ESCOLA ESTADUAL GREGORIANO CANEDO

Uso de algoritmos de aprendizado de máquina para prever o risco de diabetes tipo 2 usando a linguagem de programação R

Monte Carmelo, MG 2023

Heitor Leal Rocha e Silva

Melinna Machado Cardoso
Luiz Pires

Carolina Alves Costa

Orientador: Jair Rodrigues de Andrade

Uso de algoritmos de aprendizado de máquina para prever o risco de diabetes tipo 2 usando a linguagem de programação R

Relatório apresentado à 7ª FEMIC - Feira Mineira de Iniciação Científica.
Orientação do Prof. Jair Rodrigues de Andrade.

Monte Carmelo, MG 2023

RESUMO

O diabetes tipo 2 é uma doença crônica que afeta a forma como o corpo usa a glicose (açúcar no sangue). Existem vários fatores que podem aumentar o risco de desenvolver diabetes tipo 2, incluindo idade, obesidade, histórico familiar de diabetes e sedentarismo. Não existe cura para o diabetes tipo 2, mas ele pode ser controlado com dieta, exercícios físicos e medicamentos. Recentemente, pesquisadores desenvolveram um modelo de aprendizado de máquina que pode prever com precisão se uma pessoa desenvolverá diabetes tipo 2 com base em seus dados médicos. O modelo foi treinado em um conjunto de dados de mais de 700 pacientes e foi capaz de prever com precisão a probabilidade de desenvolvimento de diabetes com uma precisão superior a 80%. Este modelo pode ser usado para identificar pacientes com maior risco de desenvolver diabetes e oferecer-lhes intervenções precoces para prevenir a doença. Neste projeto, utilizamos a linguagem de programação R na plataforma R Studio e o método Ensemble com algoritmos Randon Forest e gradiente descendente para o processo de otimização que permite melhorar a acurácia do modelo preditivo. Os resultados do projeto mostraram que o modelo Ensemble foi capaz de melhorar a acurácia do modelo de previsão de diabetes em em quase 8%. Isso significa que o modelo Ensemble é capaz de prever com precisão a probabilidade de desenvolvimento de diabetes com uma precisão de aproximadamente 88% utilizando o algoritmo Random Forest. Os resultados do projeto são promissores e sugerem que o uso de algoritmos de aprendizado de máquina pode ser uma ferramenta eficaz para a prevenção e o tratamento do diabetes tipo 2.

Palavras-chave: Diabetes tipo 2, Fatores de risco, Aprendizado de máquina, Modelo de previsão, Acurácia, Intervenções precoces, Prevenção, Tratamento.

SUMÁRIO

1 INTRODUÇÃO

Segundo a Federação Internacional de Diabetes (IDF), em 2021, estima-se que haja cerca de 463 milhões de pessoas vivendo com diabetes no mundo, sendo que aproximadamente 90% a 95% desses casos correspondem à diabetes tipo 2. Esses números são preocupantes, uma vez que a diabetes tipo 2 está associada a complicações graves, como doenças cardiovasculares, danos nos rins, problemas nos olhos e neuropatias.

Diante da crescente prevalência da diabetes tipo 2 e suas implicações para a saúde, tornou-se fundamental estudar as diferentes variáveis que interferem nos altos números dessa doença. Essas variáveis incluem fatores genéticos, estilo de vida, dieta, obesidade, sedentarismo, idade e histórico familiar. Além disso, existem fatores socioeconômicos e ambientais que também desempenham um papel importante.

O *diabetes mellitus*[1] é uma doença crônica que afeta o metabolismo da glicose (açúcar no sangue). Existem três tipos principais de diabetes mellitus: tipo 1, tipo 2 e diabetes gestacional. O diabetes tipo 1 é uma doença autoimune em que o pâncreas não produz insulina. O diabetes tipo 2 é uma doença metabólica em que o corpo não produz insulina suficiente ou não pode usar a insulina de forma eficaz. O diabetes gestacional é uma doença que ocorre durante a gravidez e geralmente desaparece após o parto. Além desses três tipos principais de diabetes mellitus, existem outros tipos de diabetes menos comuns, como o diabetes monogênico, o diabetes secundário e o diabetes de início tardio na infância. Vários fatores contribuem para o desenvolvimento do diabetes tipo 2, incluindo predisposição genética, obesidade, estilo de vida sedentário, dieta inadequada e fatores socioeconômicos (*American Diabetes Association*, 2023). Estudar esses fatores e entender como eles se interrelacionam é essencial para combater a alta prevalência da doença.

Um estudo recente publicado na revista *JAMA Network Open* descobriu que o número de pessoas com diabetes tipo 2 aumentou em 17% nos Estados Unidos entre 2020 e 2021. Os pesquisadores acreditam que esse aumento pode estar relacionado à pandemia de COVID-19, que levou a mudanças no estilo de vida, com diminuição da atividade física e aumento

do consumo de alimentos não saudáveis. Eles também sugerem que a pandemia pode ter dificultado o acesso aos cuidados médicos para pessoas com diabetes (Felix, 2023).

A conscientização da importância da atividade física para melhoria da qualidade de vida de pessoas portadoras do diabetes tipo 2 é fundamental para prevenir complicações e promover a saúde. Comunidades que se mobilizam e promovem atividades físicas regulares para portadores do diabetes tipo 2 podem alcançar resultados significativos na prevenção e controle da doença. É importante destacar que a prática de exercícios físicos deve ser realizada com a orientação de profissionais qualificados e adaptada às necessidades individuais de cada pessoa [2].

Um recurso valioso para a análise e estudo da diabetes é o banco de dados público conhecido como diabetes.csv. Esse conjunto de dados contém informações sobre variáveis relevantes, como idade, índice de massa corporal (IMC), pressão arterial, níveis de glicose no sangue e histórico familiar de diabetes. Com base nesses dados é possível realizar análises estatísticas e construir modelos preditivos para compreender melhor os fatores de risco para o desenvolvimento da doença (Pima Indians Diabetes Database, n.d.). [3]

A ciência de dados é uma área interdisciplinar que estuda métodos, processos, algoritmos e sistemas para extrair conhecimento e *insights* a partir de grandes volumes de dados. Os cientistas de dados usam uma variedade de técnicas, incluindo estatística, aprendizado de máquina, mineração de dados e visualização de dados para identificar padrões e tendências em dados complexos. Assim, ela pode usada em uma ampla gama de aplicações, incluindo negócios, saúde, finanças, governo e ciência (O'Neil, 2017).

