APRENDIZADO SUPERVISIONADO E NÃO SUPERVISIONADO

Nesse artigo vamos falar sobre uma das maiores buzzwords do momento e finalmente falaremos de Machine Learning. Como Data Scientist, é possível imaginar que já existe uma ansiedade para começar a ensinar as máquinas a fazer todo o trabalho simples e repetitivo enquanto se concentram no que realmente importa.

MAS PRIMEIRAMENTE... O QUE É MACHINE LEARNING?

Se você procurar uma definição sobre Machine Learning, provavelmente vai se deparar com termos como “algoritmo”, “inteligência artificial” e, o mais importante e encantador, “sem ser explicitamente programado”. Em síntese, podemos afirmar que Machine Learning é:

> O ESTUDO, A CIÊNCIA OU A APLICAÇÃO DE ALGORITMOS DE COMPUTADOR QUE TÊM A HABILIDADE DE APRENDER E MELHORAR SUAS PRÓPRIAS PERFORMANCES AUTOMATICAMENTE, A PARTIR DAS EXPERIÊNCIAS, SEM SEREM EXPLICITAMENTE PROGRAMADOS.

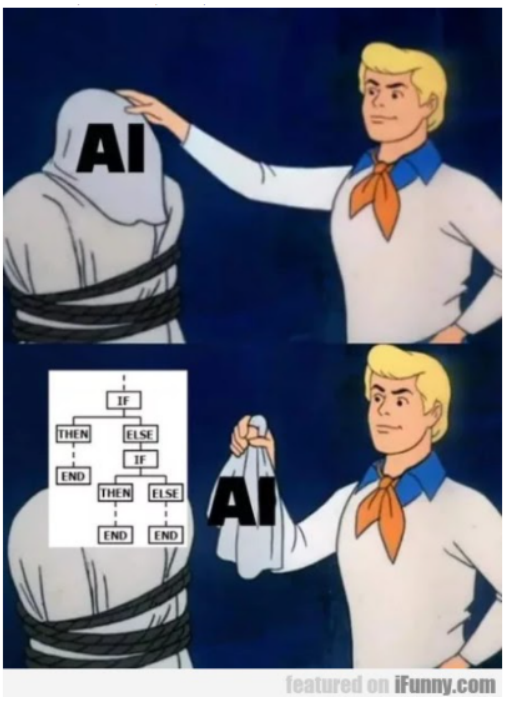
Existem três coisas que precisam de uma atenção extra nessa definição. Confira:

APRENDIZADO SUPERVISIONADO E NÃO-SUPERVISIONADO

aplicação de conceitos estatísticos em uma série de códigos e parâmetros. Em outras palavras, pense em ML como um atalho. Imagine se você tivesse que pensar cada conceito estatístico, traduzir para linguagem de código, testar e validar o código, para só então aplicar aos seus dados…

Os processos de treinamento ou aprendizado em machine learning fazem exatamente isso: analisam e processam o data set, criando um programa ou código específico que poderá ser aplicado a um novo conjunto de dados com as mesmas formas e/ou características.

Por isso, nunca esqueça: embora você não tenha programado linha por linha, é isso que temos por trás de cada resultado:



“À PARTIR DAS EXPERIÊNCIAS”

Enquanto os conceitos básicos de programação visam a construção de algoritmos e aplicações a partir de um conceito hierárquico, linear e, como o próprio nome sugere, programado, as aplicações de Machine Learning são construídas por meio de input e output. Em uma analogia com a Fórmula1, é como se, ao invés de escolher o tipo de pneu antes da corrida, o carro, após as primeiras voltas, pudesse criar o pneu ideal para a corrida de acordo com os dados do asfalto.

“A HABILIDADE DE APRENDER”

APRENDIZADO SUPERVISIONADO E NÃO-SUPERVISIONADO

como faremos com que os algoritmos aprendam, quais os tipos de output, que tipo de reshape precisamos dar aos nossos dados e a intuição por trás de como cada algoritmo foi pensado.

AVANÇAR

CATEGORIAS

Quando nos referimos aos tipos de aprendizado de máquina, estamos basicamente falando de duas categorias: **supervisionado** e **não-supervisionado**.

Esses termos são relativamente contra-intuitivos, uma vez que não se referem ao **nosso papel** em relação ao aprendizado e sim ao **tipo de autonomia que o algoritmo tem para criar os cálculos e chegar aos resultados.**

> EM TERMOS PRÁTICOS, A DIFERENÇA ENTRE APRENDIZADO SUPERVISIONADO E NÃO SUPERVISIONADO É SE OS DADOS POSSUEM OU NÃO POSSUEM RÓTULOS.

Este conceito é relativamente abstrato, mas pensem nos rótulos ou labels como a capacidade de direcionar o aprendizado a partir de um objetivo específico como “comprou” ou “não comprou”, tem uma doença ou não, a qual espécie uma observação pertence, qual o preço teórico de um bem (dado tais características),etc. Já para os dados sem rótulos, o aprendizado é mais automático, buscando similaridades e anomalias como perfis ou clusters de consumo, mau funcionamento, compras suspeitas.

Ter ou não ter labels nos seus dados, não é um detalhe ou algo insignificante. É uma dúvida existencial em Machine Learning. Vai determinar como coletar, modelar, treinar e colocar o modelo em produção. Não é uma escolha fácil, mas vamos te ajudar a fazer uma escolha mais consciente.

