line(color = "blue", size = 1) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Calibration Plot",
       x = "Mean Predicted Probability",
       y = "Mean Observed Probability") +
  theme_minimal()


# Feature Importance Plot:
# The top 15 features contributing to the XGBoost model's predictive performance are displayed. X_AGEG5Y
#
# Predicted vs. Observed Probabilities:
# This plot evaluates how well the predicted probabilities align with the observed outcomes. The predict
#
# Residuals vs. Predicted Probabilities:
# The residuals are plotted against the predicted probabilities, showing a clear "V-shaped" pattern. Thi
#
# Learning Curve (AUC vs. Iterations):
# The learning curve shows the Area Under the Curve (AUC) across boosting rounds. AUC stabilizes around
```

```
##############################################################################

### Model 4: Ridge With Over-Sampling

# - Oversampling again to help with class imbalance
# - Applies a penalty proportional to the square of coefficient magnitudes to reduce overfitting.
# - Retains all predictors, providing stability and generalizability.

# code to load the data after imputation for model building - to save time
brfss_small_knn_impute_dropped <- read.csv("data/brfss_small_filtered_subset_final.csv")

# subsetting
brfss_small_filtered <- brfss_small_knn_impute_dropped %>%
  dplyr::select(-X)

# target
brfss_small_filtered$CVDCRHD4 <- ifelse(brfss_small_filtered$CVDCRHD4 == 1, 1,
                                        ifelse(brfss_small_filtered$CVDCRHD4 == 2, 0, NA))
brfss_small_filtered$CVDCRHD4 <- as.numeric(brfss_small_filtered$CVDCRHD4)
target_var <- "CVDCRHD4"

# predictors
predictors <- setdiff(colnames(brfss_small_filtered), target_var)

# sapply(brfss_small_filtered[categorical_vars], levels)

# splitting into training and testing
train_indices <- sample(1:nrow(brfss_small_filtered), size = 0.7 * nrow(brfss_small_filtered))
train_data <- brfss_small_filtered[train_indices, ]
test_data <- brfss_small_filtered[-train_indices, ]

train_data[[target_var]] <- factor(train_data[[target_var]], levels = c(0, 1))
test_data[[target_var]] <- factor(test_data[[target_var]], levels = c(0, 1))

# Oversample the positive class
Ridge_train_data <- oversampled_train_data

Ridge_test_data <-  test_data

Ridge_test_data[[target_var]] <- factor(test_data[[target_var]], levels = c(0, 1), labels = c("class0",

train_control <- trainControl(method = "cv",
                              number = 10,
                              classProbs = TRUE,
                              summaryFunction = twoClassSummary)


# alpha = 0 for ridge regression
# lambda will be selected by the model
ridge_grid <- expand.grid(alpha = 0,
                          lambda = seq(0.001, 1, length = 10))

# ridge logistic regression model
```

```
ridge_model <- train(
  as.formula(paste(target_var, "~", paste(predictors, collapse = " + "))),
  data = Ridge_train_data,
  method = "glmnet",
  trControl = train_control,
  metric = "ROC",
  tuneGrid = ridge_grid,
  family = "binomial"
)
print(ridge_model)

# predict for the test data
predicted_prob <- predict(ridge_model, newdata = Ridge_test_data, type = "prob")[, "class1"]
predicted_class <- ifelse(predicted_prob >= 0.5, "class1", "class0")

# confusion matrix and AUC
conf_mat <- confusionMatrix(factor(predicted_class, levels = c("class0", "class1")),
                            Ridge_test_data[[target_var]])
print(conf_mat)

roc_obj <- roc(response = Ridge_test_data[[target_var]], predictor = predicted_prob)
auc_value <- auc(roc_obj)
cat("AUC:", auc_value, "\n")


## MODEL EVALUATION
# AUC: 0.854, comparable to XGBoost.
# Balanced Accuracy: 0.780, slightly lower than XGBoost
# Sensitivity (74.3%) and Specificity (81.8%): Marginally less balanced than XGBoost.
# Detection Rate: 70.1%, higher proportion of majority class identified than XGBoost.


