```
# Load the necessary libraries
library(tidyverse)
library(mice)
library(car)
library(ggplot2)
library(lattice)
library(caret)
library(glmnet)
library(Matrix)
library(pROC)

# Read in the data
df <- read.csv("/Users/preethireddy/Downloads/untitled folder 3/NSSO68.csv")

data = df
# Create the Target variable
data$non_veg <- ifelse(rowSums(data[, c('eggsno_q', 'fishprawn_q', 'goatmeat_q', 'beef_q',
'pork_q','chicken_q', 'othrbirds_q')]) > 0, 1, 0)


# Get the value counts of non_veg
non_veg_values <- data$non_veg
value_counts <- table(non_veg_values)
print(value_counts)

# Define the dependent variable (non_veg) and independent variables
y <- data$non_veg
X <- data[,(names(data) %in% c("HH_type", "Religion",
"Social_Group","Regular_salary_earner","Possess_ration_card","Sex","Age","Marital_Status","
Education","Meals_At_Home","Region","hhdsz" ,"NIC_2008","NCO_2004"))]

str(X)
# Ensure 'y' is a binary factor
y <- as.factor(y)
X$Region = as.factor(X$Region)
X$Social_Group = as.factor(X$Social_Group)
X$Regular_salary_earner = as.factor(X$Regular_salary_earner)
X$HH_type = as.factor(X$HH_type)
X$Possess_ration_card = as.factor(X$Possess_ration_card)
X$Sex = as.factor(X$Sex)
X$Marital_Status = as.factor(X$Marital_Status)
X$Education = as.factor(X$Education)
X$Region = as.factor(X$Region)


# Create the combined data frame
combined_data <- data.frame(y, X)

# Inspect the combined data
```

```
str(combined_data)
head(combined_data)
combined_data$Age
# Fit the model using glmnet with sparse matrix
probit_model <- glm(y ~ hhdsz + NIC_2008 + NCO_2004 + HH_type + Religion +
Social_Group+Regular_salary_earner+Region+Meals_At_Home+Education+Age+Sex+Possess
_ration_card,data = combined_data,
                    family = binomial(link = "probit"),
                    control = list(maxit = 1000))
data$hhdsz_scaled <- scale(data$hhdsz)
data$NIC_2008_scaled <- scale(data$NIC_2008)


# Print model summary or other relevant outputs
print(probit_model)


# Predict probabilities
predicted_probs <- predict(probit_model, newdata = combined_data, type = "response")

# Convert probabilities to binary predictions using a threshold of 0.5
predicted_classes <- ifelse(predicted_probs > 0.5, 1, 0)

# Actual classes
actual_classes <- combined_data$y

# Confusion Matrix
confusion_matrix <- caret::confusionMatrix(as.factor(predicted_classes),
as.factor(actual_classes))

print(confusion_matrix)

# ROC curve and AUC value
roc_curve <- pROC::roc(actual_classes, predicted_probs)
auc_value <- pROC::auc(roc_curve)
plot(roc_curve, col = "blue", main = "ROC Curve")
print(paste("AUC:", auc_value))

# Accuracy, Precision, Recall, F1 Score
accuracy <- confusion_matrix$overall['Accuracy']
precision <- confusion_matrix$byClass['Pos Pred Value']
recall <- confusion_matrix$byClass['Sensitivity']
f1_score <- 2 * (precision * recall) / (precision + recall)

print(paste("Accuracy:", accuracy))
print(paste("Precision:", precision))
print(paste("Recall:", recall))
print(paste("F1 Score:", f1_score))
```