

**PREDIÇÃO DE DIAGNÓSTICO DE DIABETES COM INTELIGÊNCIA ARTIFICIAL  
APLICANDO MODELOS DE MACHINE LEARNING**

Eduardo Durieux Lopes  
Renato Cardoso Zimmer

## SUMÁRIO

[Resumo](#)

[1. Descrição de dados](#)

[1.1. Origem e escopo](#)

[1.2. Dicionário Completo](#)

[1.3. Descrição da base](#)

[1.4. Valores faltantes](#)

[1.5. Variável Target](#)

[2. Métricas](#)

[3. Análise das variáveis isoladamente](#)

[3.1. Pregnancies](#)

[3.2. Glucose](#)

[3.3. BloodPressure](#)

[3.4. SkinThickness](#)

[3.5. Insulin](#)

[3.6. BMI \(Índice de massa corporal\)](#)

[3.7. DiabetesPedigreeFunction](#)

[3.8. Age](#)

[4. Matriz de correlação dos campos](#)

[4.1. Glucose x Insulim](#)

[4.2. SkinThickness x BMI](#)

[5. Modelos](#)

[5.1. Regressão linear](#)

[5.2. Regressão Logística](#)

[5.3. Árvore de decisão](#)

[5.4. Random forest](#)

[5.5. SVM](#)

[5.6. KNN](#)

[5.7. Redes Neurais](#)

[6. Conclusão](#)

[7. Referência bibliográfica](#)

## Resumo

Este trabalho descreve o desenvolvimento de uma solução de Inteligência Artificial para auxiliar no diagnóstico médico, especificamente na predição de diabetes feminina. O estudo utilizou um conjunto de dados clínicos estruturados, disponível publicamente em ["https://www.kaggle.com/datasets/pentakrishnakishore/diabetes-csv"](https://www.kaggle.com/datasets/pentakrishnakishore/diabetes-csv).

A metodologia incluiu uma análise exploratória dos dados, abrangendo a identificação e tratamento de valores ausentes, a avaliação de correlações entre variáveis e a definição precisa da variável alvo.

Foram implementados e comparados diversos modelos de Machine Learning para classificação binária, tais como Regressão Logística, Árvore de Decisão, Random Forest, SVM, KNN e Redes Neurais. A escolha e avaliação dos modelos focaram em métricas cruciais para o contexto médico e para bases desbalanceadas, como *Recall*, *Precision*, *F1-Score* e ROC-AUC, ponderando entre o alto custo de falsos positivos e a necessidade de assertividade nos verdadeiros positivos.

Os resultados obtidos nos mostraram que o desempenho da solução não depende apenas da escolha do modelo, mas também das decisões no pré-processamento e nos ajustes para confluência com as regras de negócio.

## 1. Descrição de dados

### 1.1. Origem e escopo

Fonte Primária	<a href="https://www.kaggle.com/datasets/pentakrishnakishore/diabetes-csv">https://www.kaggle.com/datasets/pentakrishnakishore/diabetes-csv</a>
Nome	Pima Indians Diabetes Dataset
Data da Extração	15/12/2025
Período	<i>Não identificado</i>
Formato	csv
Quantidade de registros	769
Quantidade de variáveis	9

## 1.2. Dicionário Completo

	Coluna	Tipo	Descrição
1	Pregnancies	int64	Número de gestações
2	Glucose	int64	Concentração de glicose no plasma
3	Blood Pressure	int64	Pressão sanguínea
4	Skin Thickness	int64	Espessura da dobra cutânea no tríceps
5	Insulin	int64	Nível de insulina no sangue
6	BMI (Body Mass Index)	float64	Índice de gordura corporal baseado na altura x peso
7	Diabetes Pedigree Function	float64	Função da taxa de probabilidade de diabetes conforme histórico familiar
8	Age	int64	Idade da Mulher
9	Outcome	int64	Indicativo da diabetes ou não 0 - Sem ocorrência 1 - com ocorrência

Estes 9 campos compõem o dataset bruto.

### 1.3. Descrição da base

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DPF	Age
25%	1	99	62	0	0	27.3	0.2437	24
50%	3	117	72	23	30.5	32	0.3725	29
75%	6	140.25	80	32	127.25	36.6	0.6262	41
min	0	0	0	0	0	0	0.078	21
max	17	199	122	99	846	67.1	2.42	81
média	3.8450	120.8945	69.1054	20.5364	79.7994	31.9925	0.4718	33.2408
desvio	3.3695	31.9726	19.3558	15.9522	115.2440	7.8841	0.3313	11.7602
nulos	0	5	35	227	0	5	35	227

## 1.4. Valores faltantes

A base continha alguns campos com valor zero onde não condiz com a realidade, indicando não coleta dos dados

Campo	Qtde nulos
Glucose	5
BloodPressure	35
SkinThickness	227
BMI	5
DiabetesPedigreeFunction	35
Age	227

Iremos alterar estes valores para Nan e posteriormente alterar novamente para mediana ou média conforme formos aplicar os modelos

## 1.5. Variável Target

A variável target definida é a coluna **OUTCOME**. Aqui temos um ponto de atenção:

- Outcome=0: 500 (65,1%) (**Não presença de diabetes**)
- Outcome=1: 268 (34,9%) (**Presença de diabetes**)

Observamos que a presença de valores negativos para presença de diabetes é o dobro da presença de valores positivos, sendo assim podemos dizer que o dataset está desbalanceado. Neste caso, a acurácia pode enganar um pouco. Usaremos **ROC-AUC**, **recall**, **F1**, **precision**.