# Model 4 diagnostics

# Manually calculate residuals
ridge_pred_response <- predict(ridge_model, Ridge_test_data, type = "prob")[, "class1"]
ridge_actual_response <- as.numeric(Ridge_test_data[[target_var]]) - 1  # Convert factor to numeric 0/1

ridge_residuals <- ridge_actual_response - ridge_pred_response  # Residuals

# Plot Residuals
plot(ridge_residuals,
     col = "blue",
     main = "Deviance Residuals for Ridge Regression",
     ylab = "Residuals",
     xlab = "Index")
abline(h = 0, col = "red", lty = 2)


# Coefficient Paths
plot(ridge_model$finalModel,
     xvar = "lambda",
     label = TRUE,
     main = "Ridge Regression Coefficient Paths")
```

```
# Predicted probabilities vs observed outcomes
plot(predicted_prob, as.numeric(Ridge_test_data[[target_var]]) - 1,
     pch = 20, col = "black",
     main = "Predicted vs Observed Probabilities",
     xlab = "Predicted Probabilities",
     ylab = "Observed Outcome")

# ROC Curve
plot.roc(roc_obj,
         main = "ROC Curve for Ridge Regression",
         col = "blue",
         lwd = 2)
abline(a = 0, b = 1, col = "red", lty = 2)

# Learning Curve: AUC vs Lambda
plot(ridge_model$results$lambda, ridge_model$results$ROC,
     type = "b", col = "darkgreen",
     xlab = "Lambda (Regularization Parameter)",
     ylab = "AUC",
     main = "AUC vs Lambda")


# Variable importance for Ridge Regression
ridge_importance <- varImp(ridge_model, scale = FALSE)
plot(ridge_importance, top = 15, main = "Top 15 Variable Importance for Ridge Regression")


# VIF Calculation
library(car)
vif_data <- lm(as.formula(paste(target_var, "~", paste(predictors, collapse = " + "))),
               data = Ridge_train_data)
vif_values <- vif(vif_data)
print(vif_values)
cat("Predictors with VIF > 5 indicate potential multicollinearity.\n")


# The Deviance Residuals plot shows that residuals are tightly clustered between -1 and 1 with no evide
#
# The coefficient paths plot highlights how coefficients shrink as lambda increases. Unlike LASSO, Ridg
#
# The Predicted vs Observed Probabilities plot shows good separation between the two outcome classes, w
#
# The ROC curve demonstrates strong discriminatory ability, with an AUC of ~0.854, confirming that the
#
# The AUC vs Lambda plot reveals that lower regularization (small lambda values) achieves the best perf
#
# The variable importance plot identifies CVDSTRK3 (previous stroke), TOLDHI3 (high cholesterol), SEXVA
#
# In summary, the Ridge model demonstrates strong predictive power (AUC = 0.854), mitigates multicollin


##############################################################################
```

```
### Model 5: LASSO With Over-Sampling

# Address class imbalance through oversampling.
# Adds a penalty proportional to the absolute value of coefficients.
# Prevents overfitting and performs variable selection by shrinking less important coefficients to zero

# code to load the data after imputation for model building - to save time
brfss_small_knn_impute_dropped <- read.csv("data/brfss_small_filtered_subset_final.csv")

# subsetting
brfss_small_filtered <- brfss_small_knn_impute_dropped %>%
  dplyr::select(-X)

# target
brfss_small_filtered$CVDCRHD4 <- ifelse(brfss_small_filtered$CVDCRHD4 == 1, 1,
                                        ifelse(brfss_small_filtered$CVDCRHD4 == 2, 0, NA))
brfss_small_filtered$CVDCRHD4 <- as.numeric(brfss_small_filtered$CVDCRHD4)
target_var <- "CVDCRHD4"

# predictors
predictors <- setdiff(colnames(brfss_small_filtered), target_var)

# sapply(brfss_small_filtered[categorical_vars], levels)

# splitting into training and testing
train_indices <- sample(1:nrow(brfss_small_filtered), size = 0.7 * nrow(brfss_small_filtered))
train_data <- brfss_small_filtered[train_indices, ]
test_data <- brfss_small_filtered[-train_indices, ]

train_data[[target_var]] <- factor(train_data[[target_var]], levels = c(0, 1))
test_data[[target_var]] <- factor(test_data[[target_var]], levels = c(0, 1))

