Aprendizado de Máquina

Emerson V. Rafael (98797361)

Itaú Unibanco
Ciência de Dados
3 de janeiro de 2023

Sumário

Links úteis

0.1 Kaggle

https://www.kaggle.com/

1 Capítulo I - O conceito de AM

1.1 Por que utilizar AM

Aprendizado de máquina é útil para cases no qual:

1. **Grande quantidade de dados**, onde há dificuldades de se observar padrões;
2. **Dados que modificam suas características ao longo do tempo**, seja pelo surgimento/remoção de classes ou por aumento ou dominuição de um valor;
3. **Problemas complexos**, que demandam soluções não triviais;

1.2 Exemplos de aplicações

Os exemplos de aplicações podem abranger:

✓ classificação de imagens - redes neurais convolucionais (CNNs);

✓ deteccão de tumores a partir de exames de imagens cerebrais - redes neurais convolucionais (CNNs);

✓ classificação automática de artigos de notícias - redes neurais recorrentes (RNNs), CNNs ou transformadores;

✓ sinalização automática de comentários ofensivos - redes neurais recorrentes (RNNs), CNNs ou transformadores;

✓ previsão de faturamento - regressão

✓ deteccão de fraude - classificação

1.3 Tipos de modelos

São tipos de modelos de aprendizado de máquina:

Figura 1. Distribuição Gaussiana

Aprendizagem de Máquina		
Supervisionada	**Não supervisionada**	**Reforço**
Classificação	Associação	
Regressão	Agrupamento	
	Detecção de desvios	
	Padrões sequenciais	
	Sumarização	

Fonte: Própria

Os modelos preditivos tem como objetivo realizar a previsão de algo que ainda está para ocorrer. Enquanto os modelos descritivos buscam estabelecer relações e obter insights sobre o que já ocorreu. Dois tipos de modelos preditivos bastante conhecidos são classificação e regressão.

1.3.1 Classificação

Sendo que a **classificação** é um modelo que **retorna uma classe** (que não necessariamente é binária), baseia-se em **prever a categoria de uma observação dada**. Aqui, procura-se estimar um "classificador" que gere como saída a classificação qualitativa de um dado não observado com base em dados de entrada (que abrangem observações com classificações já definidas). Entre os algoritmos mais famosos estão: Regressão logística (Logistic Regressor), Árvore de Decisão (Decision Tree Classifier), Floresta Aleatória (Random Forest), Máquina de Vetores de Suporte (Support Vector Machine - SVM).

1.3.2 Regressão

A **regressão** é um tipo de modelo supervisionado que **retorna um valor continuo**, ou seja, um valor numérico que não é discreto (0, 1, 2, ...). Um modelo famoso desse tipo é a Regressão Linear (Linear Regression), que pode ser aplicada sem ou com regularizações (Lasso, Ridge ou Elastic Net).

1.4 Tipos de aprendizado - Formas de aprendizado

1.4.1 Aprendizado supervisionado

No **aprendizado supervisionado**, são utilizados **dados rotulados no treinamento**, permitindo que algoritmos realizem o treinamento sabendo as predições reais. No teste, recebe-se um dado novo, o modelo realiza a decisão e o valor predito é então uma classe ou um valor continuo.

1.4.2 Aprendizado não supervisionado

No **aprendizado não supervisionado** os **dados de treinamneto não são rotulados**, ou seja, o **algoritmo deve aprender sem um professor**. Não supervisionado é utilizado para identificar padrões no conjunto de dados em tarefas como: clusterização, associação, detecção de anomalias e detecção de novidades.

1.4.3 Aprendizado semisupervisionado

O **aprendizado semisupervisionado** é aquele que **agrega etapa supervisionado e etapa não supervisionada**, exemplo:

O algoritmo do Google Fotos possui as etapas:

1. Ao realizar o upload para o Google Fotos, o algoritmo realiza a marcação de que há uma pessoa A nas fotos 1, 2 e 3 e pessoa B nas fotos 4 e 5. Esse é a etapa não supervisionado.