## 2. Métricas

### **Accuracy (Acurácia)**

*De maneira bem simples este número indica o percentual das previsões acertadas, tanto para positivo quanto para negativo. Por exemplo, para uma acurácia de 0.7, entende-se que 70% das previsões estão corretas, sem fazer distinção se são as previsões negativas ou positivas.*

### **Precision (Precisão)**

*Esta métrica indica o percentual de positivos que são verdadeiros. Aqui temos uma separação da métrica anterior. Usando exemplo do caso acima, para uma precisão de 0.7, significa que 70% das previsões que foram identificadas como positivas estão corretas. Não faz alusão às previsões negativas.*

### **Recall (Sensibilidade)**

*Esta métrica indica o percentual de positivos reais que estamos conseguindo encontrar. Continuando com a utilizar o mesmo número dos exemplos acima, um Recall de 0.7, indica que houve uma perda de 30% de resultados positivos que não conseguiram ser detectados pelo modelo aplicados..*

### **F1-Score (Pontuação F1)**

*F1-Score mede a harmonia entre Precision e Recall. Ele é prejudicado quando uma das duas métricas é muito pior que a outra. Ele busca evitar falsos positivos e falsos negativos*

### **ROC-AUC - Area Under the Curve**

*Métrica em Machine Learning para avaliar modelos de classificação binária, mostrando a capacidade de distinguir positivo e negativo, plotando a Taxa de Verdadeiros Positivos (TPR) vs. a Taxa de Falsos Positivos (FPR) em diferentes limiares, com AUC 1.0 sendo perfeito e 0.5 aleatório.*

**Curva ROC (Receiver Operating Characteristic):** Gráfico que mostra a performance de um classificador binário variando seu limiar de decisão.

**Eixo Y (vertical)** Taxa de Verdadeiros Positivos (TPR) ou Sensibilidade;

**Eixo X (horizontal)** Taxa de Falsos Positivos (FPR) ou (1 - Especificidade).

**AUC (Area Under the Curve):** A área total sob a curva ROC, fornece uma medida única do desempenho geral do modelo.

*Interpretação:*

*AUC = 1.0: Modelo perfeito (distingue classes perfeitamente).*

*AUC = 0.5: Desempenho aleatório (como jogar uma moeda).*

*AUC < 0.5: Pior que aleatório (modelo previsivelmente ruim).*

**Quanto maior a AUC (mais perto de 1), melhor o modelo.**

### **Coeficiente de assimetria**

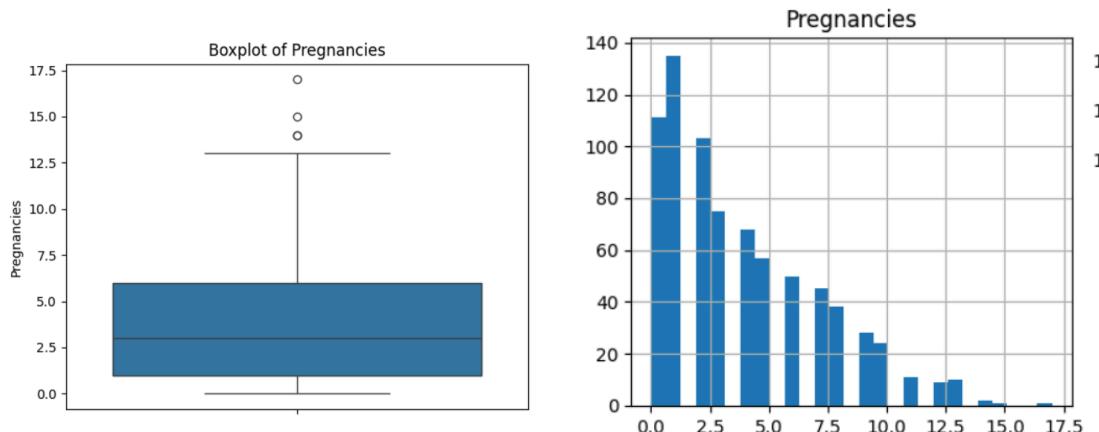
Mede o grau de distorção da simetria em uma distribuição de dados, indicando se as "caudas" da distribuição são mais longas para a direita (positiva) ou para a esquerda (negativa) em relação à média

Segundo Coeficiente de Pearson:  $A = (3 * \text{Média} - \text{Mediana}) / \text{Desvio Padrão}$

### 3. Análise das variáveis isoladamente

#### 3.1. Pregnancies

Este campo permite valores zero



- A variável Pregnancies representa o número de gestações por indivíduo.  
Trata-se de variável discreta de contagem, limitada inferiormente por zero.  
Associada a um tipo específico de diabetes, a gestacional.

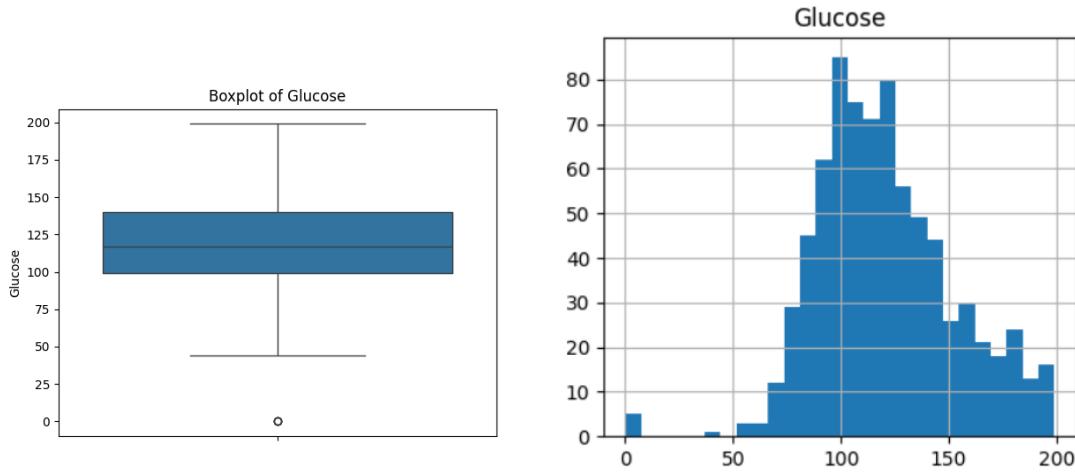
##### Boxplot

- Mediana: ~3 gestações, Q1: ~1 gestação, Q3: ~6 gestações, IQR: ~5 → variabilidade moderada
- Presença de outliers superiores, acima de 13 gestações, atingindo valores máximos próximos de 17.
- Não há outliers inferiores pois a variável é limitada em zero.

