

Ciência de Dados, Estatística e Modelagem com Python

Do zero à modelagem preditiva: Python, Estatística e Machine Learning aplicados a casos reais de negócio. Uma jornada estruturada em **20 módulos** que combina fundamentos de programação, estatística descritiva, redução de dimensionalidade, segmentação de clientes e algoritmos de regressão, classificação, recomendação e séries temporais aplicados a crédito, saúde, energia, marketing e operações — encerrada com um projeto prático de redução de custo.

20
Módulos

15+
Bibliotecas Python

12
Algoritmos de Machine Learning

01
Projeto final prático

SUA JORNADA DE FERRAMENTAS E COMPETÊNCIAS

- 1 Python
- 2 Pandas & Numpy
- 3 Datatable & Polars
- 4 Eng. de Variáveis
- 5 Estatíst.
- 6 PCA
- 7 Segment.
- 8 Regressão
- 9 Classific.
- 10 Recomend. & Séries Temp.
- 11 Projeto Final

12 algoritmos de ML

Regressão, classificação, recomendação e séries temporais

Do zero ao avançado

Base sólida em Python, Pandas e estatística

Projeto prático de portfólio

Redução de custo aplicada

100% em Python

Jupyter Notebook e bibliotecas de mercado

Transforme dados em modelos preditivos e decisões inteligentes

Saber a sintaxe de um algoritmo de Machine Learning não é o mesmo que saber quando e por que usá-lo. É comum encontrar profissionais que conseguem rodar um Random Forest ou um K-means em Python, mas travam diante da pergunta mais importante: “esse é o algoritmo certo para este problema, e o resultado é confiável o suficiente para embasar uma decisão?”. É exatamente essa lacuna que esta formação foi desenhada para fechar.

Ao longo de **20 módulos progressivos**, você vai desenvolver competências práticas em Machine Learning aplicado a negócios utilizando Python — da linguagem e das bibliotecas de análise de dados à engenharia de variáveis, passando por técnicas de redução de dimensionalidade (PCA) e segmentação de clientes, até doze algoritmos de regressão, classificação e recomendação, cada um aplicado a um estudo de caso diferente — crédito, saúde, energia, marketing e operações. A formação avança por séries temporais e é encerrada com um projeto prático integrador de redução de custo, aplicando o que foi aprendido para diagnosticar e reduzir custos operacionais de um negócio real.

A jornada segue uma lógica intencional: primeiro você consolida a base em Python e estatística descritiva; depois aprende a preparar e transformar variáveis; em seguida, explora técnicas não supervisionadas (PCA e Segmentação) que organizam e simplificam os dados; e só então avança para os algoritmos supervisionados de regressão, classificação e recomendação — a etapa em que os modelos passam a prever e decidir, não apenas descrever.

Mais do que ensinar sintaxe de algoritmos, esta formação constrói raciocínio de Machine Learning: a capacidade de escolher o algoritmo certo para cada tipo de problema, validar sua performance com métricas adequadas e traduzir a saída de um modelo estatístico em uma decisão de negócio defensável.

PARA QUEM É ESTA FORMAÇÃO

- ♦ Analistas de dados e BI que querem evoluir para Ciência de Dados
- ♦ Estudantes de estatística, computação, economia e engenharia
- ♦ Profissionais que precisam interpretar e construir modelos preditivos
- ♦ Quem já tem Python básico e quer aplicar estatística e ML em problemas reais

PRÉ-REQUISITOS

Recomenda-se conhecimento básico de lógica de programação. Não é necessária experiência prévia com Machine Learning — os Módulos 1 e 2 cobrem o essencial.

METODOLOGIA

Cada algoritmo é apresentado com o fundamento teórico mínimo necessário e imediatamente aplicado em um estudo de caso prático em Python.

O CAMINHO DA FORMAÇÃO

01	02
O que Você Precisa Saber Antes de Começar	Introdução à Linguagem Python
03	04
Pandas e Numpy para Análise de Dados	Datatable e Polars para Análise de Dados em Alta Performance
05	06
Engenharia de Variáveis e Dados Faltantes	Estatística e Teste de Hipóteses com Python

07	08
PCA — Análise Financeira	K-means — Segmentação Produtos Bancários
09	10
Regressão Logística e Linear	RandomForest — Previsão de Inadimplência em Reais
11	12
RandomForest — Classificação de Diabetes com Precisão	XGBoosting — Previsão de Consumo Energético
13	14
XGBoosting — Fraudes em Cartões de Crédito	SVR — Análise Estratégica do Comportamento do Cliente
15	16
SVM — Predict Customer Churn	Decision Tree: Satisfação do Cliente — Classificação
17	18
Apriori — Marketing Digital Orientado por Dados	KNN — Recomendações Inteligentes
19	20
Séries Temporais — Análise da Depreciação de Produtos, Máquinas e Veículos	Projeto Final: Redução de Custo

01

O que Você Precisa Saber Antes de Começar

Panorama da formação, ambiente de trabalho e mentalidade de Ciência de Dados.