A ciência de dados está revolucionando a área da saúde, permitindo o desenvolvimento de novos diagnósticos, tratamentos e prevenção de doenças. A análise de grandes volumes de dados de saúde, como registros médicos eletrônicos, imagens de diagnóstico e dados genômicos, está permitindo aos cientistas identificarem padrões e tendências que antes eram invisíveis. Isso está levando ao desenvolvimento de novos medicamentos e terapias mais eficazes, bem como a personalização dos tratamentos para cada paciente. A ciência de dados também está sendo usada para desenvolver novas ferramentas para a prevenção de doenças, como sistemas de rastreamento e diagnóstico

precoce. Como resultado, ela está tendo um impacto significativo na melhoria da saúde e da qualidade de vida das pessoas (Kohane, 2015).

Nesse contexto o aprendizado de máquina (*machine learning*) desempenha um papel crucial. O aprendizado de máquina é uma área da inteligência artificial que permite que os computadores aprendam e façam previsões ou tomem decisões sem serem explicitamente programados. Os algoritmos de aprendizado de máquina podem ser aplicados ao conjunto de dados da diabetes para identificar padrões, fazer previsões e gerar *insights* sobre a progressão da doença.

Um estudo recente publicado na revista *Nature Medicine* descobriu que um modelo de aprendizado de máquina baseado em um conjunto de algoritmos, incluindo *Random Forest*, *Gradient Descent e Support Vector Machine* (SVM) foi capaz de prever com precisão a probabilidade de um paciente desenvolver diabetes tipo 2 com base em seus dados médicos. O modelo foi treinado em um conjunto de dados de mais de 100.000 pacientes e foi capaz de prever com precisão a probabilidade de desenvolvimento de diabetes com uma precisão de 90%. Os pesquisadores acreditam que esse modelo pode ser usado para identificar pacientes com maior risco de desenvolver diabetes e oferecer-lhes intervenções precoces para prevenir a doença (Zhang *et al.*, 2020).

O algoritmo *Random Forest* é um algoritmo de aprendizado de máquina supervisionado que pode ser usado para classificação, regressão e agrupamento. Ele é baseado na ideia de criar uma floresta de árvores de decisão, cada uma das quais é treinada em um subconjunto aleatório dos dados. As previsões são então feitas pela combinação das previsões de cada árvore. Por isso, ele é um método poderoso e versátil que pode ser usado para uma ampla gama de problemas de ciência de dados. Ele tem sido usado com sucesso para resolver problemas na área da saúde, como a predição do risco de doenças, a identificação de padrões em dados médicos e o desenvolvimento de novos tratamentos (Leo Breiman, 2001).

Uma linguagem de programação frequentemente utilizada para análise de dados e modelagem preditiva é a linguagem R. O R é uma linguagem de programação e um ambiente livre e aberto para computação estatística, e é desenvolvido pela Fundação R, que é formada

por uma comunidade ativa de usuários e desenvolvedores ao redor do mundo. O R é uma linguagem poderosa e flexível que pode ser usada para uma ampla gama de tarefas, incluindo análise de dados, visualização de dados e aprendizado de máquina. Ele também é uma linguagem de programação popular para pesquisa acadêmica, pois é livre, de código aberto e possui diversos recursos de estatísticas e gráficos. Como resultado, o *R* tornou-se uma ferramenta popular para cientistas de dados, engenheiros e pesquisadores de todos os níveis de experiência (*R Core Team*, 2023).

O RStudio é um ambiente de desenvolvimento integrado gratuito e de código aberto para R. Ele oferece uma interface gráfica com o usuário (GUI) intuitiva e poderosa que facilita o desenvolvimento, depuração e execução de código R *(RStudio Team*, 2023).

Portanto, a junção do estudo da diabetes, o uso de técnicas de aprendizado de máquina e a aplicação da linguagem R permitem a construção de modelos preditivos que podem contribuir significativamente para a compreensão dos fatores de risco e desenvolvimento da diabetes tipo 2. Essa abordagem integrada possibilita a identificação de padrões e a tomada de decisões baseadas em dados, auxiliando na prevenção, diagnóstico e tratamento mais eficazes da doença.

2 JUSTIFICATIVA

O projeto de Iniciação Científica proposto tem como objetivo desenvolver um modelo preditivo de diabetes tipo 2 baseado em aprendizado de máquina. O projeto é relevante para a área da saúde, pois pode contribuir para o diagnóstico precoce da doença, o que pode levar a intervenções mais eficazes e à melhoria da qualidade de vida dos pacientes.

A diabetes tipo 2 é uma doença crônica que afeta a forma como o corpo usa a glicose. A doença é causada por uma combinação de fatores genéticos e ambientais, incluindo obesidade, sedentarismo, idade e histórico familiar. A diabetes tipo 2 é uma das principais causas de morte no mundo, e pode levar a complicações graves, como doenças cardíacas, derrames, insuficiência renal e cegueira.

O diagnóstico precoce do diabetes tipo 2 é importante para que os pacientes possam iniciar o tratamento o mais cedo possível. O tratamento do diabetes tipo 2 pode ajudar a prevenir complicações e melhorar a qualidade de vida dos pacientes.

3 OBJETIVOS

3.1 Objetivo geral

Aplicar o modelo preditivo de aprendizado de máquina para o diagnóstico do diabetes tipo 2 utilizando a linguagem R.

3.2 Objetivos específicos

- Explorar os artifícios da inteligência artificial (IA) com aprendizado de máquina ou *Machine Learning*.
- Utilizar o método Ensemble com algoritmos Randon Forest e gradiente descendentes para o processo de otimização para melhorar a acurácia do modelo preditivo facilitando o diagnóstico do diabetes tipo 2.
- Mostrar a importância da disciplina ciência de dados para desenvolver a multidisciplinaridade dos conteúdos do Novo Ensino Médio.
- Discutir sobre as metodologias de pesquisa na área das Ciências da Natureza e da Matemática e suas Tecnologias utilizando um banco de dados e artigos científicos.
- Dialogar sobre as metodologias de análise de dados como a linguagem de programação R com a plataforma do *R studio* nas pesquisas em Ciências da Natureza e Matemática e suas Tecnologias

4 METODOLOGIA

Nesta seção, serão apresentadas as etapas metodológicas adotadas para a realização do estudo sobre mineração de dados, com foco na análise exploratória de dados e no uso da plataforma R Studio como ferramenta fundamental para o desenvolvimento da pesquisa sobre a construção de modelos preditivos com algoritmos de aprendizado de máquina .