##### Histogram

- Distribuição assimétrica à direita
- Alta concentração de valores baixos, especialmente entre 0 e 3 gestações.
- Frequência decrescente à medida que o número de gestações aumenta.
- Cauda longa à direita, com valores chegando a aproximadamente 17 gestações.
- Padrão típico de variáveis de contagem onde quais eventos elevados são raros..
- Coeficiente de assimetria = 0.7524

### 3.2. Glucose



- Esta variável representa a concentração de glicose no sangue (mg/dL), sendo um indicador clínico relevante. O organismo com diabetes não consegue quebrar a glicose, fazendo com que este índice suba

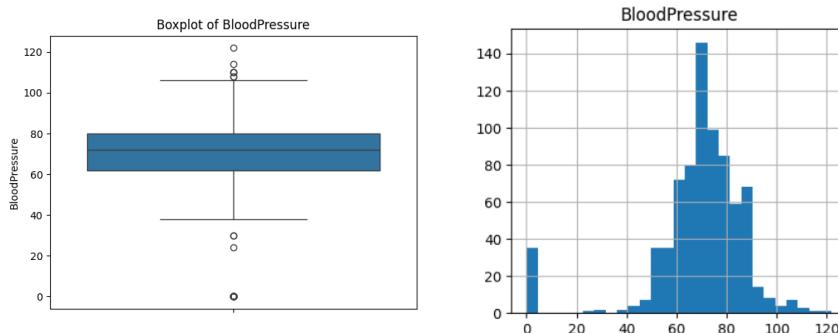
#### Boxplot

- Mediana: Aproximadamente entre 115 e 120 mg/dL, indicando que metade da amostra apresenta valores acima desse intervalo.
- Primeiro quartil (Q1): Em torno de 100 mg/dL.
- Terceiro quartil (Q3): Aproximadamente 140 mg/dL.
- Intervalo interquartil (IQR): Cerca de 40 mg/dL.
- Valores mínimos e máximos sem atipicidade.
- Existe um valor próximo a zero, o que é inviável fisiologicamente. Provável erro ou não existência de coleta.

#### Histogram

- A distribuição assimétrica positiva à direita
- Maior concentração de valores entre 90 e 140 mg/dL.
- Há uma extensão gradual até valores elevados (180–200 mg/dL), sugerindo presença de indivíduos com hiperglicemia.
- Coeficiente de assimetria = 0.3654

### 3.3. BloodPressure



- Pressão arterial diastólica (mmHg) é muito utilizada como indicador de risco cardiovascular.
- Pessoas com diabetes mais chances de desenvolver pressão alta e a pressão alta pode agravar ainda mais a diabetes

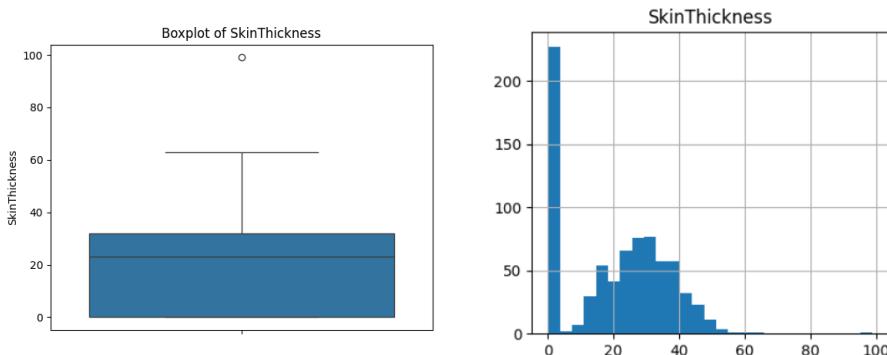
#### Boxplot

- Mediana: ~72 mmHg; Q1: ~62 mmHg; Q3: ~80 mmHg; IQR: ~18 mmHg → baixa a moderada variabilidade
- Outliers inferiores: valores próximos de 0 mmHg e entre 20–30 mmHg
- Outliers superiores: valores acima de 105 mmHg, chegando a ~122 mmHg
- Presença de zeros indicando também, não coleta ou erro.

#### Histogram

- A maior concentração de observações está entre 60 e 85 mmHg.
- A distribuição apresenta cauda mais longa à esquerda, causada pelos valores próximos de zero e leve alongamento à direita devido aos valores elevados (>100 mmHg).
- A presença de múltiplos outliers e assimetria indica que a variável não segue perfeitamente uma distribuição normal.
- Coeficiente de assimetria = -0.4486 (principal causa são os valores zerados - que neste caso são valores artificiais)

### 3.4. SkinThickness



- Espessura da prega cutânea do tríceps (mm) - indicativo de adiposidade corporal
- A dobra cutânea do tríceps (DCT) está indiretamente relacionada com diabetes. É indicador da gordura corporal e quantidades elevadas de gordura estão associadas a maior risco de resistência à insulina e diabetes tipo 2.

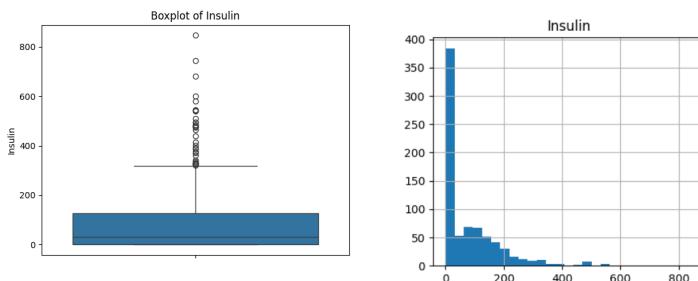
#### Boxplot

- Mediana: ~23 mm; Q1: ~0 mm; Q3: ~32 mm; IQR: elevado, refletindo mistura de zeros com valores reais
- Outliers superiores: valores próximos de 100 mm
- Sem outliers inferiores reais, pois os zeros dominam a base da distribuição

#### Histogram

- Alta concentração de valores em zero
- Pico muito elevado em 0 mm representando uma fração significativa da amostra
- Distribuição principal entre ~10 e 50 mm
- Pico central em torno de 20–30 mm
- Cauda longa à direita, alcançando valores próximos de 100 mm
- Coeficiente de assimetria = -0.4633 (principal causa são os valores zerados - que neste caso são valores artificiais e a longa cauda a direita)

### 3.5. Insulin



- Concentração sérica de insulina ( $\mu\text{U}/\text{mL}$ ) é utilizada em estudos de metabolismo glicídico, resistência à insulina e diabetes mellitus.
- Relação muito importante pois diabetes ocorre quando há produção insuficiente de insulina levando a dificuldade de processar a glicose.

