

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CAMPUS FLORIANÓPOLIS  
BACHARELADO EM MATEMÁTICA E COMPUTAÇÃO  
CIENTÍFICA

Renan Stolf Farhat

**Floresta Aleatória Isotrópica:** Um algoritmo de classificação

Florianópolis - SC  
2023

Renan Stolf Farhat

**Floresta Aleatória Isotrópica: Um algoritmo de classificação**

Trabalho de Conclusão de Curso de Graduação em Bacharelado em Matemática e Computação Científica do Campus Florianópolis da Universidade Federal de Santa Catarina para obtenção do título de Bacharelado em Matemática e Computação Científica.

Orientador: Prof. Vladimir Pestov, Dr.

Coorientadora: Prof. Maria Inez Cardoso Gonçalves, Dra.

Florianópolis - SC

2023

### Ficha de identificação da obra

A ficha de identificação é elaborada pelo próprio autor.

Orientações em:

<http://portalbu.ufsc.br/ficha>

Renan Stolf Farhat

**Floresta Aleatória Isotrópica:** Um algoritmo de classificação

Este Trabalho Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharelado em Matemática e Computação Científica e aprovado em sua forma final pelo Curso de Bacharelado em Matemática e Computação Científica.

Florianópolis - SC, 06 de julho de 2023.

---

Prof. Felipe Lopes Castro, Dr.  
Coordenador do Curso

**Banca Examinadora:**

---

Prof. Vladimir Pestov, Dr.  
Orientador  
Instituição UFSC

---

Prof. Maria Inez Cardoso Gonçalves, Dra.  
Avaliadora  
Instituição UFSC

---

Prof. Aldrovando Luis Azeredo Araújo, Dr.  
Avaliador  
Instituição UFSC

Este trabalho é dedicado aos meus colegas, Professores,  
familiares e amigos.

## **AGRADECIMENTOS**

Marilza Aparecida Stolf, Dra.

Prof. Irene Pereira Nobre Stolf

Nestor Stolf

Nestor Stolf Filho, Dr.

Mauricio Zacarias Farhat, Dr.

Prof. Maria Inez Cardoso Gonçalves, Dra.

Prof. Vladimir Pestov, Dr.

Prof. Silvia Martini de Holanda, Dra.

Bruno Yamamoto

Kaique Belluco

Thiago Gonçalves

Daniel Vezozzo Farhat

Silmara Barnabé

Prof. Ivan Pontual Costa e Silva, Dr.

## RESUMO

Essa pesquisa objetiva apresentar e estudar uma variação algoritmo de Aprendizado de Máquina(Machine Learning) Supervisionado para a classificação de novos casos, o Isotropic Random Forest. O algoritmo Isotropic Random Forest realiza transformações isotrópicas na base de dados e a partir de cada uma delas gera um classificador Árvore de Decisão, o conjunto final de classificadores votam democraticamente no rotulo de maior representatividade para cada novo caso. Somente após a finalização deste trabalho, descobrimos que algoritmo já havia sido sugerido em 2016 por Rico Blaser e Piotr Fryzlewicz sob o nome “Random Rotation Ensembles”.(Blaser, Rico e Fryzlewicz, Piotr. “Random Rotation Ensembles”. Journal of Machine Learning Research, 17, 2016, 1-26)[8]

**Palavras-chave:** Algorítmico. Supervisionado. Isotópico. Random-Forest. Inferência. Classificação. Árvore. Floresta. Acurácia. Entropia. Gain.

## ABSTRACT

This research aims to present and study a variation of a Supervised Machine Learning algorithm for classifying new cases, the Isotropic Random Forest. The Isotropic Random Forest algorithm performs isotropic transformations in the database and from each one of them generates a Decision Tree classifier, the final set of classifiers democratically vote on the most representative label for each new case. Only after finishing this work, we discovered that algorithm had already been suggested in 2016 by Rico Blaser and Piotr Fryzlewicz under the name “Random Rotation Ensembles”.(Blaser, Rico e Fryzlewicz, Piotr. “Random Rotation Ensembles”. Journal of Machine Learning Research, 17, 2016, 1-26)[8]

**Keywords:** Algorithmic. Supervised. Isotopic. Random-Forest. Inference. Classification. Tree. Forest. Accuracy. Entropy. Gain.



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>9</b>
1.0.1	Aprendizado de Máquina	9
1.0.2	Aprendizado de Máquina Supervisionado	9
1.0.3	Amostra Rotulada e não Rotulada	10
1.0.4	Classificador	10
1.0.5	Regra de Aprendizagem	10
1.0.6	Isotropia e Anisotropia	11
1.0.7	Algoritmo Floresta Aleatória Isotrópica	12
1.0.8	Histórico do Trabalho	12
<b>2</b>	<b>CLASSIFICADORES ÁRVORE DE DECISÃO E FLORESTA ALEATÓRIA</b>	<b>14</b>
2.1	ENTROPIA	14
2.2	GANHO DE INFORMAÇÃO (GAIN)	15
2.3	ALGORITMO ÁRVORE DE DECISÃO	15
2.4	CATEGORIZAÇÃO DE VARIÁVEIS NUMÉRICAS	18
2.5	ALGORITMO FLORESTA ALEATÓRIA (RANDOM FOREST)	18
<b>3</b>	<b>FLORESTA ALEATÓRIA ISOTRÓPICA</b>	<b>22</b>
3.1	MATRIZES ORTOGONAIS	23
3.2	ROTAÇÕES ALEATÓRIAS DO CONJUNTO DE DADOS	25
<b>4</b>	<b>CONCLUSÃO</b>	<b>27</b>
	<b>REFERÊNCIAS</b>	<b>29</b>