CONTEXTO

Antes de escrever a primeira linha de código, é preciso entender o mapa da jornada: o que é Ciência de Dados na prática, como um cientista de dados pensa diante de um problema de negócio, e como está estruturado o ambiente de trabalho (Python, Jupyter/Anaconda) que você vai usar em todos os módulos seguintes.

PRINCIPAIS TÓPICOS

- O que é Ciência de Dados na prática
- Instalação do Python e Anaconda
- Como navegar esta formação
- O ciclo de um projeto de dados
- Jupyter Notebook e boas práticas

NA PRÁTICA

Você vai configurar seu ambiente (Anaconda/Jupyter) e explorar a estrutura de um projeto de dados do início ao fim, entendendo onde cada módulo desta formação se encaixa nesse ciclo.

DESAFIO DO MÓDULO

Instale o ambiente Python/Jupyter e execute seu primeiro notebook, documentando cada etapa do ciclo de um projeto de dados com suas próprias palavras.

ERRO COMUM A EVITAR

Pular direto para os algoritmos sem entender o ciclo completo de um projeto de dados é a causa mais comum de modelos tecnicamente corretos, mas inúteis para o negócio.

DICA DO AUTOR

Antes de codar, escreva em uma frase qual pergunta de negócio você está tentando responder — é ela que vai guiar todas as decisões técnicas dos módulos seguintes.

O QUE VOCÊ VAI CONQUISTAR

Você vai sair deste módulo com o ambiente pronto e uma visão clara do caminho que vai percorrer até o último módulo.

- ♦ Ambiente Python/Jupyter configurado
- ♦ Visão do ciclo de um projeto de dados
- ♦ Mentalidade orientada a problemas de negócio

Início da jornada → **01 Fundamentos** → 02 Python

FERRAMENTAS DO MÓDULO

[Anaconda](#)

[Jupyter Notebook](#)

02

Introdução à Linguagem Python

Sintaxe, estruturas de dados e lógica de programação para Ciência de Dados.

CONTEXTO

Toda análise de dados em Python começa pelo domínio das estruturas básicas da linguagem. Este módulo apresenta variáveis, listas, dicionários, estruturas de repetição e funções — a base que vai sustentar a manipulação de dados em todos os módulos seguintes.

PRINCIPAIS TÓPICOS

- Variáveis e tipos de dados
- Estruturas condicionais
- Funções e boas práticas
- Listas, tuplas e dicionários
- Laços de repetição (for/while)
- List comprehensions

NA PRÁTICA

Você vai praticar a criação e manipulação de listas e dicionários, e automatizar tarefas repetitivas com laços e funções — habilidades que vai reaproveitar em praticamente todos os módulos da formação.

DESAFIO DO MÓDULO

Escreva uma função em Python que receba uma lista de valores de vendas e retorne um dicionário com total, média e maior valor, sem usar bibliotecas externas.

ERRO COMUM A EVITAR

Confundir listas mutáveis com estruturas imutáveis (tuplas) é um deslize comum de quem começa em Python — e pode gerar bugs silenciosos difíceis de rastrear.

DICA DO AUTOR

Pratique list comprehensions desde já — é a forma mais 'pythonica' de transformar dados, e você vai encontrá-la o tempo todo em código de mercado.

O QUE VOCÊ VAI CONQUISTAR

Você vai dominar a sintaxe do Python e as estruturas de dados que sustentam toda a formação.

- ♦ Sintaxe fluente em Python
- ♦ Scripts reprodutíveis
- ♦ Base para Pandas e Numpy

01 Fundamentos → **02 Python** → 03 Pandas & Numpy

FERRAMENTAS DO MÓDULO

[Python 3](#)

[Jupyter Notebook](#)

CONTEXTO

Pandas e Numpy são o coração da análise de dados em Python: Numpy fornece a base numérica vetorizada, e Pandas organiza esses dados em DataFrames — a estrutura que você vai usar para carregar, explorar, filtrar e transformar dados em toda a formação.

