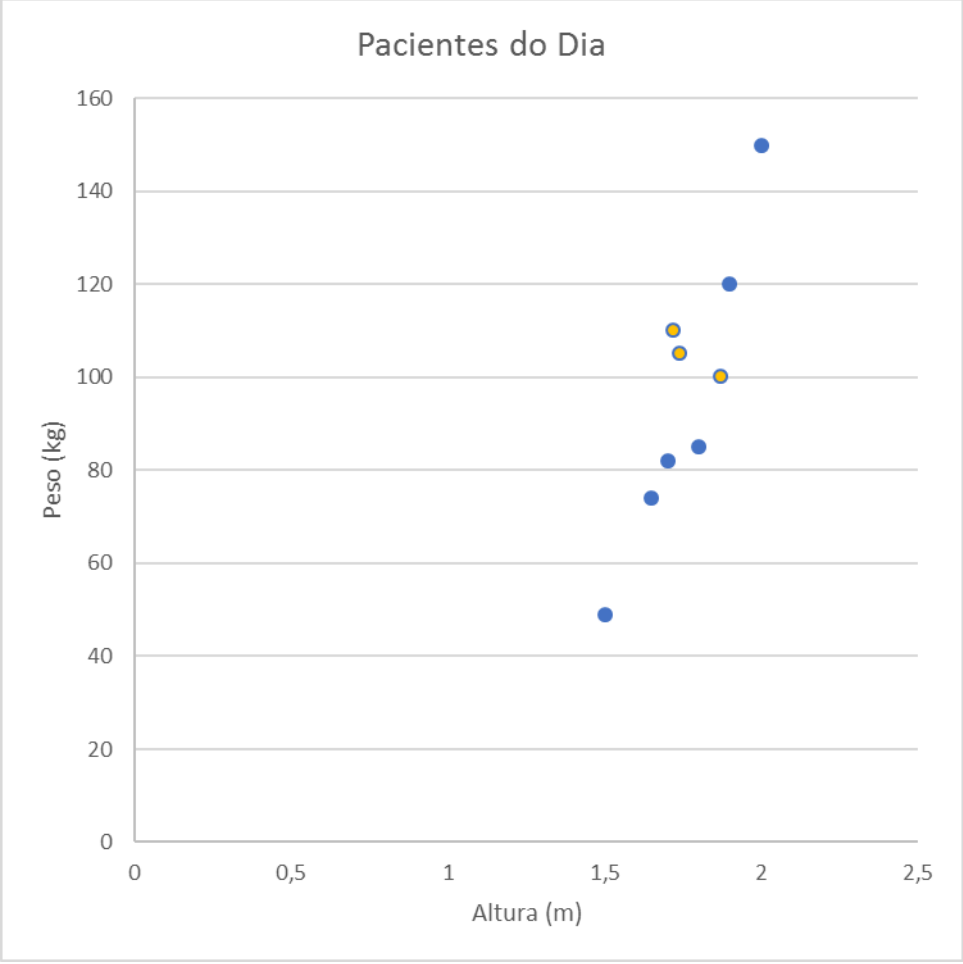
Engenharia e seleção de características em bases de dados

Normalização

Quando fala-se em *normalização* de dados geralmente deseja-se, como finalidade, garantir que todas as informações (no caso, os *atributos* ou *colunas*) obedecem a mesma escala.