## 1 INTRODUÇÃO

### 1.0.1 Aprendizado de Máquina

Aprendizado de Máquina (Machine Learning) é o processo que visa ensinar uma máquina/inteligência a inferir sobre determinado evento, através dos dados disponíveis. Nesse contexto, a matemática e a estatística desempenham um papel fundamental, fornecendo uma variedade de soluções e algoritmos sofisticados para análise de padrões e dados. Por exemplo, é possível usar essas técnicas para compreender as diferenças entre o perfil de uma pessoa saudável, calcular o tempo estimado de um jogador de beisebol chegar à quarta base sob certas circunstâncias, como mostra no filme MoneyBall, entre outras aplicações, [6].

### 1.0.2 Aprendizado de Máquina Supervisionado

O aprendizado de Máquina Supervisionado envolve o uso de algoritmos computacionais para aprender uma função capaz de mapear dados/variáveis de entrada para suas respectivas saídas (rótulos), com base em exemplos prévios já rotulados nessa aplicação, [7].

O objetivo é realizar previsões com precisão e minimizar erros de classificação ao rotular nossos casos.

Isso nos ajuda em diversas áreas e tarefas na atualidade, como por exemplo prever quando uma peça de um avião precisará ser trocada, ou identificar quando alguém poderá não pagar um empréstimo. Informações as quais são vitais para muitas empresas e organizações entenderem melhor os seus clientes, produtos, mercados e, assim, avançarem suas ações para outro patamar mais inteligente e estratégico.

### 1.0.3 Amostra Rotulada e não Rotulada

Uma amostra rotulada consiste em elementos que estão associados a rótulos. No caso binário, cada elemento estaria associando ao rótulo "0" ou "1", ou de forma mais geral, às classes  $c_1$  ou  $c_2$ .

Assim, definimos nossa amostra rotulada da seguinte maneira:

$\mathbb{T} := [\mathbb{X}\mathbb{Y}]$  tal que

$$\mathbb{X} := (x_1, \dots, x_n) \in \mathbb{R}^{l \times n},$$

$$\mathbb{Y} := (y_1, \dots, y_n) \in \{c_1, c_2\}^l := C^l.$$

Por outro lado, amostras não rotuladas não possuem tal associação, [7].

### 1.0.4 Classificador

O conjunto de rótulos pode ser qualquer, mas neste trabalho será considerado o caso binário, ou seja, com dois rótulos.

Assim, definimos a função classificador(a função de predição ou função de transferência ) como

$$h : \mathbb{T} \longrightarrow \{c_1, c_2\},$$

onde  $\mathbb{T}$  é o conjunto de dados de entrada e  $\{c_1, c_2\}$  representam os dois rótulos possíveis.

### 1.0.5 Regra de Aprendizagem

Uma regra da aprendizagem é uma aplicação associando a cada amostra rotulada,  $\mathbb{T}$ , um classificador,  $h$ .

Dado uma amostra  $\mathbb{T} := [\mathbb{X}\mathbb{Y}]$ , podemos dizer formalmente que uma regra de aprendizagem é uma família  $L = (L_n)^\infty$  com  $n = 0, 1, 2, 3, \dots$ . Onde, aplicada a um ponto  $x \in \mathbb{X}$

$$L_n : \mathbb{T} \times \mathbb{X} \longrightarrow \{c_1, c_2\}, [7].$$

Um exemplo de regra da aprendizagem é o classificador k-NN(k-vizinhos mais próximos), [7].

### 1.0.6 Isotropia e Anisotropia

“No contexto da ciência de dados, os termos isotropia e anisotropia podem ser usados de maneira um pouco diferente, mas ainda se referem a características de dados ou modelos.

A isotropia em ciência de dados pode se referir à igualdade ou uniformidade nas características dos dados em todas as direções ou dimensões. Isso implica que não há preferência direcional nos dados ou que não há correlação específica entre as variáveis.

Por outro lado, a anisotropia em ciência de dados indica a existência de uma variação direcional nas características dos dados. Isso pode ser observado quando há correlações diferentes entre variáveis em diferentes direções ou quando as características dos dados têm uma distribuição não uniforme ou variam de forma desigual em diferentes dimensões.

A detecção de anisotropia ou isotropia em dados pode ser útil para entender a estrutura dos dados, identificar correlações direcionais ou determinar quais modelos são mais adequados para representar e analisar os dados. Dependendo do contexto, a anisotropia ou isotropia dos dados pode ter implicações importantes para a escolha de técnicas de modelagem, como a seleção de algoritmos de aprendizado de máquina ou a aplicação de métodos específicos de pré-processamento de dados.”[2]

Um exemplo da isotropia em ciência de dados é o próprio algoritmo Floresta Aleatória Isotrópica (Isotropic Random Forest), o qual será apresentado ao longo deste trabalho. Esse é um novo algoritmo e quase não há nenhuma pesquisa disponível na literatura.

Por outro lado, um exemplo da anisotropia em ciência de dados

é, o algoritmo Floresta Aleatória (Random Forest) que depende da escolha das coordenadas, dado que as partições são feitas nas direções das coordenadas/variáveis.