PRINCIPAIS TÓPICOS

- Arrays e operações vetorizadas com Numpy
- Leitura de CSV/Excel
- Junção de bases (merge/join)
- Séries e DataFrames com Pandas
- Filtros, seleção e agrupamento (groupby)
- Estatísticas descritivas rápidas

NA PRÁTICA

Você vai carregar uma base real em um DataFrame, aplicar filtros e agrupamentos, e calcular estatísticas descritivas — o fluxo de trabalho que antecede qualquer modelagem estatística ou de Machine Learning.

DESAFIO DO MÓDULO

Carregue uma base de vendas em Pandas e responda, usando groupby, qual produto e qual região tiveram o maior faturamento médio no último trimestre.

ERRO COMUM A EVITAR

Usar laços for para percorrer linha a linha de um DataFrame, em vez de operações vetorizadas do Pandas/Numpy, é um erro de performance muito comum e evitável.

DICA DO AUTOR

Sempre que estiver percorrendo um DataFrame com um for, pare e pergunte: existe uma operação vetorizada do Pandas que faz isso em uma linha?

O QUE VOCÊ VAI CONQUISTAR

Você será capaz de carregar, explorar e transformar bases de dados reais com Pandas e Numpy.

- ♦ Manipulação vetorizada de dados
- ♦ Agrupamentos e junções de bases
- ♦ Base para engenharia de variáveis

02 Python → 03 Pandas & Numpy → 04 Datatable & Polars

FERRAMENTAS DO MÓDULO

pandas

numpy

04

Datatable e Polars para Análise de Dados em Alta Performance

Alternativas de alta performance ao Pandas para bases de grande volume.

CONTEXTO

Quando a base de dados cresce para milhões de linhas, o Pandas pode se tornar lento. Este módulo apresenta Datatable e Polars, bibliotecas otimizadas para processamento em alta performance, mantendo uma lógica de manipulação de dados semelhante à do Pandas.

PRINCIPAIS TÓPICOS

- Quando migrar de Pandas para Polars/Datatable
- Execução lazy vs. eager
- Boas práticas de memória
- Sintaxe de seleção e filtros no Polars
- Agrupamentos de alta performance

NA PRÁTICA

Você vai processar a mesma base de dados com Pandas e com Polars, comparando tempo de execução, e vai decidir em que cenários a migração para uma ferramenta de alta performance realmente se justifica.

DESAFIO DO MÓDULO

Compare o tempo de execução de um agrupamento (groupby) da mesma base em Pandas e em Polars, e explique em que situações essa diferença passaria a importar na prática.

ERRO COMUM A EVITAR

Migrar todo o pipeline para Polars/Datatable por modismo, mesmo em bases pequenas onde o Pandas já é suficientemente rápido, é um esforço de engenharia desnecessário.

DICA DO AUTOR

Meça antes de otimizar: só migre para Polars/Datatable quando o tempo de processamento em Pandas já for um problema real e mensurável.

O QUE VOCÊ VAI CONQUISTAR

Você será capaz de decidir quando e como usar ferramentas de alta performance para bases de grande volume.

- ♦ Processamento de dados em alta performance
- ♦ Leitura de execução lazy
- ♦ Critério para escolher a ferramenta certa

03 Pandas & Numpy → **04 Datatable & Polars** → 05 Eng. de Variáveis

FERRAMENTAS DO MÓDULO

[datatable](#)

[polars](#)

05

Engenharia de Variáveis e Dados Faltantes

Transforme e trate variáveis para que os algoritmos funcionem corretamente.

CONTEXTO

Nenhum algoritmo de Machine Learning funciona bem com dados brutos e incompletos. Este módulo ensina a tratar dados faltantes, criar variáveis derivadas e transformar variáveis (log, padronização, encoding) — a etapa que mais impacta a qualidade final de qualquer modelo.

PRINCIPAIS TÓPICOS

- Tipos de dados faltantes (MCAR, MAR, MNAR)
- Estratégias de imputação
- Criação de variáveis derivadas
- Encoding de variáveis categóricas
- Padronização e normalização
- Detecção de outliers

NA PRÁTICA

Você vai diagnosticar o padrão de dados faltantes de uma base real, escolher a estratégia de imputação adequada, e criar variáveis derivadas que aumentam o poder preditivo antes de qualquer modelagem.

