

”

E-fólio B | Folha de resolução para E-fólio



UNIDADE CURRICULAR: Raciocínio e Representação do Conhecimento

CÓDIGO: 21097

DOCENTE: Vitor Rocio

NOME: Inês Correia Gonçalves Marques

N.º DE ESTUDANTE: 2103589

CURSO: Licenciatura em Engenharia Informática

DATA DE ENTREGA: 20 de Maio de 2024

Índice

Introdução.....	3
1. Indicador escolhido e variáveis independentes	3
Indicador Escolhido	3
Variáveis Independentes.....	3
2. Tratamento dos Dados:	4
Discretização e Normalização dos Dados.....	4
3. Métodos	5
3.1 Árvore de decisão	5
3.2 K vizinhos mais próximos.....	5
3.3 Redes neurais	5
4. Reflexão sobre resultados	6
Conclusão.....	8
Bibliografia	9
Outros recursos	9
ANEXO – Código R e Tabelas de resultados	10

TRABALHO / RESOLUÇÃO:

Introdução

No E-Fólio B explorou-se a utilização de técnicas de aprendizagem automática para prever a participação de adultos na aprendizagem. Utilizaram-se três algoritmos: Árvores de Decisão, K Vizinhos Mais Próximos (KNN) e Redes Neurais, aplicados aos dados fornecidos pelo Pordata.

1. Indicador escolhido e variáveis independentes

O objetivo principal é prever a participação de adultos na aprendizagem com base em diferentes indicadores sociais e económicos. Os indicadores selecionados incluem acesso à internet, disparidade salarial e taxa de desemprego de longa duração.

Indicador Escolhido

Participação de adultos na aprendizagem (ODS Educação de Qualidade: Objetivo 4)

Este indicador é fundamental para compreender a adesão dos adultos a atividades educativas contínuas, essencial para a adaptação às mudanças no mercado de trabalho e para o desenvolvimento pessoal.

Variáveis Independentes

- Taxa de desemprego de longa duração: total e por sexo

O desemprego de longa duração pode motivar indivíduos a procurar educação adicional para melhorar as suas qualificações e reentrar no mercado de trabalho. No âmbito deste E-Fólio, considerou-se apenas a taxa total.

- Disparidade salarial entre homens e mulheres

As disparidades salariais podem influenciar a motivação para participar em programas de educação, especialmente se a educação for vista como uma via para alcançar a igualdade salarial.

- Assinaturas do acesso à Internet

O acesso à Internet é um facilitador na educação de adultos, especialmente para programas de aprendizagem online e cursos à distância.

2. Tratamento dos Dados:

Foram carregados os dados das quatro fontes, em csv:

- Assinaturas do Acesso à Internet
- Disparidade Salarial entre Homens e Mulheres
- Participação de Adultos na Aprendizagem
- Taxa de Desemprego de Longa Duração

Os dados foram transformados através da função *pivot_longer* para garantir que cada observação (país/ano) fosse uma linha e cada variável fosse uma coluna.

Os dados foram então unidos com base nas colunas comuns de país e ano.

Eliminaram-se os anos consecutivos, mantendo apenas anos pares para evitar dependências entre observações consecutivas.

Discretização e Normalização dos Dados

A variável *target* (participação de adultos na aprendizagem) foi discretizada para classificação binária, utilizando a mediana como ponto de corte para definir classes "low" e "high".

Dividiram-se os dados disponíveis num conjunto de treino (70% dos dados) e num conjunto de teste (30% restantes) para validar a eficácia dos modelos:

Data	
data	29 obs. of 6 variables
dataTest	8 obs. of 6 variables
dataTest_no...	8 obs. of 3 variables
dataTrain	21 obs. of 6 variables

As variáveis de *input* foram normalizadas para garantir que todas tivessem uma escala comparável.

Verificou-se também se havia valores NA ou NaN nos dados normalizados.

Para detalhes adicionais basta consultar o código em anexo. Desta forma, obtiveram-se as tabelas necessárias, organizadas e limpas.