### 1.0.7 Algoritmo Floresta Aleatória Isotrópica

O algoritmo Floresta Aleatória Isotrópica usa sistemas de coordenadas e transformações de recursos aplicados a floresta aleatória, o que nos permite analisar os dados por diversos ângulos, proporcionando melhores condições de detectar os padrões e reduzir os impactos de dados desbalanceados nas análises.

Aborda as seguintes limitações: Viés em conjuntos de dados desbalanceados, dificuldade em capturar padrões complexos, sensibilidade à estrutura geométrica de recursos em relação à estrutura de grade dos dados.

Esse algoritmo pode oferecer vantagens como robustez, melhorias de desempenho, versatilidade, entre outras.

Existem métodos melhores para a seleção de transformações ortogonais, os quais consomem menos recursos computacionais do que o método que foi aplicado neste trabalho. Por exemplo, a utilização de matrizes gaussianas e matrizes de Bernoulli.

Também há a necessidade de testar o algoritmo para conjuntos de dados mais complexos e fazer estudos comparativos sobre o assunto.

### 1.0.8 Histórico do Trabalho

Este TCC é dedicado ao desenvolvimento de um algoritmo que incorpora rotações aleatórias no algoritmo clássico de classificação conhecido como Floresta Aleatória.

No final 2019, começamos a trabalhar no algoritmo utilizando matrizes ortogonais, como um primeiro passo.

Um segundo passo seria utilizar matrizes gaussianas, pois são muito mais fáceis de gerar do que matrizes ortogonais. E, como terceiro, passo utilizar matrizes de Bernoulli.

Somente após a finalização deste trabalho, descobrimos que algoritmo já havia sido sugerido em 2016, por Rico Blaser e Piotr Fryzlewicz, sob o nome “Random Rotation Ensembles”, [8].

## 2 CLASSIFICADORES ÁRVORE DE DECISÃO E FLORESTA ALEATÓRIA

Em problemas de classificação, o algoritmo classificador Árvore de Decisão divide o espaço de dados  $\Omega_i$  em regiões disjuntas conhecidas como folhas, com base em uma métrica específica. O objetivo é que o número de elementos em cada folha sejam o menor possível. São muito utilizadas as métricas Gini e Entropia,[5]

Neste estudo, utilizaremos a entropia.

### 2.1 ENTROPIA

A entropia de uma amostra rotulada  $S$  é uma medida do grau de não homogeneidade (desordem, "caos") dessa amostra, [5].

É um métrica utilizada na classificação de dados e pode ser calculada da seguinte maneira:

$$E(S) = \sum_i p_i \log_2(p_i). \quad (1)$$

Onde  $S$  é um conjunto de treinamento,  $p_i := \frac{|c_i|}{|S|}$  é proporção da classe  $c_i$  no conjunto  $S$  e  $|\bullet|: S \rightarrow \mathbb{N}$  é a aplicação que leva um conjunto em sua cardinalidade.

Como nesse trabalho trataremos de classificação binária, a entropia será dada por

$$\begin{aligned} E(S) &= -(p_1 \log_2(p_1) + p_2 \log_2(p_2)) = -\log_2(p_1^{p_1} p_2^{p_2}) = \\ &= -\log_2(p_1^{p_1} (1 - p_1)^{1-p_1}) \end{aligned}$$

Note que se a amostra rotulada  $S$  for homogênea, ou seja, há somente elementos de uma única classe, então  $E(S) = 0$ . E atinge o seu máximo quando  $p_1 = p_2 = \frac{1}{2}$ , ocorre apenas em casos de duas classes.

## 2.2 GANHO DE INFORMAÇÃO (GAIN)

O Ganho de Informação (Gain)  $G(S, \bar{x})$  representa a redução esperada da entropia do conjunto  $S$  quando este é particionado pela variável categórica  $\bar{x}$  de  $i$  categorias  $\bar{x}_i$ . Pode ser calculado por:

$$G(S, \bar{x}) = E(S) - \sum_i \frac{|\bar{x}_i|}{|S|} E(\bar{x}_i). \quad (2)$$

Onde  $E(S)$  é a entropia da amostra rotulada  $S$ ,  $|\cdot|$  é a cardinalidade, indicando a proporção de categorias  $\bar{x}_i$  em relação ao total de linhas/casos  $|S|$ . Já  $E(\bar{x}_i)$  indica a entropia restrita aos elementos  $\bar{x}_i$ , ou seja, é a entropia ponderada de cada classe, com  $\sum_i \frac{|\bar{x}_i|}{|S|} = 1$ .

Logo, quanto maior for  $G(S, \bar{x})$ , melhor é o poder de classificação de  $\bar{x}$  em relação aos elementos de  $S$ , [5]

## 2.3 ALGORITMO ÁRVORE DE DECISÃO

O Algoritmo Árvore de Decisão é um classificador que particiona de modo hierárquico o espaço das variáveis preditoras  $\mathbb{R}^n$  em um conjunto de  $F$  regiões retangulares  $R = \{R_1, \dots, R_F\}$  disjuntas, também denominadas de folhas, [5].

Assim, quando um novo caso  $x_0$  surge determinamos em qual região  $R_j$  ele se encontra.

Os hiperplanos que dividem o espaço podem ser visto como os ramos da árvore, e os conjuntos como os nós.

Nesse trabalho, nas árvores desenvolvidas utilizaremos apenas 2 ramificações por nós a cada iteração.