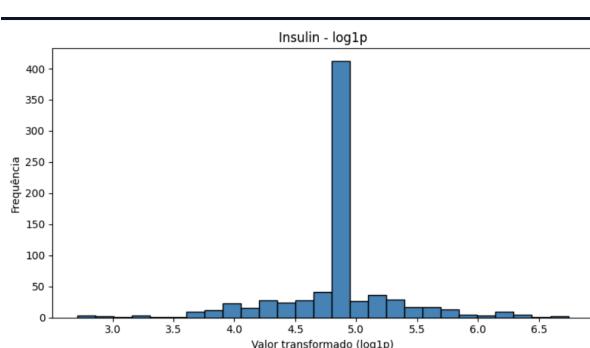
#### Boxplot

- Mediana: baixa, próxima da região inferior da escala ( $\approx 30 \mu\text{U}/\text{mL}$ )
- Q1: muito próximo de zero.
- Q3: aproximadamente entre 120 e 130  $\mu\text{U}/\text{mL}$ .
- IQR elevado, refletindo alta variabilidade.
- Número elevado de outliers superiores, acima de **300  $\mu\text{U}/\text{mL}$** , com máximas próximas de **850  $\mu\text{U}/\text{mL}$** .
- Esses outliers indicam possivelmente indivíduos com resistência à insulina.

#### Histogram

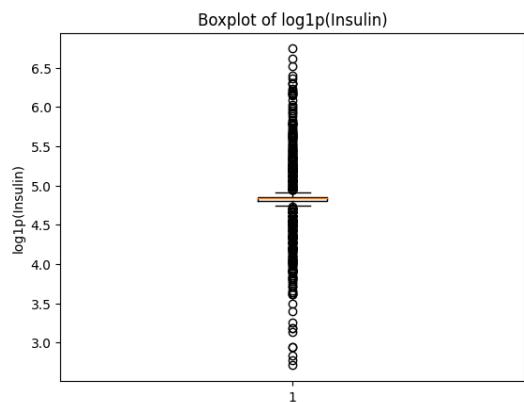
- Padrão fortemente assimétrico à direita
- Grande concentração de valores próximos de zero, formando um pico muito elevado na região inicial.
- A maioria dos valores observados encontra-se abaixo de 150  $\mu\text{U}/\text{mL}$ .
- Presença de cauda longa à direita, estendendo-se até valores extremamente elevados (acima de 800  $\mu\text{U}/\text{mL}$ ).
- Valores zero ou próximos de zero para insulina não são fisiologicamente possíveis - indicando falta de coleto ou erro.
- Coeficiente de assimetria = 1.2834 (principal causa são os valores zerados e a longa cauda a direita)

#### Ações Pré-processamento

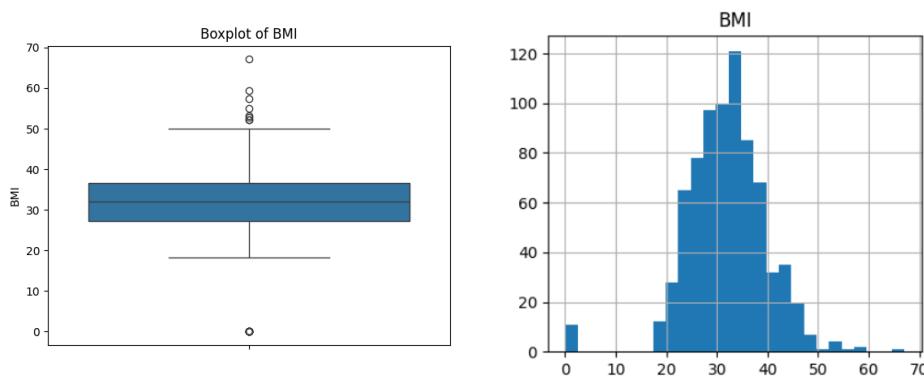


Aplicando  $\log_{10}$  a esta variável e plotando novamente os gráficos vemos que a assimetria foi reduzida. Aquela cauda a direita não está mais presente. Existe um pico um pouco antes de 5.0 no histograma, este pico está presente pois havia muitos zeros que foram substituídos pela mediana - isso acabou gerando um acúmulo de valores iguais que está agora representado por este pico no gráfico.

Usaremos esta variável com esta alteração na regressão logística, knn e svm. Nos outros modelos manteremos sem esta modificação.



### 3.6. BMI (Índice de massa corporal)



- O Índice de Massa Corporal (BMI) é uma medida utilizada para classificar o estado nutricional de indivíduos, sendo calculado pela razão entre o peso (kg) e o quadrado da altura (m<sup>2</sup>). É amplamente empregado em estudos associados a obesidade, risco cardiometabólico e diabetes mellitus.  
BMI elevado é um dos principais fatores de risco para diabetes.

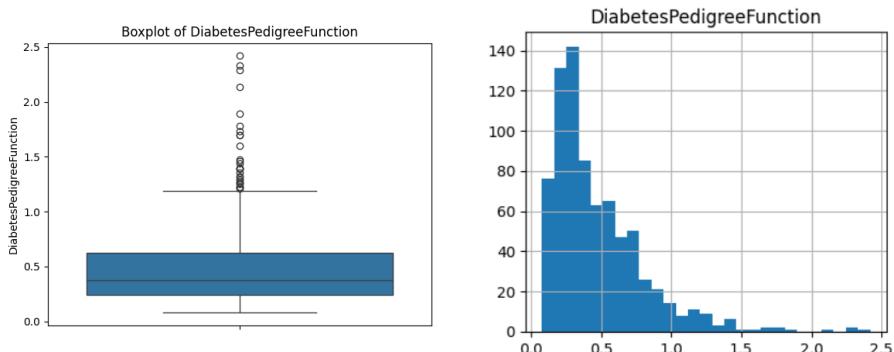
#### Boxplot

- Mediana: ~31 kg/m<sup>2</sup>, Q1: ~27 kg/m<sup>2</sup>, Q3: ~36 kg/m<sup>2</sup> e IQR: ~9 kg/m<sup>2</sup> → variabilidade moderada
- Outlier inferior: valor próximo de 0 kg/m<sup>2</sup>, não plausível clinicamente.
- Outliers superiores: valores acima de 50 kg/m<sup>2</sup>, alcançando aproximadamente 67 kg/m<sup>2</sup>, indicativos de obesidade mórbida.