3. Métodos

3.1 Árvore de decisão

Para as Árvores de Decisão, utilizou-se o algoritmo *Random Forest* com uma única árvore ($ntree = 1$) para alinhamento com o exemplo do professor. Os dados foram discretizados e categorizados conforme necessário. A variável alvo foi transformada em binária (0 e 1) para classificação. A taxa de erro OOB foi de 50%, o que indica a necessidade de melhorias.

```
rf_model <- randomForest(x = dataTrain[, 3:5], y =  
factor(dataTrain$adult_learning_participation, levels = c(0, 1)), ntree = 1, importance =  
TRUE)  
  
rf_predictions <- predict(rf_model, dataTest[, 3:5])  
  
rf_accuracy <- confusionMatrix(factor(rf_predictions, levels = c(0, 1)),  
factor(dataTest$adult_learning_participation, levels = c(0, 1)))$overall['Accuracy']
```

3.2 K vizinhos mais próximos

Para o KNN, utilizou-se a função `knn3` da biblioteca *caret*. Os dados foram normalizados para assegurar que todas as variáveis tivessem igual peso no cálculo das distâncias.

```
knn_model <- knn3(x = dataTrain_norm, y = factor(dataTrain$adult_learning_participation,  
levels = c(0, 1)), k = 5)  
  
knn_predictions <- predict(knn_model, dataTest_norm)  
  
knn_predictions_class <- ifelse(knn_predictions[, 2] > 0.5, 1, 0)  
  
knn_accuracy <- confusionMatrix(factor(knn_predictions_class, levels = c(0, 1)),  
factor(dataTest$adult_learning_participation, levels = c(0, 1)))$overall['Accuracy']
```

3.3 Redes neurais

Para redes neurais, utilizou-se a função *nnet*. O modelo foi treinado com um neurónio na camada oculta. A variável alvo foi transformada em binária e os dados foram normalizados.

```
nn_model <- nnet(x = dataTrain_norm, y = dataTrain$adult_learning_participation, size = 1,  
maxit = 500)
```

```
nn_predictions <- predict(nn_model, dataTest_norm, type = "raw")

nn_predictions_class <- ifelse(nn_predictions > 0.5, 1, 0)

nn_accuracy <- confusionMatrix(factor(nn_predictions_class, levels = c(0, 1)),
factor(dataTest$adult_learning_participation, levels = c(0, 1)))$overall['Accuracy']
```

4. Reflexão sobre resultados

Os resultados dos três modelos foram os seguintes:

```

      Model Accuracy
1  Decision Tree   0.375
2 K-Nearest Neighbors 0.500
3  Neural Network  0.375
> |
```

Método	Precisão
Árvore de Decisão (Random Forest com ntree=1)	0.375
K Vizinhos Mais Próximos (k=5)	0.500
Redes Neurais (neurónio com size=1)	0.375

Os resultados indicam que, embora todos os modelos tenham apresentado precisão abaixo do ideal, o algoritmo KNN apresentou a melhor performance, o que pode ser atribuído à natureza dos dados e à simplicidade dos modelos de árvore de decisão utilizado com apenas uma árvore, e de redes neurais apenas com um neurónio.

Em redes neurais, a vantagem de utilizar um único neurónio na camada oculta é ser rápido de treinar e interpretar, o que é útil para dados relativamente simples, como é o caso. No entanto, modelos tão simples podem não capturar padrões complexos nos dados, resultando em menor precisão comparado com redes neurais mais profundas com múltiplos neurónios e camadas.

Da mesma forma, em árvores de decisão, a utilização de apenas uma árvore simplifica a implementação e permite uma visualização direta da árvore de decisão, mas com desvantagens em termos de robustez e precisão.

A análise das variáveis mostrou que a taxa de desemprego e acesso à Internet influenciam diretamente a necessidade e a capacidade de aceder a oportunidades educacionais.

Para melhorar a performance, poderia fazer-se um tratamento diferente dos valores ausentes e considerar mais variáveis que possam influenciar a participação na aprendizagem. Também seria recomendável testar diferentes configurações e explorar outros algoritmos.

Assim sendo, alteraram-se as configurações e verificaram-se os novos resultados:

Método	Precisão
Árvore de Decisão (Random Forest com ntree=10)	0.375
K Vizinhos Mais Próximos (k=1)	1.000
Redes Neurais (neurónios com size=5)	0.750

Alterar k no algoritmo KNN não deve, em teoria, impactar diretamente a precisão das redes neurais. No entanto, variações na divisão dos dados de treino e teste, a sensibilidade dos dados e outras possíveis interdependências podem explicar diferenças nos resultados observados.