Algoritmo Árvore de Decisão [5]

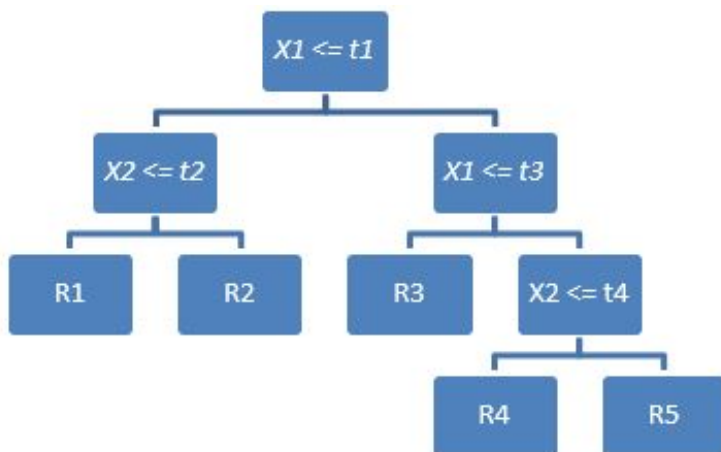
I) Ao começar com todos os exemplos de  $\mathbb{T}$ , escolha a variável  $x_i$  que melhor divide as observações, ou seja, produz subconjuntos



heterogêneos entre si. Nesse trabalho utilizamos a entropia e o ganho de informação para realizar essa avaliação.

II) Em seguida, criamos um nó para cada subconjunto disjunto gerado a partir de  $x_i$ , particionando a amostra rotulada  $S$  em folhas.

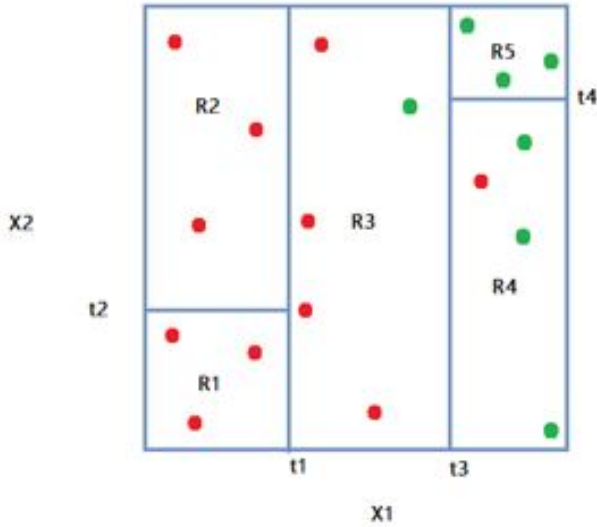
III) Repetimos esse procedimento para cada novo subconjunto disjunto(folha) que ainda não seja homogêneo, até atender algum critério de parada.



Defina

$$I_A : X \rightarrow \mathbb{Z}$$
$$I_A(x) = \begin{cases} 1; & \text{se } x \in A; \\ 0; & \text{se } x \notin A; \end{cases}$$

Dado isso, uma vez determinada a região  $R_j$  a qual uma nova observação  $a_0$  pertence, classificamos esse elemento como sendo da classe mais frequente em  $R_j$ .



Formalmente, defina  $a_l := (x_{l1}, \dots, x_{ln}) \in \mathbb{T}$  uma observação de  $\mathbb{T}$ .

Seja  $N_j := \sum_j I_{R_j}(x_i)$  o número de observações contidas na região  $R_j$ .

Assim, a proporção de observações de classe  $c_i$  em  $R_j$  é dada por

$$\hat{\rho}(c_k, R_j) = \frac{1}{N_j} \sum_{l: a_l \in R_j} I_{\{c_j\}}(y_l).$$

Por conseguinte, a função de classificação para uma nova observação  $a_0$  é dada por

$$h(a_0, \mathbb{T}) = \sum_{j=1}^F \rho_j I_{R_j}(a_0). \tag{3}$$

onde  $\rho_j := \max_{k: c_k \in \{c_1, c_2\}} \{\hat{\rho}(c_k, R_j)\}$ .

E assim, classificamos novos casos.

## 2.4 CATEGORIZAÇÃO DE VARIÁVEIS NUMÉRICAS

Note que o *gain* avalia variáveis categóricas. No caso de uma variável numérica  $x_i$ , aplicamos um processo de categorização para cada  $x_i$ , de forma a gerar a variável categórica correspondente  $\tilde{x}_i$ , através a partição que fornece o melhor *gain* entre as partições consideradas e, a partir disso, encontramos o melhor atributo para a iteração em questão, [3].

Primeiramente, geramos  $(n + 1)$  partições de  $x_i$  com volumetria/distribuição iguais, considerando os conjunto de pontos de corte como os extremos máximos de cada partição. Logo, se desejamos testar  $n$  pontos de corte, devemos solicitar  $n + 1$  pontos, pois o conjunto gerado pelo extremo máximo da última partição é vazio, trivialmente.

Assim, basta encontrar o *gain* correspondente a cada ponto de corte e selecionar o maior entre eles, ou seja, o ponto de corte que melhor divide  $x_i$ .