APRENDIZADO SUPERVISIONADO E NÃO-SUPERVISIONADO

AVANÇAR

APRENDIZADO SUPERVISIONADO

Já vimos que a presença ou não de rótulos nos dados é o que diferencia o aprendizado supervisionado do não supervisionado.

MAS O QUE É ISSO NA PRÁTICA?

Vamos olhar para o dataset **Pima Indians Diabetes**, disponível como desafio no Kaggle.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Apesar de estarem em um único csv, em cada linha ou observação, temos dois tipos de dados: as características ou features e um rótulo ou label. Nem sempre eles vão ser chamados features e labels, mas sempre que estamos falando de aprendizado supervisionado, nós teremos essa lógica. Perceba abaixo algumas das principais formas nas quais se convencionou chamar estes dois tipos de dados:

Features

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	6	148	72	35	0	33.6	0.627	50
1	1	85	66	29	0	26.6	0.351	31
2	8	183	64	0	0	23.3	0.672	32
3	1	89	66	23	94	28.1	0.167	21
4	0	137	40	35	168	43.1	2.288	33

Labels

Outcome
1
0
1
0
1

Features
=
Predictor variables
=
Independent variables

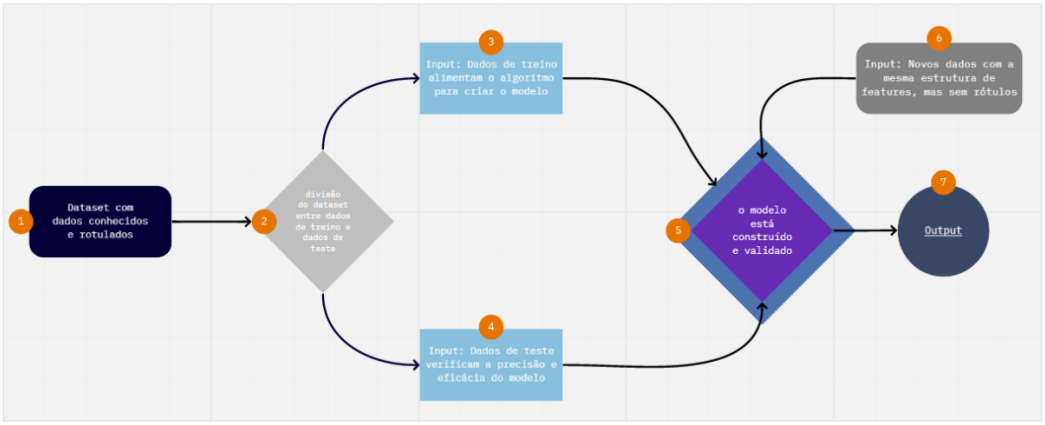
Target
=
Response variable
=
Dependent variable

PROCESSO DE APRENDIZADO SUPERVISIONADO

No aprendizado supervisionado nós direcionamos o aprendizado ao informar ao algoritmo que determinadas features significam um determinado valor. Dessa forma ele precisa aprender o porquê. Qual a relação entre as features que fazem seu resultado ser aquele. Mas isso não é tudo. Uma vez que colocamos os dados e treinamos o algoritmo, precisamos testar esse

É nesse momento que descobrimos o quão **preciso** (e não “assertivo”, que nesse contexto quer dizer “com ênfase” e não “com precisão”) é o nosso modelo e está pronto para receber dados novo, sem labels, e gerar os outputs desejados.

Representando todo este processo em um fluxograma:



Fluxo do processo de treinamento para modelos supervisionados

OUTPUTS

Os resultados dos modelos de ML com aprendizado supervisionado estão relacionados diretamente com o tipo de target que desejamos obter.

TARGET COM DADOS DISCRETOS

Se o target é uma variável discreta (como no exemplo do dataset Pima Indians Diabetes, visto acima) temos um modelo de classificação, em que o algoritmo vai calcular e classificar, a partir dos features, se o nosso target tem ou não diabetes. É importante dizer que, apesar desse modelo ser binário, com duas variáveis, os modelos de classificação podem ter 3 ou mais classes. Este é o caso do famoso Iris Dataset, que possui dados de três tipos de flores iris: iris setosa, iris versicolor e iris virginica, com features de comprimento e largura das pétalas e das sépalas das três flores.