DESAFIO DO MÓDULO

Em uma base com dados faltantes, diagnostique se o padrão é aleatório ou sistemático, escolha uma estratégia de imputação e justifique a escolha.

ERRO COMUM A EVITAR

Imputar todos os dados faltantes com a média, sem investigar por que aquele dado está faltando, é um erro comum que pode introduzir viés silencioso no modelo.

DICA DO AUTOR

Antes de imputar, sempre pergunte: existe um padrão em quem tem o dado faltante? Isso pode ser, em si, uma informação valiosa para o modelo.

O QUE VOCÊ VAI CONQUISTAR

Você será capaz de preparar dados adequados aos algoritmos das próximas etapas, tratando dados faltantes e criando variáveis relevantes.

- ♦ Diagnóstico e tratamento de dados faltantes
- ♦ Engenharia de variáveis derivadas
- ♦ Pré-requisito direto para PCA e modelagem

04 Datatable & Polars → **05 Eng. de Variáveis** → 06 Estatística

FERRAMENTAS DO MÓDULO

pandas

scikit-learn
(SimpleImputer)

06

Estatística e Teste de Hipóteses com Python

Os fundamentos estatísticos que sustentam qualquer modelo de Machine Learning.

CONTEXTO

Antes de qualquer algoritmo de Machine Learning, é preciso entender o comportamento estatístico das variáveis. Este módulo revisa estatística descritiva, distribuições e testes de hipótese em Python — a base conceitual que orienta decisões de modelagem em toda a formação.

PRINCIPAIS TÓPICOS

- Medidas de tendência central e dispersão
- Testes de hipótese (t-Student, qui-quadrado)
- Intervalos de confiança
- Distribuições de probabilidade
- Correlação e causalidade

NA PRÁTICA

Você vai aplicar testes de hipótese em Python para responder perguntas de negócio com rigor estatístico — por exemplo, se uma campanha realmente aumentou as vendas ou se a diferença observada é apenas ruído.

DESAFIO DO MÓDULO

Use um teste de hipótese em Python para avaliar se a diferença de conversão entre dois grupos de clientes é estatisticamente significativa ou apenas variação aleatória.

ERRO COMUM A EVITAR

Interpretar qualquer diferença numérica entre dois grupos como 'significativa' sem aplicar um teste de hipótese formal é um erro estatístico grave e recorrente em análises de negócio.

DICA DO AUTOR

Sempre reporte o p-valor junto com o tamanho do efeito — significância estatística sem relevância prática não ajuda ninguém a decidir.

O QUE VOCÊ VAI CONQUISTAR

Você será capaz de aplicar testes de hipótese e interpretar medidas estatísticas para embasar decisões de negócio com rigor.

- ♦ Leitura crítica de medidas estatísticas
- ♦ Testes de hipótese aplicados
- ♦ Base para validação de modelos de ML

05 Eng. de Variáveis → 06 Estatística → 07 PCA

FERRAMENTAS DO MÓDULO

[scipy.stats](#)

[statsmodels](#)

CONTEXTO

Bases financeiras costumam ter dezenas de indicadores correlacionados entre si. O PCA (Análise de Componentes Principais) resolve esse problema, reduzindo a dimensionalidade dos dados sem perder informação relevante — fundamental tanto para visualização quanto para alimentar outros modelos com menos ruído.

PRINCIPAIS TÓPICOS

- Redução de dimensionalidade com PCA
- Interpretação dos componentes principais
- Variância explicada e Scree Plot
- Geração de scores financeiros

NA PRÁTICA

Você vai aplicar PCA a um conjunto de indicadores financeiros de empresas ou clientes, interpretar o Scree Plot para decidir quantos componentes reter, e gerar scores que resumem o perfil financeiro de cada observação.

DESAFIO DO MÓDULO

Aplique PCA a uma base de indicadores financeiros e decida, com base na variância explicada, quantos componentes principais são necessários para representar a maior parte da informação.

ERRO COMUM A EVITAR

Aplicar PCA em variáveis não padronizadas é um erro comum e custoso — variáveis em escalas maiores (como faturamento em milhões) dominam artificialmente o resultado.

DICA DO AUTOR

Os componentes principais não têm significado de negócio automático — é seu trabalho interpretar o que cada componente representa na prática financeira.

O QUE VOCÊ VAI CONQUISTAR

Você será capaz de reduzir a dimensionalidade de indicadores financeiros complexos, preservando a informação relevante para análises seguintes.