## 2.5 ALGORITMO FLORESTA ALEATÓRIA (RANDOM FOREST)

A Aprendizagem ensemble (ensemble learning) é uma estratégia importante para criar novos algoritmos de aprendizagem e efetuar decisões por meio do voto maioritário entre várias versões de regras de aprendizagem, geralmente do mesmo tipo mas com os parâmetros variáveis. Por exemplo o algoritmo Random Forest que usa o voto maioritário entre um alto número (por exemplo, 500) de árvores de decisão construídos usando parâmetros escolhidos aleatoriamente, [7].

O algoritmo Floresta Aleatória (Random Forest) é um algoritmo de aprendizagem supervisionada que identifica padrões nos dados com base em um conjunto de classes associadas aos casos/dados históricos amostrados.

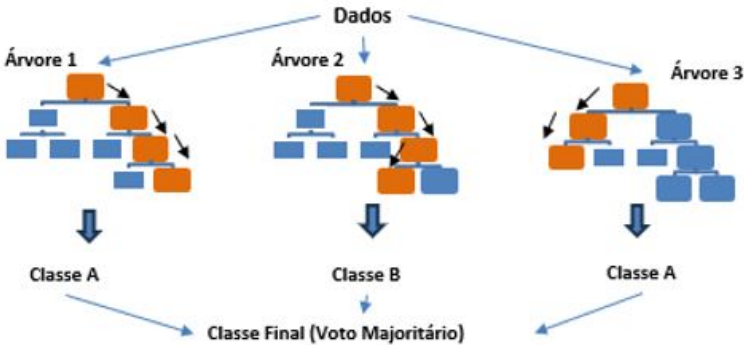
Devido aos excelentes resultados promovidos por esse modelo,

ele é muito utilizado em problemas de classificação e regressão.

O algoritmo cria um conjunto de classificadores Árvore de Decisão a partir de amostras sem reposição (bagging) do conjunto de dados históricos destinado ao treinamento do modelo. Onde, através do voto majoritário sobre as classes, elege a classe representante para um novo caso, [1, 7].

Esse estudo tem o objetivo de estudar uma variação do algoritmo Random Forest, chamada Isotropic Random Forest, desenvolvida através de rotações aleatórias sobre o espaço de dados.

O algoritmo proposto, Isotropic Random Forest, gera as Árvores de Decisão através de rotações aleatórias sobre o espaço de dados, ao invés do processo bagging.



Seja

$$\mathbb{T} := \begin{bmatrix} x_{11} & \cdots & x_{1n} & y_1 \\ \vdots & \ddots & \vdots & \vdots \\ x_{l1} & \cdots & x_{ln} & y_l \end{bmatrix} := \begin{bmatrix} \mathbb{X} & \mathbb{Y} \end{bmatrix},$$

uma matriz de dados com  $n$  variáveis (colunas) e  $l$  linhas (observações) classificada, onde

$\mathbb{X} := (x_1, \dots, x_n) \in \mathbb{R}^{l \times n}$  é a matriz contendo as  $n$  variáveis  $x_i := (x_{1i}, \dots, x_{li})$  de  $\mathbb{T}$ .

$\mathbb{Y} := (y_1, \dots, y_n) \in \{0, 1\}^n$  tal que  $c_1, c_2$  são os rótulos.

A Floresta Aleatória (Random Forest) é uma combinação de preditores Árvore de Decisão, em que cada árvore é gerada a partir de um subconjunto  $\Omega_k$  amostrado de forma independente e representativa a partir do conjunto dados de treinamento  $\mathbb{X}$ .

Usualmente, esse conjunto de subamostras  $\Omega_k$  é construído a partir do método de ensacamento (bootstrap ou bagging - Breiman, 1996), que consiste em amostragens com substituição obtidas a partir do conjunto de dados de treinamento  $\mathbb{T}$ , [1].

O princípio comum incorporado nessas metodologias é que, para o desenvolvimento da  $k$ -ésima árvore classificadora  $h(x, \Omega_k)$ , a amostra de treinamento  $\Omega_k$  associada a essa árvore é gerada de forma independente das amostras de treinamento anteriores  $\{\Omega_1, \Omega_2, \dots, \Omega_{k-1}\}$ , sendo  $x$  novo vetor de dados de entrada a ser classificado.

Assim, para classificar um novo caso, o conjunto de árvores geradas votam na classe mais popular. Esse procedimento é chamado Floresta Aleatória (Random Forest).

Uma floresta aleatória é um classificador que consiste em uma coleção de classificadores árvore de decisão  $\{h(x, \Omega_k), k = 1, \dots, K\}$  onde os  $\Omega_k$  são amostras rotuladas fixas e cada árvore vota unicamente na classe mais popular do novo vetor de entrada  $x$ , [1, 4].

Algoritmo Random Forest:

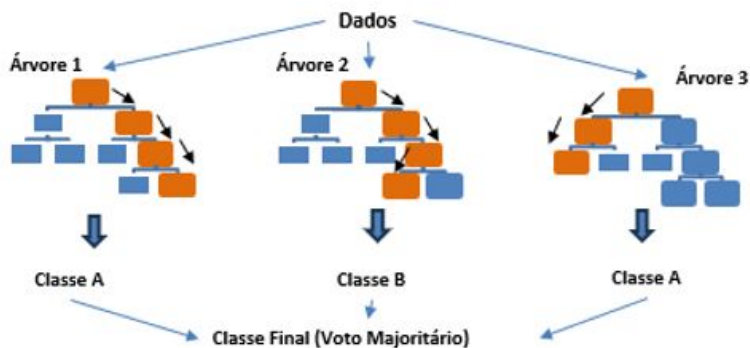
I) Gere um conjunto de  $K$  subamostras rotuladas  $\Omega = \{\Omega_1, \Omega_2, \dots, \Omega_K\}$  a partir de  $\mathbb{T}$ , utilizando o método bootstrap (amostragem com reposição) e com tamanho amostral  $\eta$ ;

