Diabetes Risk Monitor

Use case:

Diabetes is among one of the most prevalent chronic conditions, affecting millions of Americans each year and posing a financial burden on the economy. It can cause various life-threatening complications such as blindness, kidney disease, lower limb amputation, and heart disease. According to the CDC, diabetes is the eighth leading cause of death in the United States, 98 million Americans already have prediabetes and more than 8 out of 10 are unaware that they have prediabetes. The following are the specific stakeholders involved in addressing diabetes risk factors in the United States:

- **General Population**: The main stakeholder for this data visualization is the general population as the main aim of this visualization is to spread awareness regarding the major risk factors involved in the development of diabetes. People usually have risk factors such as high cholesterol, high BMI, and lack of physical activity but they are unaware of how these risk factors can add up to become a fatal chronic disease.
- **Public Health Authorities**: The risk factors of diabetes are diverse. There is a difficulty in identifying and addressing these risk factors. There are gaps in the data analysis and surveillance systems used by public health authorities to track the risk factors of diabetes. Early diagnosis can help to adopt lifestyle changes and start treatment thus making predictive diabetes risk tools for public health officials very important.
- **Healthcare workers (Doctors and Nurses):** The healthcare providers in their busy schedule can sometimes miss individuals with high-risk factors of diabetes. Visualizations can help to highlight the patients who are at high risk of developing diabetes based on their characteristics. This can help doctors to start early screening and preventive interventions.

The solutions to these specific problems can be:

The data visualization gives a visual representation of which risk factors are more involved in the development of diabetes with the help of bar charts and heat plots. This data analysis tool can also help to provide a prediction of whether an individual has diabetes or not. Simply entering your gender, age and other risk factors that a person has the tool can give a prediction of diabetes.

- **For Public Health Authorities:**
  Such data visualizations will enable the authorities to do timely surveillance and assess the different changing trends in risk factors of diabetes. They will be able to better visualize which group of gender and age is at high risk of developing diabetes and focus resource allocation on that group. They can also promote healthy lifestyle changes through policy implementation.
- **Healthcare workers:**

The integration of user-friendly data visualizations can help to facilitate risk assessment of diabetes in a primary care setting. The healthcare professionals should be trained on how to use the data visualization for diabetes risk factor management. By changing one parameter such as gender, age, or BMI the doctors would be able to identify how a change in one risk factor can impact on overall risk of diabetes.

- **General Population:**
  Interactive visualization of diabetes risk factors can help an individual identify how close they are to developing diabetes and can help them to make early preventive lifestyle changes.
- **App developers:** App developers can be involved in designing an upscale data visualization involving various other geographical regions and populations.

Implementation:

Tools:

Diabetes Risk Monitor app is designed using Shiny framework for R. Data analysis is done using R language.

Dataset:

Dataset is collected by the CDC under Behavioural Risk Factor Surveillance System (BRFSS) from over 400,000 Americans. Dataset used for this application is available on Kaggle from year 2015. Dataset contains 253,680 samples with the target variable containing three classes i.e., no diabetes, prediabetes, and diabetes. The dataset has 21 feature variables which can be potential risk factors for diabetes.

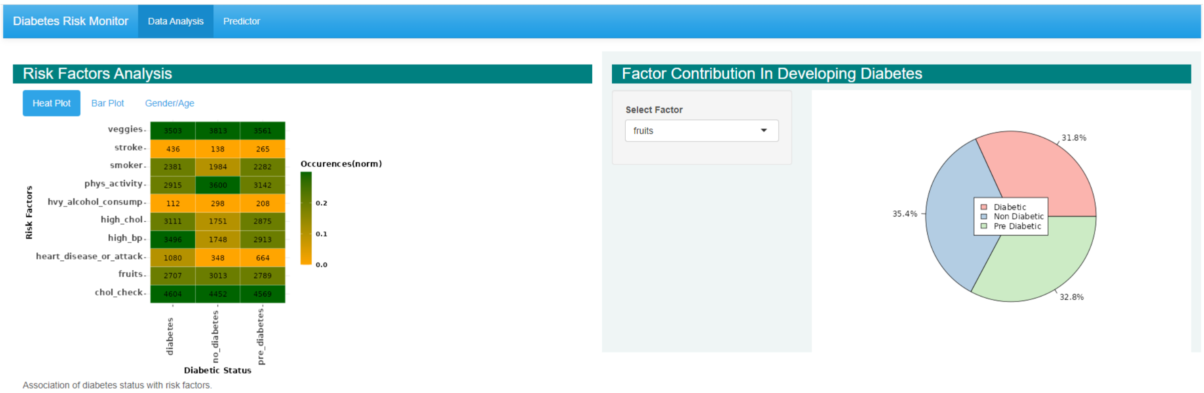
diabetes_012	high_bp	high_chol	chol_check	bmi	smoker	stroke	heart_diseaseor_attack	phys_activity	fruits	veggies	hvy_alcohol_consump	any_healthcare	no_docbc_cost	gen_hlth	ment_hlth	phys_hlth	diff_walk	sex	age	education	income
0	1	1	1	40	1	0	0	0	0	1	0	1	0	5	18	15	1	0	9	4	3
0	0	0	0	25	1	0	0	1	0	0	0	0	1	3	0	0	0	0	7	6	1
0	1	1	1	28	0	0	0	0	1	0	0	1	1	5	30	30	1	0	9	4	8
0	1	0	1	27	0	0	0	1	1	1	0	1	0	2	0	0	0	0	11	3	6
0	1	1	1	24	0	0	0	1	1	1	0	1	0	2	3	0	0	0	11	5	4
0	1	1	1	25	1	0	0	1	1	1	0	1	0	2	0	2	0	1	10	6	8
0	1	0	1	30	1	0	0	0	0	0	0	1	0	3	0	14	0	0	9	6	7
0	1	1	1	25	1	0	0	1	0	1	0	1	0	3	0	0	1	0	11	4	4
2	1	1	1	30	1	0	1	0	1	1	0	1	0	5	30	30	1	0	9	5	1
0	0	0	1	24	0	0	0	0	0	1	0	1	0	2	0	0	0	1	8	4	3
2	0	0	1	25	1	0	0	1	1	1	0	1	0	3	0	0	0	1	13	6	8
0	1	1	1	34	1	0	0	0	1	1	0	1	0	3	0	30	1	0	10	5	1
0	0	0	1	26	1	0	0	0	0	1	0	1	0	3	0	15	0	0	7	5	7
2	1	1	1	28	0	0	0	0	0	1	0	1	0	4	0	0	1	0	11	4	6
0	0	1	1	33	1	1	0	1	0	1	0	1	1	4	30	28	0	0	4	6	2
0	1	0	1	33	0	0	0	1	0	0	0	1	0	2	5	0	0	0	6	6	8
0	1	1	1	21	0	0	0	1	1	1	0	1	0	3	0	0	0	0	10	4	3
2	0	0	1	23	1	0	0	1	0	0	0	1	0	2	0	0	0	1	7	5	6
0	0	0	0	23	0	0	0	0	0	1	0	1	0	2	15	0	0	0	2	6	7
0	0	1	1	28	0	0	0	0	0	0	1	1	0	2	10	0	0	1	4	6	8
0	1	1	1	22	0	1	1	0	1	0	0	1	0	3	30	0	1	0	12	4	4
0	1	1	1	38	1	0	0	0	1	1	0	1	0	5	15	30	1	0	13	2	3
0	0	0	1	28	1	0	0	0	0	1	0	1	0	3	0	7	0	1	5	5	5
2	1	0	1	27	0	0	0	1	1	1	0	1	0	1	0	0	0	0	13	5	4