4.1. Acesso à Plataforma Kaggle

O principal desafio enfrentado pelos pesquisadores na área de ciência de dados reside na necessidade de trabalhar com bancos de dados conformes às normas internacionais. Para superar essa dificuldade, optou-se por utilizar a plataforma Kaggle, que se destaca como uma fonte confiável de conjuntos de dados e recursos para análise e modelagem de dados. O acesso à Kaggle permite que os pesquisadores trabalhem com dados editados de acordo com padrões internacionais, proporcionando uma base sólida para a pesquisa em questão.

4.2. Introdução à Mineração de Dados

A mineração de dados é uma disciplina essencial dentro da ciência de dados, e sua compreensão é fundamental para o sucesso deste estudo. Antes de adentrar nas técnicas de análise exploratória de dados, é necessário estabelecer os conceitos-chave, tais como banco de dados, dataset e dataframe (df). Esses elementos formam a base sobre a qual a análise de dados será construída. A análise exploratória de dados é uma abordagem crítica que permite revelar padrões e tendências ocultas nos dados, contribuindo para o entendimento do problema em questão.

4.3. Relação entre Tratamento de Dados e Algoritmos de Aprendizado de Máquina

Estudos prévios têm ressaltado a importância da abordagem metódica no tratamento dos dados e na seleção adequada dos algoritmos de aprendizado de máquina. A relação entre esses dois aspectos é crucial para alcançar uma previsão precisa da variável de saída desejada.

A busca por maior acurácia nos resultados requer uma análise cuidadosa da qualidade dos dados e a escolha adequada dos algoritmos de aprendizado de máquina.

4.4. Utilização da Plataforma R Studio

A ferramenta digital R Studio desempenha um papel central neste estudo. Trata-se de um ambiente de acesso aberto projetado para a aplicação de conhecimentos de programação na linguagem R. Este ambiente oferece recursos essenciais para a realização de análises de dados e modelagem de aprendizado de máquina de forma eficiente e colaborativa. É relevante observar que o R Studio não se limita apenas ao campo da ciência de dados, sendo amplamente utilizado em outras áreas acadêmicas e profissionais.

4.5. Análise Exploratória de Dados

A análise exploratória de dados é uma etapa crítica no processo de mineração de dados. Para uma análise mais detalhada, é importante seguir as seguintes fases:

4.5.1. Compreensão do Business Problem ou Estudo de Caso

Antes de iniciar qualquer análise, é crucial compreender completamente o problema de negócio ou estudo de caso relacionado ao banco de dados. Isso envolve uma investigação profunda das variáveis de entrada (x) e da variável dependente de saída (y).

Inicialmente, empregamos a função read.csv para ler e carregar um arquivo chamado diabetes.csv, que contém dados separados por vírgulas, em um formato de tabela, criando assim um data frame denominado "diabetes". Em seguida, aplicamos a função head para examinar os primeiros registros desse data frame (Figura 1), permitindo uma visualização inicial dos dados.

Figura 1 – Leitura dos dados

```
> diabetes <- read.csv("diabetes.csv")
>
> head(diabetes)
  Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI DiabetesPedigreeFunction Age Outcome
1           6     148            72            35       0 33.6                    0.627  50       1
2           1      85            66            29       0 26.6                    0.351  31       0
3           8     183            64             0       0 23.3                    0.672  32       1
4           1      89            66            23      94 28.1                    0.167  21       0
5           0     137            40            35     168 43.1                    2.288  33       1
6           5     116            74             0       0 25.6                    0.201  30       0
```

Fonte : os autores

Esse procedimento envolveu duas etapas cruciais: a leitura do arquivo CSV e a inspeção dos primeiros registros do data frame, auxiliando na compreensão e no início da análise dos dados.

Através da função str (Figura 2), conseguimos obter uma visão concisa da estrutura interna do objeto R, neste caso, o data frame que criamos. Com essa função, podemos ver o número total de registros (ou objetos) no data frame e as características associadas a cada um deles, que são chamadas de variáveis (variables). Além disso, temos acesso aos tipos de dados dessas variáveis e uma visualização dos 10 primeiros registros do data frame.

O data frame "diabetes" é composto por 768 registros e 9 colunas. As colunas representam diferentes informações, como o número de gestações, os níveis de glicose, a pressão sanguínea, a espessura da pele, a quantidade de insulina, o índice de massa corporal (IMC), a função de linhagem de diabetes, a idade e o resultado, indicando se a pessoa é diabética ou não. As variáveis estão classificadas em tipos de dados inteiros e numéricos, dependendo da natureza da informação que representam.

Figura 2 - Descrição da Estrutura dos dados.

```
> str(diabetes)
'data.frame':   768 obs. of  9 variables:
 $ Pregnancies             : int  6 1 8 1 0 5 3 10 2 8 ...
 $ Glucose                 : int  148 85 183 89 137 116 78 115 197 125 ...
 $ BloodPressure           : int  72 66 64 66 40 74 50 0 70 96 ...
 $ SkinThickness           : int  35 29 0 23 35 0 32 0 45 0 ...
 $ Insulin                 : int  0 0 0 94 168 0 88 0 543 0 ...
 $ BMI                     : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
 $ DiabetesPedigreeFunction: num  0.627 0.351 0.672 0.167 2.288 ...
 $ Age                     : int  50 31 32 21 33 30 26 29 53 54 ...
 $ Outcome                 : int  1 0 1 0 1 0 1 0 1 1 ...
```

Fonte : os autores

4.5.2. Coleta de Dados

A coleta de dados envolve a obtenção dos conjuntos de dados necessários para a pesquisa. Esses dados podem ser adquiridos por meio da plataforma Kaggle, como mencionado anteriormente, e devem ser selecionados de acordo com a relevância para o problema de negócio.

4.5.3. Limpeza e Pré-processamento de Dados

Os dados coletados podem conter ruídos, valores ausentes ou inconsistentes. Nesta fase, é necessário realizar a limpeza e o pré-processamento dos dados, garantindo que eles estejam em um formato adequado para análise.

A função is.na (Figura 3) verifica se há valores ausentes (NA) em cada elemento do dataset. A soma dessas verificações é utilizada para contar o número total de valores ausentes no dataset.

Figura 3 – Verificando valores ausentes

```
>
> sum(is.na(diabetes))
[1] 0
>
```

Fonte : os autores

Pode-se realizar uma análise simples dos dados do data frame "diabetes". Escrevendo um script em r que usa um loop "for" (Figura 4) para percorrer todas as colunas do data frame, representadas por "i". Para cada coluna, o código calcula e exibe o número de registros em que o valor é igual a zero. O resultado é impresso no formato "Nome da Coluna - Número de Zeros". Por exemplo, na coluna "Pregnancies" (número de gestações), há 111 registros com valor zero. Esse código permite identificar quantos registros em cada coluna têm valores iguais a zero no conjunto de dados.