2. Para pessoas não conhecidas pelo algoritmo, deve-se realizar a marcação identificando quem são. Essa é a etapa supervisionada.

1.4.4 Aprendizado por reforço

O aprendizado por reforço é melhor explicado através das palavras chaves como:

✓ **Agente**: é o aprendizado em si, responsável por observar o ambiente e executar as ações, obtendo recompensas ou penalidades.

✓ **ações**: tomadas de decisão realizadas pelo agente.

✓ **políticas**: ações aprendidas pelo agente.

1.5 Tipos de aprendizado - Formas de treinamento

Há duas formas de um modelo ser treinado com novos dados: Em batch ou online.

1.5.1 Aprendizado em batch

Para aprendizado em batch:

1. O modelo é inicialmente treinado com todos os dados disponíveis;
2. O modelo é colocado em produção;
3. Após um certo delta t, o modelo pode:
   - ✓ Possuir um novo volume de dados considerável;
   - ✓ Uma coluna conter novas classes;
   - ✓ O modelo ter uma queda no desempenho;
4. O modelo é retreinado com os novos dados **offline** e então substituido pelo modelo em produção.

*A máquina deve ter capacidade de treinar o modelo com todo o conjunto de dados disponíveis.

1.5.2 Aprendizado online

Para aprendizado online, os dados são adicionados de forma incremental (os pequenos grupos de dados são chamados de mini-batchs):

1. O modelo deve ter uma rápida adaptabilidade a novos cenários, devendo atender as rápidas mudanças de cenário.
2. Cenários em que há pouca capacidade computacional para atender grande volume de dados, e portanto, adota-se a estratégia de receber dados que não cabem na memória (out_of_core)[1]

*Deve haver uma inteligência para medir o desempenho do modelo com o aprendizado com o novo batch, decidindo se há ou não substituição do modelo antigo, dado performance do modelo recém treinado.

1.6 Tipos de modelo

1.6.1 Aprendizado baseado em instância ou aprendizado em modelo

O **aprendizado** baseado em **instância** é aquele no qual **memorizam-se** regras de acordo com os dados de treinamento, ex: Um modelo que baseia-se no aparecimento de n palavras previamente memorizadas para a identificação de SPAM.

Um aprendizado inteligente, **aprendizado baseado em modelo** é capaz de realizar generalizações e dessa forma, no exemplo do SPAM, ser capaz de classificar um email como SPAM, além

[1]O out_of_cor, em específico, é utilizado para máquinas com pouca capacidade computacional, porém usado em algoritmos offlines.

das palavras dadas no treinamento, porque conseguiu identificar padrões nos emails que constitutem um SPAM (ex: combinação gramatical das palavras).

Para o exemplo de PIB:

1. Se o modelo é baseado em memorização, ele buscará o país com PIB mais próximo do PIB da nova instância, e usará esse país para determinar a satisfação de vida.
2. Se o modelo é baseado em modelo, ele usará os dados de treinamento para construir uma equação, tal como y = ax + b), para dado uma nova instância, com seu dado de x, obter o valor de y.

1.7 Principais desafios do Aprendizado de Máquina

Os principais desafios do AM são:

✓ **Qualidade dos dados de treinamento**: Dados de treinamento que fornecem poucas informações para o que deseja-se modelar. Para contornar a qualidade dos dados de treinamento, pode-se utilizar:

- combinação de várias variáveis em uma única: feature engineering;
- seleção de características: feature selection;
- criação de novas características ao coletar dados novos.

✓ **Quantidade insuficiente dos dados de treinamento**: Para um modelo possuir a capacidade de generalização, deve se fornecer a ele **quantidade suficiente de dados** para que não sejam utiizados poucos dados, que podem possuir ruídos, que invabializarão generalizações.

✓ **Dados de treinamento não representativos**: Ao usar um conjunto de **dados não representativos**, o modelo poderá performar muito bem no treinamento, mas **falhará nos testes**, bem como **não representar a vida real**. Ex: disputa de Landon contra Roosevelt.