K = 1, size = 5

	Model	Accuracy
1	Decision Tree	0.375
2	K-Nearest Neighbors	1.000
3	Neural Network	0.375

K = 5, size = 5

	Model	Accuracy
1	Decision Tree	0.375
2	K-Nearest Neighbors	0.500
3	Neural Network	0.750

Consideraram-se os maiores valores de precisão de cada modelo nesta análise.

Independentemente de utilizar 1 árvore ou 100, o valor da precisão não se alterou no modelo da árvore de decisão, indicando que pode haver limitações nos dados ou na seleção de atributos.

No entanto, reduzir o k no algoritmo de K Vizinhos mais próximos aumentou a precisão para 1.000, mas isso pode indicar que ocorreu *overfitting*.

Aumentar o número de neurónios para 5 no algoritmo de redes neuronais também aumentou a precisão para 0.750. Essa melhoria sugere que uma arquitetura mais complexa pode ser benéfica.

Mais detalhes e análises sobre os resultados, e também sobre o “problema” encontrado para a árvore de decisão, em Anexo.

Conclusão

Os resultados deste estudo indicam que a previsão da participação de adultos na aprendizagem através de técnicas de aprendizagem automática ainda apresenta desafios significativos. A análise comparativa dos modelos de Árvores de Decisão, K Vizinhos Mais Próximos (KNN) e Redes Neuronais revelou que, embora todos os modelos tenham apresentado precisão abaixo do ideal, o KNN demonstrou a melhor performance entre os métodos analisados, especialmente com um valor de $k=1$.

No entanto, observou-se que a utilização de apenas uma árvore de decisão ou de uma rede neural com um único neurónio não foi suficiente para capturar a complexidade dos dados, resultando em menor precisão. Este estudo destaca a importância da escolha adequada dos parâmetros e da complexidade dos modelos de aprendizagem para obter resultados mais precisos.

Futuras melhorias podem incluir a inclusão de variáveis adicionais e a utilização de modelos mais complexos. A análise dos dados também sugere que fatores externos e políticas governamentais desempenham um papel significativo na participação em programas de aprendizagem, o que pode justificar a inclusão de variáveis contextuais em futuras pesquisas.

Este estudo reforça a necessidade de uma abordagem iterativa na modelagem preditiva, onde a experimentação e o ajuste contínuo dos modelos são essenciais para alcançar melhores resultados preditivos.

Bibliografia

Russell, S., Norvig, P., Artificial Intelligence: A Modern Approach, Prentice-Hall.

Outros recursos

PORDATA, 2024. PORDATA

Exercício disponibilizado pelo professor: 'Aprendizagem automática: usando bibliotecas em R', com exemplos de código e explicações sobre o funcionamento da linguagem R.

OpenAI, ChatGPT, 2024. Utilizado para suporte na identificação de variáveis para o indicador escolhido.

Caraça, G. (2023). Raciocínio e Representação do Conhecimento. Relatório de E-Fólio B. Curso de Licenciatura em Engenharia Informática, Universidade Aberta. Entregue a 29 de Maio de 2023.

Veríssimo da Cruz, R. M. G. (2023). Raciocínio e Representação do Conhecimento. Relatório de E-Fólio B. Curso de Licenciatura em Engenharia Informática, Universidade Aberta. Entregue a 29 de Maio de 2023.

ANEXO – Código R e Tabelas de resultados

Algumas das tabelas apresentam apenas parte dos resultados. Para consultar todos os dados, executar o programa em R.

Anos	country	internet_access	wage_gap	adult_learning_participation	unemployment_rate
1 2009	AT - Austria	NA	24.3	13.9	1.4
2 2009	BE - Bélgica	NA	10.1	7.1	3.2
3 2009	BG - Bulgária	998	13.3	1.6	3.4
4 2009	CH - Suíça	NA	18.4	23.9	NA
5 2009	CY - Chipre	NA	17.8	NA	0.6
6 2009	CZ - República Checa	NA	25.9	7.1	2.0