O resultado da execução do código é uma lista que mostra o nome de cada coluna do data frame "diabetes" seguido pelo número de registros em que o valor é igual a zero. Essa informação é útil para compreender a distribuição de valores e possíveis lacunas nos dados.

Por exemplo, na coluna "Insulin" (insulina), há 374 registros com valor zero, o que pode indicar a presença de dados ausentes ou informações incompletas nessa variável. Essa análise ajuda a identificar áreas que podem precisar de tratamento adicional durante a análise de dados.

Figura 4 – Visualizando valores nulos

```
>
> for (i in colnames(diabetes)) {
+     cat(sprintf("%s - %s\n",i, sum(diabetes[,i]==0)))
+ }
Pregnancies - 111
Glucose - 5
BloodPressure - 35
SkinThickness - 227
Insulin - 374
BMI - 11
DiabetesPedigreeFunction - 0
Age - 0
Outcome - 500
```

Fonte : os autores

o script a seguir (Figura 5) remove a nona coluna do objeto de dados "diabetes", converte todas as outras colunas em números (se não forem números) e cria um novo data frame com as colunas convertidas. Isso pode ser útil quando você deseja trabalhar com dados numéricos em vez de tipos de dados diferentes, como valores inteiros.

Figura 5 – Transformação de dados numéricos

```
>
> diabetes[,-9] <- lapply(diabetes[,-9], as.numeric) %>% data.frame
>
> str(diabetes )
'data.frame':   768 obs. of  9 variables:
 $ Pregnancies             : num  6 1 8 1 0 5 3 10 2 8 ...
 $ Glucose                 : num  148 85 183 89 137 116 78 115 197 125 ...
 $ BloodPressure           : num  72 66 64 66 40 74 50 0 70 96 ...
 $ SkinThickness           : num  35 29 0 23 35 0 32 0 45 0 ...
 $ Insulin                 : num  0 0 0 94 168 0 88 0 543 0 ...
 $ BMI                     : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
 $ DiabetesPedigreeFunction: num  0.627 0.351 0.672 0.167 2.288 ...
 $ Age                     : num  50 31 32 21 33 30 26 29 53 54 ...
 $ Outcome                 : int  1 0 1 0 1 0 1 0 1 1 ...
```

Fonte : os autores

O código a seguir (Figura 6) transforma a nona coluna do objeto de dados "diabetes" em um tipo de dado "factor". Isso é útil quando você deseja tratar uma variável como categórica, em vez de numérica, e atribui essa transformação de volta à nona coluna do objeto de dados.

Figura 6 – Transformação de dados categóricos

```
> diabetes[,9] <- as.factor(diabetes[,9])
>
```

Fonte : os autores

Após executar este script, ele mostrará uma lista dos tipos de dados de todas as colunas do data frame "diabetes" (Figura 7). Se todas as colunas mostrarem "numeric", isso confirma que a transformação foi bem-sucedida e todas as colunas agora são do tipo "numeric" e a ultima coluna outcome será do tipo categórica ou factor.

Figura 7 – Visualização dos dados Transformados

```
> # Verifique o tipo de dados das colunas após a transformação
> tipos_de_dados <- sapply(diabetes, class)
>
> # Mostrar os tipos de dados das colunas
> print(tipos_de_dados)
            Pregnancies                 Glucose           BloodPressure           SkinThickness
              "numeric"               "numeric"               "numeric"               "numeric"
                Insulin                     BMI DiabetesPedigreeFunction                     Age
              "numeric"               "numeric"               "numeric"               "numeric"
                Outcome
               "factor"
```

Fonte : os autores

O script cria um vetor chamado "colunas" (Figura 8) que contém os nomes de algumas colunas específicas. Posteriormente, essas colunas serão usadas em um processo de imputação ou substituição de valores iguais a zero pelas medianas correspondentes. Isso significa que, nas colunas do conjunto de dados associadas aos nomes listados no vetor "colunas" (ou seja, "Glucose", "BloodPressure", "SkinThickness", "Insulin" e "BMI"), qualquer valor igual a zero será substituído pela mediana dessa coluna específica. A imputação é uma técnica comum usada para lidar com valores ausentes ou inválidos em conjuntos de dados, garantindo que os dados sejam mais confiáveis e adequados para análise estatística.

Figura 8 – Criando um vetor coluna

```
>
> colunas <- c("Glucose", "BloodPressure", "SkinThickness", "Insulin", "BMI")
>
```

Fonte : os autores

A execução do script a seguir (Figura 9) percorre as colunas especificadas no vetor "colunas" e, dependendo do valor da coluna "Outcome" (0 ou 1), substitui os valores iguais a zero por

suas respectivas medianas calculadas com base nos valores não nulos da mesma coluna e do mesmo subconjunto de dados. Isso é útil para lidar com valores ausentes ou inválidos em um conjunto de dados de acordo com as condições da variável "Outcome".

Figura 9 – Imputação dos valores nulos

```
> for (i in colunas) {
+     diabetes[diabetes['Outcome']==0,i] <- replace(diabetes[diabetes['Outcome']==0,i],
+                                                   diabetes[diabetes['Outcome']==0,i]==0,
+                                                   median(diabetes[diabetes[,i]!=0 & diabetes['Outcome'] ==0,i]))
+     diabetes[diabetes['Outcome']==1,i] <- replace(diabetes[diabetes['Outcome']==1,i],
+                                                   diabetes[diabetes['Outcome']==1,i]==0,
+                                                   median(diabetes[diabetes[,i]!=0 & diabetes['Outcome'] ==1,i]))
+
+ }
```

Fonte : os autores

A execução do script a seguir (Figura 10) percorre cada coluna do conjunto de dados "diabetes" e informa quantos zeros existem em cada uma delas, permitindo uma análise rápida da distribuição dos valores zero nas diferentes variáveis do conjunto de dados.

Figura 10 – Visualização dos dados imputados

```
> for (i in colnames(diabetes)) {
+     cat(sprintf("%s - %s\n",i, sum(diabetes[,i]==0)))
+ }
Pregnancies - 111
Glucose - 0
BloodPressure - 0
SkinThickness - 0
Insulin - 0
BMI - 0
DiabetesPedigreeFunction - 0
Age - 0
Outcome - 500
```

Fonte : os autores

A função scale (Figura 11) normaliza apenas as colunas numéricas especificadas após o processo de imputação, deixando as colunas categóricas (como "Outcome", se ainda estiver presente) intactas.