II) Para cada subamostra  $\Omega_k$  em  $\Omega$ , crie um classificador árvore

de decisão  $h(x, \Omega_k)$ ;

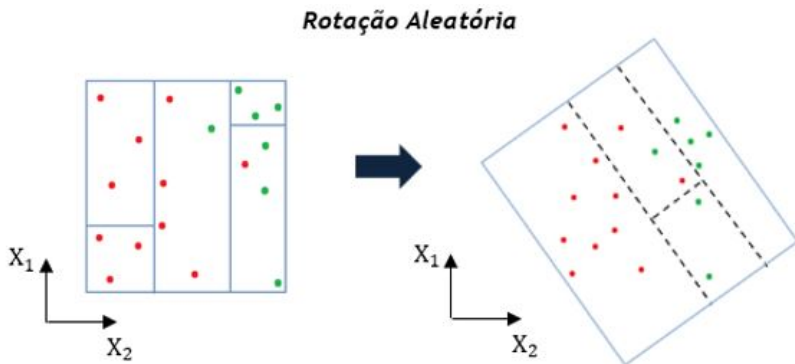
III) Calcule a classe por meio da votação das árvores, considerando apenas um voto por árvore.

Pelo fato dos cortes serem feitos diretamente ao longo das coordenadas, o algoritmo possui comportamento anisotrópico.[1][4]



### 3 FLORESTA ALEATÓRIA ISOTRÓPICA

O algoritmo Floresta Aleatória Isotrópica utiliza rotações aleatórias para obter um comportamento isotropico. Ou seja, que as propriedades do algoritmo não dependam das coordenadas/direções.



Seja  $\mathbb{T}$  uma matriz de dados com  $n$  variáveis (colunas) e  $l$  linhas (observações) classificada, onde

$\mathbb{X} := (x_1, \dots, x_n) \in \mathbb{R}^{l \times n}$  é a matriz contendo as  $n$  variáveis  $x_i := (x_{1i}, \dots, x_{li})$  de  $\mathbb{T}$ .

$\mathbb{Y} := (y_1, \dots, y_n) \in \{c_1, c_2\}^l := C^l$  tal que  $c_1, c_2$  são os rótulos.

O algoritmo Floresta Aleatória Isotrópica (Isotropic Random Forest) consiste em uma combinação de classificadores do tipo Árvores de Decisão  $h(x, \Omega_k)$  que votam na classe mais popular para classificar um novo caso.

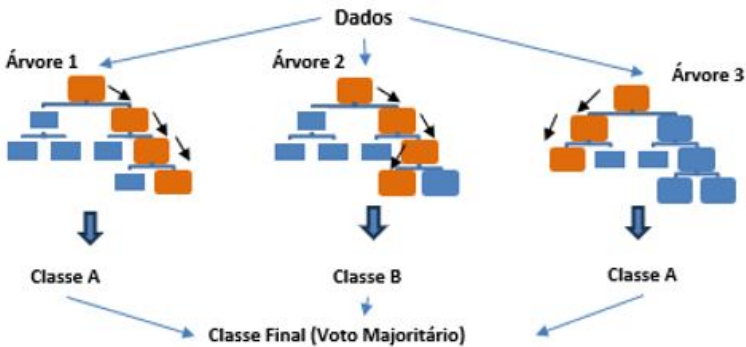
Todavia, o algoritmo Floresta Aleatória Isotrópica gera os conjuntos de treinamento  $\{\Omega_1, \Omega_2, \dots, \Omega_K\}$  a partir de rotações aleatórias da matriz de dados original  $\mathbb{X}$  no espaço  $\mathbb{R}^{l \times n}$ .

Algoritmo Isotropic Random Forest:

I) Gere, a partir de  $\mathbb{T}$ , um conjunto  $\Omega = \{\Omega_1, \Omega_2, \dots, \Omega_K\}$  com  $K$  rotações aleatórias rotuladas;

II) Para cada matriz de dados rotacionada  $\Omega_k$  em  $\Omega$ , desenvolva um classificador árvore de decisão  $h(x, \Omega_k)$ ;

III) Calcule a classe mais popular a partir da votação única dos conjunto de árvores  $\{h(x, \Omega_1), h(x, \Omega_2), \dots, h(x, \Omega_K)\}$ .