#### Histogram

- Formato aproximadamente unimodal, com concentração central entre 25 e 35 kg/m<sup>2</sup>.
- Pico principal em torno de 30–32 kg/m<sup>2</sup>, sugerindo predominância de indivíduos com sobrepeso ou obesidade grau I.
- Cauda à direita, estendendo-se até valores elevados (acima de 60 kg/m<sup>2</sup>), indicando presença de indivíduos com obesidade severa.
- Um pequeno pico próximo de 0 kg/m<sup>2</sup>, que não é fisiologicamente plausível.
- Coeficiente de assimetria = -0.0028 - pode-se dizer que esta variável é simétrica

### 3.7. DiabetesPedigreeFunction



- Esta variável indica a predisposição genética a desenvolver diabetes - Histórico familiar
- Histórico familiar aumenta significativamente o risco de desenvolver a doença

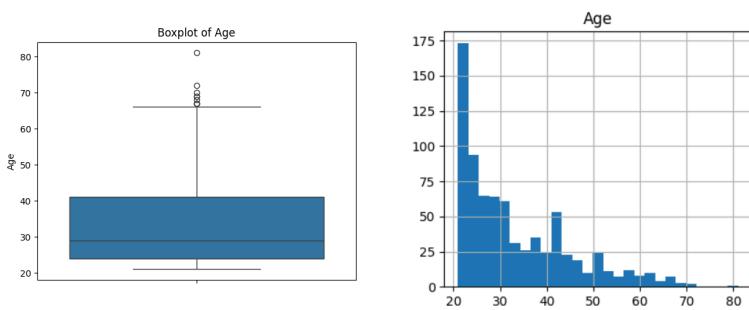
#### Boxplot

- Mediana: ~0,35–0,40, Q1: ~0,25, Q3: ~0,60, IQR: ~0,35 → variabilidade moderada
- Grande quantidade de outliers superiores, acima de 1,2, chegando a valores próximos de 2,5.
- Não há outliers inferiores relevantes, pois a variável é limitada inferiormente por valores próximos de zero.

#### Histogram

- Distribuição fortemente assimétrica à direita
- Alta concentração de observações em valores baixos, principalmente entre 0,2 e 0,6.
- Frequência decrescente à medida que os valores aumentam.
- Cauda longa à direita, estendendo-se até aproximadamente 2,5.
- Essa configuração indica que a maior parte da população apresenta baixo risco hereditário, enquanto poucos indivíduos concentram valores elevados de predisposição genética.
- Coeficiente de assimetria = 0.8998 - Grande cauda à direita

### 3.8. Age



- Idade dos pacientes em anos
- Existe um maior risco de diabetes tipo 2 em pessoas acima de 40 anos de idade

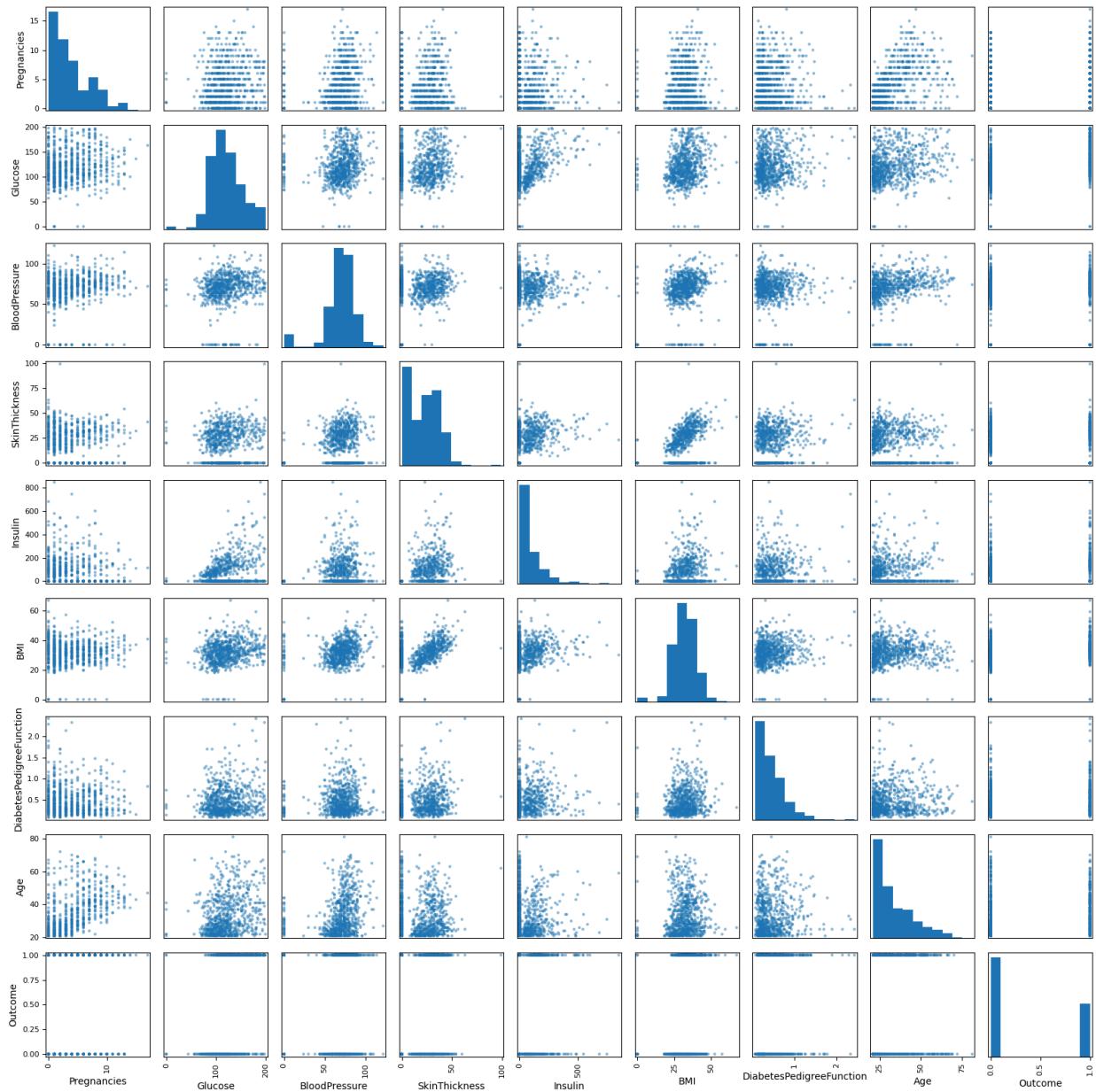
#### Boxplot

- Mediana: ~29 anos, Q1: ~24 anos, Q3: ~41 anos, IQR: ~17 anos → variabilidade moderada
- Observam-se outliers superiores a partir de aproximadamente 66 anos, chegando a valores acima de 80 anos.
- Não há outliers inferiores relevantes, pois a idade mínima inicia próximo a 20 anos