1.8 Resultados de um modelo

Ao executar um modelo, ele pode:

✓ Performar bem nos exemplos de treinamento e nos exemplos de teste.

✓ Performar bem nos exemplos de treinamento e mal nos exemplos de teste:

Efeito chamado Overfitting, no qual provavelmente os **dados de treinamento possuam características distintas dos dados de teste (ex: haver dados ruidos nos dados de treinamento ou nos dados de testes)**, impedindo que o modelo generaliza-se corretamente para esses novos dados. Uma das maneiras de corrigir esse problema é incluir

dados de treinamento que permitam melhor generalização. Outra forma de evitar o overfitting é utilizado **modelos menos complexos/regularizados, que evitem super ajuste aos dados de treinamento.** Por tais motivos, é tão importante uma base de treinamento de quantidade representativa que consiga abranger as características dos dados como um todo. Ex de dado ruidoso: Países com W na base de life satisfaction (Norway, Switzerland, New Zeland and Sweden), no qual um modelo muito complexo pode identificar que países com W são significados de alta satisfação de vida.

A quantidade de regularização aplicada a um modelo, é um **hiperparâmetro do modelo**, ou seja, uma característica que é definida, antes do treinamento do modelo e permanece constante ao longo dele.

✓ Performar mal nos exemplos de treinamento e mal nos exemplos de teste:

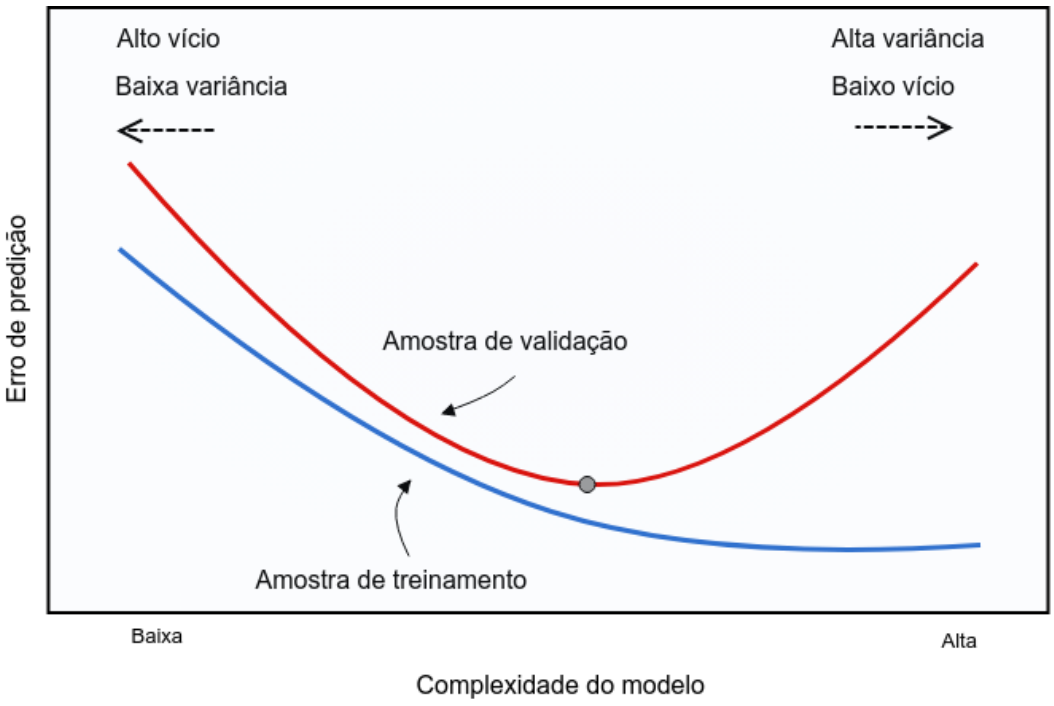
Efeito chamado underfitting, no qual o **modelo escolhido não consegue generalizar** em nenhum dos cenários (treinamento e teste, necessitando: da **substituição por um modelo mais complexo, capaz de identificar mais sutilezas** nos dados ou **incluir novas características que forneçam maior quantidade de informações (que pode ser feito com feature engineering)** e/ou **minimizar as restrições do modelo**.