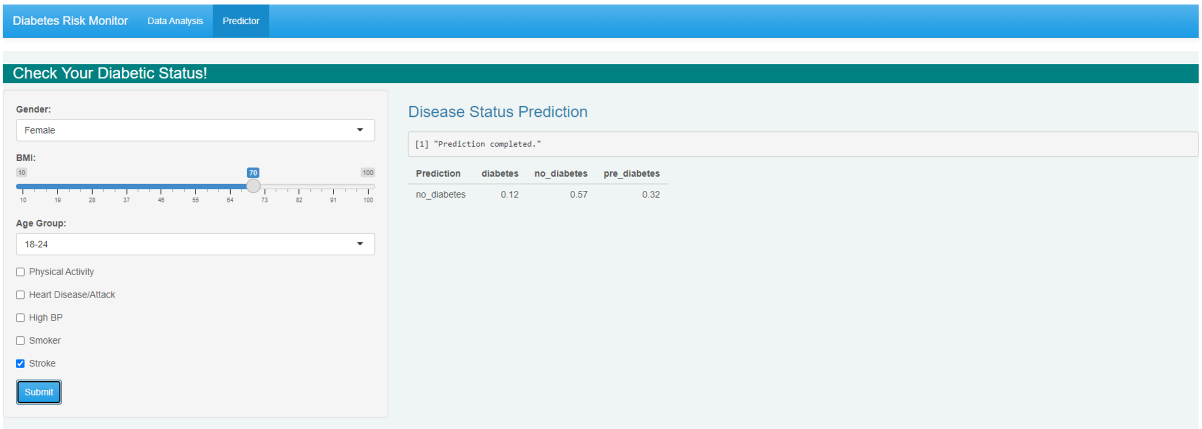
Application Design:

Diabetes Risk Monitor is designed using shiny user interface components. The home page has a navigational theme. A user can navigate between different sections of the app either a user can explore the data analysis part or can use the predictor for disease status prediction. The design theme used is “cerulean” from shiny themes with navigation bars for different sections.

```
ui <- fluidPage(theme = shinytheme('cerulean'),
                navbarPage(
                  "Diabetes Risk Monitor",
```

The following image shows the main user interface:





Code snippet is as follows.

```
ui <- fluidPage(theme = shinytheme('cerulean'),
                navbarPage(
                  "Diabetes Risk Monitor",
                  tabPanel("Data Analysis",
                           fluidRow(
                             column(width = 6,

                                    h3("Risk Factors Analysis",
                                       style='background-color:#008080;
                                              padding-left: 15px;
                                              color: white'),

                                    mainPanel(
                                      navset_card_pill(
                                        nav_panel("Heat Plot",
                                                  plotOutput("riskFactorHeatPlot"),
                                                  p("Association of diabetes status
                                                    with risk factors.")),

                                        nav_panel("Bar Plot",
                                                  plotOutput("riskFactor"),
                                                  p("Distribution of risk
                                                    factors among population.")),

                                        nav_panel("Gender/Age",
                                                  radioButtons("plotType",
                                                               "Select Plot Type:",
                                                               choices = c("pre_diabetes",
                                                                           "diabetes"),
                                                               selected = "diabetes",
                                                               inline = TRUE),
                                                  plotOutput("pyramidPlot"),
                                                  p("Distribution of diabetic
                                                  and prediabetic population with
                                                    respect to age groups"))
                                      )
                                    )
                             ),
```

```
                        column(width = 6,

                               style = "background-color: #eff5f5;",

                               h3("Factor Contribution In Developing Diabetes",
                                  style='background-color:#008080;
                                      padding-left: 15px;
                                      color: white'),

                               sidebarLayout(
                                 sidebarPanel(
                                   selectInput("option", "Select Factor",
                                               choices = c("fruits",
                                                           "heart_disease_or_attack",
                                                           "high_bp",
                                                           "phys_activity",
                                                           "smoker",
                                                           "stroke", "veggies"),
                                               selected = "fruits")
                                 ),
                                 mainPanel(
                                   plotOutput("bubblePlot")
                                 )
                               )
                        )
                      )),
             tabPanel("Predictor",
                      fluidRow(
                        style = "background-color: #eff5f5;",

                        h3("Check Your Diabetic Status!",
                           style='background-color:#008080;
                               padding-left: 15px;
                               color: white'),

                        sidebarLayout(
                          sidebarPanel(
                            selectInput("Gender", label = "Gender:",
                                        choices = list("Female" = "0", "Male" = "1"),
                                        selected = "Female"),

                            sliderInput("bmi", "BMI:",
                                        min = 10, max = 100,
                                        value = 70),

                            selectInput("age", label = "Age Group:",
                                        choices = list("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"),
                                        selected = "25-29"),

                            checkboxInput(inputId = "PhysicalActivity",
                                          label = "Physical Activity",
                                          value = FALSE),

                            checkboxInput(inputId = "HeartDisease",
                                          label = "Heart Disease/Attack",
                                          value = FALSE),

                            checkboxInput(inputId = "Highbp",
                                          label = "High BP",
                                          value = FALSE),

                            checkboxInput(inputId = "smoker",
                                          label = "Smoker",
                                          value = FALSE),

                            checkboxInput(inputId = "stroke",
                                          label = "Stroke",
                                          value = FALSE),

                            actionButton("submitbutton", "Submit",
                                         class = "btn btn-primary")
                          ),

                          # Show a plot of the generated distribution
                          mainPanel(
                            tags$label(h3('Disease Status Prediction')), # Status/Output Text Box
                            verbatimTextOutput('contents'),
                            tableOutput('tabledata') # Prediction results table
                          )
                        )
                      )

             )# tabPanel

  ) # navbarPage
)
```