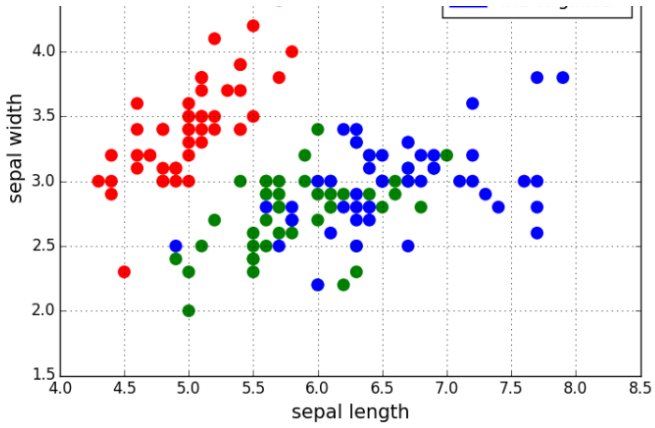


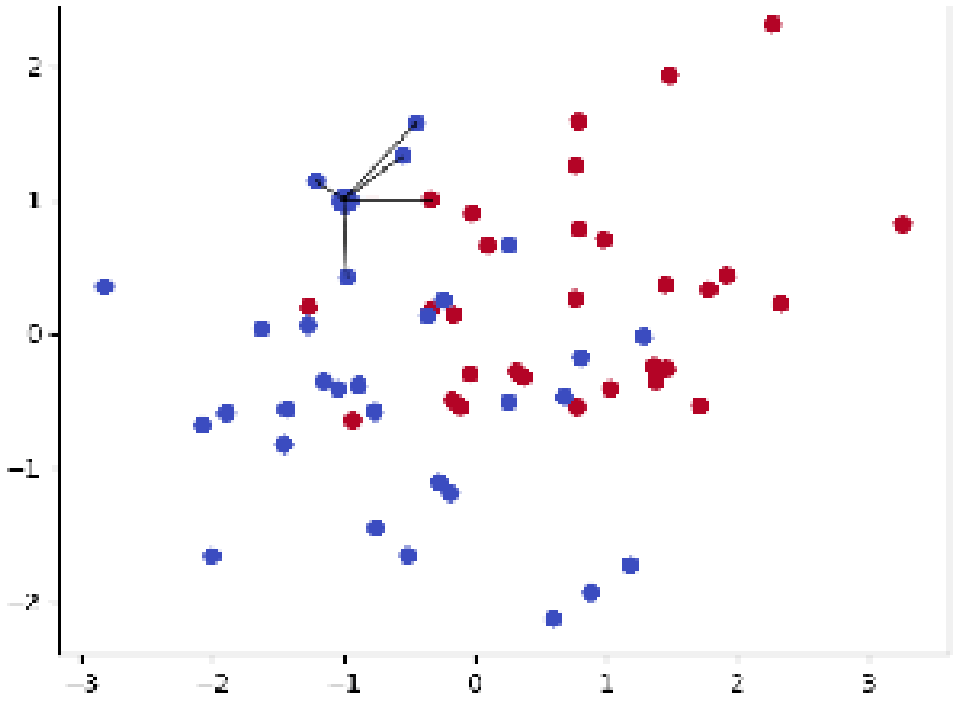
Imagem via: Pybloggers.com

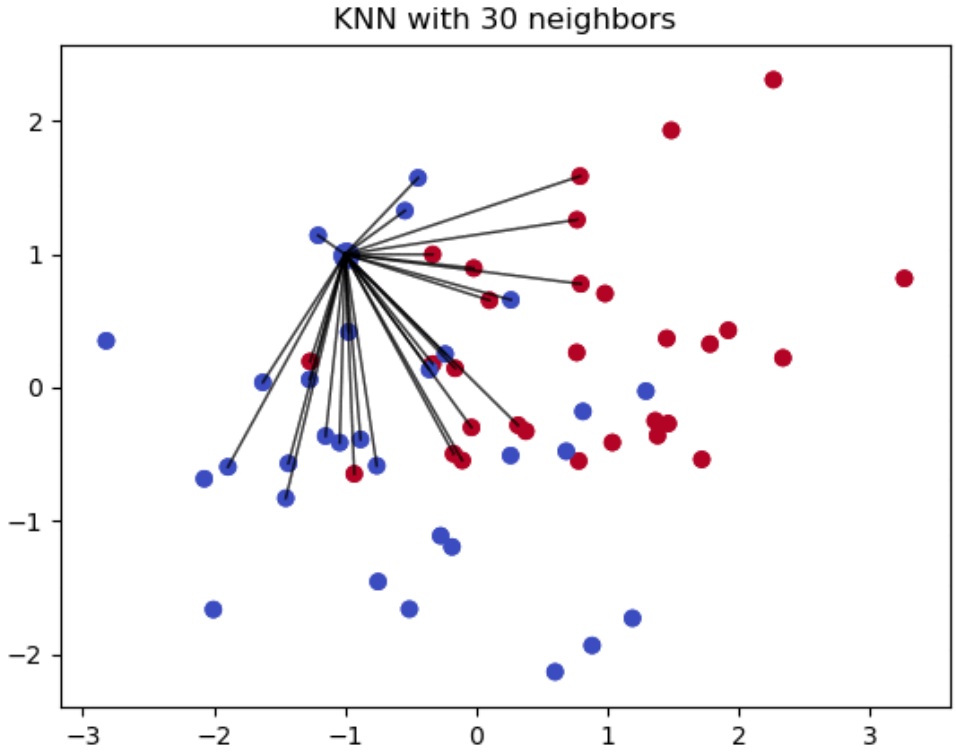
EXEMPLO DE ALGORITMO DE CLASSIFICAÇÃO (KNN)

Um dos mais clássicos algoritmos de classificação é o K-Nearest Neighbours. Esse é um algoritmo de votação. Ele distribui os dados em espaço n-dimensional (de acordo com o número de features) e, através de um um processo de proximidade, os novos dados são arranjados por similaridade. Através dos hiperparâmetros, personalizamos a quantidade necessária de votos para determinar a qual classe pertencem os novos dados.

No exemplo a seguir temos os dados novos entrando e sendo classificados a partir de dois cenários: de acordo com os 5 dados mais próximos e de acordo com os 30 dados mais próximos.