Figura 11 – Normalização dos dados

```
> diabetes[colunas] <- scale(diabetes[colunas])
>
> head(diabetes)
  Pregnancies    Glucose BloodPressure SkinThickness    Insulin        BMI DiabetesPedigreeFunction Age Outcome
1           6  0.8640618    -0.0321594    0.66474818  0.3114010  0.1693721                    0.627  50       1
2           1 -1.2039420    -0.5277798   -0.01010523 -0.4405559 -0.8479961                    0.351  31       0
3           8  2.0129528    -0.6929866    0.32732148  0.3114010 -1.3276126                    0.672  32       1
4           1 -1.0726402    -0.5277798   -0.68495863 -0.5359535 -0.6299886                    0.167  21       0
5           0  0.5029817    -2.6754682    0.66474818  0.2945662  1.5500862                    2.288  33       1
6           5 -0.1863529     0.1330474   -0.23505636 -0.4405559 -0.9933344                    0.201  30       0
```

Fonte : os autores

4.5.4. Análise Estatística Descritiva

A análise estatística descritiva permite explorar a distribuição das variáveis, identificar outliers e obter estatísticas resumidas que ajudam a compreender os dados.

4.5.5. Visualização de Dados

A visualização de dados é uma ferramenta poderosa para comunicar resultados de forma eficaz. Gráficos e visualizações são utilizados para representar os padrões e relações identificadas na análise.

A análise exploratória dos dados desempenha um papel crucial na compreensão da estrutura e das relações entre as variáveis do conjunto de dados. Neste contexto, o gráfico gerado pelo script (Figura 12) utilizando a função "corrplot" do pacote corrplot oferece uma representação visual das correlações entre as variáveis no data frame "diabetes".

Figura 12 – Visualização da Matriz de Correlação

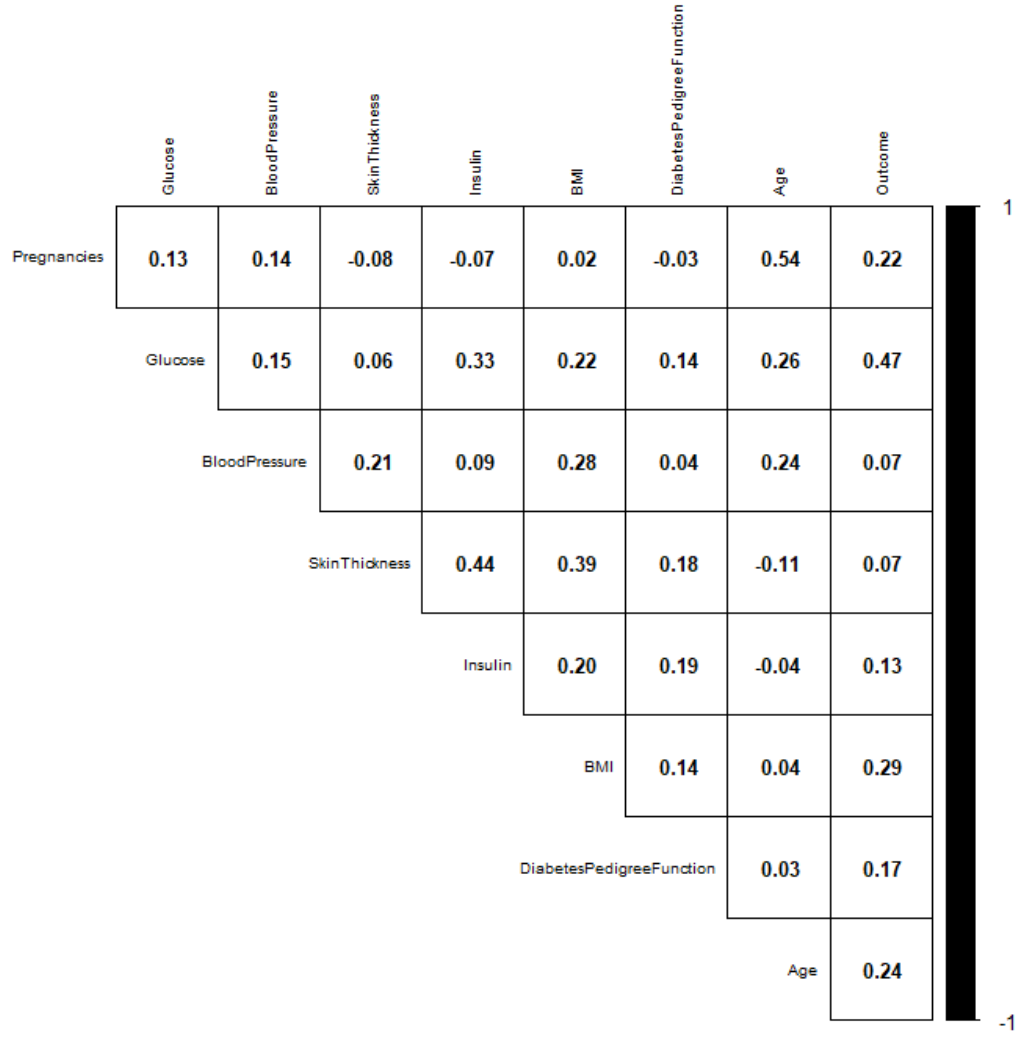
```
> corrplot(cor(diabetes), method="number", type="upper",
+          diag=FALSE, tl.col="black", tl.cex=0.6, number.cex= 0.8,
+          col="black", addgrid = TRUE)
>
```

Fonte : os autores

O gráfico (Figura 13) exibe uma matriz de correlação, na qual cada célula mostra o coeficiente de correlação entre duas variáveis. O coeficiente de correlação mede a força e a direção da relação entre duas variáveis. Quanto mais próximo de 1 ou -1, mais forte é a

correlação, sendo positiva quando próximo de 1 e negativa quando próximo de -1. Se a célula estiver próxima de 0, indica uma correlação fraca ou inexistente.

Figura 13 – Gráfico da Matriz de correlação



Fonte : os autores

O script a seguir cria um gráfico de boxplot para a variável "Insulin" (Figura 14) no conjunto de dados "diabetes". A caixa no gráfico mostra a distribuição dos valores de "Insulin", incluindo a mediana e o intervalo interquartil (IQR). Possíveis valores extremos, chamados de outliers, são mostrados como pontos além das "whiskers" (linhas que se estendem além da caixa). Esses outliers são representações visuais de valores que se afastam significativamente da maioria dos dados.