#### Histogram

- Distribuição assimétrica à direita
- Alta concentração de indivíduos entre 21 e 35 anos, formando o núcleo da distribuição.
- Frequência progressivamente menor à medida que a idade aumenta.
- Cauda longa à direita, com observações chegando a aproximadamente 81 anos.
- Coleta representação de indivíduos jovens com menor presença de idosos.
- Coeficiente de assimetria = 1.0818 - Grande cauda à direita

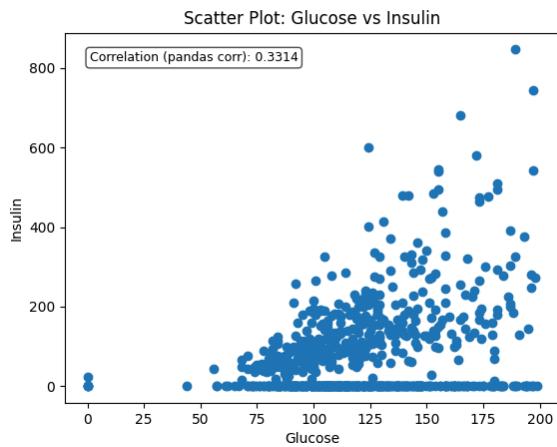
#### 4. Matriz de correlação dos campos



Nesta matriz identificamos 2 gráficos de correlação que podem ser analisados com mais cuidado.

- 2.●. Glucose x Insulim
- 2.●. SkinThickness x BMI

#### 4.1. Glucose x Insulin

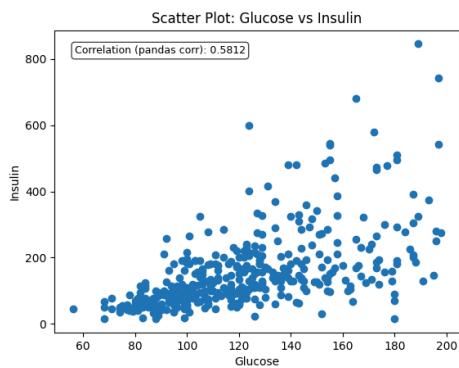


À medida que a glicose aumenta, a insulina tende a aumentar, isso é fisiologicamente esperado

O gráfico apresenta correlação positiva não muito forte, parece existir um pouco de dispersão quando os valores tendem a aumentar

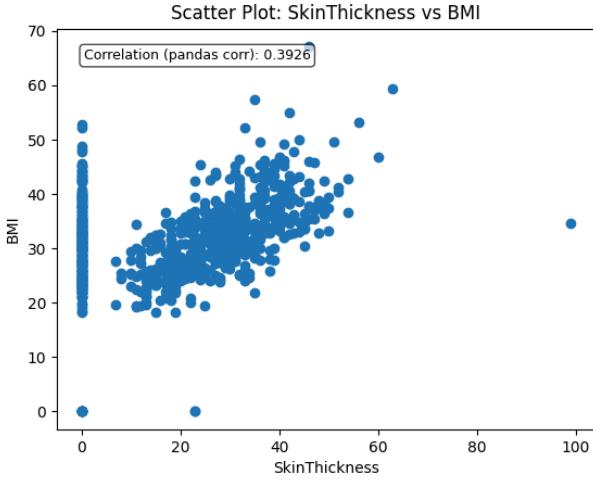
Presença de outliers no canto superior direito do gráfico e presença dos zeros inválidos.

Após o tratamento dos zeros



A correlação fica um pouco mais evidente e percebe-se que não existe mais a presença da linha horizontal no valor zero do eixo y

#### 4.2. SkinThickness x BMI



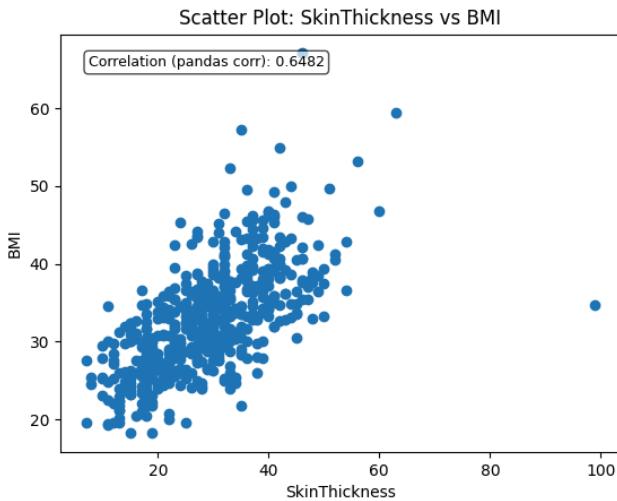
outliers (ex.: SkinThickness  $\approx$  100 mm), que podem influenciar a correlação.

Tendência ascendente confirmando a correlação positiva.  
A maior concentração de pontos está aproximadamente entre, SkinThickness: 15–45 mm e BMI: 25–40 kg/m<sup>2</sup>

Existe dispersão considerável em torno da tendência média.

Há uma concentração vertical em SkinThickness = 0, associada a dados ausentes codificados como zero.

Observa-se também a presença de outliers (ex.: SkinThickness  $\approx$  100 mm), que podem influenciar a correlação.



A remoção dos zeros eliminou a coluna vertical artificial no eixo X.

O padrão reflete melhor a relação real entre gordura subcutânea e massa corporal relativa.

Ainda aparecem alguns outliers (ex.: SkinThickness  $\approx$  100 mm),

## 5. Modelos

### 5.1. Regressão linear

Este modelo foi aplicado e posteriormente removido pois não é adequado seu uso quando a variável Target é binária.