The page design contains following components:

- Navigation Bar
- Tab Panels (Data Analysis, Predictor)
- Navigation panels
- Fluid Row
- Two Columns

- Headers
- Side bar layout
- Main panel

Input and output components are following:

- Radio buttons
- Input selector from drop down menu.
- Checkbox input
- Action button
- Table output
- Plot outputs

Styling is done using CSS components.

```
fluidRow(
  style = "background-color: #eff5f5;",

  h3("Check Your Diabetic Status!",
     style='background-color:#008080;
         padding-left: 15px;
         color: white'),
```

Application Functionality:

Application contains two R files. First file is data_processing.R which is responsible for data loading, data cleaning and preparation, feature selection.

Data Pre-processing:

1. Raw dataset is loaded, and columns values are adjusted from the data analysis.

```
age_gender_data <- raw_data %>%
  select(age, sex, diabetic_status) %>%
  mutate(age_cat=case_when(age == 1 ~ "18-24",
                           age == 2 ~ "25-29",
                           age == 3 ~ "30-34",
                           age == 4 ~ "35-39",
                           age == 5 ~ "40-44",
                           age == 6 ~ "45-49",
                           age == 7 ~ "50-54",
                           age == 8 ~ "55-59",
                           age == 9 ~ "60-64",
                           age == 10 ~ "65-69",
                           age == 11 ~ "70-74",
                           age == 12 ~ "75-79",
                           age == 13 ~ "80+"),
         age_cat=factor(age_cat),
         sex=case_when(sex == 0 ~ "female",
                       sex == 1 ~ "male"))

data <- raw_data %>%
  select(diabetes_012, high_bp, high_chol, chol_check, smoker, stroke, heart_diseaseor_attack, phys_activity, fruits, veggies, hvy_alcohol_consump, diabetic_status)

colnames(data)[which(names(data) == "heart_diseaseor_attack")] <- "heart_disease_or_attack"
```

2. For the statistical analysis based on modifiable risk factors, a subset of columns is selected from raw data.

```
data <- raw_data %>%
  select(diabetes_012, high_bp, high_chol, chol_check, smoker, stroke,
         heart_diseaseor_attack, phys_activity, fruits, veggies,
         hvy_alcohol_consump, diabetic_status)
```

3. As data is highly unbalanced so a min sampling technique is applied to equalized samples from all three classes.

```
# Aggregate the data
min_samples <- min(table(data$diabetes_012))

equal_data <- data %>%
  group_by(diabetes_012) %>%
  sample_n(min_samples)


agg_data <- equal_data %>%
  group_by(diabetic_status) %>%
  summarise(high_bp = sum(high_bp),
            high_chol = sum(high_chol),
            chol_check = sum(chol_check),
            smoker = sum(smoker),
            stroke = sum(stroke),
            heart_disease_or_attack = sum(heart_disease_or_attack),
            phys_activity = sum(phys_activity),
            fruits = sum(fruits),
            veggies = sum(veggies),
            hvy_alcohol_consump = sum(hvy_alcohol_consump))

# Reshape the data for ggplot
plot_data <- agg_data %>%
  gather(key = "factor", value = "value", -diabetic_status)

> table(raw_data$diabetic_status)

  diabetese  no_diabetese pre_diabetese
      35346        213703          4631
> table(equal_data$diabetic_status)

  diabetese  no_diabetese pre_diabetese
       4631          4631          4631
```

Diabetic Prediction Model:

A random forest tree model is trained on the features of the risk factors to predict probability of a user belonging to one of the three classes i.e., diabetic, no diabetic and pre diabetic. A subset of the risk factors has been used to train the predictor. This subset has BMI, physical activity, heart disease or attack, high blood pressure, smoker, storke, age and gender. It shows which risk factors are most predictive of diabetes risk.