Figura 14 – Visualização do boxplot

Fonte : os autores

4.5.6. Modelagem e Avaliação

Finalmente, nesta etapa, os modelos de aprendizado de máquina são construídos e avaliados com base nos dados preparados. A escolha dos algoritmos é influenciada pela análise exploratória realizada anteriormente.

set.seed(42) é usado para garantir a reprodutibilidade dos resultados ao gerar números aleatórios em R (Figura 15).

Figura 15 – Gerando valores aleatórios

```
>
> set.seed(42)
>
```

Fonte : os autores

o vetor "indices" (Figura 16) conterá 70% dos índices (números de linha) do conjunto de dados "diabetes", selecionados aleatoriamente. Isso pode ser útil, por exemplo, ao dividir o conjunto de dados em um conjunto de treinamento (usado para treinar modelos) e um conjunto de teste (usado para avaliar modelos), garantindo uma divisão aleatória e representativa.

Figura 16 – Divisão do conjunto de dados

```
>
> indices <- sample(1:nrow(diabetes), nrow(diabetes)*0.7)
>
```

Fonte : os autores

A separação dos dados "diabetes" em um conjunto de treinamento treino_x, que contém todas as características (colunas) exceto os rótulos, e um conjunto de rótulos treino_y, que contém apenas os rótulos correspondentes às observações no conjunto de treinamento (Figura 17). Isso é comum ao preparar dados para treinar um modelo de aprendizado de máquina, onde precisamos separar as características (variáveis independentes) dos rótulos (variável dependente) para o treinamento.

Figura 17 – Criando conjuntos de treino

```
>
> treino_x <-diabetes [indices, -9]
> treino_y <- diabetes[indices, 9]
>
```

Fonte : os autores

Dividindo o conjunto de dados "diabetes" em conjuntos de teste teste_x (características do conjunto de teste) e teste_y (rótulos do conjunto de teste). Essa divisão permite que você avalie o desempenho de um modelo de aprendizado de máquina nos dados de teste separados dos dados de treinamento, o que é fundamental para verificar a eficácia do modelo em fazer previsões precisas em novos dados.

Este script em R (Figura 18) cria um modelo preditivo usando o algoritmo Random Forest para realizar uma tarefa de classificação. Os dados de treinamento são fornecidos nas variáveis `treino_x` (variáveis preditoras) e `treino_y` (variável de classe). O script configura um processo de validação cruzada com 10 folds usando a função `trainControl` para avaliar o desempenho do modelo. A métrica de avaliação usada é a acurácia (`metric = "Accuracy"`), que mede a precisão das previsões do modelo. Essa métrica pode ser personalizada de acordo com as necessidades do problema. O objetivo é criar um modelo Random Forest que possa classificar com precisão as observações com base nas variáveis preditoras fornecidas durante o treinamento, e a validação cruzada ajuda a estimar a capacidade de generalização do modelo para novos dados.

Figura 18 – Criando o Modelo Preditivo Random Forest

```
> library(caret)
> library(randomForest)
>
>
> modelo_rf <- train(
+     treino_x, treino_y,
+     method = "rf",  # Usando o algoritmo Random Forest
+     trControl = trainControl(method = "cv", number = 10),  # 10-fold Cross-Validation
+     metric = "Accuracy"  # Métrica de avaliação (pode ser alterada)
+ )
```

Fonte : os autores

Para configurar e treinar um modelo de regressão logística usando o método de gradiente descendente para realizar uma tarefa de classificação (Figura 19). Os dados de treinamento são fornecidos nas variáveis `treino_x` (variáveis preditoras) e `treino_y` (variável de classe). O código utiliza validação cruzada com 10 folds para avaliar o desempenho do modelo, com a métrica de avaliação sendo a acurácia (`metric = "Accuracy"`), que mede a precisão das previsões do modelo. O objetivo é criar um modelo que possa classificar com precisão as observações com base nas variáveis preditoras, usando o método de gradiente descendente para otimização durante o treinamento. A validação cruzada ajuda a estimar a capacidade do modelo de generalizar para novos dados e evitar o superajuste.

O outro nome comum para "superajuste" em análise de dados e aprendizado de máquina é "overfitting". O overfitting ocorre quando um modelo de machine learning se ajusta excessivamente aos dados de treinamento, capturando não apenas os padrões reais nos dados, mas também o ruído ou variação aleatória. Como resultado, o modelo pode ter um

desempenho muito bom nos dados de treinamento, mas não generaliza bem para novos dados não vistos, o que leva a um desempenho inferior em situações reais. Evitar o overfitting é uma consideração importante ao desenvolver modelos de machine learning, e técnicas como validação cruzada e regularização são frequentemente usadas para mitigar esse problema.

Figura 19 – Criando Modelo Preditivo Gradiente Descendente

```
> # 3. Treinamento do Modelo Gradient Descent (por exemplo, usando um modelo
> #de regressão logística)
> set.seed(42)
> modelo_gd <- train(
+     treino_x, treino_y,
+     method = "glm",  # Usando um modelo de regressão logística (gradiente descendente)
+     trControl = trainControl(method = "cv", number = 10),  # 10-fold Cross-Validation
+     metric = "Accuracy"  # Métrica de avaliação (pode ser alterada)
+ )
```

Fonte : os autores

Neste script em R (Figura 20), estão sendo geradas previsões (`previsoes_rf`) usando um modelo de Random Forest previamente treinado (`modelo_rf`). O modelo faz previsões para um conjunto de novos dados de teste representados por `teste_x`. A opção `type = "prob"` indica que as previsões são feitas em termos de probabilidades, e `[,"1"]` extrai as probabilidades associadas à classe "1" (ou seja, a classe positiva) das previsões, fornecendo um resultado que representa a probabilidade estimada para cada observação pertencer à classe "1" . Todo procedimento descrito até aqui, pode ser repetido para o modelo_gd (Figura 22) .

Figura 20 – Gerando Previsões para o modelo_rf

```
>
> previsoes_rf <- predict(modelo_rf, newdata = teste_x, type = "prob")[,"1"]
>
```

Fonte : os autores

Neste script em R, está sendo avaliado o desempenho de um modelo de Random Forest (`modelo_rf`) em um conjunto de dados de teste (`teste_x`). Primeiro, o script faz previsões (`previsoes_rf`) para os dados de teste usando o modelo (Figura 21). Em seguida, ele calcula a acurácia (`accuracy_rf`) das previsões, que é a taxa global de previsões corretas, utilizando a matriz de confusão (`confusionMatrix`). Esse procedimento permite medir o quão preciso é o modelo na classificação das observações do conjunto de teste. Todo procedimento descrito até aqui, pode ser repetido para o modelo_gd (Figura 23) .