1.9 Teste e validação

A **taxa de erro nos novos casos** (não vistos no treinamento) é chamado de **erro de generalização**. Essa etapa é útil para **ajuste de hiperparâmetros e seleção de modelo**. Há duas formas de **validar um modelo**: **holdout de validação cruzada** e **k-folds**.

Métodos de reamostragem são ferramentas indispensáveis na estatística moderna. As técnicas envolvem particionar os dados de treino e reajustar os modelos em competição para cada subamostra, a fim de obter informações adicionais sobre o ajuste do modelo (que não seria possível com os dados completos). Como exemplo, podemos estimar o erro de teste associado a determinado modelo (avaliação do modelo), selecionar o nível de flexibilidade adequado (seleção do modelo) etc.

Figura 2. Validação cruzada - HoldOut



Fonte: Própria

Na prática, treinamos o algoritmo com a amostra de treinamento (curva em azul da figura abaixo) e avaliamos a qualidade do treinamento com a amostra de validação (baseando-se na curva em vermelho). Note que se utilizarmos algoritmos muito simples (pouco flexíveis), teremos um erro de predição alto na amostra de treino (curva em azul), e a medida que aumentamos sua complexidade, o erro de treinamento tende a diminuir. Entretanto, essa melhora aparente vem acompanhada da redução na capacidade de generalizar a informação, ou seja, com a chegada de novos exemplos (amostra de validação), o desempenho não é satisfatório.

Frente a essa questão, o desafio é encontrar um meio-termo entre o modelo simples, que aprendeu pouco (viciado), e o complexo que aprendeu demais (alta variabilidade), tal que com a chegada de novos dados, se erre o mínimo possível (haja capacidade de generalização).

1.9.1 Holdout de validação cruzada

A atividade envolve **dividir aleatoriamente o conjunto de dados em (i) treinamento: utilizado para preparar o modelo (treiná-lo) e (ii) validação: utilizado para avaliar o desempenho do modelo treinado** (neste ponto é que consideramos as funções acima descritas). É importante constatar que algumas literaturas não fazem distinção entre validação e teste, muito embora sejam entidades diferentes. Os dados de validação são rotulados, assim como os de treinamento, mas são utilizados para testar o desempenho do modelo. Neste texto, consideramos

os dados de teste como informações novas, ainda não rotuladas (veja a figura abaixo).

Figura 3. Holdout - Datasets

Fonte: Própria

Portanto:

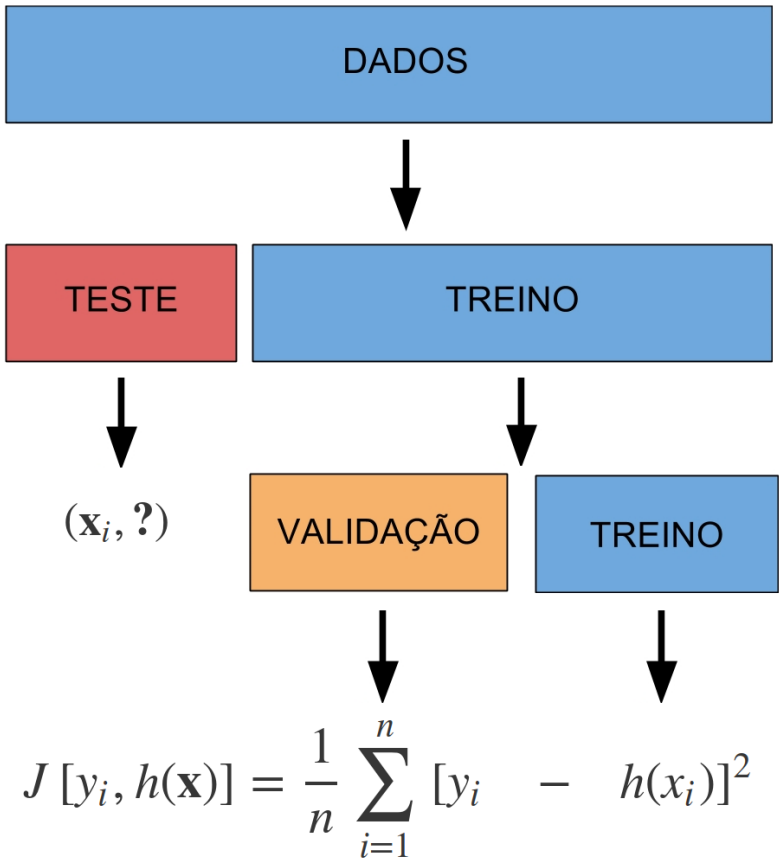
✓ **train + validation dataset**: utilizado para escolha do melhor modelo e dos hiperparâmetros.