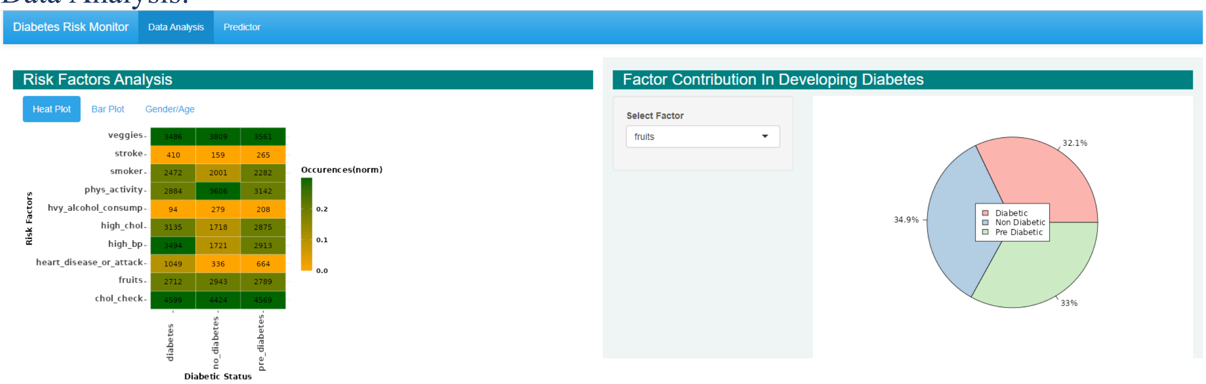
```
clustering_equal_data <- raw_data %>%
  select(diabetes_012, bmi, phys_activity, heart_diseaseor_attack, high_bp, smoker, stroke, age, sex, diabetic_status) %>%
  group_by(diabetes_012) %>%
  sample_n(min_samples)

model <- randomForest(diabetic_status ~ ., data = clustering_equal_data, ntree = 500, mtry = 8, importance = TRUE)
```

Random forest is trained of 8 input variables to predict probability of three classes.

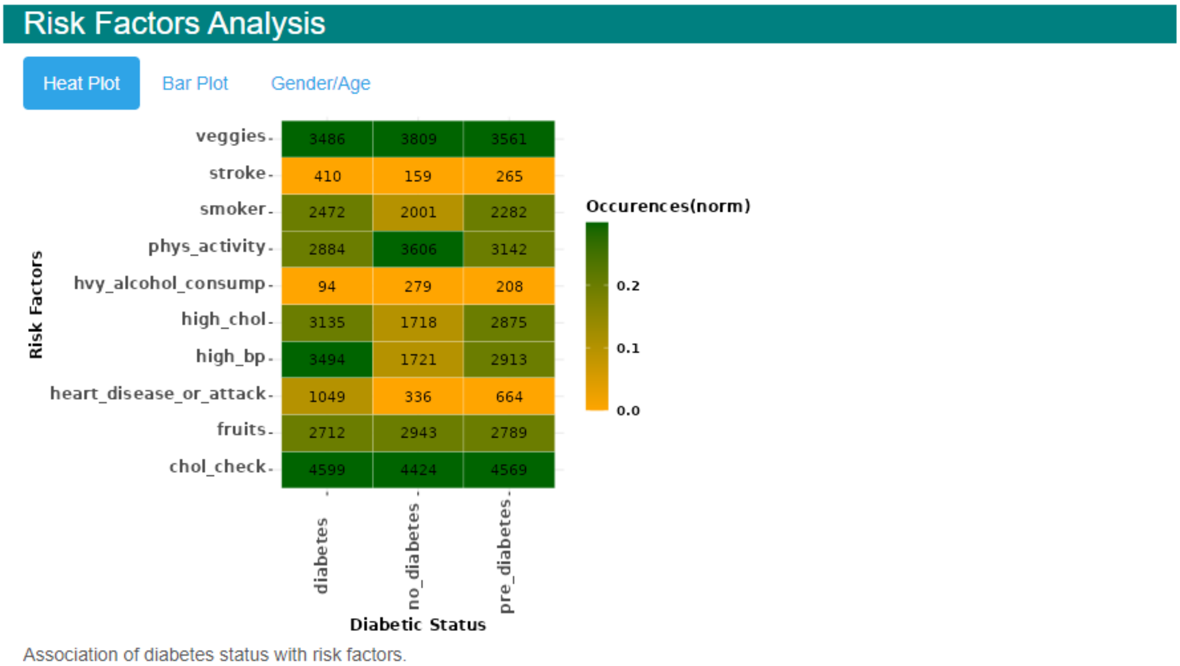
Main Application and Data Visualization:

Main application layout code is present in app.R. This code module contains design, plotting and interactivity. Main app is divided in to tow following sections:

Data Analysis:



Association of diabetes status with risk factors.

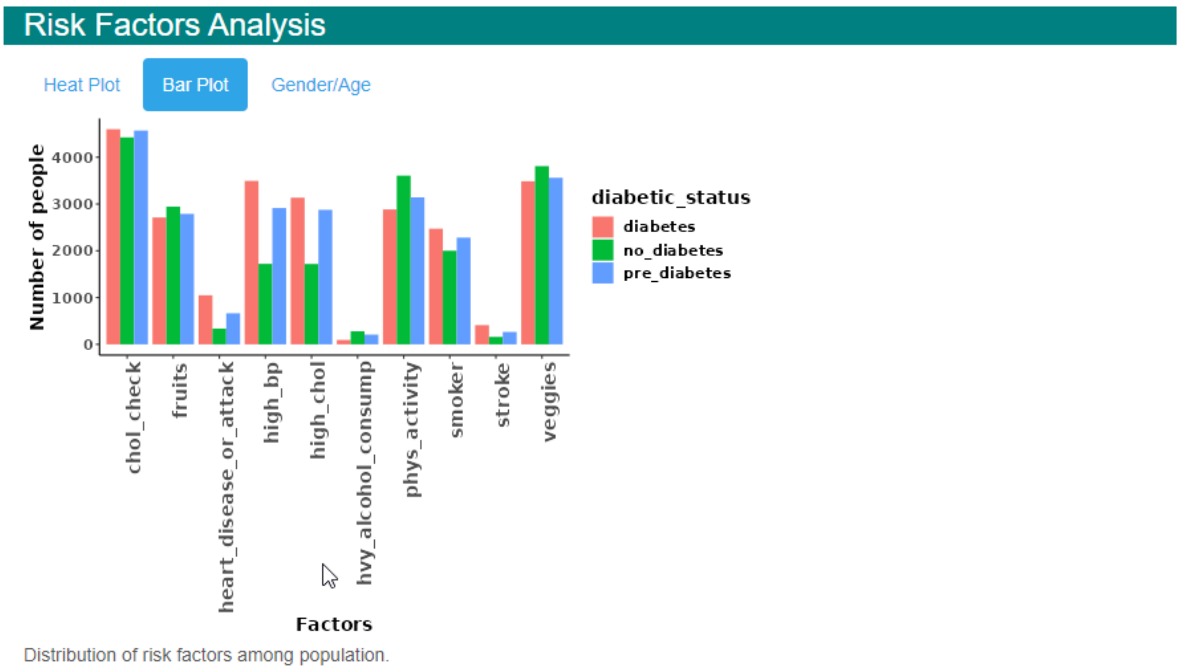


UI code:

```
mainPanel(
  navset_card_pill(
    nav_panel("Heat Plot",
              plotOutput("riskFactorHeatPlot"),
              p("Association of diabetes status
                with risk factors.")),
```