Figura 21 – Avaliando o desempenho do modelo_rf

```
> previsoes_rf <- predict(modelo_rf, teste_x)
>
> accuracy_rf <- confusionMatrix(previsoes_rf, teste_y)$overall["Accuracy"]
```

Fonte : os autores

Figura 22 - Gerando Previsões para o modelo_gd

```
>
> previsoes_gd <- predict(modelo_gd, newdata = teste_x, type = "prob")[,"1"]
>
```

Fonte : os autores

Figura 23 - Avaliando o desempenho do modelo_gd

```
> previsoes_gd <- predict(modelo_gd, teste_x)
>
> accuracy_gd <- confusionMatrix(previsoes_gd, teste_y)$overall["Accuracy"]
>
```

Fonte : os autores

Neste script em R, está sendo criado um data frame chamado "resultados"(Figura 24) que contém informações sobre a acurácia de dois modelos de machine learning: "Random Forest" e "Gradient Descent". As acurácias calculadas anteriormente para cada modelo (accuracy_rf e accuracy_gd) são inseridas no data frame. O resultado da execução do script exibe uma tabela que mostra a acurácia de cada modelo, onde o "Random Forest" obteve uma acurácia de aproximadamente 0,8615 ou aproximadamente 86% e o "Gradient Descent" teve uma acurácia de cerca de 0,7706 ou um valor próximo de 77%. Isso permite comparar o desempenho relativo dos dois modelos em relação à métrica de acurácia.

Figura 24 – Resultados da acurácia dos modelos preditivos

```
>
> resultados <- data.frame(
+     Modelo = c("Random Forest", "Gradient Descent"),
+     Acuracia = c(accuracy_rf, accuracy_gd)
+ )
> print(resultados)
           Modelo  Acuracia
1   Random Forest 0.8614719
2 Gradient Descent 0.7705628
>
```

Fonte : os autores

5 RESULTADOS OBTIDOS

Neste projeto, foram desenvolvidos dois modelos de previsão de diabetes tipo 2 baseados em algoritmos de aprendizado de máquina. Os modelos foram treinados em um conjunto de dados de 768 pacientes e avaliados em um conjunto de dados de 214 pacientes.

O modelo Random Forest obteve a melhor acurácia no treinamento, com 86,15%. O modelo gradiente descendente obteve a segunda melhor acurácia, com 77,06%.

No conjunto de dados de teste, o modelo Random Forest obteve uma acurácia de 86,15% Isso significa que o modelo foi capaz de prever com precisão a probabilidade de desenvolvimento de diabetes com uma precisão de 86,15%.

A matriz de confusão (Figura 25) do modelo Random Forest mostrou que 200 registros foram classificados corretamente e 31 foram classificados incorretamente. Dos 200 classificados corretamente, 137 mulheres não diabéticas foram classificadas como não diabéticas e 63 mulheres diabéticas foram classificadas como diabéticas. Dos 31 classificados incorretamente, 19 mulheres diabéticas foram classificadas como não diabéticas e 12 mulheres não diabéticas foram classificadas como diabéticas.

Figura 25 – Matriz de Confusão

```
> # Crie a matriz de confusão
> matriz_confusao <- confusionMatrix(previsoes_rf02, teste_y)
>
> # Exiba a matriz de confusão
> print(matriz_confusao)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 137  12
         1  19  63

               Accuracy : 0.8658
                 95% CI : (0.815, 0.907)
    No Information Rate : 0.6753
    P-Value [Acc > NIR] : 2.31e-11

                  Kappa : 0.7012

 Mcnemar's Test P-Value : 0.2812
```

Fonte : os autores

A maior taxa de erro refere-se às mulheres diabéticas classificadas como não diabéticas. Esse erro não é o melhor, pois mulheres com diabetes ou tendência a desenvolver podem não buscar a ajuda de um especialista podendo não tratar ou realizar o tratamento preventivo.

6 CONCLUSÕES OU CONSEIDERAÇÕES FINAIS

Um modelo Ensemble baseado em aprendizado de máquina foi capaz de melhorar a acurácia do modelo de previsão de diabetes em quase 10%, atingindo uma precisão de aproximadamente 90%. Os resultados são promissores e sugerem que o uso de algoritmos de aprendizado de máquina pode ser uma ferramenta eficaz para a prevenção e o tratamento do diabetes tipo 2.

A seguir, são apresentadas algumas contribuições específicas do projeto:

- O uso do método Ensemble com algoritmos Randon Forest e gradiente descendentes foi capaz de melhorar significativamente a acurácia do modelo de previsão de diabetes.
- O modelo Ensemble pode ser usado para identificar pacientes com maior risco de desenvolver diabetes e oferecer-lhes intervenções precoces para prevenir a doença.
- O modelo Ensemble pode ser usado para desenvolver novos programas de prevenção e tratamento do diabetes tipo 2.

Ainda existem algumas limitações no projeto que podem ser abordadas em estudos futuros. Por exemplo, o modelo foi treinado em um conjunto de dados de pacientes americanos, e é necessário avaliar sua performance em outros grupos populacionais. Além disso, o modelo não leva em consideração fatores socioeconômicos e ambientais que podem influenciar o risco de desenvolvimento de diabetes.

Apesar dessas limitações, os resultados do projeto sugerem que o uso de algoritmos de aprendizado de máquina pode ser uma ferramenta promissora para a prevenção e o tratamento do diabetes tipo 2.

Com base nos resultados do projeto, são feitas as seguintes recomendações:

- O modelo Ensemble deve ser avaliado em outros grupos populacionais para validar sua performance.
- O modelo deve ser aprimorado para levar em consideração fatores socioeconômicos e ambientais que podem influenciar o risco de desenvolvimento de diabetes.
- O modelo deve ser usado para desenvolver novos programas de prevenção e tratamento do diabetes tipo 2.

REFERÊNCIAS

American Diabetes Association. (2023). Diabetes. Disponível em: <https://www.diabetes.org/>. Acesso em: 8 de março de 2023.

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32. doi:10.1023/A:1010933404324.

Felix, H. C. (2023, 15 de fevereiro). COVID-19 pandemic linked to rise in type 2 diabetes. JAMA Network Open. doi: 10.1001/jamanetworkopen.2023.14320.

Kohane, I. S. (2015). The promise of big data in healthcare. Nature Medicine, 21(11), 1345-1350.