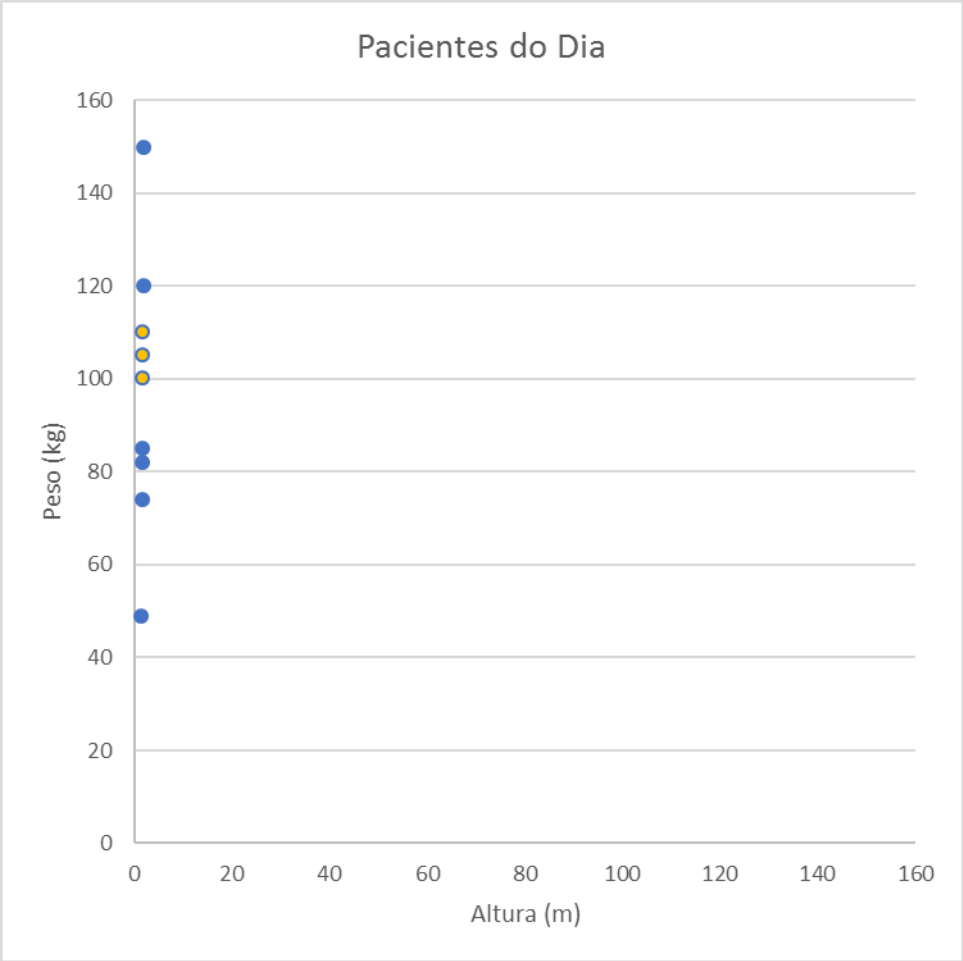
Vamos primeiro fazer um pequeno exercício mental: suponhamos que queiramos trabalhar com uma base de dados de pacientes que passam por um hospital. Veja a amostra abaixo com dois atributos: o peso, em quilogramas, e a altura, em metros de alguns deles:



Como pode ver, cada ponto representa um paciente. Preste atenção no máximo e no mínimo de cada um dos eixos: o eixo x (da altura) vai de 0 até 2.5, enquanto que o eixo y (do peso) vai de 0 até 160. Peguei como exemplo três pacientes que estão muito próximos tanto em peso como em altura (ao menos

seriam os dois mais próximos considerando este gráfico sob um ponto de vista visual) e pintei-os de amarelo.

Além disso, preste atenção nesta informação em específico: os pontos máximos dos dois eixos são 2.5 e 160, respectivamente. Meio desigual, não? Vamos ver o mesmo gráfico, mas agora com ambos os eixos limitados a 160:



Analisando novamente sob um ponto de vista visual veja que aqueles três pontos em amarelo não são mais os mais próximos um do outro, existindo outros casos. Agora, coloque-se no lugar de um algoritmo que deveria escolher os pontos mais próximos uns dos outros: a tomada de decisão mudaria conforme a forma na qual nós (e o algoritmo) visualizaríamos esses dados, não?

