Machine Learning

Mas afinal, o que é *machine learning*? De acordo com Nelli (2015), é uma disciplina que utiliza uma variedade de procedimentos e algoritmos para identificar padrões, agrupamentos ou tendências e extrair informações úteis para análise de dados, de forma automatizada. Em termos simples, são métodos matemáticos usados para treinar algoritmos a identificar padrões. Géron (2019) oferece uma definição mais acessível: "Machine learning é a ciência (e arte) de programar computadores para que eles possam aprender a partir dos dados".
Os modelos de *machine learning* podem ser classificados de acordo com suas características. Duas das principais categorias são o aprendizado supervisionado e o não supervisionado:

- No aprendizado supervisionado, durante o treinamento do modelo, são fornecidas as respostas corretas. Por exemplo, ao identificar se uma transação é fraudulenta ou não, o algoritmo aprende os padrões dessas transações ilícitas e pode classificar transações futuras. Esse é um exemplo clássico de modelo de classificação, no qual a tarefa é classificar algo entre duas ou mais categorias. Outro tipo de modelo supervisionado é a regressão, que produz como saída do algoritmo um valor numérico. Um exemplo desse tipo de modelo é prever o valor de uma casa com base em suas características.
- No aprendizado não supervisionado, não há respostas corretas fornecidas ao modelo durante o treinamento. Um exemplo desse tipo de modelo é o algoritmo de clusterização, que pode ser usado para segmentar os clientes de uma empresa com base em suas características.

Guia de boas práticas de modelagem

Os resultados de um modelo podem ser questionáveis devido a vários fatores, como:

- Descuido com os dados de entrada.
- Calibração e validação insuficientes.
- Trabalho fora do escopo do projeto.
- Hipótese imprecisa do modelo.

Para abordar esses desafios, foi desenvolvido o Guia de Boas Práticas de Modelagem (*Good Modelling Practice* - GMP). Trata-se de um conjunto de diretrizes e práticas recomendadas para a criação, implementação e documentação de modelos numéricos. Desenvolvido por Van Waveren e outros especialistas, ele visa garantir a qualidade, transparência e replicabilidade dos modelos utilizados em diversos campos, como ciência, engenharia, economia e ciências sociais. O GMP abrange várias etapas do processo de modelagem, mas pode ser resumido em 6 passos:

- **Definição do problema (passo 1):** nesta etapa, é fundamental ter uma compreensão clara do problema que o modelo visa resolver. Isso inclui

identificar o objetivo do projeto, as partes envolvidas, as estimativas de recursos necessários e a definição de reuniões para alinhamento. A análise do problema deve ser feita em dois níveis: cliente e modelador, para garantir um entendimento comum sobre o que será modelado.

- **Configuração do modelo (passo 2):** após compreender o problema, esta etapa envolve a análise dos dados disponíveis e necessários, a descrição do sistema a ser modelado e a definição do modelo conceitual. O modelo conceitual descreve a relação funcional entre os componentes do sistema de forma simplificada, seja por meio de texto ou de equações matemáticas. Isso é crucial para explicar a base do modelo e permitir que outros compreendam o processo.
- **Análise do modelo (passo 3):** aqui, é importante manter um registro detalhado de todas as etapas realizadas, incluindo calibração e testes. A calibração é fundamental para ajustar os parâmetros do modelo com precisão. Testes como rodar com dados padrão, análises de sensibilidade e testes de estabilidade ajudam a garantir que o modelo esteja sendo usado corretamente e possa reproduzir observações com um grau pré-estipulado de ajuste.
- **Simulações de produção (passo 4):** nesta etapa, é essencial descrever como o modelo será utilizado na prática, incluindo os dados de entrada, a versão do modelo, o período a ser simulado e os desvios do modelo de referência. É importante distinguir entre as questões propostas pelo cliente e as modificações feitas pelo modelador ou pelo modelo. As simulações devem responder se o modelo corresponde ao propósito, atende aos requisitos e se a definição do sistema e do modelo conceitual estão corretas.
- **Interpretação dos resultados (passo 5):** antes de interpretar os resultados, é necessário descrevê-los e sumarizá-los estatisticamente, incluindo as suas incertezas e restrições. É importante verificar se os procedimentos adotados resultaram em um modelo que capaz de responder à pergunta inicial. As conclusões devem estar diretamente relacionadas ao motivo da pesquisa e aos resultados obtidos.
- **Reporte e arquivamento do projeto de modelagem (passo 6):** nesta etapa final, é necessário descrever completamente as características do modelo, a localização dos dados utilizados e obtidos, a produção científica resultante e os relatórios do processo. A qualidade dos relatórios deve permitir a terceiros reproduzir o modelo ou continuar o trabalho a partir de onde foi suspenso. Os relatórios devem indicar a validade, usabilidade e restrições do modelo.