# Oversample the minority class
Lasso_train_data <- oversampled_train_data

Lasso_test_data <- test_data

Lasso_test_data[[target_var]] <- factor(test_data[[target_var]], levels = c(0, 1), labels = c("class0",

train_control <- trainControl(method = "cv",
                              number = 10,
                              classProbs = TRUE,
                              summaryFunction = twoClassSummary)


# alpha = 1 for LASSO penalty
# lambda is the penalty strength
lasso_grid <- expand.grid(alpha = 1,
                          lambda = seq(0.001, 1, length = 10))

# Train the LASSO logistic regression model
lasso_model <- train(
  as.formula(paste(target_var, "~", paste(predictors, collapse = " + "))),
```

```
  data = Lasso_train_data,
  method = "glmnet",
  trControl = train_control,
  metric = "ROC",
  tuneGrid = lasso_grid,
  family = "binomial"
)
print(lasso_model)

lasso_coefficients <- coef(lasso_model$finalModel, s = lasso_model$bestTune$lambda)

lasso_coefficients_df <- as.data.frame(as.matrix(lasso_coefficients))
colnames(lasso_coefficients_df) <- c("Coefficient")
lasso_coefficients_df$Predictor <- rownames(lasso_coefficients_df)
print(lasso_coefficients_df)

# predictions for the test set
predicted_prob <- predict(lasso_model, newdata = Lasso_test_data, type = "prob")[, "class1"]
predicted_class <- ifelse(predicted_prob >= 0.5, "class1", "class0")

# confusion matrix and AUC
conf_mat <- confusionMatrix(factor(predicted_class, levels = c("class0", "class1")),
                            Lasso_test_data[[target_var]])
print(conf_mat)

roc_obj <- roc(response = Lasso_test_data[[target_var]], predictor = predicted_prob)
auc_value <- auc(roc_obj)
cat("AUC:", auc_value, "\n")


# Model 5 diagnostics

### LASSO Model Diagnostics

# Deviance Residuals
lasso_predictions <- predict(lasso_model$finalModel,
                             newx = as.matrix(Lasso_test_data[, predictors]),
                             s = lasso_model$bestTune$lambda,
                             type = "response")
deviance_residuals <- residuals(lasso_model$finalModel,
                                s = lasso_model$bestTune$lambda,
                                type = "deviance")
plot(deviance_residuals,
     col = "blue",
     main = "Deviance Residuals for LASSO Regression",
     ylab = "Residuals",
     xlab = "Index")

# Coefficient Paths
plot(lasso_model$finalModel, xvar = "lambda", label = TRUE, main = "LASSO Coefficient Paths")

# Predicted vs Observed Probabilities
plot(predicted_prob, as.numeric(Lasso_test_data[[target_var]]) - 1,
     xlab = "Predicted Probabilities",
```

```
     ylab = "Observed Outcome",
     main = "Predicted vs Observed Probabilities",
     pch = 16, col = "black")

# Residuals vs Fitted Probabilities
residuals <- as.numeric(Lasso_test_data[[target_var]]) - predicted_prob
plot(predicted_prob, residuals,
     col = "red",
     main = "Residuals vs Predicted Probabilities",
     xlab = "Predicted Probabilities",
     ylab = "Residuals")
abline(h = 0, lty = 2)

# ROC Curve
plot.roc(roc_obj,
         col = "blue",
         main = "ROC Curve for LASSO Regression",
         print.auc = TRUE)

# AUC vs Lambda
plot(lasso_model$results$lambda, lasso_model$results$ROC,
     type = "o", col = "green",
     main = "AUC vs Lambda",
     xlab = "Lambda (Regularization Parameter)",
     ylab = "AUC")


# 1. Deviance Residuals Plot
# The deviance residuals plot shows residuals distributed fairly symmetrically around zero, which is a g
#
# 2. Coefficient Paths
# The coefficient paths plot shows how LASSO shrinks predictors' coefficients as the regularization para
#
# 3. Predicted vs Observed Probabilities
# The predicted vs observed probabilities plot demonstrates clear separation between the predicted proba
#
# 4. Residuals vs Predicted Probabilities
# The residuals vs predicted probabilities plot shows that the residuals are largely symmetric but exhil
#
# 5. ROC Curve
# The ROC curve shows strong discriminatory ability, with an AUC of 0.856. This performance is consister
#
# 6. AUC vs Lambda
# The AUC vs Lambda plot indicates that the model achieves its best performance (highest AUC) at very lo