✓ **test dataset**: utilizado para avaliação do erro de generalização

Sendo a **sequência de passos**:

1. Seleção aleatória dos datasets de train, validation e test.
2. Escolha do modelo e hiperparâmetros no dataset de train e validation
3. Retreino com dataset completo (train + validation).
4. Obtenção do erro de generalização (com alguma função de custo) no test dataset.

Figura 4. Validação cruzada - Fluxo com Datasets

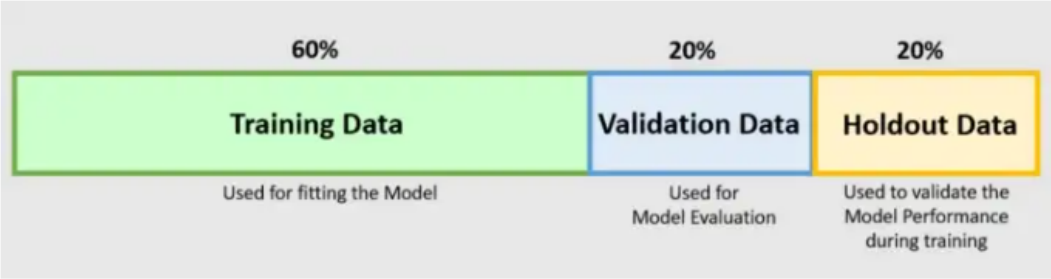


Fonte: https://towardsdatascience.com/when-training-a-model-you-will-need-training-validation-and-holdout-datasets-7566b2eaad80

Em resumo:

- ✓ Essa solução geramente *funciona muito bem, se estamos considerando que todos os datasets terão amostragens* suficientes para permitir identificar ruidos (que estejam em todas as instâncias), evitando enviesamento e conjuntos ruidosos.
- ✓ Um exemplo de problema pode ser visto quando o dataset de validação é muito pequeno ou não contempla todas as características contidas no conjunto de dados e portanto, o modelo não será bem avaliado, e provavelmente haverá overfitting.

Figura 5. Holdout - Datasets - Tamanhos



Fonte: https://towardsdatascience.com/when-training-a-model-you-will-need-training-validation-and-holdout-datasets-7566b2eaad80