Server Code:

```
output$riskFactorHeatPlot <- renderPlot({

  plot_data$norm <- round(plot_data$value /
                            nrow(equal_data), 1)
  # Create the plot
  ggplot(plot_data, aes(x = diabetic_status, y = factor, fill = norm))+
    geom_tile(colour="white")+
    geom_text(aes(label = value)) +
    scale_fill_gradient(low = "orange", high = "darkgreen", na.value = "grey80")+
    theme_minimal()+
    theme(
      legend.title = element_text(size=12, face="bold"),
      legend.text  = element_text(size=10, face="bold"),
      legend.key.height = grid::unit(1,"cm"),
      legend.key.width  = grid::unit(0.6,"cm"),
      axis.text.x = element_text(angle = 90, vjust=0.05, size=12, face='bold'),
      axis.text.y = element_text(size=12, vjust=0.2, face='bold'),
      axis.ticks = element_line(size=0.4),
      axis.title = element_text(size=12, face="bold")
    )+
    labs(x = "Diabetic Status",
         y = "Risk Factors",
         fill = "Occurences(norm)")
})
```



UI Code:

```
                                        nav_panel("Bar Plot",
                                                  plotOutput("riskFactor"),
                                                  p("Distribution of risk
                                                    factors among population.")),
```

Server Code:

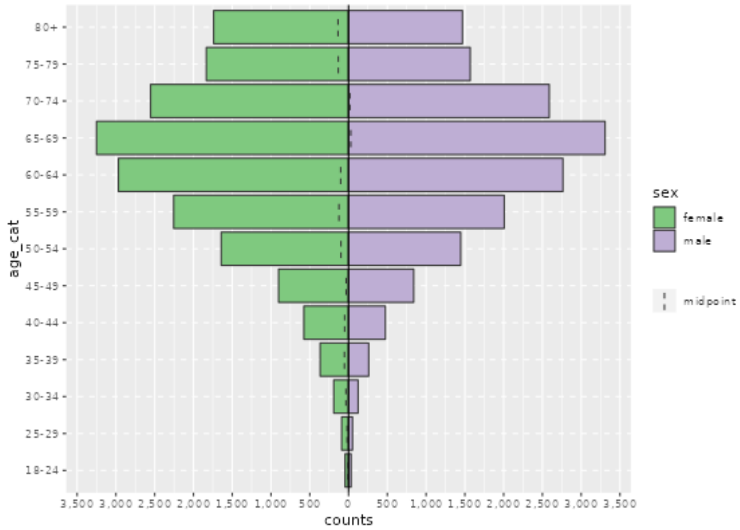
```
  output$riskFactor <- renderPlot({

    # Create the plot
    ggplot(plot_data, aes(x = factor, y = value, fill = diabetic_status)) +
      geom_bar(stat = "identity", position=position_dodge()) +
      labs(x = "Factors", y = "Number of people") +
      theme_classic() +
      theme(axis.text.x = element_text(angle = 90, hjust = 1, size = 14,
                                       face = "bold"),
            text = element_text(size = 14, face = "bold"))
  })

```

Risk Factors Analysis

Heat Plot | Bar Plot | Gender/Age

Select Plot Type:

○ pre_diabetes ◉ diabetes



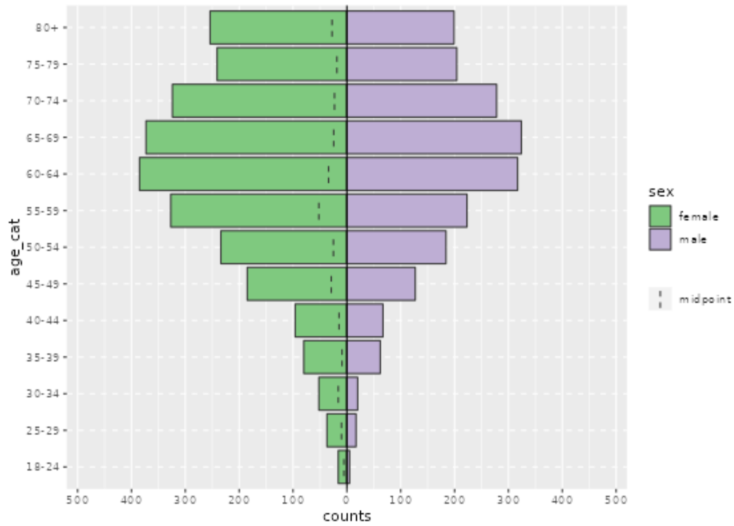
Distribution of diabetic and prediabetic population with respect to age groups

Risk Factors Analysis

Heat Plot | Bar Plot | Gender/Age

Select Plot Type:

◉ pre_diabetes ○ diabetes



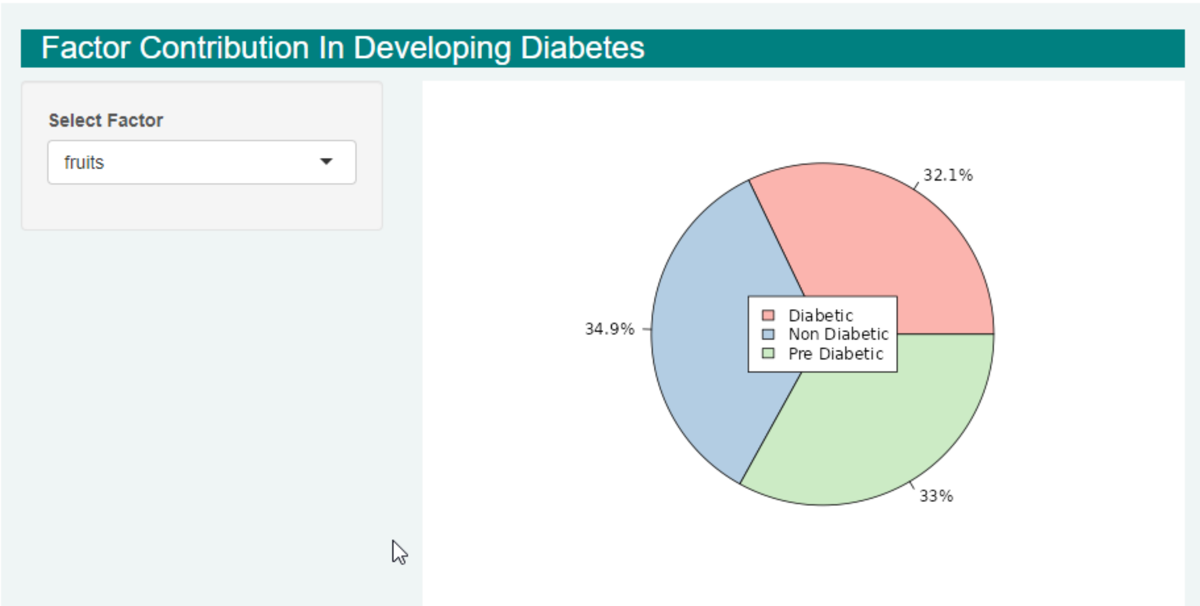
Distribution of diabetic and prediabetic population with respect to age groups

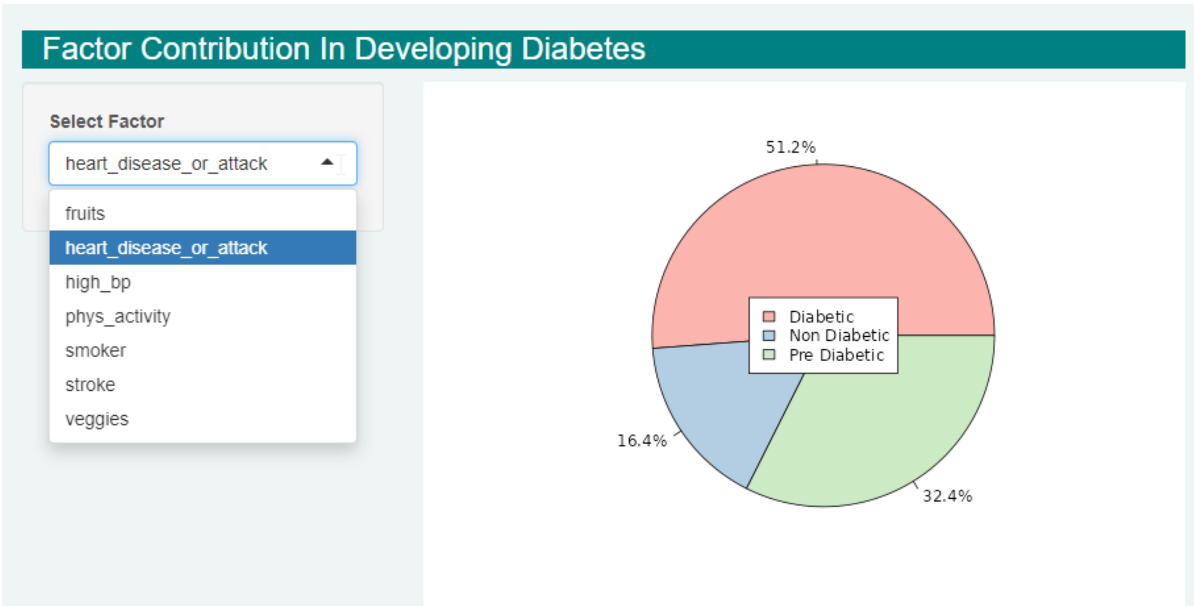
UI code:

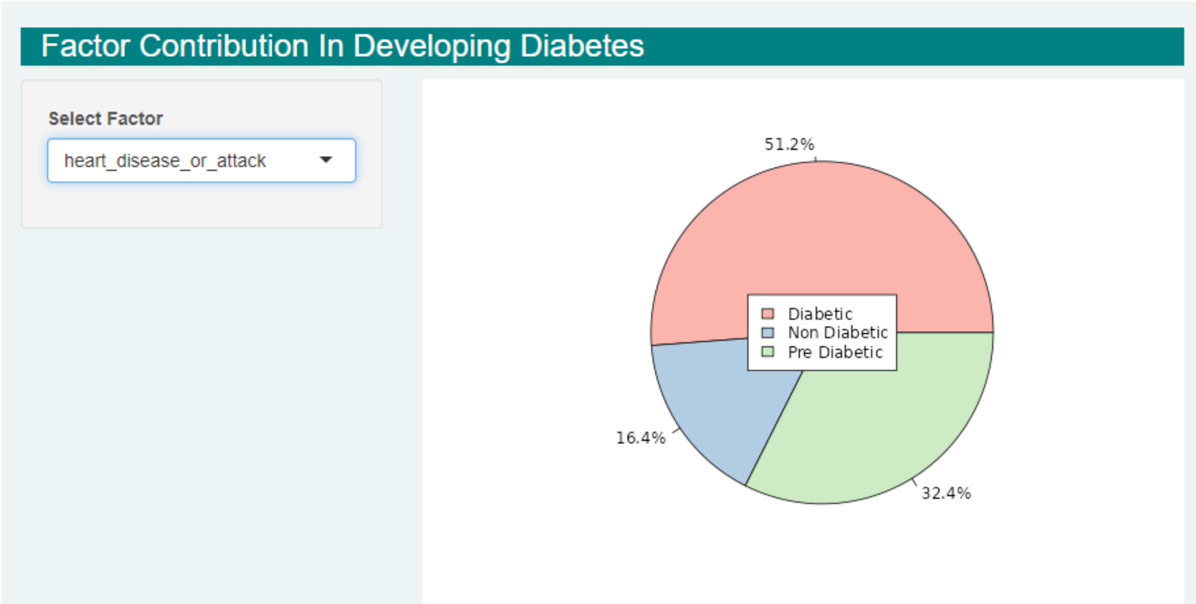
```
nav_panel("Gender/Age",
          radioButtons("plotType",
                       "Select Plot Type:",
                       choices = c("pre_diabetes",
                                   "diabetes"),
                       selected = "diabetes",
                       inline = TRUE),
          plotOutput("pyramidPlot"),
          p("Distribution of diabetic
and prediabetic population with
  respect to age groups"))
)
)
```