APRENDIZADO SUPERVISIONADO E NÃO-SUPERVISIONADO





Imagens retiradas do blog ***limportq.wordpress.com***

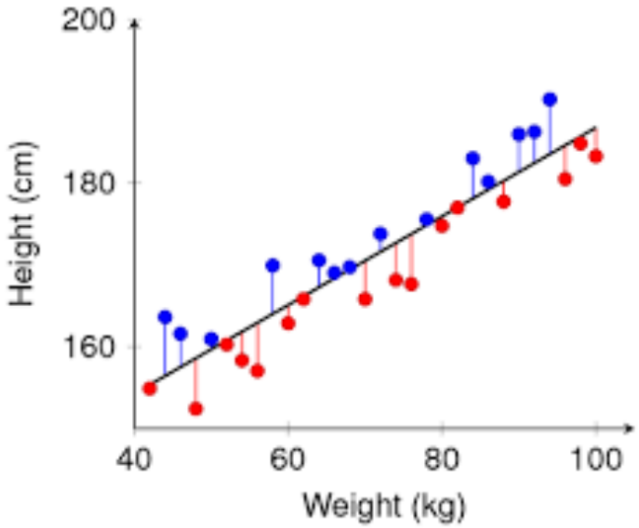
TARGET COM DADOS CONTÍNUOS

APRENDIZADO SUPERVISIONADO E NÃO-SUPERVISIONADO

mais impactam naquele valor e traçar uma regressão linear que vai ser capaz de prever o impacto de uma ou mais features em dados novos (sem rótulo).

Imagem via: Anoop Singh

Essa é a intuição por trás dos modelos lineares, mas a verdade é que essa versão mais simples está muito sujeita a underfitting (quando a complexidade capaz de ser interpretada pelo modelo é menor do que a complexidade intrínseca do conjunto de dados, produzindo predições com alto viés e comprometendo a sua aplicabilidade em novos dados.). Por isso, existem alguns algoritmos mais complexos, que funcionam como uma camada de complexidade, e permitem que o modelo consiga fazer previsões corretas mesmo para pontos mais distantes da linha.



TUDO BEM ATÉ AQUI?

APRENDIZADO SUPERVISIONADO E NÃO-SUPERVISIONADO

APRENDIZADO NÃO-SUPERVISIONADO

Calma! Não é bruxaria… mas, diferente do aprendizado supervisionado, os algoritmos de Machine Learning com aprendizado não supervisionados não precisam de rótulos. Se no aprendizado supervisionado nós damos uma espécie de tarefas para os algoritmos, no aprendizado não supervisionado nós permitimos que os algoritmos busquem padrões, similaridades e anomalias nos dados de forma autônoma.

OUTPUTS

O fato de não termos dados rotulados, torna o tipo de output dos modelos não supervisionados um pouco menos versátil. Os dois tipos de output mais usados a partir desse tipo de aprendizado são: **clusterização** e **redução de dimensionalidade.**

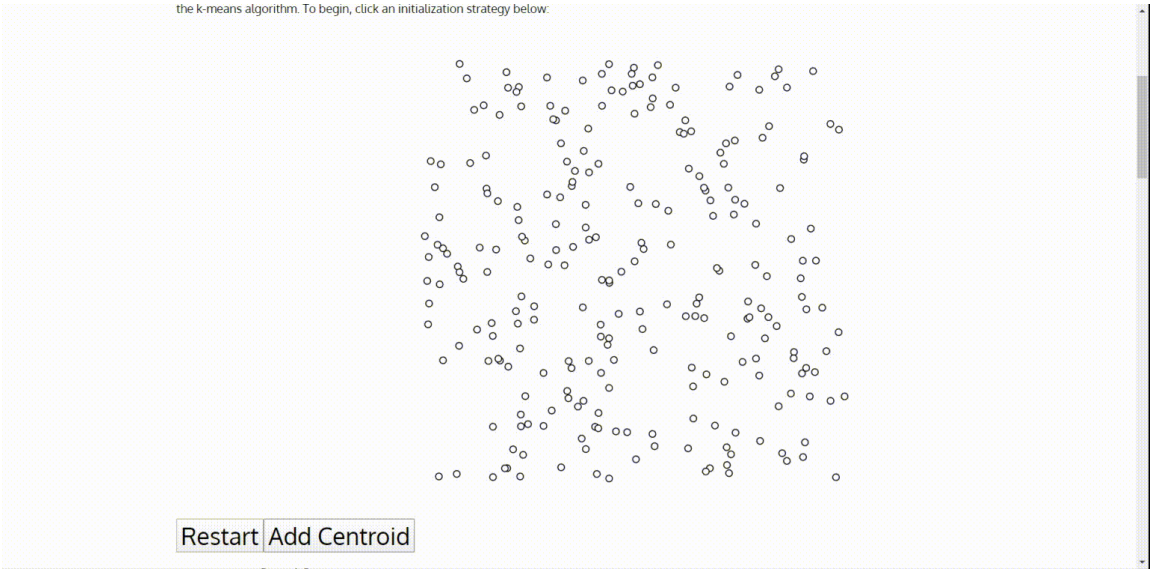
APRENDIZADO SUPERVISIONADO E NÃO-SUPERVISIONADO

É o agrupamento de dados por similaridade. Não é muito difícil ter uma intuição sobre o que é clusterização a partir de dados, mas ela se dá de diferentes formas, dependendo do algoritmo usado.