### 3.1 MATRIZES ORTOGONAIS

**Definição 3.1.1** (Matriz Aleatória Real). *Uma matriz real  $A$  é considerada aleatória se suas entradas forem reais e geradas de forma aleatória.*

**Definição 3.1.2** (Independência Linear). *Um conjunto  $\{v_1, \dots, v_n\}$  é linearmente independente se nenhum  $v_i$  se escreve como combinação linear dos demais, [6].*

**Teorema 3.1.1** (Teorema de Gram-Schmidt). *Seja  $B = \{v_1, \dots, v_n\}$  um conjunto de vetores linearmente independentes em um espaço vetorial  $V$  com produto interno. Então, existe um conjunto ortonormal*



$O_A := \{o_1, \dots, o_n\}$  em  $V$ , dado por

$$o_1 = v_1,$$

$$o_k = \frac{v_k - \sum_{j=1}^{k-1} \text{proj}_{o_j}(v_k)}{\left\| v_k - \sum_{j=1}^{k-1} \text{proj}_{o_j}(v_k) \right\|}.$$

Onde

$$\text{proj}_{o_j}(v_k) = \frac{\langle v_k, o_j \rangle}{\langle o_j, o_j \rangle} o_j \quad e \quad k = 2, \dots, n.$$

Tal que  $\text{span}\{v_1, \dots, v_n\} = \text{span}\{o_1, \dots, o_n\}$ , onde  $\text{span}\{v_1, \dots, v_n\}$  denota o subespaço de  $V$  gerado pelos vetores  $v_1, \dots, v_n$ , [6].

**Definição 3.1.3** (Matriz Ortogonal). *Uma matriz  $O$  é dita ortogonal se possui inversa  $M^{-1}$  e  $M^{-1} = M^t$ , onde  $M^t$  transposta de  $M$ .*

**Propriedades 3.1.1** (Colunas da Matriz Ortogonal). *Uma matriz é ortogonal se, e somente se, seus vetores coluna (ou linha) formam uma base ortonormal, [6].*

**Teorema 3.1.2** (Teorema da Isometria das Matrizes Ortogonais). *Seja  $O := [o_1 \dots o_n]$  uma matriz ortogonal em  $\mathbb{R}^{n \times n}$ , onde  $o_n := O(*, i)$  representa o  $i$ -ésimo vetor coluna de  $O$ . Então, para quaisquer  $v, w \in \mathbb{R}^n$  temos que*

(i)  $\langle Ov, Ow \rangle = \langle v, w \rangle$ ;

(ii)  $\| Ov \| = \| v \|$ ;

(iii) *O ângulo entre  $Ov$  e  $Ow$  é igual ao ângulo entre  $v$  e  $w$ , [6].*

Note que pelo Teorema da Isometria das Matrizes Ortogonais 3.1.2, a aplicação associada a uma matriz ortogonal preserva ângulos e comprimentos.

### 3.2 ROTAÇÕES ALEATÓRIAS DO CONJUNTO DE DADOS

Agora, seja  $A_i$  uma matriz aleatória real qualquer em  $\mathbb{R}^{n \times n}$ . Devido a aleatoriedade das entradas de  $A$ , as colunas serão linearmente independentes.

Assim, pelo Teorema de 3.1.1, podemos encontrar uma base ortornormal  $O_i := O(A_i) := \{o_1, \dots, o_n\}$  de  $\mathbb{R}^n$  ortogonalizando os vetores colunas da matriz  $A_i$ .

Pela propriedade 3.1.1, como as colunas de  $O_i$  são ortogonais então a matriz  $O_i$  formada por essa colunas é uma matriz ortogonal.

Seja  $K$  o número de classificadores árvore de decisão a serem desenvolvidos e  $i \in \{1, \dots, K\}$ .

Defina

$$\Omega_i := \mathbb{X}O_i.$$

Note que

$$\begin{aligned} \Omega_i &:= \begin{bmatrix} x_{11} & \dots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{l1} & \dots & x_{ln} \end{bmatrix} \begin{bmatrix} o_{11} & \dots & o_{1n} \\ \vdots & \ddots & \vdots \\ o_{l1} & \dots & o_{ln} \end{bmatrix} := \\ & \begin{bmatrix} L(1, *) \\ \vdots \\ L(l, *) \end{bmatrix} \begin{bmatrix} O(*, 1) \dots O(*, n) \end{bmatrix} = \\ &= \begin{bmatrix} \langle L(1, *), O(*, 1) \rangle & \langle L(1, *), O(*, 2) \rangle & \dots & \langle L(1, *), O(*, n) \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle L(l, *), O(*, 1) \rangle & \langle L(l, *), O(*, 2) \rangle & \dots & \langle L(l, *), O(*, n) \rangle \end{bmatrix}. \end{aligned}$$

Onde  $\mathbb{X}$  é a matriz de dados de treinamento e  $O_i$  corresponde à matriz  $A_i$  ortogonalizada através do Teorema de Gram-Schmidt 3.1.1.

Dado que  $X(i, *)O = (XO)(i, *)$  e  $y_i$  é rótulo de  $X(i, *)$ , então também será rótulo de  $(XO)(i, *)$ , com

$$(XO)(i, j) = \sum_{h=1}^n x_{ih}o_{hj}.$$

Note também que cada linha (observação, caso estudado) de  $\Omega_i(i, *) := (XO)(i, *)$  é uma "mistura" dos valores  $\{x_{i1}, \dots, x_{in}\}$  das variáveis na linha  $i$  em  $\mathbb{X}$  com os valores dos vetores coluna  $\{o_1, \dots, o_n\}$  da matriz ortogonal  $O$  gerada.

Como  $O$  é uma matriz ortogonal, então o conjunto  $\{o_1, \dots, o_n\}$  é linearmente independente pelas propriedades das colunas da Matriz Ortogonal apresentadas em 3.1.1.

Assim, não se espera correlação entre os classificadores, pois cada entrada  $\Omega_i(i, j) := (XO)(i, j) = \langle L(i, *), O(*, j) \rangle$  não pode ser escrita como combinação linear das demais, uma vez que, se isso ocorresse, então implicaria que  $\{o_1, \dots, o_n\}$  ou  $\{x_{i1}, \dots, x_{in}\}$  seria linearmente dependente.