Para tal, uma das formas de garantirmos que os algoritmos tomariam as tomadas de decisões mais “honestas” quanto possível seria se normalizássemos os dados. Para tal, uma das estratégias seria o seguinte:

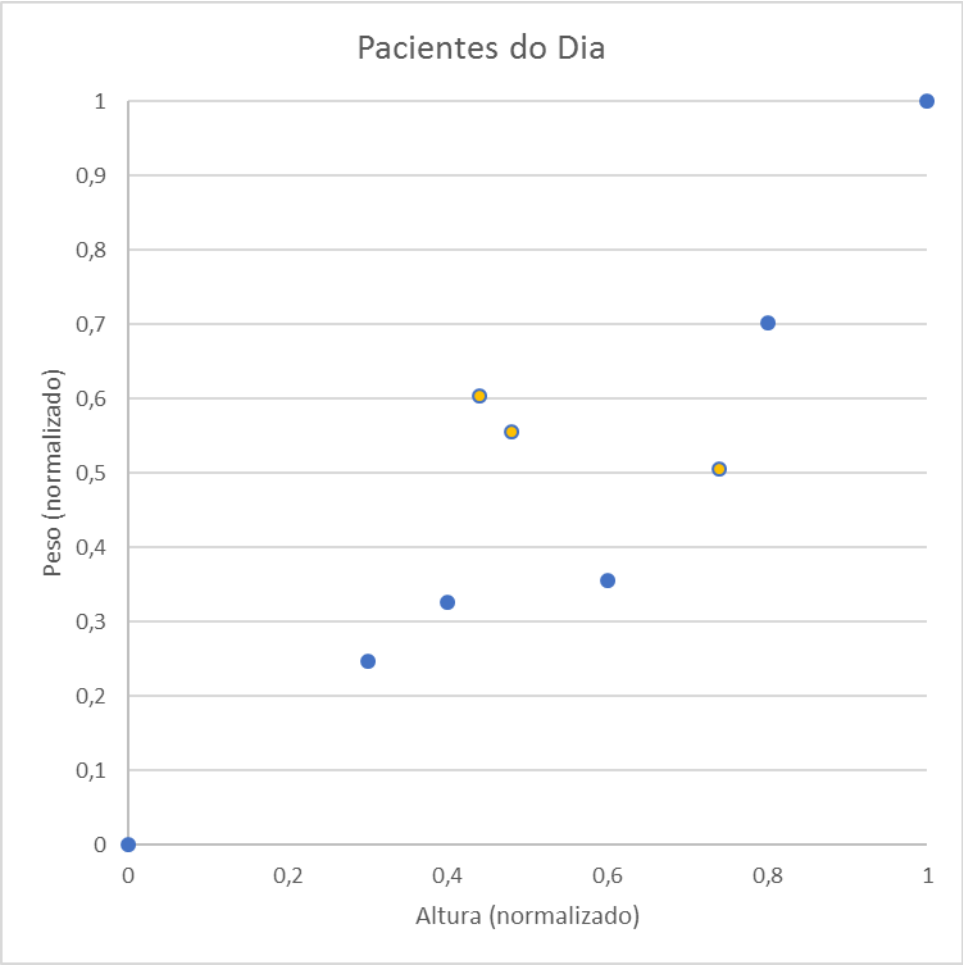
$$x_{novo} = \frac{x - x_{mínimo}}{x_{máximo} - x_{mínimo}}$$

Como isto funcionaria? Peguemos os pesos, por exemplo: o menor dos pesos registrados ($x_{mínimo}$) é igual a 49kg. O maior dos pesos registrados ($x_{máximo}$) é 150kg. Assim, a equação para os peso, para qualquer peso x, seria:

$$x_{novo} = \frac{x - 49}{150 - 49} = \frac{x - 49}{101}$$

Faça alguns testes e veja quanto daria x_{novo} se x fosse igual a 80kg, ou a 110kg, por exemplo. Veja que qualquer peso entre 49kg e 150kg estaria entre 0 e 1. O mesmo ocorreria com a altura e qualquer outra medida – em outras palavras, a **normalização garante que todos os valores numéricos estejam na faixa de números entre 0 e 1**.