Tabela 1 - Dados Combinados

Anos	country	internet_access	wage_gap	unemployment_rate	adult_learning_participation
1 2010	AT - Austria	NA	24.0	1.5	13.8
2 2012	AT - Austria	NA	22.9	1.5	14.2
3 2014	AT - Austria	NA	22.2	1.9	14.3
4 2016	AT - Austria	NA	20.8	2.4	14.9
5 2018	AT - Austria	NA	20.4	1.7	15.1
6 2010	BE - Bélgica	NA	10.2	3.7	7.4
7 2012	BE - Bélgica	NA	8.3	3.1	6.9
8 2014	BE - Bélgica	NA	6.6	3.9	7.4
9 2016	BE - Bélgica	NA	6.0	3.7	7.0
10 2018	BE - Bélgica	NA	5.8	2.6	8.5
11 2010	BG - Bulgária	NA	13.0	NA	NA
12 2012	BG - Bulgária	NA	15.1	7.2	1.7
13 2014	BG - Bulgária	NA	14.2	7.4	2.1
14 2016	BG - Bulgária	NA	14.6	5.0	2.2
15 2018	BG - Bulgária	NA	13.9	3.6	2.5

Tabela 2 - Dados Tratados

Anos	country	internet_access	wage_gap	unemployment_rate	adult_learning_participation
22 2012	CY - Chipre	221	15.6	3.6	7.7
23 2014	CY - Chipre	244	14.2	7.7	7.1
24 2016	CY - Chipre	278	12.3	5.8	6.9
25 2018	CY - Chipre	313	10.4	2.7	6.7
42 2012	EE - Estónia	365	29.9	5.4	12.8
43 2014	EE - Estónia	386	28.1	3.3	11.6
44 2016	EE - Estónia	414	24.8	2.2	15.3
45 2018	EE - Estónia	441	21.8	1.3	19.3
68 2014	HR - Croácia	952	8.7	10.1	2.8
92 2012	LT - Lituânia	732	11.9	6.6	5.4
93 2014	LT - Lituânia	802	13.3	4.8	5.1
94 2016	LT - Lituânia	858	14.4	3.0	6.0
95 2018	LT - Lituânia	789	14.0	2.0	6.6
97 2012	LU - Luxemburgo	171	7.0	1.6	14.2
98 2014	LU - Luxemburgo	187	5.4	1.6	14.5
99 2016	LU - Luxemburgo	203	3.9	2.2	16.8
100 2018	LU - Luxemburgo	225	1.4	1.4	18.0
102 2012	LV - Letónia	472	14.9	8.6	7.2
103 2014	LV - Letónia	500	17.3	5.1	5.6
104 2016	LV - Letónia	512	19.7	4.4	7.3
105 2018	LV - Letónia	519	19.6	3.4	6.7
107 2012	MT - Malta	136	9.5	3.8	7.2
108 2014	MT - Malta	152	10.6	2.9	7.7
109 2016	MT - Malta	171	11.6	2.4	7.8
110 2018	MT - Malta	192	13.0	1.9	10.9
142 2012	SI - Eslovénia	531	4.5	4.3	13.8
143 2014	SI - Eslovénia	572	7.0	5.3	12.1
144 2016	SI - Eslovénia	627	8.1	4.3	11.6
145 2018	SI - Eslovénia	672	9.3	2.2	11.4

Tabela 3 - NAs Removidos

```

Type of random forest: classification
Number of trees: 1
No. of variables tried at each split: 1

OOB estimate of error rate: 50%
Confusion matrix:
  0 1 class.error
0 0 4          1
1 0 4          0

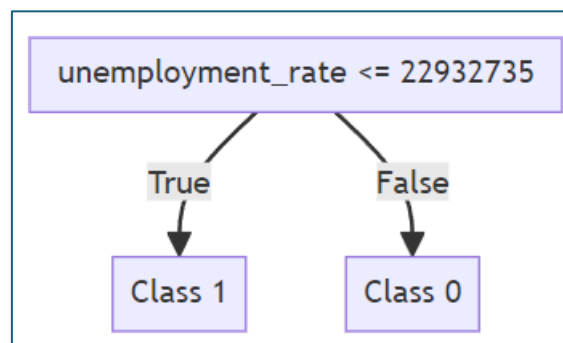
```

Tabela 4 - Random Forest OOB e Matriz de Confusão

	left daughter	right daughter	split var	split point	status	prediction
1	2	3	unemployment_rate	22932735	1	<NA>
2	0	0	<NA>	0	-1	1
3	0	0	<NA>	0	-1	0