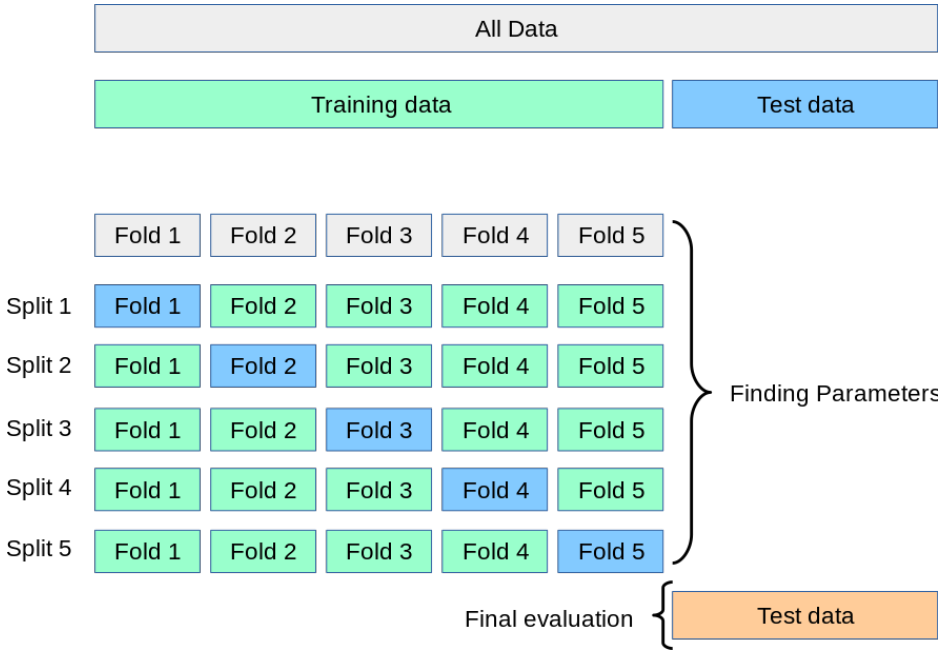
Geralmente, a taxa de divisão frequentemente usada é 60:20:20 (60% para dados de treinamento, 20% para dados de validação e 20% para dados de retenção) ou 50:25:25. No entanto, isso também depende do tamanho e do tipo de dados usados. É importante garantir que o conjunto de dados seja bem particionado com cada conjunto de dados contendo os padrões ou tendências dos dados originais ou podemos acabar selecionando um modelo que é tendencioso com base nos padrões ou tendências nos dados de validação.

1.9.2 K-Folds

Os KFolds é uma técnica de avaliação do modelo utilizando subconjuntos de datasets, no qual um subconjunto é utilizado uma vez para teste e como treinamento nas vezes restantes, dessa forma, o modelo pode ser avaliado k vezes e em vários subconjuntos menores. Ao final do teste, poderá se observar se o modelo performou muito indistintamente em um determinado fold, podendo identificar uma característica a ser analisada.

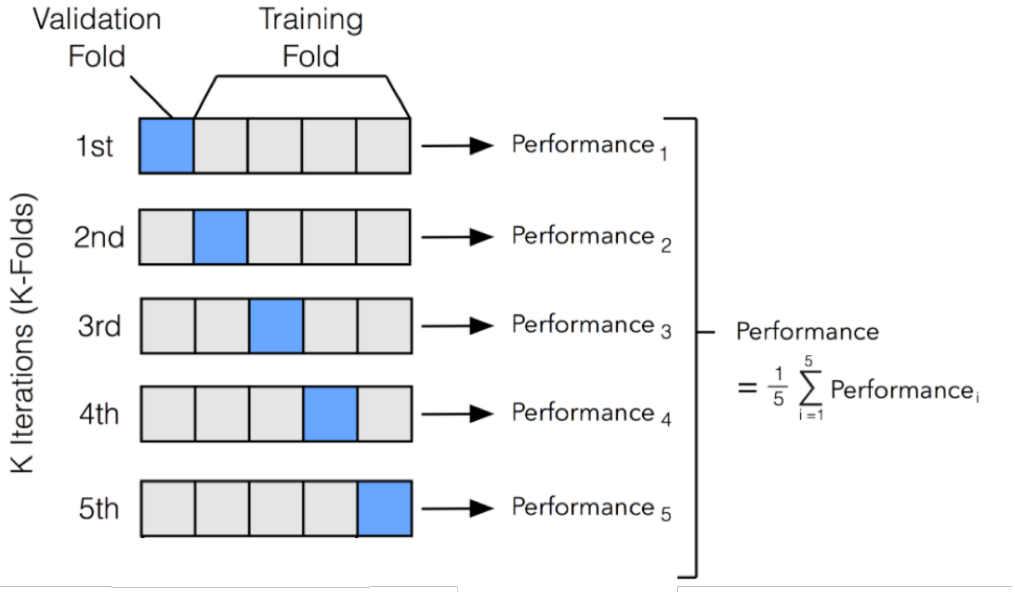
A primeira etapa é separar o conjunto de dados em treinamento e teste. O conjunto de treinamento é subdividido em kfolds.