*Este modelo retorna uma predição linear - como o próprio nome diz - sendo mais adequado, por exemplo, quando vamos determinar uma probabilidade, um valor de vendas, uma temperatura,etc. Resumidamente ele calcula a relação entre 2 ou mais variáveis para prever valores futuros ou entender tendências.*

## 5.2. Regressão Logística

*Regressão Logística - de maneira bem sucinta - calcula a probabilidade de algo acontecer e dados este valor transforma em 0 ou 1, para isto ele se vale de um limiar (0.5) . Quanto mais longe deste limiar mais confiança existe no resultado.*

É sensível à escala, sofre com outliers e é fortemente afetada por valores inválidos e nulos.

Para este modelo criamos um Pipeline executando algumas alterações nos dados

Para as colunas "**Glucose**", "**BloodPressure**", "**SkinThickness**", "**Insulin**", "**BMI**", trocamos os zeros por NaN e em seguida pelas medianas para garantir que não tenhamos dados “sujos” que possam prejudicar o uso do modelo.

Para a coluna "**Insulin**" foi aplicado log1p para melhorar a questão da assimetria. Por ser neste caso positiva a direita causada pela presença de valores elevados associados a resistência à insulina e uma distribuição concentrada em valores baixos. Tentamos com isso diminuir a influência dos valores extremos diminuindo a cauda direita.

Efetuamos normalização das variáveis Pregnancies, SkinThickness, BMI, BloodPressure, Glucose, Insulin, e Age utilizando StandardScaler e DiabetesPedigreeFunction utilizando robustScaler - esta última utilizamos um método diferente por ter uma assimetria mais forte

Rodando o modelo aplicando para base de testes 20%, obtivemos as métricas

Accuracy: 0.7078  
Precision: 0.6000  
Recall: 0.5000  
F1-Score: 0.5455  
ROC-AUC: 0.8130

O Modelo apresentou uma baixa acurácia, 70,78% das previsões são acertadas. Precisão baixa e uma Recall muito ruim, Não temos um número muito ruim mas estamos longe de podermos afirmar que este modelo é bom. A harmonia - Precision x Recall - acima de 0.5 pode ser considerada razoável, mesmo com ROC-AUC indicando boa capacidade de separar positivos de negativos

Analisando todos os números o modelo tem baixa acurácia e recall - temos um grande problema nos falsos negativos - como tratamos de diabetes, falsos negativos são muito ruins.

Num cenário onde não podemos perder positivos verdadeiros, o modelo está inadequado.

### 5.3. Árvore de decisão

A árvore de decisão é uma técnica que toma uma decisão numa série de perguntas, semelhante a uma brincadeira de crianças existentes que inicia fazendo perguntas simples e a partir das respostas faz perguntas mais específicas até que consiga tomar uma decisão.

Para este modelo criamos um Pipeline executando algumas alterações nos dados

Para as colunas "**Glucose**", "**BloodPressure**", "**SkinThickness**", "**Insulin**", "**BMI**", trocamos os zeros por NaN e em seguida pelas medianas.

Rodando o modelo e utilizamos os parâmetros

- max\_depth=10 (A profundidade máxima da árvore)
- min\_samples\_split=5 (O número mínimo de amostras necessárias para dividir um nó interno)
- min\_samples\_leaf=2 (O número mínimo de amostras necessárias para estar em um nó folha - nó terminal)
- max\_features="sqrt" (O número de features a considerar na busca pela melhor divisão)
- class\_weight="balanced" (pesos de classe)

Estas foram as métricas retornadas inicialmente

Accuracy: 0.6948

Precision: 0.5479

Recall: 0.7407

F1-score: 0.6299

ROC AUC: 0.7294

Fizemos uma análise da maneira que construímos o modelo está apenas razoável.

Esperávamos um resultado bem melhor para este modelo; estamos perdendo quase 25% dos positivos; quase metade dos valores que estamos identificando como positivo, na verdade não o são e também baixa acurácia. Os números mostram ainda pouca harmonia na aplicação do modelo (F1-Score).

Pesquisando um pouco mais nos deparamos com o método **GridSearchCV** que faz ajuste (**fit**) e treinamento do modelo exaustivamente para encontrar a melhor combinação de parâmetros. Efetuamos as alterações e rodamos novamente o modelo mantendo o threshold padrão 0.5. Esta função nos retornou que os melhores parâmetros são:

```
classifier__ccp_alpha: 0.0  
classifier__class_weight: 'balanced',  
classifier__max_depth: 5,  
classifier__max_features: None,  
classifier__min_samples_leaf: 5  
classifier__min_samples_split: 2
```

Accuracy: 0.7338

Precision: 0.6066

Recall: 0.6852

F1-score: 0.6435

ROC AUC: 0.7226

Depois que utilizamos o gridSearchCV, o modelo teve pequena perda no Recall ( $\approx 0,69$ ), porém com ganho em todas as outras métricas.

Apresentou resultados melhores que o modelo anterior em relação ao Recall  
Modelo razoável para capturar positivos.

## 5.4. Random forest

A Random Forest toma uma decisão a partir da utilização de diversas Árvores de decisão, mas aqui há uma diferença, em vez de utilizar diversas árvores iguais, ele cria as árvores de decisão utilizando “partes dos dados” - ou seja, cada Árvore utilizada trabalha de maneira diferente. Com as respostas de cada uma destas árvores ele toma a sua decisão.

Este modelo não exige normalização ou padronização porém sensível com os zeros artificiais.

Único pré processamento utilizado foi nas colunas "**Glucose**", "**BloodPressure**", "**SkinThickness**", "**Insulin**", "**BMI**", trocamos os zeros por NaN e em seguida pelas medianas.

Aqui começamos a perceber mais necessidade de capacidade computacional para execução do modelo. Cada execução começou a demorar em torno de 40 minutos.

O *GridSearchCV* nos retornou que os melhores parâmetros são:

```
classifier__bootstrap: True  
classifier__class_weight: None  
classifier__max_depth: 5  
classifier__max_features: sqrt  
classifier__min_samples_leaf: 4  
classifier__min_samples_split: 2  
classifier__n_estimators: 400
```

Accuracy: 0.7208

Precision: 0.6410

Recall: 0.4630

F1-score: 0.5376

ROC AUC: 0.6615

Este modelo apresentou uma precisão um pouco melhor que os modelos anteriores mas com Recall muito ruim, ou seja, uma perda muito grande de positivos verdadeiros além de alto custo computacional.