Essas práticas são essenciais para garantir a confiabilidade dos modelos e a tomada de decisões baseadas em suas previsões ou análises. Ao seguir o GMP, os modeladores podem evitar armadilhas comuns, como viés nos dados, *overfitting* (sobreajuste) do modelo, interpretação errônea dos resultados e falta de transparência na metodologia utilizada.
O GMP é amplamente reconhecido como uma ferramenta valiosa para melhorar a qualidade e a credibilidade dos modelos numéricos, contribuindo para avanços significativos em diversas áreas de pesquisa e aplicação prática.

Variável categórica

Uma variável categórica representa categorias ou grupos discretos e finitos. Elas não têm uma ordem intrínseca, e cada valor representa uma categoria distinta e não mensurável em termos numéricos. Por exemplo, uma variável categórica poderia representar cores (vermelho, verde, azul), estados civis (solteiro, casado, divorciado) ou tipos de veículos (carro, moto, caminhão). Em contraste, variáveis numéricas representam quantidades mensuráveis ou contínuas, como idade, altura ou peso. Para a variável *Embarked*, que é uma variável categórica, uma estratégia viável é imputar a categoria mais frequente para os dados faltantes.

As variáveis *Name*, *Sex*, *Ticket*, *Cabin* e *Embarked* estão formatadas como *string*. As variáveis *Name*, *Ticket* e *Cabin* serão descartadas e não precisamos nos preocupar com elas. As variáveis *Sex* e *Embarked* serão utilizadas no treinamento do modelo e precisam ser transformadas em um valor numérico. Existem duas técnicas utilizadas com frequência para tratar dados categóricos: *Label Encoder* e *One Hot Encoder*.

Normalização

Alguns algoritmos de aprendizado de máquina não tem um bom desempenho quando as variáveis numéricas estão em escalas muito diferentes. Como indica Géron (2019), para resolver essa questão, existem dois métodos comuns: padronização e normalização dos dados.
A normalização, em que os valores são colocados em uma escala de 0 a 1, é o método mais simples. A biblioteca *Scikit-Learn* possui uma classe chamada *MinMaxScaler* para realizar essa operação.
A padronização dos dados resulta em valores com média 0 e desvio padrão igual a 1. A biblioteca *Scikit-Learn* implementa a padronização dos dados através da classe *StandardScaler*.

Scikit-learn

Uma das bibliotecas de *machine learning* mais conhecidas para a linguagem *Python* é o *scikit-learn*, o qual utilizaremos ao longo desta aula para treinar nosso algoritmo de classificação.
De acordo com Géron (2019), o *scikit-learn* segue o princípio de que todos os objetos compartilham uma interface simples e consistente, podendo ser classificados em três tipos:

- **Estimadores:** são objetos capazes de estimar parâmetros com base em um conjunto de dados. A estimativa é realizada pelo método *fit()*.

- **Transformadores:** são objetos que podem transformar um conjunto de dados por meio do método *transform()*. Alguns estimadores também podem realizar essas transformações, que normalmente são baseadas nos parâmetros aprendidos. Todos os transformadores também possuem o método *fit_transform()*, que é equivalente a chamar os métodos *fit()* e *transform()* sequencialmente.

- **Previsores:** alguns estimadores são capazes de fazer previsões com base em um conjunto de dados. Neste contexto, por exemplo, utilizaremos o modelo *LogisticRegression* para prever se um determinado passageiro do Titanic sobreviverá ao desastre. Um previsor possui o método *predict()*, que recebe um novo conjunto de dados — diferente dos utilizados durante o treinamento do modelo — e retorna um conjunto de previsões correspondentes.

Relações entre variáveis

A análise das relações entre variáveis é uma parte fundamental da ciência de dados e do processo de modelagem em *machine learning*. Ela consiste em investigar como as diferentes variáveis em um conjunto de dados estão relacionadas entre si e com a variável alvo que estamos tentando prever ou entender. Isso envolve a identificação de padrões, correlações e dependências entre as variáveis, o que pode fornecer insights importantes para a construção de modelos mais precisos e interpretação dos resultados. Métodos como análise de correlação, análise de regressão e técnicas de visualização de dados são frequentemente utilizados para explorar e entender essas relações. Essa análise é crucial para tomar decisões informadas sobre quais variáveis incluir ou excluir em um modelo, bem como para compreender melhor os mecanismos subjacentes aos dados estudados.