Figura 6. Validação cruzada - KFolds



Para cada split, avalia-se a performance do modelo, podendo após n rodadas, obter-se a média de performance:

Figura 7. Validação cruzada - KFolds



1.10 Exercícios - Cap 1 - Hands on Machine Learning with Scikit-Learn and Tensorflow

1.10.1 1.

O aprendizado de máquina pode ser definido como a ciência da máquina aprender com os dados.

1.10.2 2.

O AM é otimo para problemas que envolvem:

- ✓ Soluções que envolvem muitas configurações e condições, ex: o caso dos spammers que podem mudar a forma de escrever uma palavra, para tentar dar o bypass em uma condição como classificar como spam emails com "só para você".
- ✓ Problemas que possuem dados flutuantes, ou seja, que recebem dados com novas características;
- ✓ Problemas complexos que envolvem soluções não triviais, ex: reconhecimento de voz ou reconhecimento de caracteres;
- ✓ Grande quantidade de dados, pois nesses casos, o AM pode ser a melhor forma de encontrar padrões e tendências.

1.10.3 3.

Um conjunto de dados rotulado envolve dados que incluem as soluções desejadas.

1.10.4 4.

As duas tarefas supervisionadas mais comuns são Regressão e Classificação. A regressão pode ser explicada também como a previsão de um alvo de valor numérico.

1.10.5 5.

Tarefas comuns sem supervisão são: Clustering, Visualização, Redução de dimensionalidade, regras de associação e detecção de anomalias.

No aprendizado não supervisionado, os dados de treinamento não são rotulados, ou seja, o algoritmo tentará aprender sem "um professor". O objetivo desses algoritmos é descobrir associações entre os dados do modelo.[2][3] O aprendizado não supervisionado é útil em tarefas como:

- ✓ **Agrupar clientes com perfis semelhantes de uma população de clientes**, nesse caso, se utilizar um algotimo de clustering hierárquico, ele também poderá subdividir os grupos em grupos cada vez menores.

[2] Não há mapeamento de uma função f(x) = y

[3] Normalmente busca-se similaridade e dissimilaridade calculando-se a distância entre os dados

- ✓ **Sistemas de recomendação** com base no perfil de consumo do cliente. Nesse caso, são algoritmos comuns: Apriori e Eclat. Aprendizado não supervisionado é parte das técnicas utilizadas para resolução desse tipo de case e um exemplo bastante comum é nas prateleiras de supermercado[4].

- ✓ **Redução de dimensionalidade**, O PCA é uma das técnicas para redução de dimensionalidade, na qual combinam-se linerarmente as variáveis independentes, a fim de remover variáveis que fornecem pouca informação ao modelo (para isso, pode-se realizar o processo de adição ou remoção de variáveis e cálculo do R2 ajustado) e merge de variáveis altamente correlacionadas.

Outras duas tarefas bastante comuns em NS são:

- ✓ **Detecção de anomalias | Detecção de novidades**: Nesses algoritmos um dataset inicial possui dados "normais"e dados "anômalos", o algoritmo deve ser capaz de identificar esses padrões (dois grupos distintos). Para uma nova instância, o algoritmo deve detectar a qual cluster essa nova instância pertence (normal ou anômalo).

1.10.6 6.

Para um robô andar em vários terrenos desconhecidos, pode-se utilizar o aprendizado por reforço.

Esse tipo de aprendizado possui palavras chaves como:

- ✓ **Agente**: o agente nesse contexto tem sinônimo de sistema de aprendizado, no qual o agente pode assistir o ambiente, selecionar e executar ações e obter recompensas ou penalidades.