Server code:

```
output$pyramidPlot <- renderPlot({
  plot_choice <- input$plotType
  filtered_data <- age_gender_data %>%
    filter(diabetic_status == plot_choice) #%>%

  apyramid::age_pyramid(data = filtered_data,
                        age_group = "age_cat",
                        split_by = "sex")
})

```


Factor Contribution In Developing Diabetes
Select Factor
fruits
32.1%
34.9%
33%
Diabetic
Non Diabetic
Pre Diabetic


Factor Contribution In Developing Diabetes
Select Factor
heart_disease_or_attack
fruits
heart_disease_or_attack
high_bp
phys_activity
smoker
stroke
veggies
51.2%
16.4%
32.4%
Diabetic
Non Diabetic
Pre Diabetic



UI code:

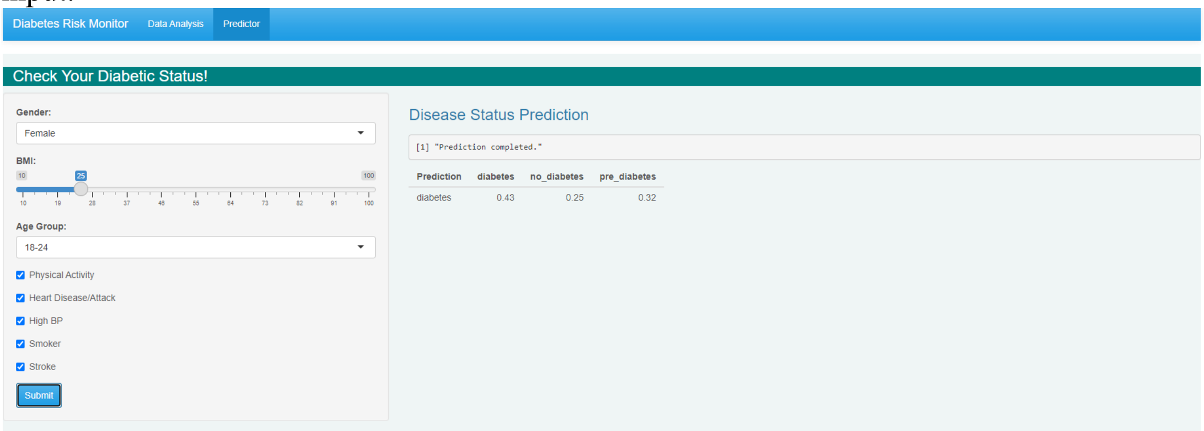
```
column(width = 6,

       style = "background-color: #eff5f5;",

       h3("Factor Contribution In Developing Diabetes",
          style='background-color:#008080;
             padding-left: 15px;
             color: white'),

       sidebarLayout(
         sidebarPanel(
           selectInput("option", "Select Factor",
                       choices = c("fruits",
                                   "heart_disease_or_attack",
                                   "high_bp",
                                   "phys_activity",
                                   "smoker",
                                   "stroke", "veggies"),
                       selected = "fruits")
         ),
         mainPanel(
           plotOutput("bubblePlot")
         )
       )
)
)),
```

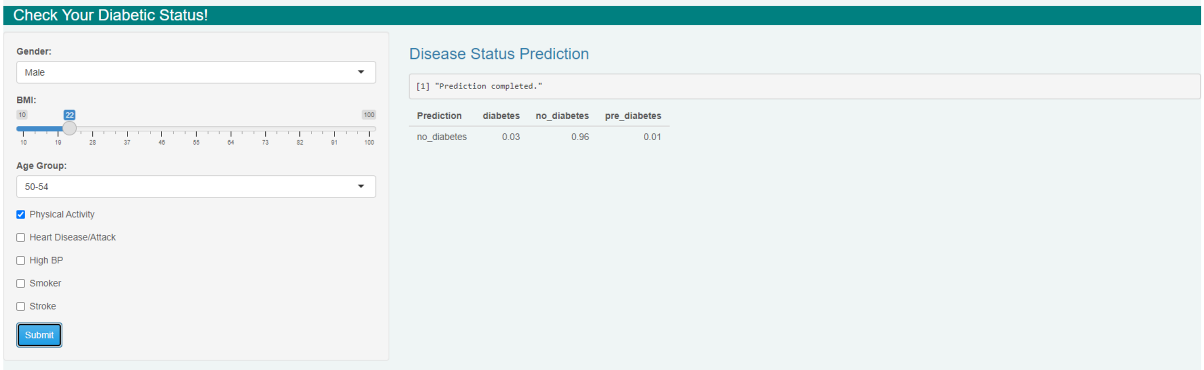
Server code:

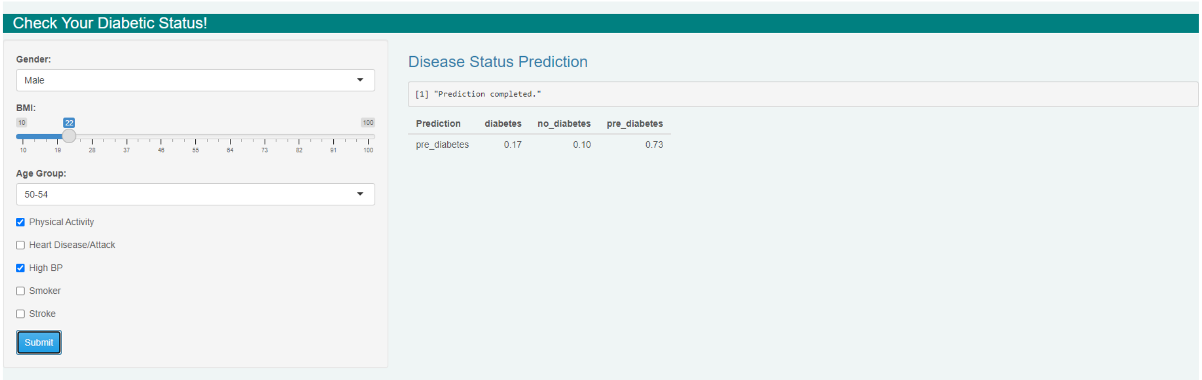
```r
output$piePlot <- renderPlot({
  # Get selected option
  selected_option <- input$option

  # Based on the selected option, choose the corresponding data
  filtered_data <- plot_data %>% filter(factor == selected_option)

  # Create pie plot
  filtered_data$percent <- round(filtered_data$value /
                                 sum(filtered_data$value)*100, 1)
  filtered_data$percent_label <- paste0(filtered_data$percent, "%")
  pie(x = filtered_data$value, labels = filtered_data$percent_label,
      angle = 180,
      col = brewer.pal(n = nrow(filtered_data), name = "Pastel1"),
      radius = 1.05)
  legend("center", legend = c("Diabetic", "Non Diabetic", "Pre Diabetic"),
         fill = brewer.pal(n = nrow(filtered_data), name = "Pastel1"))
})
```

Diabetic Status Predictor:

Diabetic status predictor allows user to input their variables like gender, age, physical activity, high blood pressure status, smoker, stroke, and BMI. The predictor is a random forest trained model, this will predict the status of diabetes for the user with respect to their input.







UI Code:

```
                    tabPanel("Predictor",
                             fluidRow(
                               style = "background-color: #eff5f5;",

                               h3("Check Your Diabetic Status!",
                                  style='background-color:#008080;
                                      padding-left: 15px;
                                      color: white'),

                               sidebarLayout(
                                 sidebarPanel(
                                   selectInput("Gender", label = "Gender:",
                                               choices = list("Female" = "0", "Male" = "1"),
                                               selected = "Female"),

                                   sliderInput("bmi", "BMI:",
                                               min = 10, max = 100,
                                               value = 70),

                                   selectInput("age", label = "Age Group:",
                                               choices = list("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"),
                                               selected = "25-29"),

                                   checkboxInput(inputId = "PhysicalActivity",
                                                 label = "Physical Activity",
                                                 value = FALSE),

                                   checkboxInput(inputId = "HeartDisease",
                                                 label = "Heart Disease/Attack",
                                                 value = FALSE),

                                   checkboxInput(inputId = "Highbp",
                                                 label = "High BP",
                                                 value = FALSE),

                                   checkboxInput(inputId = "smoker",
                                                 label = "Smoker",
                                                 value = FALSE),

                                   checkboxInput(inputId = "stroke",
                                                 label = "Stroke",
                                                 value = FALSE),

                                   actionButton("submitbutton", "Submit",
                                                class = "btn btn-primary")
                                 ),
```

```
                                # Show a plot of the generated distribution
                                mainPanel(
                                  tags$label(h3('Disease Status Prediction')), # Status/Output Text Box
                                  verbatimTextOutput('contents'),
                                  tableOutput('tabledata') # Prediction results table
                                )
                              )
                            )

                   )# tabPanel

                 ) # navbarPage
```

Server code:

```
# Input Data
datasetInput <- reactive({

  # outlook,temperature,humidity,windy,play
  df <- data.frame(
    Name = c("bmi",
             "phys_activity",
             "heart_diseaseor_attack",
             "high_bp",
             "smoker",
             "stroke",
             "age",
             "sex"),
    Value = as.character(c(input$bmi,
                           input$PhysicalActivity,
                           input$HeartDisease,
                           input$Highbp,
                           input$smoker,
                           input$stroke,
                           input$age,
                           input$Gender)),
    stringsAsFactors = FALSE)

  diabetic_status <- "diabetic_status"
  df <- rbind(df, diabetic_status)
  print(df)
  input <- as.data.frame(t(as.matrix(df)))
  # Set column names to the first row of the transposed dataframe
  colnames(input) <- input[1, ]
  # Remove the first row (column names row)
  transposed_df <- input[-1, ]
  print(input)
  write.table(input,"input.csv", sep=",", quote = FALSE, row.names = FALSE, col.names = FALSE)

  test <- read.csv(paste("input", ".csv", sep=""), header = TRUE)

  test$bmi <- as.numeric(test$bmi)
  test$phys_activity <- as.numeric(as.logical( test$phys_activity))
  test$heart_diseaseor_attack <- as.numeric(as.logical( test$heart_diseaseor_attack))
  test$high_bp <- as.numeric(as.logical( test$high_bp))
  test$smoker <- as.numeric(as.logical( test$smoker))
  test$stroke <- as.numeric(as.logical( test$stroke))
  test$age <- as.numeric(test$age)
  test$sex <- as.numeric(test$sex)

  Output <- data.frame(Prediction=predict(model,test), round(predict(model,test,type="prob"), 3))
  #print(Output)
})

# Status/Output Text Box
output$contents <- renderPrint({
  if (input$submitbutton>0) {
    isolate("Prediction completed.")
  } else {
    return("Ready to predict.")
  }
})

# Prediction results table
output$tabledata <- renderTable({
  if (input$submitbutton>0) {
    isolate(datasetInput())
  }
})
```