Dessa forma obtemos o conjunto  $\{\Omega_1, \Omega_2, \dots, \Omega_K\}$  e, a partir dele, a floresta de classificadores  $\{h(x, \Omega_1), h(x, \Omega_2), \dots, h(x, \Omega_K)\}$  Árvore de Decisão que classificam novos casos a partir do voto único majoritário das árvores. Conforme abaixo, [5].

$$h(x, \Omega_k) = \sum_{j=1}^F \rho_{k,j} I_{R_{k,j}}(x),$$

onde  $\rho_j := \max_{k:c_k \in \{c_1, c_2\}} \{\hat{\rho}(c_k, R_j)\}$  e  $x \in \mathbb{R}$  é uma nova entrada de dados de um novo caso a ser classificado.

Esse procedimento é chamado de Floresta Aleatória Isotrópica.

## 4 CONCLUSÃO

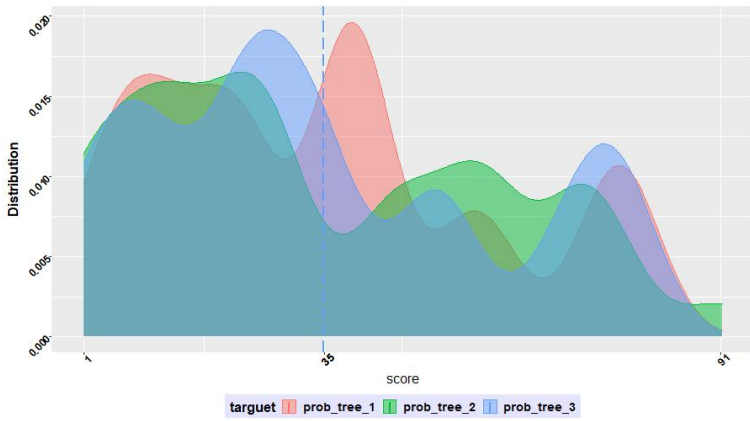
O novo algoritmo de classificação Isotropic Random Forest foi testado em uma base de dados com 34.9% dos pacientes contendo diabetes e o restante sem a doença, utilizando a linguagem R. Cada árvore foi criada de modo que não haja mais de 10 níveis ou menos que 5% do volume total em cada folha, utilizando quebra binária.

A tabela abaixo apresenta o número de classificadores árvore de Decisão utilizados para classificar e suas respectivas taxas de acerto no rótulo 1, no rótulo 0 e a taxa de acerto geral (ambos os rótulos).

n_trees	acerto_rotulo_1	acerto_rotulo_0	acerto_total
1	0.869403	0.564	0.6705729
2	0.7537313	0.714	0.7278646
3	0.8358209	0.634	0.7044271
4	0.7873134	0.702	0.7317708
5	0.8507463	0.644	0.7161458
6	0.7947761	0.686	0.7239583
7	0.8134328	0.666	0.7174479
8	0.7835821	0.708	0.734375
9	0.8208955	0.682	0.7304688
10	0.8059701	0.712	0.7447917
20	0.8171642	0.712	0.7486979

Note que quanto mais classificadores, melhor a performance dos algoritmos Isotropic Random Forest, aumentando a acurácia em média 7%.

Curiosamente, o fato de as árvores serem diferentes entre si é vital para que haja variação positiva na performance. Assim, podemos observar a diferença entre as árvores ao analisar a distribuição de probabilidades delas conforme exemplo abaixo.



## REFERÊNCIAS

- [1] Leo Breiman. “Random Forests”. Em: *Machine Learning* 45 (out. de 2001), pp. 5–32.
- [2] ChatGPT (OpenIA). *ChatGPT respondendo à questão "Que pode significar a diferença entre isotropia e anisotropia no contexto particular da ciência de dados?"* <https://chat.openai.com>. Accessed: 2023-07-10.
- [3] Sergio Queiroz e Cleber Zanchettins. *Categorização de Variáveis Numéricas, UFPE*. <https://www.cin.ufpe.br/~if684/EC/aulas/Aula-arvores-decisao-SI.pdf>. Accessed: 2023-06-24.
- [4] Trevor Hastie, Robert Tibshirani e Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. 2<sup>a</sup> ed. Springer, 2009. Disp. em: <http://www-stat.stanford.edu/~tibs/ElemStatLearn/>.
- [5] Paulo C. Marques F. e Hedibert F. Lopes. *Árvore de Decisão - Regiões Retangulares, Insper*. <http://hedibert.org/wp-content/uploads/2018/06/CART.pdf>. Accessed: 2023-06-24. 2017.
- [6] S.J. Leon. *Linear Algebra with Applications*. Prentice Hall, 1998.
- [7] Vladimir G. Pestov. *Elementos da teoria de aprendizagem de máquina supervisionada*. Rio de Janeiro: Instituto Nacional de Matemática Pura e Aplicada (IMPA), 2019, pp. 1–476. Disp. em:

---

[https://impa.br/wp-content/uploads/2022/03/32CBM07\\_eBook.pdf](https://impa.br/wp-content/uploads/2022/03/32CBM07_eBook.pdf).

- [8] Rico Blaser e Piot Fryzlewicz. “Random Rotation Ensembles”. Em: *Journal of Machine Learning Research* 17 (2016), pp. 1–26.