Para ilustrar os diferentes processos de clusterização, vou trazer três dos mais populares algoritmos não-supervisionados que realizam essa operação.

a) K-Means

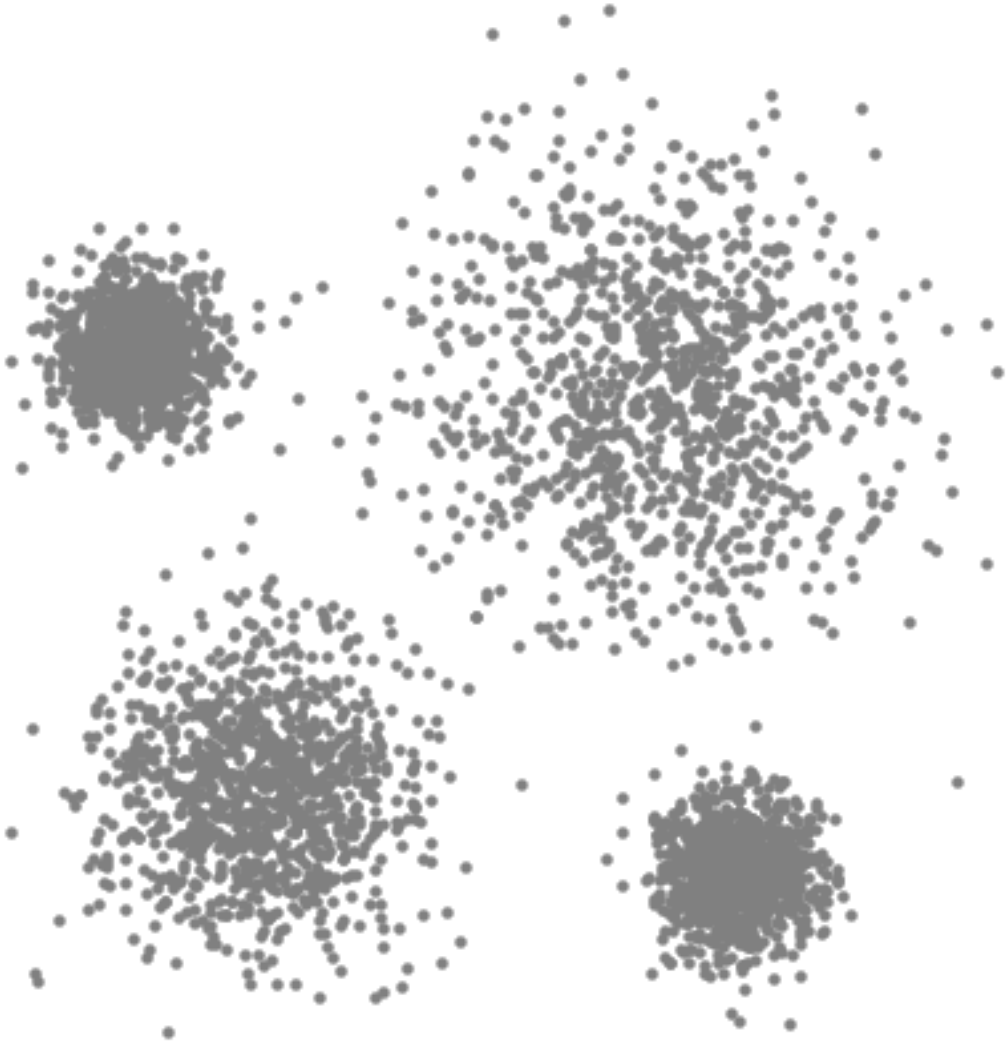
Nós determinamos a quantidade de clusters que o algoritmo deve encontrar nos dados. De forma randômica, ele inicializa a quantidade de center points igual ao número de clusters que determinamos. Através de cálculos vetoriais, **cada center point busca pelo centro do grupo através da média das distâncias de todos os pontos do grupo**. O cálculo para quando os centros dos grupos não se alteram significativamente. Como é um cálculo randômico, os resultados podem ser diferentes de acordo com os pontos iniciais, então pode-se repetir tudo o processo algumas vezes e testar seus resultados. O gif abaixo ilustra o funcionamento do K-Means:



*Simulador de **K-Means***

b) Mean-Shift

Ao contrário do K-Means, este algoritmo não precisa que seja dito a ele quantos clusters procurar. Para iniciar, ele transforma a concentração dos dados em uma topografia. Quanto maior a densidade mais alto é o pico. A busca do Mean-Shift é sempre pelos picos de concentração de dados. Ou



*Simulação do Mean-Shift em ação. Retirado do site **primo.ai***

c) (DBSCAN) Density-Based Spatial Clustering of Applications with Noise

Em português: Clusterização Espacial Baseada em Densidade de Aplicações com Ruído

Esse algoritmo inicia em um ponto qualquer a sua busca e classifica todos os data points por proximidade, de acordo com um parâmetro de minPoints (mínimo de pontos de dados ou vizinhos) e com o epsilon ou o tamanho de sua área de busca (distância máxima alcançada). Uma vez que não consegue dar sequência em sua trilha, ele recomeça sua busca em outro ponto aleatório. Os pontos que não estiverem próximos o suficiente ou em concentração suficiente, são considerados ruídos. Esse algoritmo é