## 5.5. SVM

O próximo modelo implementado foi SVM

*O SVM (Support Vector Machine) é um algoritmo utilizado para classificação binária. (Sim e não; 0 e 1). Utiliza uma ideia bem simples, ele desenha uma linha que irá “separar” os dados.*

*A ideia é encontrar um padrão diferenciador e desenhar uma linha que consiga da melhor maneira possível separar os dados em 2 conjuntos distintos para que quando chegue um novo dados ele consiga, a partir dessa delimitação, determinar em qual dos conjuntos o dado será colocado.*

*Os pontos que ficam mais próximos dessa linha são os mais importantes para o SVM. Eles são chamados de vetores de suporte, porque são eles que “seguram” a posição da linha. Se você mover esses pontos, a linha muda; se mover outros pontos mais distantes, quase nada acontece.*

*Fazendo sempre o mesmo roteiro de transformações com apenas 1 alteração, na variável `DiabetewsPedigreeFunction` mudamos o escalonamento, no lugar de `standardScaler` utilizamos `RobustScaler`, para as outras variáveis mantivemos como estava.*

Voltamos a utilizar `log1p` para melhorar assimetria de insulin

Grid search nos retornou que os melhores parâmetros são:

```
classifier__C:1  
classifier__class_weight:None  
classifier__gamma:0.01  
classifier__kernel:rbf
```

Accuracy: 0.7078

Precision: 0.6098

Recall: 0.4630

F1-score: 0.5263

ROC AUC: 0.6515

Resultado bem próximo do Random Forest. Recall muito ruim, indicando dificuldade de detectar positivos verdadeiros.

## 5.6. KNN

O *k-nearest neighbors (KNN)*, é o algoritmo classificador mais utilizado atualmente. É um algoritmo de aprendizado supervisionado que utiliza a proximidade para fazer classificações ou previsões sobre agrupamentos. De uma maneira bem simples pode-se dizer que ele recebe um número *k* que irá usar para comparação e para decidir a qual conjunto um dado pertence ele utiliza os *k* dados mais próximos e compara, conforme a maioria determinar será feita a escolha.

Aplicamos o Knn usando as mesmas transformações de pré-processamento do SVM. As justificativas se mantém Rodamos o modelo com threshold padrão 0.5, 0.3 e 0.15 para comparação:

O grid search nos retorna que os melhores parâmetros são:

```
classifier__n_neighbors:15  
classifier__p: 2  
'classifier__weights':'uniform'
```

Accuracy: 0.7143

Precision: 0.6042

Recall: 0.5370

F1-score: 0.5686

ROC AUC: 0.6735

Recall um pouco melhor que a maioria dos modelos exceto Árvore de decisão. Mas de uma maneira geral ficou na média geral entre os outros modelos..

## 5.7. Redes Neurais

Aqui no tópico redes neurais utilizamos o *MLPClassifier* (*Multilayer Perceptron Classifier*): um tipo de algoritmo de aprendizado de máquina supervisionado usado para tarefas de classificação. Ela é inspirada na estrutura do cérebro e é uma das formas mais fundamentais de redes neurais.

Para este modelo utilizamos robustScaler em DiabetesPedigreeFunction (maior problema de assimetria) e nas outras variáveis utilizamos StandardScaler, uma vez que outras variáveis com assimetria foram corrigidas via log1p(Insulin) e Age não consideramos presença de outliers

A Variável Insulin tratamos com log1p na tentativa de correção do longa cauda.

Foi efetuada a correção dos zeros nas colunas de sempre.

Accuracy: 0.7208

Precision: 0.6078

Recall: 0.5741

F1-score: 0.5905

ROC AUC: 0.6870

De uma maneira geral foi o modelo que apresentou os melhores resultados. Melhor equilíbrio geral entre Recall e Precision e ROC-AUC.

## 6. Conclusão

Este trabalho serviu muito como uma construção de conhecimento. Nos deparamos com situações adversas que nos desafiaram, como demorar a perceber numa das bases utilizadas os valores nulos estavam populados com a palavra Missing e neste como zero. Descobrimos funções (gridSearchCV) que nos auxiliou na busca de melhores parâmetros para rodar os modelos e também aprendemos algumas técnicas para melhorar variáveis (log1p).

Entendemos que não existe um modelo ideal mas sim, uma melhor relação entre objetivo do negócio, pré-processamento e escolha do modelo.

Podemos dizer que neste nosso caso há 2 possibilidades:

- Não perder positivos verdadeiros

O modelo de rede neural utilizado foi mais adequado na identificação de positivos pois conseguiu maximizar o número de positivos detectados sem elevar muito a taxa de falsos positivos

- Minimizar a identificação de falsos positivos.

Nos casos onde falsos positivos irão elevar os custos podemos dizer que Random Forest e SVM se mostraram mais adequados pois houve um melhor resultado evitando a detecção dos falsos positivos, mesmo aumentando a chance de perder um positivo verdadeiro.

## 7. Referência bibliográfica

- <https://scikit-learn.org/stable/modules/preprocessing.html>
- <https://www.ibm.com/br-pt/think/topics/logistic-regression>
- <https://didatica.tech/o-que-e-e-como-funciona-o-algoritmo-knn/>
- <https://www.ibm.com/br-pt/think/topics/knn>
- <https://medium.com/brasil-ai/knn-k-nearest-neighbors-1-e140c82e9c4e>
- <https://www.ibm.com/br-pt/think/topics/support-vector-machine>
- <https://www.ibm.com/br-pt/think/topics/decision-trees>
- <https://medium.com/data-hackers/%C3%A1rvore-de-decis%C3%A3o-88c7d0fd7a31>
- <https://www.ibm.com/br-pt/think/topics/random-forest>
- <https://didatica.tech/o-que-e-e-como-funciona-o-algoritmo-randomforest/>
- <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
- <https://ebaconline.com.br/blog/random-forest-seo>
- <https://www.cienciaedados.com/normalizacao-em-machine-learning/>
- <https://medium.com/@carlosalbertoff/mlp-classifier-526978d1c638>
- <https://www.deeplearningbook.com.br/as-principais-arquiteturas-de-redes-neurais/>