##############################################################################


## MODEL EVALUATION
AUC: 0.85-0.86, similar to previous.
Balanced Accuracy: 0.780, similar to previous
Sensitivity (74.4%) and Specificity (81.7%): Falls between Ridge and XGBoost metrics.
Detection Rate: 70.3%, effective at identifying the majority class.
```

```
## Model Comparison and Selection

# performance and evaluation metrics, as well as AICs, to compare the built models.

# XGBoost Predictions
xgb_pred_probs <- predict(xgb_model, test_data, type = "prob")[, 2]
xgb_pred_labels <- ifelse(xgb_pred_probs > 0.5, 1, 0)
xgb_pred_labels <- factor(xgb_pred_labels, levels = c(0, 1))

# Ridge Predictions
ridge_pred_probs <- predict(ridge_model, test_data, type = "prob")[, 2]
ridge_pred_labels <- ifelse(ridge_pred_probs > 0.5, 1, 0)
ridge_pred_labels <- factor(ridge_pred_labels, levels = c(0, 1))

# LASSO Predictions
lasso_pred_probs <- predict(lasso_model, test_data, type = "prob")[, 2]
lasso_pred_labels <- ifelse(lasso_pred_probs > 0.5, 1, 0)
lasso_pred_labels <- factor(lasso_pred_labels, levels = c(0, 1))

actual <- factor(test_data[[target_var]], levels = c(0, 1))

compute_metrics <- function(model, data, target_var) {
  if ("train" %in% class(model)) {
    pred_probs <- predict(model, data, type = "prob")[, 2]
    pred_labels <- ifelse(pred_probs > 0.5, 1, 0)
  } else if ("xgb.Booster" %in% class(model)) {
    mat <- as.matrix(data[, setdiff(names(data), target_var)])
    pred_probs <- predict(model, mat)
    pred_labels <- ifelse(pred_probs > 0.5, 1, 0)
  } else {
    pred_probs <- predict(model, data, type = "response")
    pred_labels <- ifelse(pred_probs > 0.5, 1, 0)}

  actual <- factor(data[[target_var]], levels = c(0, 1))
  pred_labels <- factor(pred_labels, levels = c(0, 1))

  if (any(is.na(pred_labels)) || any(is.na(actual))) {
    warning("Predicted or actual labels contain NA. Metrics cannot be computed.")
    return(list(Sensitivity = NA, Specificity = NA, PPV = NA, NPV = NA, Accuracy = NA,
                F1 = NA, AUC = NA, FP_Rate = NA, FN_Rate = NA))}

  cm <- confusionMatrix(data = pred_labels, reference = actual)
  sensitivity <- as.numeric(cm$byClass["Sensitivity"])
  specificity <- as.numeric(cm$byClass["Specificity"])
  ppv <- as.numeric(cm$byClass["Pos Pred Value"])
  npv <- as.numeric(cm$byClass["Neg Pred Value"])
  accuracy <- as.numeric(cm$overall["Accuracy"])
  f1 <- ifelse((ppv + sensitivity) > 0, 2 * (ppv * sensitivity) / (ppv + sensitivity), NA)

  roc_curve <- roc(as.numeric(as.character(actual)), pred_probs)
  auc_val <- as.numeric(auc(roc_curve))

  fp_rate <- 1 - specificity
```

```
  fn_rate <- 1 - sensitivity

  return(list(
    Sensitivity = sensitivity,
    Specificity = specificity,
    PPV = ppv,
    NPV = npv,
    Accuracy = accuracy,
    F1 = f1,
    AUC = auc_val,
    FP_Rate = fp_rate,
    FN_Rate = fn_rate))}