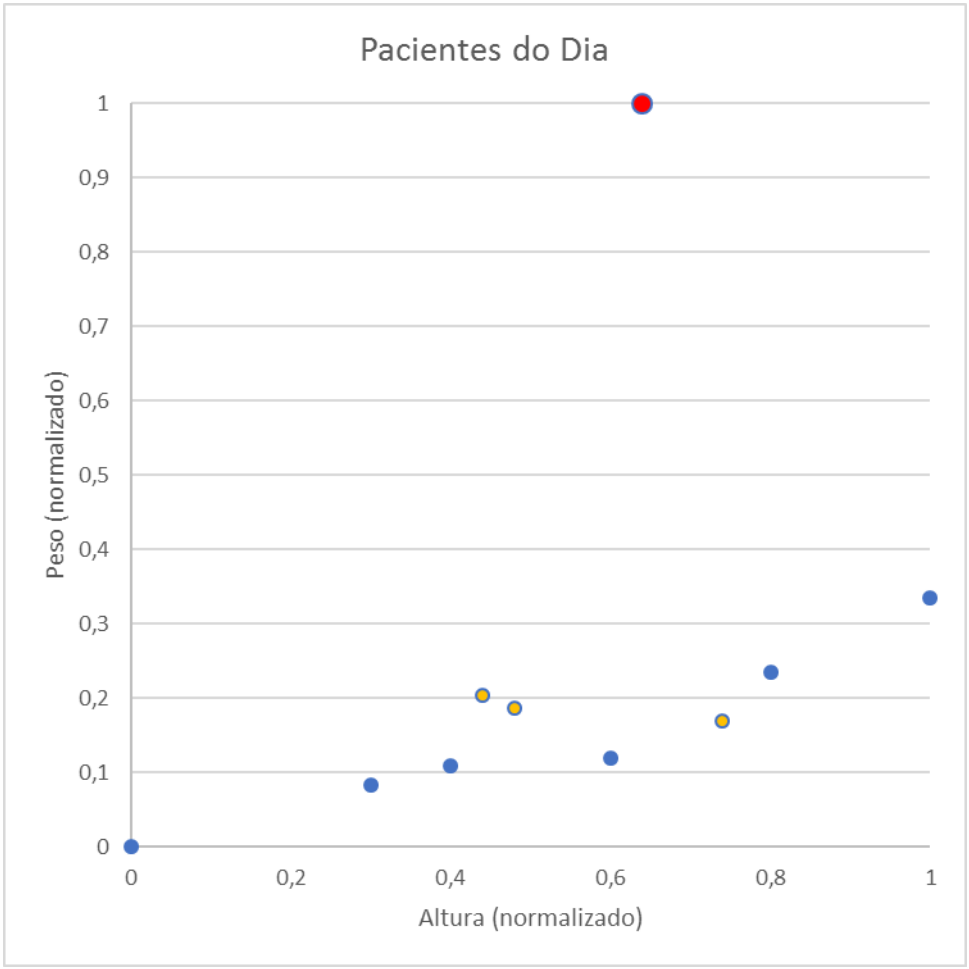
Vamos ver como fica o nosso gráfico, mas agora devidamente normalizado?



Veja onde estão os pontos em amarelo agora. Consegue perceber que seria possível criar agrupamentos “mais honestos” com as informações normalizadas do que com esses três pontos que escolhemos lá no começo?

Perceba que o comportamento dos algoritmos buscam, em muitas vezes, trabalhar com conceitos humanos e visuais – isto é, da mesma forma que nós procuramos encontrar comportamentos e padrões nos dados em um gráfico ou em uma tabela usando os nossos olhos, a mesma coisa acontece muitas vezes com um algoritmo. Por esta razão, note que a normalização é importante para ter resultados mais precisos. Por outro lado, cuidado: caso o seu dado possua muitos *outliers* – isto é, valores muito fora da realidade (como, por exemplo, um paciente com 350kg), a normalização poderá deixar em intervalos muito próximos os demais valores, o que acabará atrapalhando o desempenho (em outras palavras, a taxa de acerto) do seu algoritmo ao invés de ajudar. Vou adicionar este paciente com 350kg junto com

os demais (em vermelho) e normalizar os dados para você ver o resultado e tomar as suas próprias conclusões. Veja:



Reduzindo a dimensionalidade

Um dos conjuntos de dados mais clássicos é o das flores iris (https://archive.ics.uci.edu/ml/datasets/iris). Ele é composto por quatro atributos: o comprimento e largura das sépalas e o comprimento e largura das pétalas. Dependendo da combinação desses quatro atributos nós (e os algoritmos) conseguiriam saber qual dos três tipos de flor cada uma das linhas do conjunto de dados representa.