APRENDIZADO SUPERVISIONADO E NÃO-SUPERVISIONADO

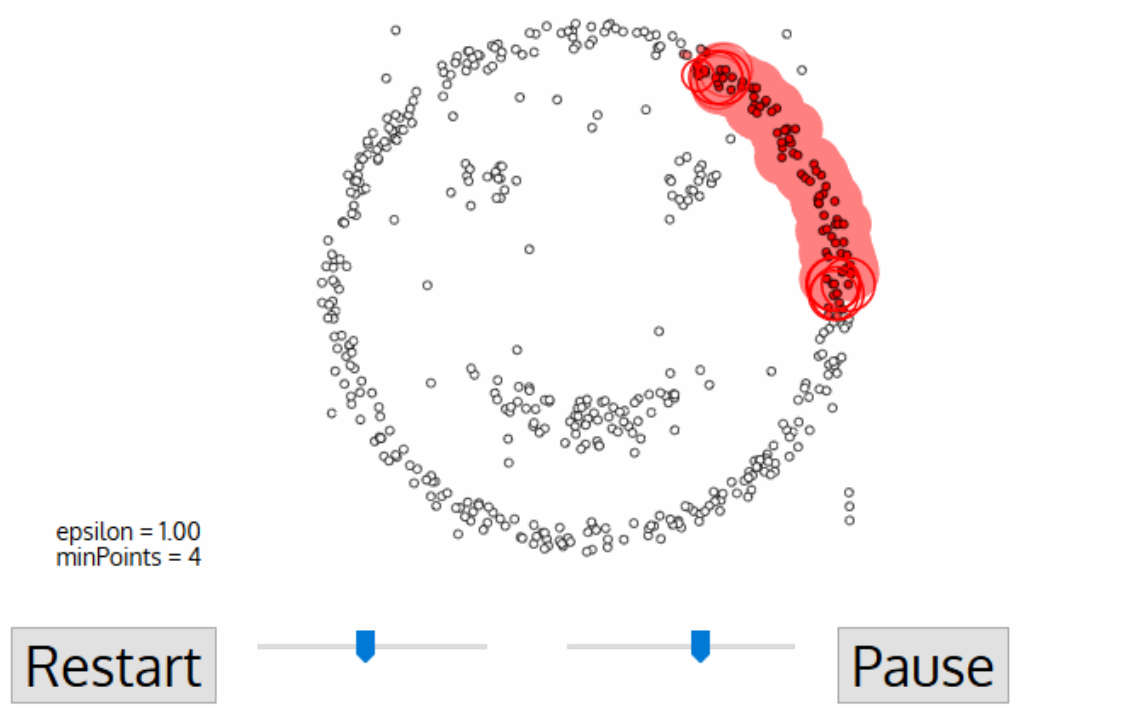


Imagem do blog ***naftaliharris.com/blog***

2. REDUÇÃO DE DIMENSIONALIDADE

Para entender o que é redução de dimensionalidade e porque ela é importante, é preciso deixar muito claro o conceito de dimensionalidade dos dados. Cada nova coluna ou feature de uma observação adiciona uma nova dimensão aos dados. Se, em um dataset de dados de jogadores profissionais de futebol, tivermos como features Altura, Peso, Idade, Gols Marcados, Velocidade e Condicionamento Físico, por exemplo, estaríamos falando de um dataset com 6 dimensões.

Computar e interpretar dados com muitas dimensões é uma problema. Primeiro porque o grau de complexidade aumenta exponencialmente a cada nova dimensão e isto implica tanto em maior necessidade de poder computacional quanto em dificuldades na interpretabilidade. Segundo porque dificilmente as features são puras, ou seja, possuem co-variância entre si, como a velocidade e o condicionamento físico, por exemplo.

Além de co-variância, elas podem conter **multicolinearidade,** que é quando uma ou mais features variam juntas, mas existem outros elementos que não estão no dataset que capturariam este fator. Um exemplo é o consumo de calorias e o metabolismo. Por último, mas não menos importante, muitas features aumentam muito o risco de overfitting do modelo.

APRENDIZADO SUPERVISIONADO E NÃO-SUPERVISIONADO

casos isso pode ser uma boa saída. Mas quando não temos certeza, muitas dessas colunas podem conter informações relevantes. Na maioria dos casos, vamos querer capturar toda informação relevante das colunas, ou fazer uma **feature extraction**.

AVANÇAR

APRENDIZADO SUPERVISIONADO E NÃO-SUPERVISIONADO

PCA - PRINCIPAL COMPONENT ANALYSIS

Em português: Análise de Componente Principal

O **PCA** é um dos principais algoritmos de extração de features. Basicamente, ele vai plotar todos os dados em um espaço n-dimensional (de acordo com a quantidade de features) e rotacionar os dados até encontrar um ângulo em que consiga projetar e capturar a maior variância possível dos dados e a menor distância ao quadrado dos resíduos em um espaço linear (não confundir com uma regressão linear).

Seria mais ou menos isso:

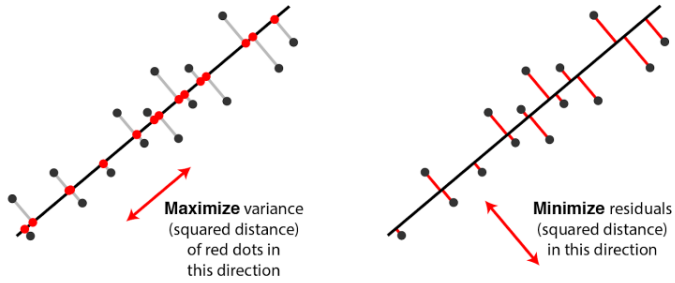


Imagem via artigo **Everything you did and didn't know about PCA**

Everything you did and didn't know about PCA

PCA OPTIMIZATION

O objetivo de fazer isso é compactar a redundância dos dados mais densos sem excluir informações relevantes, porém mais dispersas.