Tabela 5 - Árvore de Decisão

O *split point* indica o ponto de divisão (threshold) da *split var*. O valor de 22932735 parece um erro, pois deveria ser um valor dentro do intervalo esperado para *unemployment_rate*. Este valor sugere que pode haver um problema no processo de treino ou um erro de escala. A coluna *status* mostra se o nó é terminal (-1) ou não (1). Os nós 2 e 3 são terminais (*status* = -1), enquanto o nó 1 não é terminal (*status* = 1). *Prediction* vai indicar a classe prevista para os nós terminais, ou seja, o nó 2 prevê a classe 1, e o nó 3 prevê a classe 0.



```

num [1:8, 1:2] 0 0 1 1 1 1 1 0 1 1 ...
- attr(*, "dimnames")=List of 2
..$ : NULL
..$ : chr [1:2] "0" "1"

```

Tabela 6 - Estrutura K Vizinhos Mais Próximos $k = 1$

```

num [1:8, 1:2] 0.6 0.4 0.6 0.6 0.4 0.4 0.2 0.4 0.4 0.6 ...
- attr(*, "dimnames")=List of 2
 ..$ : NULL
 ..$ : chr [1:2] "0" "1"

```

Tabela 7 - Estrutura K Vizinhos Mais Próximos $k = 5$

A previsão gerada pelo modelo K-Nearest Neighbors (KNN) usando o valor de $k=5$ é uma matriz numérica com 8 linhas e 2 colunas, ou seja, há 8 observações (ou exemplos) no conjunto de teste e 2 classes previstas (neste caso, "0" e "1"). Existem, por exemplo, 0.6, 0.4, 0.6, 0.6, etc., valores que representam as probabilidades de cada observação pertencer a cada uma das duas classes. Se uma linha da matriz é [0.6, 0.4], isso significa que o modelo KNN prevê que a probabilidade da observação pertencer à classe "0" é 60% e à classe "1" é 40%.

```

> nn_model <- nnet(x = dataTrain_norm, y = dataTrain$adult_learning_participation, size = 1, maxit = 500)
# weights: 6
initial value 5.538301
final value 5.238095
converged
> print(nn_model)
a 3-1-1 network with 6 weights
options were -
>
> # Mostrar os pesos da rede neural
> print(nn_model$wts)
[1] -0.69780351 0.11390875 -0.47895635 -0.19736726 -0.04351698 0.13876603

```

Tabela 6 - Redes Neurais size = 1

```

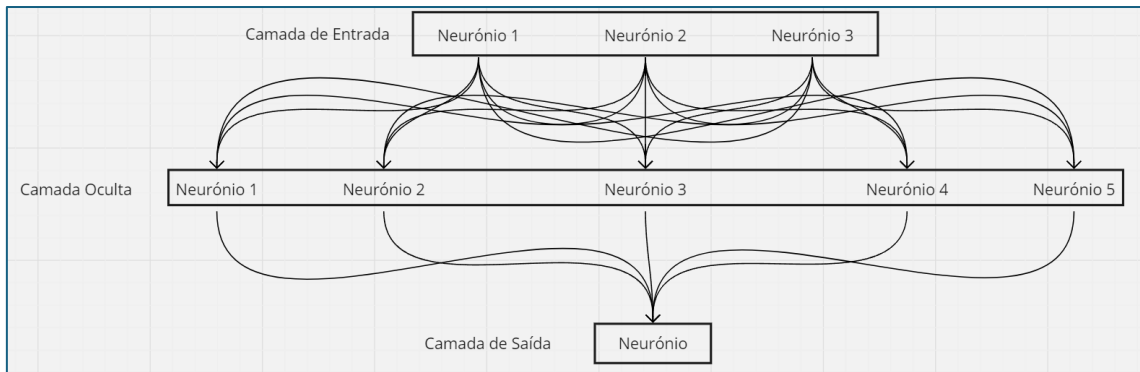
> nn_model <- nnet(x = dataTrain_norm, y = dataTrain$adult_learning_participation, size = 5, maxit = 500)
# weights: 26
initial value 5.266704
final value 5.238094
converged
> print(nn_model)
a 3-5-1 network with 26 weights
options were -
>
> # Mostrar os pesos da rede neural
> print(nn_model$wts)
[1] -0.69780459 0.11374929 -0.47896756 -0.19737055 0.20388463 0.38615271 0.08910558 -0.37281524 -0.57402728 -0.58014311
[11] -0.27269428 0.23439712 -0.69966554 -0.40800206 0.60624778 0.59590265 0.32773202 -0.23369922 0.02108866 0.34156450
[21] 0.21607025 0.22600575 -0.34673191 -0.39680576 -0.15547696 0.61943797

