# Introduction to Data Science [A]

Final Term Project Report- Fall 23-24

Report Prepared by,

Hadiur Rahman Nabil [20-42095-1]

Khan Nushrat Sultana Netu [20-43191-1]

# Dataset

The cardiovascular disease risk prediction dataset derived from the CDC's (Centers for Disease Control and Prevention) Behavioral Risk Factor Surveillance System (BRFSS) in 2021 encompasses various attributes related to health indicators and behaviors. This dataset was collected from Kaggle. This cardiovascular disease risk prediction dataset contains 308854 samples. Additionally, it contains 19 attributes including the target attribute. Also here, 7 are numerical and 12 are categorical attributes.

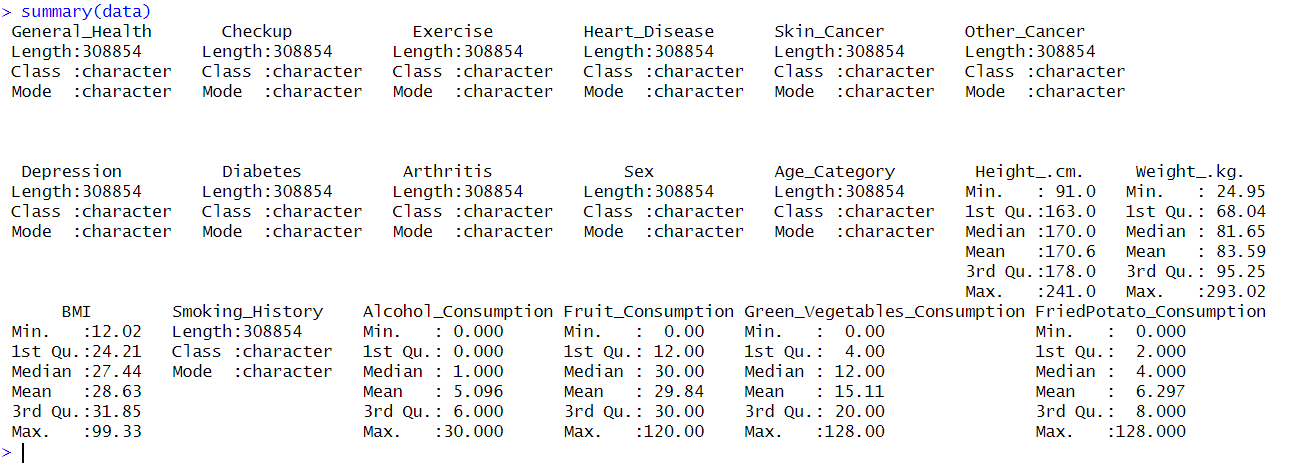
Dataset Link: <https://www.kaggle.com/datasets/alphiree/cardiovascular-diseases-risk-prediction-dataset>

# Load The Dataset

RScript:



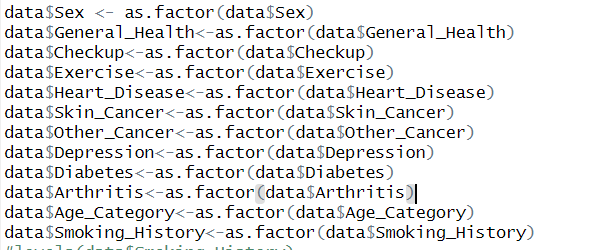
Consol:



After doing a summary of the dataset we got the details of each attribute.

# Convert from Categorical to factor

R Script:



Here we convert all the categorical variables into factors which is crucial to applying Naïve Bayes. This factor converts the data into a level. For example, In the General\_Health attribute, 5 different variations apply within 308854 values. Factors support making these values into different levels and store numeric values among them.