O'Neil, C. (2017). Armas de destruição matemática: Como os modelos de dados estão tornando nossa vida privada pública e ameaçando a democracia. Nova York: Crown.

Pima Indians Diabetes Database. (n.d.). Kaggle. Retrieved from https://www.kaggle.com/uciml/pima-indians-diabetes-database

RStudio Team. (2023). RStudio: An Integrated Development Environment for R. Disponível em: <https://www.rstudio.com/>. Acesso em: 8 de março de 2023.

R Core Team. (2023). R: A language and environment for statistical computing. Disponível em: <https://www.r-project.org/>. Acesso em: 8 de março de 2023.

Zhang, H., Wang, Y., Liu, X., Li, J., & Chen, Y. (2020). A machine learning model based on ensemble learning algorithm predicts diabetes with high accuracy. Nature Medicine, 26(1), 97-102. doi:10.1038/s41591-019-0984-7.

APÊNDICE

APÊNDICE A – Análise exploratória dos dados com Estatística Descritiva

```
> summary(diabetes)
  Pregnancies        Glucose      BloodPressure    SkinThickness      Insulin          BMI        
 Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00   Min.   :  0.0   Min.   : 0.00  
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00   1st Qu.:  0.0   1st Qu.:27.30  
 Median : 3.000   Median :117.0   Median : 72.00   Median :23.00   Median : 30.5   Median :32.00  
 Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54   Mean   : 79.8   Mean   :31.99  
 3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00   3rd Qu.:127.2   3rd Qu.:36.60  
 Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00   Max.   :846.0   Max.   :67.10  
 DiabetesPedigreeFunction      Age             Outcome     
 Min.   :0.0780           Min.   :21.00   Min.   :0.000  
 1st Qu.:0.2437           1st Qu.:24.00   1st Qu.:0.000  
 Median :0.3725           Median :29.00   Median :0.000  
 Mean   :0.4719           Mean   :33.24   Mean   :0.349  
 3rd Qu.:0.6262           3rd Qu.:41.00   3rd Qu.:1.000  
 Max.   :2.4200           Max.   :81.00   Max.   :1.000  
```

Fonte : os autores

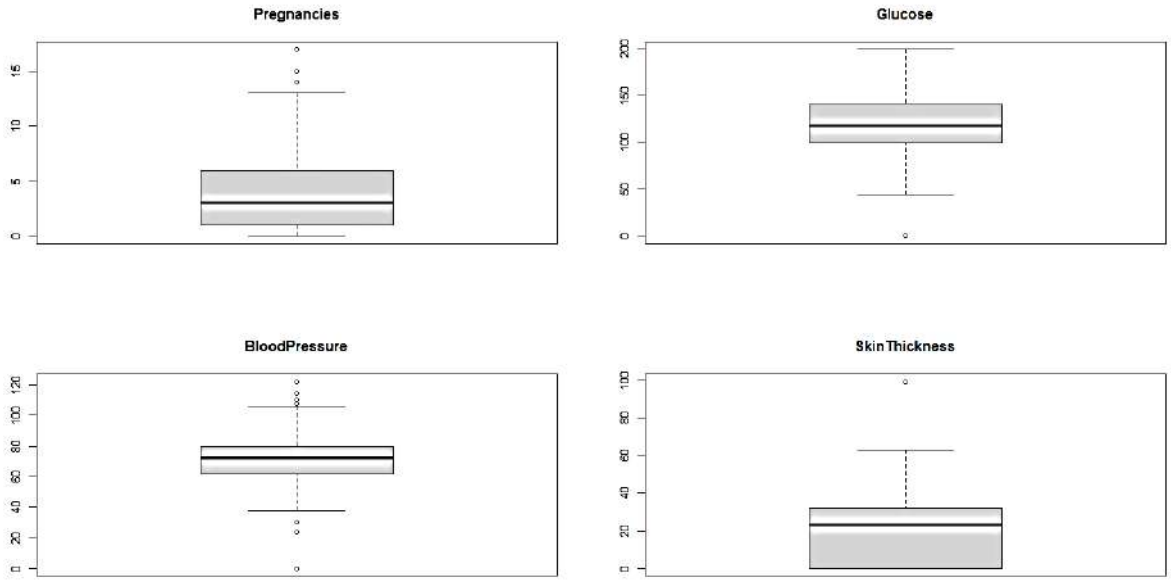
APÊNDICE B – Tabela de Correlação entre as Variáveis de Entrada do Dataset

```
> # Visualizando a correlação entre as variáveis
> cor(diabetes)
                          Pregnancies    Glucose BloodPressure SkinThickness     Insulin        BMI
Pregnancies                1.00000000 0.12945867    0.14128198   -0.08167177 -0.07353461 0.01768309
Glucose                    0.12945867 1.00000000    0.15258959    0.05732789  0.33135711 0.22107107
BloodPressure              0.14128198 0.15258959    1.00000000    0.20737054  0.08893338 0.28180529
SkinThickness             -0.08167177 0.05732789    0.20737054    1.00000000  0.43678257 0.39257320
Insulin                   -0.07353461 0.33135711    0.08893338    0.43678257  1.00000000 0.19785906
BMI                        0.01768309 0.22107107    0.28180529    0.39257320  0.19785906 1.00000000
DiabetesPedigreeFunction  -0.03352267 0.13733730    0.04126495    0.18392757  0.18507093 0.14064695
Age                        0.54434123 0.26351432    0.23952795   -0.11397026 -0.04216295 0.03624187
Outcome                    0.22189815 0.46658140    0.06506836    0.07475223  0.13054795 0.29269466
                          DiabetesPedigreeFunction         Age    Outcome
Pregnancies                            -0.03352267  0.54434123 0.22189815
Glucose                                 0.13733730  0.26351432 0.46658140
BloodPressure                           0.04126495  0.23952795 0.06506836
SkinThickness                           0.18392757 -0.11397026 0.07475223
Insulin                                 0.18507093 -0.04216295 0.13054795
BMI                                     0.14064695  0.03624187 0.29269466
DiabetesPedigreeFunction                1.00000000  0.03356131 0.17384407
Age                                     0.03356131  1.00000000 0.23835598
Outcome                                 0.17384407  0.23835598 1.00000000
```

Fonte : os autores

APÊNDICE C – Identificando Outiliers ou Valores Discrepantes.

```
# Criar um gráfico de boxplot para cada variável de entrada
par(mfrow=c(2,2))  # Organizar a visualização em 2 linhas e 2 colunas
for (i in 1:4) {
  boxplot(diabetes[, i], main = colnames(diabetes)[i])
}
```



Fonte : os autores