```

Tabela 7 - Redes Neurais size = 5

A rede neuronal foi executada com sucesso, atingindo o valor final de 5.238094, com uma redução do erro em relação ao valor inicial de 5.266704.

Contém 3 neurónios na camada de entrada, 5 na camada oculta e 1 na camada de saída.



O valor obtido para a precisão do treino foi de 0.750 (75.0%), o que significa que o modelo foi capaz de acertar mais de metade das previsões do conjunto de treino. Os resultados obtidos para `size = 5` são bastante mais aceitáveis do que para `size = 1`.

Código R (com `ntree=1` e `size=1`):

```
# Verificar e instalar pacotes necessários

if (!require('randomForest')) install.packages('randomForest', dependencies = TRUE)

if (!require('caret')) install.packages('caret', dependencies = TRUE)

if (!require('nnet')) install.packages('nnet', dependencies = TRUE)


# Carregar as bibliotecas necessárias

library(randomForest)

library(caret)

library(nnet)

library(readr)

library(dplyr)

library(tidyr)

library(class)


# Função para carregar e ajustar os tipos de dados

load_data <- function(file_path, value_name) {
```

```

data <- read_csv(file_path, show_col_types = FALSE)

data <- data %>%

  mutate(across(-Anos, as.character)) %>%

  pivot_longer(cols = -Anos, names_to = "country", values_to = value_name)

data[[value_name]] <- as.numeric(data[[value_name]])

return(data)
}

# Carregar os dados

internet_data <- load_data('C:/Users/nokas/Desktop/UAb 23-24/2º Semestre/Raciocínio e
Representação do Conhecimento/e-folio B/final/AssinaturasAcessoInternet.csv',
'internet_access')

wage_gap_data <- load_data('C:/Users/nokas/Desktop/UAb 23-24/2º Semestre/Raciocínio e
Representação do Conhecimento/e-folio B/final/DisparidadeSalarialHomensMulheres.csv',
'wage_gap')

learning_data <- load_data('C:/Users/nokas/Desktop/UAb 23-24/2º Semestre/Raciocínio e
Representação do Conhecimento/e-folio B/final/ParticipacaoAdultosAprendizagem.csv',
'adult_learning_participation')

unemployment_data <- load_data('C:/Users/nokas/Desktop/UAb 23-24/2º Semestre/Raciocínio e
Representação do Conhecimento/e-folio B/final/TaxaDesempregoLongaDuracao.csv',
'unemployment_rate')

# Verificar os dados carregados

print("Dados de Acesso à Internet")

print(head(internet_data))

print("Dados de Disparidade Salarial")

print(head(wage_gap_data))

print("Dados de Participação de Adultos na Aprendizagem")

print(head(learning_data))

print("Dados de Taxa de Desemprego de Longa Duração")

print(head(unemployment_data))

# Unir os dados por um identificador comum (Anos e country)

data <- merge(merge(merge(internet_data, wage_gap_data, by = c("Anos", "country")),

```

```

        learning_data, by = c("Anos", "country")),

        unemployment_data, by = c("Anos", "country"))

# Verificar o conjunto de dados combinado

print("Dados Combinados")

print(head(data))

# Selecionar as colunas relevantes e remover anos consecutivos

data <- data %>%

  select(Anos, country, internet_access, wage_gap, unemployment_rate,
adult_learning_participation) %>%

  arrange(country, Anos)

# Eliminar anos consecutivos (mantendo apenas anos pares para simplificação)

data <- data %>% filter(as.integer(Anos) %% 2 == 0)

# Tratar valores em falta

print("Dados Antes de Remover NAs")

print(data)

data <- na.omit(data)

print("Dados Depois de Remover NAs")

print(data)

# Verificar que países estão incluídos

print("Países Incluídos")

print(unique(data$country))

# Discretizar a variável target para classificação

data$adult_learning_participation <- cut(data$adult_learning_participation,