RFE e Seleção de variáveis

Em projetos de *machine learning*, é comum lidarmos com muitas variáveis preditoras possíveis ou criarmos variáveis que não têm um grande poder preditivo. Ambos os casos devem ser evitados: um grande número de variáveis preditoras irá aumentar a complexidade do modelo, em muitos casos desnecessariamente, e pode fazer com que o algoritmo treinado tenha dificuldade de fazer previsões acuradas para novos dados. O mesmo pode acontecer quando temos variáveis com baixo poder preditivo.
Existem algumas técnicas para selecionar as melhores variáveis a serem utilizadas para treinar um modelo. Uma delas é o RFE (Recursive Feature Elimination), um método de seleção de características utilizado em problemas de *machine learning* para escolher automaticamente as melhores características (ou variáveis) para o modelo (KUHN, 2019). Ele funciona de forma iterativa, removendo as características menos importantes em cada iteração, com base em um modelo de *machine learning* ajustado aos dados. O processo continua até que o número desejado de características seja atingido. O RFE ajuda a reduzir o *overfitting*, a melhorar a generalização do modelo e a reduzir o tempo de treinamento, ao eliminar características irrelevantes ou redundantes (O QUE, 2024).
O *scikit-learn* possui a classe RFECV. De acordo com sua documentação, trata-se de uma classe de eliminação recursiva de recursos com validação cruzada para selecionar recursos (RFECV, 2024).

Existem várias técnicas para facilitar essa interpretação:

- *Feature Importance* (Importância das Variáveis): esta técnica avalia a importância de cada variável no modelo. Algoritmos como árvores de decisão

e *random forests* fornecem naturalmente uma medida de importância das variáveis.

- *Partial Dependence Plots* (Gráficos de Dependência Parcial): esses gráficos mostram como uma variável específica afeta as previsões do modelo, enquanto mantém outras variáveis constantes.
- SHAP Values (Valores SHAP): essa técnica atribui um valor de importância a cada variável para cada previsão individual, ajudando a entender como cada uma contribui para a decisão do modelo.
- LIME (*Local Interpretable Model-agnostic Explanations*): o LIME é uma técnica que explica as previsões de um modelo de *machine learning* de forma local, ou seja, para instâncias específicas de dados.
- *Model Summaries* (Resumos do Modelo): tem como objetivo resumir as principais características do modelo, como métricas de desempenho, coeficientes de regressão, árvores de decisão simplificadas, entre outros; pode fornecer insights sobre como o modelo está funcionando.

Matriz de confusão

Uma alternativa para medir a performance de um modelo de classificação é utilizar a matriz de confusão, a qual apresenta a comparação entre os valores reais e os previstos por um modelo supervisionado. Os valores reais positivos, previstos como tal, são chamados de verdadeiro positivo (VP), e os valores reais negativos, previstos como tal, são chamados de verdadeiro negativo (VN). Os valores reais positivos, previstos como negativos, são chamados de falso negativo (FN), e os valores reais negativos, previstos como positivos, são chamados de falso positivo (FP).

Precision, recall e *F-score*

Também podemos usar as métricas *precision*, *recall* e *F-score* para avaliar um modelo de classificação. A métrica *precision* mede a acurácia das previsões positivas, e o *recall* mede a taxa de instâncias que pertencem à classe positiva e foram detectadas pelo modelo. O *F-score* combina o *precision* e *recall* em uma única métrica, por meio da *média harmônica* desses dois valores (GÉRON, 2019).

$$Precision = \frac{(VP)}{(VP + FP)}$$

$$Recall = \frac{(VP)}{(VP + FN)}$$

$$F - score = 2 \times \frac{(precision \times recall)}{(precision + recall)}$$

Curva ROC

A última métrica que veremos — e que é utilizada com frequência — é a área sob a curva *Receiver Operating Characteristic* (ROC). A curva ROC é um gráfico que mostra a relação entre a taxa de verdadeiros positivos (ou *recall*) e a taxa de falsos positivos, que é a proporção de instâncias negativas classificadas incorretamente como positivas. Podemos utilizar a função *roc_curve()* da biblioteca *scikit-learn* para calcular a taxa de verdadeiros e falsos positivos, e a classe *RocCurveDisplay* para plotar um gráfico com a curva ROC.