*summary(data$General\_Health)*

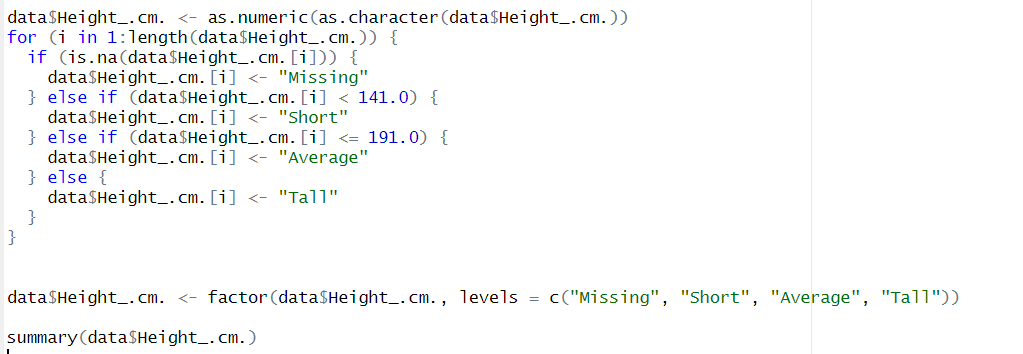
"Excellent" "Fair" "Good" "Poor" "Very Good"