Para saber mais, confira o artigo A tutorial on principal component analysis, de Jon Shlens.

APRENDIZADO SUPERVISIONADO E NÃO-SUPERVISIONADO

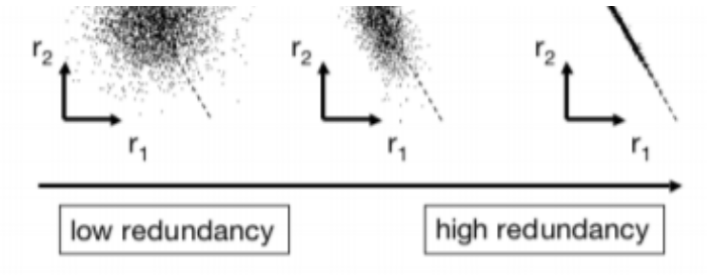
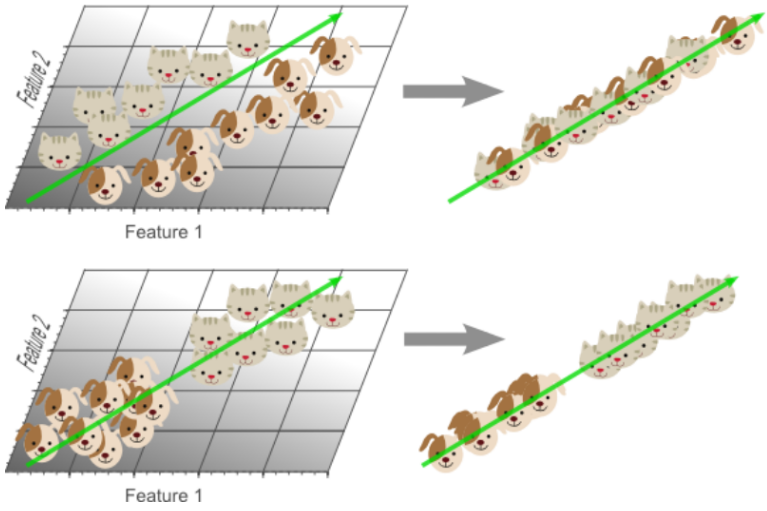


Imagem via artigo A tutorial on principal component analysis.

No final desse processo, a PCA extrai os features de todos os dados e os armazena no espaço linear PC1 ou Principal Component 1. Então ele está pronto para continuar a buscar uma PC2, PC3, PC4… mas, como o algoritmo tenta extrair o máximo de informação de todos os dados, dificilmente haverá informações relevantes acima da PC3. O que é excelente em termos computacionais e de performance para os modelos a serem treinados.

É importante perceber que as novas PCs ou eixos ortogonais capturam as informações de vários features, fazendo uma verdadeira fusão dessas informações:



Exemplo de como funciona a extração de features com PCA. Imagem via: visiondummy.com

visiondummy.com

Para deixar um último exemplo, imagine que temos uma dataset de Pokemóns com 4 dimensões e você precisa clusteriza-lo. Sem reduzir a dimensionalidade vai ser muito difícil você completar essa tarefa. Mas com uma dispersão em 2D fica bem mais rápido e simples para os algoritmos de custerização.

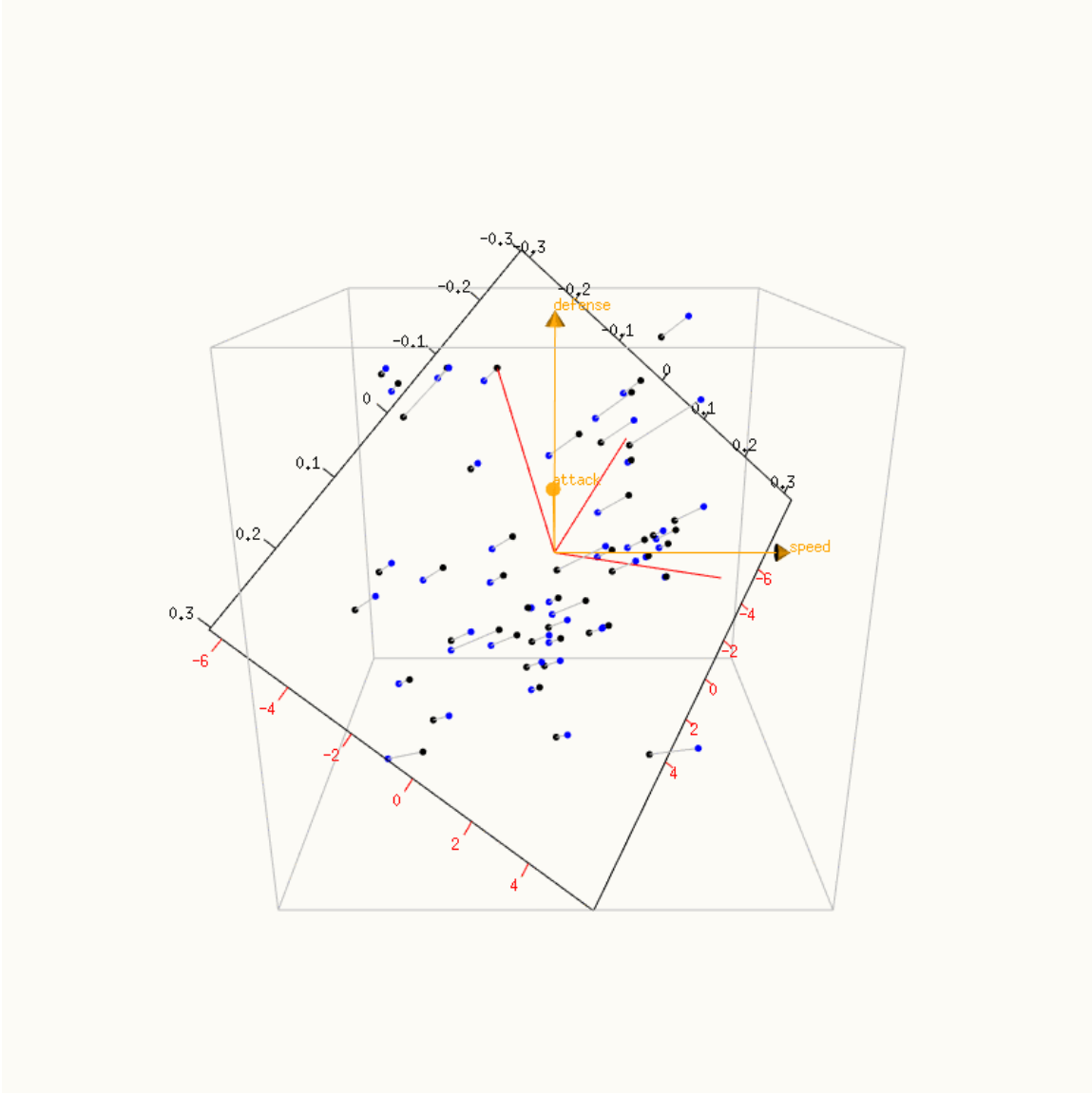
O dataset:

APRENDIZADO SUPERVISIONADO E NÃO-SUPERVISIONADO

Chandelure	60	55	90	80
Helioptile	44	38	33	70
MeloettaAria Forme	100	77	77	90
MetagrossMega Metagross	80	145	150	110
Sawsbuck	80	100	70	95
Probopass	60	55	145	40
GiratinaAltered Forme	150	100	120	90
Tranquill	62	77	62	65
Simisage	75	98	63	101
Scizor	70	130	100	65
Jigglypuff	115	45	20	20
Carracosta	74	108	133	32
Ferrothorn	74	94	131	20
Kadabra	40	35	30	105
Sylveon	95	65	65	60
Golem	80	120	130	45
Magnemite	25	35	70	45
Vanillish	51	65	65	59
Unown	48	72	48	48
Snivy	45	45	55	63
Tynamo	35	55	40	60
Duskull	20	40	90	25
Beautifly	60	70	50	65

Banco de dados de Pokémons com 4 dimensões

O PCA em ação (PC1 e PC2 sobre eixos 3D):



PCA1, PCA2, resíduos em uma projeção relativa a 3Dimenões (attack, defense, speed)

APRENDIZADO SUPERVISIONADO E NÃO-SUPERVISIONADO

Para saber mais sobre PCA, assista o vídeo Principal Component Analysis (PCA)com o passo a passo (22 min).

21:57

AVANÇAR

CONCLUSÃO

Nesse texto abordamos conceitos que nos ajudam a tangibilizar os tipos de aprendizado de máquina e nos dão uma intuição sobre como funcionam e de que forma profissionais de data science podem usá-los.

Em um breve resumo, vimos que:

Os algoritimos de Machine Learning se dividem em dois tipos de aprendizado, caracterizados pela presença ou não de rótulos. A partir daí vimos que os tipos de tarefas/tasks que eles são capazes de realizar também variam e, por último, vimos exemplos de algoritmos para cada task.

Relação entre aprendizado supervisionado e não-supervisionado		
Supervisionado	X	**Não-supervisionado**
SIM	Presença de rótulos	**NÃO**
Classificação/ Regressão	Tasks	Clusterização/ Red. Dimensionalidade
KNN/Regressão Linear	Algorítmos	K-Means/MeanShift/ DBscan/PCA
Train/Test/Score	Processo de Aprendizado	Automático
Classificação: Acurácia Regressão: Erros	Medidas de Performance	———
(Este quadro é apenas um resumo, exitem outros algorítmos, outras medidas de performance e outras tasks possíveis para modelos supervisionados e não supervisionados)		

Agora você já sabe como os algoritmos de ML aprendem e teve uma intuição sobre o funcionamento de alguns deles.

Esperamos que esse texto tenha sido claro e direto nos conceitos e exemplos. Mas se você ficou com alguma dúvida ou precisa compartilhar algo conosco, não deixe para depois! Tem sempre alguém do time Tera pronto para te ajudar. Nos vemos no próximo texto 😉

AVANÇAR

APRENDIZADO SUPERVISIONADO E NÃO-SUPERVISIONADO

APRENDIZADO SUPERVISIONADO E NÃO-SUPERVISIONADO

PARA SE APROFUNDAR

The Curse of Dimensionality in classification (artigo, em inglês - 8 min)

Feature extraction using PCA (CompViz) (artigo, em inglês - 10 min)

PCA: Principal Component Analysis (artigo, em inglês - 10 min)

Interpreting PCA figures in layman terms (artigo, em inglês - 6 min)

O QUE ACHOU DESTA AULA?

Deixe seu feedback para continuarmos melhorando sua experiência.

1 MIN

AVALIAR

VOLTAR PARA O CURSO