```

```

                                breaks = c(-Inf,
median(data$adult_learning_participation, na.rm = TRUE), Inf),

                                labels = c("low", "high"))

# Converter variáveis em fatores

data[, 3:5] <- lapply(data[, 3:5], factor)

# Subtrair 1 da variável de resposta para ter valores binários 0 e 1

data$adult_learning_participation <- as.numeric(data$adult_learning_participation) - 1

# Dividir os dados em conjuntos de treino e teste

set.seed(42)

trainIndex <- createDataPartition(data$adult_learning_participation, p = 0.7,

                                list = FALSE,

                                times = 1)

dataTrain <- data[ trainIndex,]

dataTest  <- data[-trainIndex,]

# Normalizar os dados

preProcValues <- preProcess(dataTrain[, -c(1, 2, 6)], method = c("center", "scale"))

dataTrain_norm <- predict(preProcValues, dataTrain[, -c(1, 2, 6)])

dataTest_norm <- predict(preProcValues, dataTest[, -c(1, 2, 6)])

# Verificar se há valores NA ou NaN nos dados normalizados

sum(is.na(dataTrain_norm))

sum(is.na(dataTest_norm))

# Garantir que todas as colunas de input para knn são numéricas

str(dataTrain_norm)

str(dataTest_norm)

```



```

# Modelos de aprendizagem

##### Árvores de Decisão #####

rf_model <- randomForest(x = dataTrain[, 3:5], y =
factor(dataTrain$adult_learning_participation, levels = c(0, 1)), ntree = 1, importance =
TRUE)

print(rf_model)

# Mostrar a árvore de decisão da Random Forest

tree_info <- getTree(rf_model, 1, labelVar = TRUE)

print(tree_info)

# Verificar o tipo de modelo Random Forest para validar que foi uma classificação e não
uma regressão

print(rf_model$type)

# Prever usando o modelo Random Forest

rf_predictions <- predict(rf_model, dataTest[, 3:5])

rf_accuracy <- confusionMatrix(factor(rf_predictions, levels = c(0, 1)),
                                factor(dataTest$adult_learning_participation, levels = c(0,
1)))$overall['Accuracy']

# Verificar o valor de OOB error rate

cat("OOB Error Rate:", rf_model$err.rate[nrow(rf_model$err.rate), "OOB"], "\n")

##### K Vizinhos Mais Próximos usando knn3 #####

knn_model <- knn3(x = dataTrain_norm, y = factor(dataTrain$adult_learning_participation,
levels = c(0, 1)), k = 5)

knn_predictions <- predict(knn_model, dataTest_norm)

# Verificar a estrutura da previsão

str(knn_predictions)

```

```

# Converter as probabilidades em classes binárias 0 e 1

knn_predictions_class <- ifelse(knn_predictions[, 2] > 0.5, 1, 0)

# Avaliar a precisão do modelo KNN

knn_accuracy <- confusionMatrix(factor(knn_predictions_class, levels = c(0, 1)),
                                factor(dataTest$adult_learning_participation, levels =
c(0, 1)))$overall['Accuracy']

##### Redes Neurais #####

nn_model <- nnet(x = dataTrain_norm, y = dataTrain$adult_learning_participation, size = 1,
maxit = 500)

print(nn_model)

# Mostrar os pesos da rede neural

print(nn_model$wts)

# Prever usando o modelo de Redes Neurais

nn_predictions <- predict(nn_model, dataTest_norm, type = "raw")

nn_predictions_class <- ifelse(nn_predictions > 0.5, 1, 0)

# Avaliar a precisão do modelo de Redes Neurais

nn_accuracy <- confusionMatrix(factor(nn_predictions_class, levels = c(0, 1)),
                                factor(dataTest$adult_learning_participation, levels = c(0,
1)))$overall['Accuracy']

# Resultados

results <- data.frame(

  Model = c("Decision Tree", "K-Nearest Neighbors", "Neural Network"),

  Accuracy = c(rf_accuracy, knn_accuracy, nn_accuracy)

)

```

```
# Mostrar resultados
```

```
print(results)
```