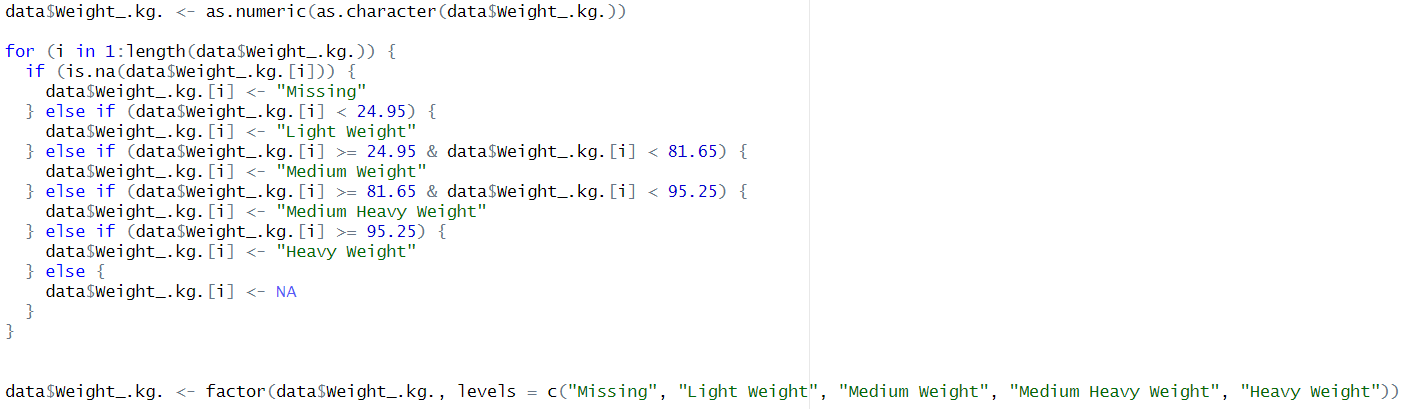
Excellent Fair Good Poor Very Good

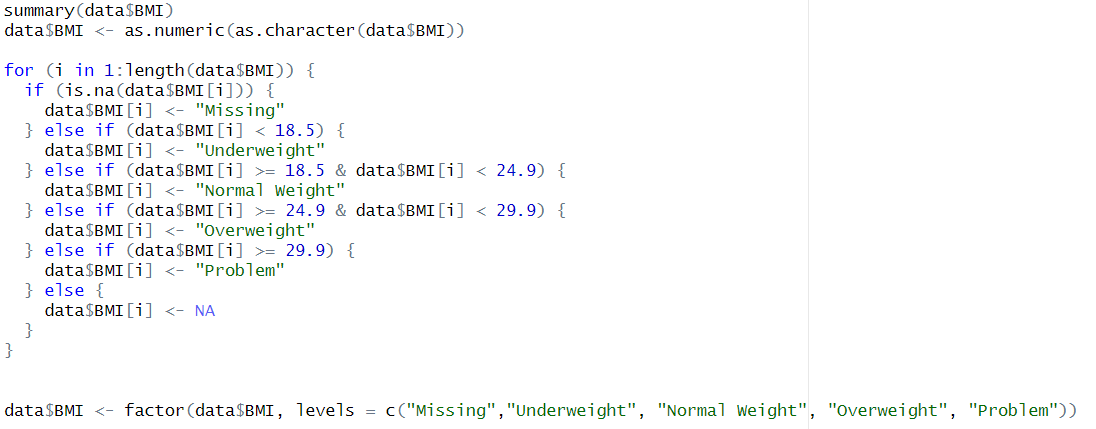
55954 35810 95364 11331 110395

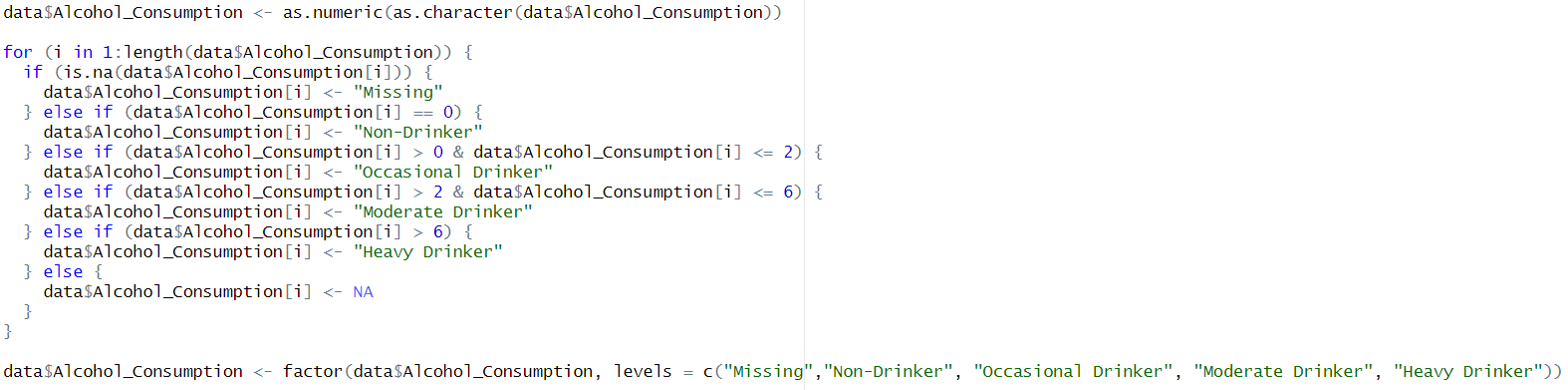
# Convert Numeric to Factor

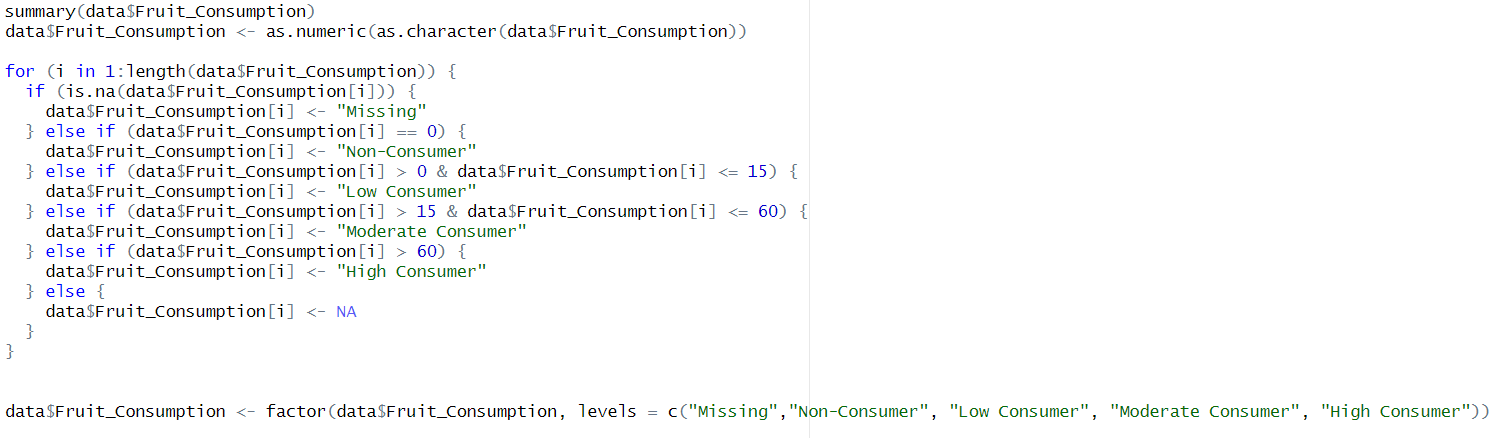
RScript:

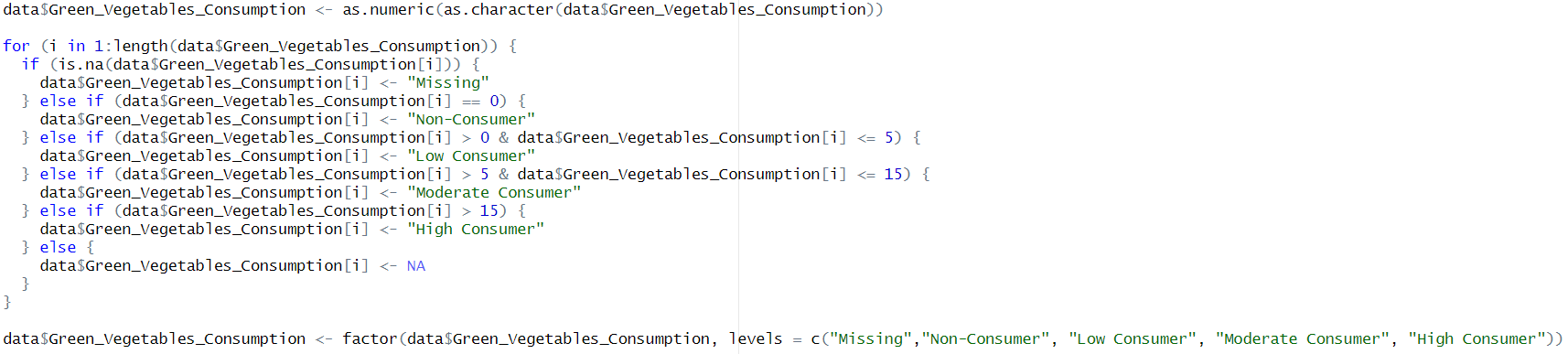


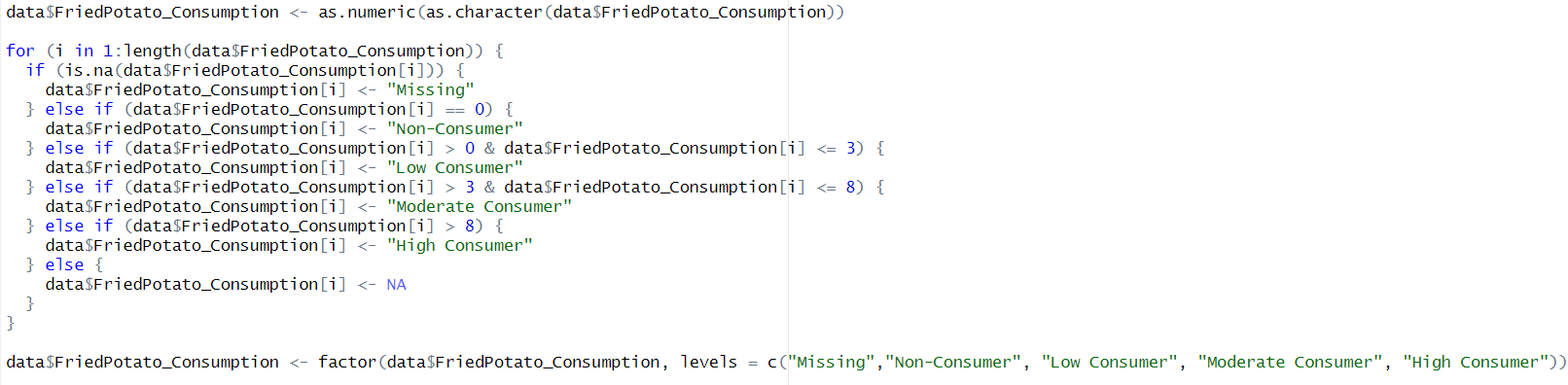




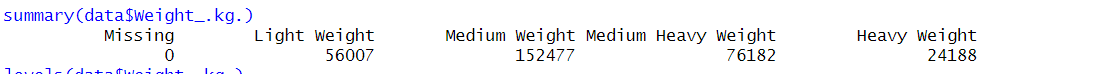


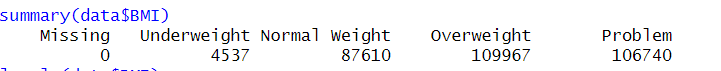


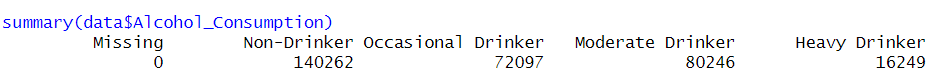




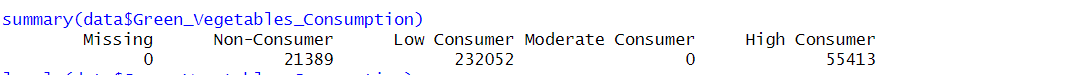
Consol:









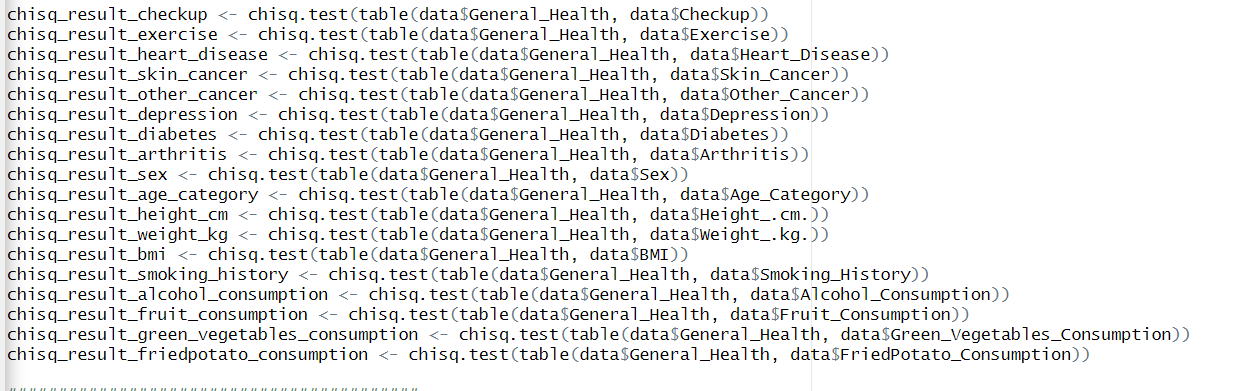


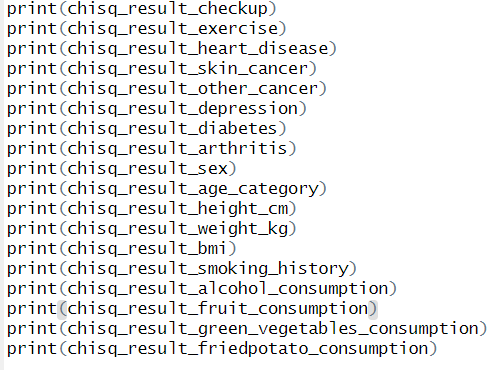


Here, we convert the numeric values to characters first using the *as.numeric(as.Character())* function. Then we divide each level of the values considering their quantile using a For loop. If there are any missing values it shows “Missing”, and based on their conditions it divides 308854 values into different levels. For example, if we consider *Height\_.cm.* attribute it has 4 different levels which are Missing, Short, Average, and Tall. If the height is less than 141.00 cm then we consider it as short, If the height is between 141.00 to 191.00 we consider it to be average, and lastly, if the height is greater than 191.00 then we consider it as Tall. Additionally, we iterate the For loop to the end of the length which is 308854. We repeat this iteration for every attribute that contains Numeric values. At the end of this, we convert Categorical values to the factor based on their level.

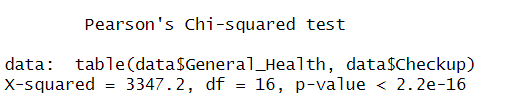
# Pearson’s Chi-Squared Correlation Technique

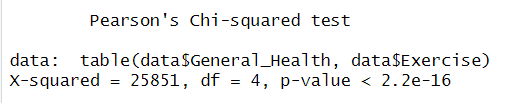
RScript:

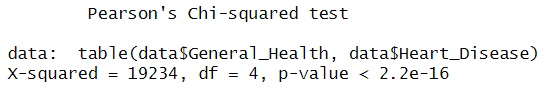


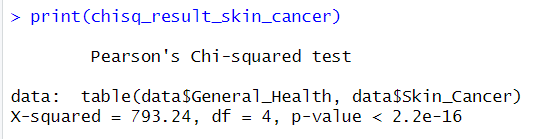


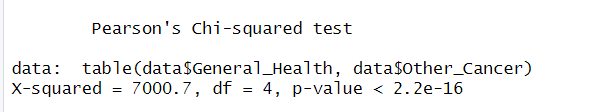
Consol:

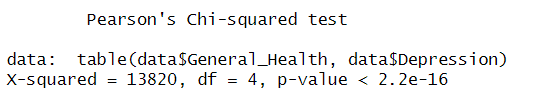


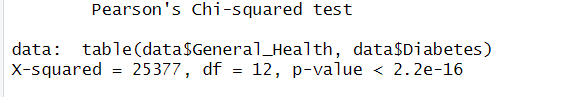


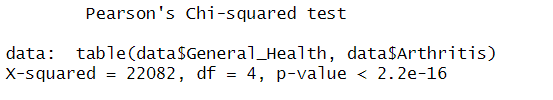


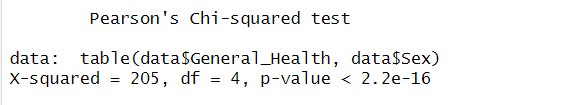


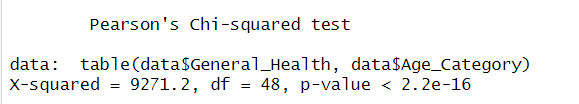


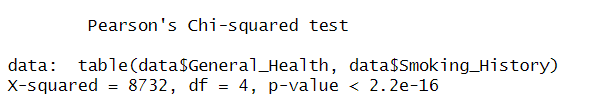










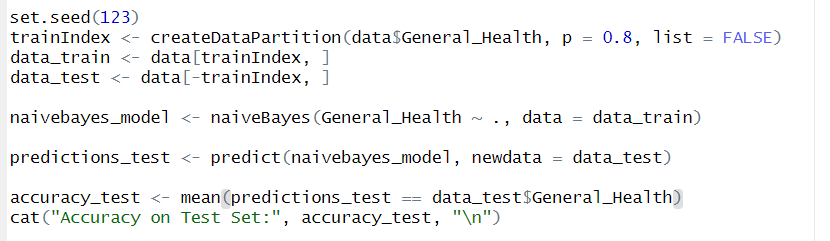


For every attribute, we apply Pearson’s Chi-squared test to identify the association between two categorical attributes from the dataset and print them. Here we did it with the predictor variable and the target variable to understand the relationship between them.

# Remove that value that is less significant



# Train and Test Value Split



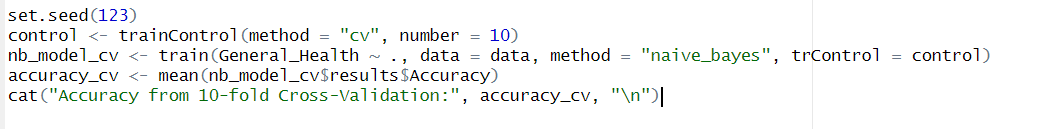


The dataset is divided into two subsets where a training set with 80% of the data and a testing set with the remaining 20%. A Naive Bayes model was trained using `General\_Health` as the target variable and all other available variables as predictors. The trained model was then used to predict the values of `General\_Health` in the test dataset. The accuracy of the model on the test set was calculated by comparing these predictions with the actual values of `General\_Health`. The resulting accuracy score (0.408085) indicates the proportion of correct predictions made by the model on the data, serving as a metric for evaluating the performance of the model.

Here, *set.seed(123)* is used to run the same code multiple times which is randomly split each time.

*createDataPartition(data$General\_Health, p= 0.8, list = FALSE)* uses the createDataPartition function to create an index for splitting the data. It divides the dataset into training and testing subsets, where 80% of the data is allocated for training.

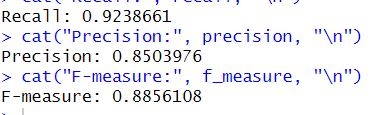
# 10-Fold Cross Validation



This code section performs a 10-fold cross-validation on a Naive Bayes model for General\_Health prediction. It evaluates the model's accuracy by repeatedly splitting the data into 10 subsets. And after using *mean* it provides its average value.

# Confusion Matrix

# 



*TP <- conf\_matrix$table[2, 2] provides* the count of true positives from the confusion matrix. *FP <- conf\_matrix$table[1, 2]* provides the count of false positives from the confusion matrix. *FN <- conf\_matrix$table[2, 1]* provides the count of false negatives from the confusion matrix.