compute_aic <- function(model, data, target_var) {
  if ("train" %in% class(model)) {
    pred_probs <- predict(model, data, type = "prob")[, 2]
    actual <- as.numeric(data[[target_var]])
    pred_probs[pred_probs == 0] <- 1e-15
    pred_probs[pred_probs == 1] <- 1 - 1e-15
    log_likelihood <- sum(actual * log(pred_probs) + (1 - actual) * log(1 - pred_probs))
    if (is.null(model$finalModel)) return(NA)
    coefs <- coef(model$finalModel)
    if (inherits(coefs, "dgCMatrix")) {
      num_params <- length(which(coefs != 0))
    } else {
      num_params <- length(coefs)}
    return(2 * num_params - 2 * log_likelihood)
  } else if ("glm" %in% class(model)) {
    return(as.numeric(AIC(model)))
  } else {
    return(NA)}}

all_metrics <- list()
all_metrics[["Logistic (All Features)"]] <- compute_metrics(logistic_model_cv, test_data, target_var)
all_metrics[["Logistic (Stepwise)"]] <- compute_metrics(stepwise_model, test_data, target_var)
all_metrics[["XGBoost"]] <- compute_metrics(xgb_model, test_data, target_var)
all_metrics[["Ridge"]] <- compute_metrics(ridge_model, test_data, target_var)
all_metrics[["LASSO"]] <- compute_metrics(lasso_model, test_data, target_var)

metrics_table <- do.call(rbind, all_metrics) %>%
  as.data.frame(stringsAsFactors = FALSE)

metrics_table$Model <- rownames(metrics_table)
rownames(metrics_table) <- NULL

aic_values <- c(
  compute_aic(logistic_model_cv, test_data, target_var),
  compute_aic(stepwise_model, test_data, target_var),
  compute_aic(xgb_model, test_data, target_var),
  compute_aic(ridge_model, test_data, target_var),
  compute_aic(lasso_model, test_data, target_var))

metrics_table$AIC <- aic_values
```

```
numeric_cols <- setdiff(names(metrics_table), "Model")
metrics_table[numeric_cols] <- lapply(metrics_table[numeric_cols], function(x) as.numeric(unlist(x)))

metrics_table <- metrics_table[, c("Model", setdiff(names(metrics_table), "Model"))]

writexl::write_xlsx(metrics_table, "metrics_table_with_AIC.xlsx")

saveRDS(metrics_table,
        "/Users/seshat/Documents/GitHub/DATA621/final-project/tables/metrics_table.rds")


## Building the ROC curves plot as well:

# Reshape data for plotting
metrics_long <- metrics_table %>%
  tidyr::pivot_longer(
    cols = -Model,
    names_to = "Metric",
    values_to = "Value")

# Plot metrics
metrics_plot <- ggplot(metrics_long, aes(x = Metric, y = Value, fill = Model)) +
  geom_bar(stat = "identity", position = position_dodge()) +
  labs(
    title = "Model Performance Metrics",
    x = "Metric",
    y = "Value"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "bottom"
  )

ggsave("plots/metrics_plot.png", plot = metrics_plot, width = 8, height = 6, dpi = 300)

## ROC Curves

roc_curves <- list(
  Logistic_All = roc(train_data[[target_var]], predict(logistic_model_cv, train_data, type = "prob")[, 1
  Logistic_Stepwise = roc(train_data[[target_var]], predict(stepwise_model, train_data, type = "respons
  XGBoost = roc(train_data[[target_var]], predict(xgb_model, train_data, type = "prob")[, 2]),
  Ridge = roc(train_data[[target_var]], predict(ridge_model, train_data, type = "prob")[, 2]),
  LASSO = roc(train_data[[target_var]], predict(lasso_model, train_data, type = "prob")[, 2])
)

# plotting
roc_plot <- ggroc(roc_curves)

final_roc_plot <- roc_plot +
  aes(color = name) +
  geom_line(size = 1.2) +
  labs(
    title = "ROC Curves for All Models",
```

```
    x = "False Positive Rate",
    y = "True Positive Rate"
  ) +
  theme_minimal() +
  scale_color_discrete(name = "Model")

# Display the plot
print(final_roc_plot)

# Save the plot as a PNG
ggsave("plots/roc_curves_plot.png", plot = final_roc_plot, width = 8, height = 6, dpi = 300)
```