- ✓ **ações**: As ações são possíveis tomadas de decisão que um agente pode tomar.

- ✓ **políticas**: São as estratégias aprendidas.

Um exemplo é o AlphaGo, que venceu o campeão mundial no jogo Go em 2017.

1.10.7 7.

A segmentação de clientes em vários grupos pode ser feita utilizando aprendizado não superivisionado do tipo clustering.

[4] produtos que devem estar próximos em um mercado

1.10.8 8.

Um problema de detecção de spam pode ser modelado como aprendizado supevisionado de classificação. Esse tipo de técnica pode aprender com as palavras: part of speech, palavras mais frequentes (freqdist ou countvectorizer), a junção de palavras mais comuns (bigram, trigram, fourgram) e mais. Um ruim sistema de aprendizado para spam é o aprendizado baseado em instâncias (ou memorização), ou seja, aprender quais as palavras mais utilizadas ou a quantidade de palavras comuns em emails classificados como spam, com ambos aprendizados o modelo poderá possuir dificuldades em generalizar em novos casos. O problema dos modelos baseados em instâncias é a dificuldade que esses modelos possuem em generalizar, dado uma possível mudança de perfil dos dados (tal como a mudança de perfil dos emails dos spammers).

1.10.9 9.

Um sistema de aprendizado online (também chamado de incremental learning ou out-of-core) é um tipo de sistema que possui uma taxa de aprendizado em relação à dados novo. Essa taxa de aprendizado significa que o modelo vai recebendo dados novos e realizando novos treinamentos, modelando-se novamente.

1.10.10 10.

O out-of-core learning pode ser explicado pelas etapas:

- ✓ Recebe um mini-lote de dados e realiza um treinamento.
- ✓ Recebe um novo mini-lote de dados e realiza um novo treinamento.

Esses passos são realizados de modo sucessivo, sendo útil para problemas com grande quantidade de dados e para máquinas que não possuem grande poder de processamento.

1.10.11 11.

Medidas de similaridade são utilizados por modelos baseados em instância, ou seja, que aprendem por memorização e depois usam similaridade. Exemplo: KNN.

1.10.12 12.

Os parâmetros do modelo são as propriedades dos dados aprendidos durante o treinamento.

Os hiperparâmetros são comuns para modelos semelhantes e não podem ser aprendidos durante o treinamento, mas são definidos previamento, ex: nº e tamanho das camadas oculta, taxa de aprendizado.

1.10.13 13.

Os algoritmos baseados em modelos procuram identificar características que permitam a formulação de uma equação, plano ou hiperplano, que expliquem modelem o problema.

1.10.14 14.

Os desafios envolvem principalmente: Dados não representativos, dados de má qualidade (outliers e ruidosos), overfitting (sobreajuste) e características irrelevantes para explicar os dados.

1.10.15 15.

Esse é um caso de overfitting, ou seja, sobreajuste de dados. Para atuar com esses casos, pode-se:
- Escolher um modelo mais simples; - Escolher características mais relevantes (pode-se usar feature engineering); - Coletas mais dados; - Reduzir dados de má qualidade.

1.10.16 16.

Um conjunto de testes são dados separados do treinamento, para verificar se o modelo está generalizando bem.

1.10.17 17.

Um conjunto de validação permite uma maior segurança na verificação de quanto o modelo está generalizando bem para subconjuntos diferentes.

1.10.18 18.

Ajustando os hiperparâmetros do modelo, o conjunto pode se tornar muito simples, passando de um problema de overfitting para underfitting, ou seja, não funcionar bem para os dados de treino e teste.

1.10.19 19.

A validação cruzada é uma técnica para comparação de modelos e ajustes de hiperparâmetros, sem a necessidade de um conjunto de validação. Nessa técnica são formados conjuntos de dados e a cada treino, são escolhidos conjuntos de dados diferentes para treino e teste, sendo o processo repetindo por n vezes, fornecendo o desempenho em cada uma das vezes.