Pense em você mesmo: como você conseguiria diferenciar uma flor de outra? Como você consegue diferenciar, por exemplo, uma margarida de uma camomila e uma camomila de um girassol? Ora, você provavelmente toma a sua decisão em base do tamanho da flor, da cor, do formato das pétalas, entre outros. Com um algoritmo é a mesma coisa, e esse banco das flores iris é bem simples para entendermos o conceito.

Mas, e se estivermos trabalhando com dados reais de clientes de uma seguradora para sabermos quem pertence a uma ou outra faixa de risco, por exemplo? *O que define se uma pessoa deve pertencer a determinada faixa enquanto que a outra não?* A idade? O sexo? A renda? Uma combinação de tudo isso? O estado civil? Será que o tempo de empresa interfere mais do que a renda? Será que o endereço onde mora interfere mais do que a idade? Será que o nome da pessoa importa? Será que o número de calçado que a pessoa usa importa?

Veja que, nesse simples exercício mental, você já consegue admitir certas informações: *alguns atributos possuem mais peso do que outros, e* ***alguns atributos não influenciam em nada na decisão***. Ora, se não influenciam em nada, eles merecem ser removidos para facilitar a tomada de decisão, tal qual um filtro, não é?

Esta técnica se chama **seleção de atributos**, cuja intenção é remover aqueles atributos que não são importantes na nossa tomada de decisão o que, por sua vez, culmina na **redução da dimensionalidade**. No exemplo da seguradora,

imagine que são 100 mil clientes (ou instâncias ou, ainda, “linhas” em uma planilha do Excel) com 200 características (ou colunas ou, ainda, “colunas” na planilha). 100 mil linhas multiplicadas por 200 colunas equivalem a **20 milhões de valores**! Agora, se das 200 características, descobríssemos que somente 10 seriam realmente relevantes, teríamos **1 milhão de valores** para testarmos, ou, em outras palavras, **20 vezes menos dados** para concentrarmos os nossos esforços. Melhor, certo?

Ainda que vejamos mais sobre isto ainda nesta disciplina, é importante que entenda desde já a importância da seleção de atributos no contexto de machine learning – certamente, não nos interessa aplicar técnicas de redução de dimensionalidade em todo e qualquer conjunto de dados (vide o caso das flores iris, onde já temos tudo o que precisamos no conjunto e, se removermos alguma daquelas informações, corremos o risco de tomarmos a decisão errada), mas principalmente naqueles conjuntos onde:

1. Temos um grande volume de informações;
2. Temos atributos (colunas) os quais não influenciam (e/ou podem atrapalhar ao invés de ajudar) na tomada de decisão.

SAIBA MAIS

Kaggle. Bases de dados para testes de diferentes tipos e complexidades. Disponível em: <https://www.kaggle.com/datasets>

Minerando dados. um exemplo simples de normalização usando o scikit-learn no Python. Disponível em: <http://minerandodados.com.br/index.php/2017/12/28/pre-processamento-standartization/>

Data Science. Veja o gráfico de exemplo de um conjunto de dados não-normalizado e o mesmo conjunto normalizado utilizando estratégias diferentes do scikit-learn no Python (StandardScaler, MinMaxScaler, RobustScaler e Normalizer). Disponível em: <https://python-data-science.readthedocs.io/en/latest/normalisation.html>

Scikit-learn - Machine Learning in Python. Este link fornece mais exemplos quanto à normalização de dados segundo o exemplo visual mostrado no link anterior. Disponível em: <https://scikit-learn.org/stable/modules/preprocessing.html>

PUCPR

GRUPO MARISTA

PROFESSOR CONTEUDISTA

Wellington Rodrigo Monteiro