- ♦ Redução de dimensionalidade com PCA
- ♦ Interpretação de variância explicada
- ♦ Geração de scores para outros modelos

06 Estatística → 07 PCA → 08 K-means

FERRAMENTAS DO MÓDULO

[scikit-learn \(PCA\)](#)

08

K-means — Segmentação Produtos Bancários

Agrupe clientes por perfil de consumo de produtos bancários.

CONTEXTO

Depois de reduzir a dimensionalidade com PCA, o próximo passo natural é agrupar clientes com comportamentos semelhantes entre si. Este módulo ensina o K-means aplicado à segmentação de clientes por uso de produtos bancários — um dos casos de uso mais valiosos de Machine Learning em marketing financeiro.

PRINCIPAIS TÓPICOS

- Análise de cluster e distância euclidiana
- K-means passo a passo
- Método do cotovelo (elbow method)
- Interpretação de perfis de cluster

NA PRÁTICA

Você vai segmentar uma base de clientes bancários em grupos com perfis de consumo semelhantes — por exemplo, por uso de crédito, investimentos e conta corrente — e vai interpretar o que cada cluster representa para o negócio.

DESAFIO DO MÓDULO

Segmente uma base de clientes bancários em 3 a 5 clusters com K-means e escreva, para cada cluster, uma frase que resuma o perfil de consumo predominante daquele grupo.

O QUE VOCÊ VAI CONQUISTAR

Você será capaz de segmentar clientes em grupos homogêneos por comportamento e de interpretar o perfil de cada cluster formado.

- ♦ Segmentação de clientes bancários
- ♦ Escolha do número ideal de clusters
- ♦ Interpretação de perfis para o negócio

07 PCA → 08 K-means → 09 Regressão

FERRAMENTAS DO MÓDULO

[scikit-learn \(KMeans\)](#)

ERRO COMUM A EVITAR

Escolher o número de clusters (K) de forma arbitrária, sem usar o método do cotovelo ou outro critério objetivo, é um erro frequente que gera segmentações pouco interpretáveis.

DICA DO AUTOR

Um cluster de clientes só tem valor se puder orientar uma ação concreta — uma oferta, um produto, uma campanha. Se você não imagina essa ação, o cluster ainda não está bem definido.

09

Regressão Logística e Linear

Preveja valores numéricos e probabilidades com os dois modelos mais usados no mercado.

CONTEXTO

Este módulo marca a entrada nos algoritmos preditivos supervisionados: Regressão Linear para prever valores numéricos contínuos, e Regressão Logística para prever probabilidades e classificar categorias — os dois modelos mais usados e mais fáceis de explicar a uma diretoria de negócio.

PRINCIPAIS TÓPICOS

- Regressão Linear simples e múltipla
- Regressão Logística e odds ratio
- Interpretação de coeficientes
- Métricas de avaliação (R^2 , matriz de confusão)

NA PRÁTICA

Você vai construir um modelo de Regressão Linear para prever uma métrica de negócio contínua, e um modelo de Regressão Logística para estimar a probabilidade de um evento binário, interpretando os coeficientes de cada um.

DESAFIO DO MÓDULO

Construa um modelo de Regressão Linear e um de Regressão Logística sobre o mesmo conjunto de dados de negócio, e explique em que cenário cada um seria mais adequado.

O QUE VOCÊ VAI CONQUISTAR

Você será capaz de construir, avaliar e interpretar modelos de regressão e classificação com os dois algoritmos mais usados no mercado.

- ♦ Modelagem com Regressão Linear e Logística
- ♦ Interpretação de coeficientes
- ♦ Benchmark para modelos mais complexos

08 K-means → **09 Regressão** → 10 RF
Inadimplência

FERRAMENTAS DO MÓDULO

[statsmodels](#)

[scikit-learn](#)

ERRO COMUM A EVITAR

Interpretar um R^2 alto como garantia de um bom modelo, sem validar em dados de teste, é um erro comum — o modelo pode se ajustar bem ao treino e generalizar mal.

DICA DO AUTOR

Regressão Linear e Logística são os modelos mais fáceis de explicar a uma diretoria — comece sempre por eles antes de justificar um modelo mais complexo.

10

RandomForest — Previsão de Inadimplência em Reais

Estime o valor esperado da perda por inadimplência com um modelo de ensemble.

CONTEXTO

Nem toda relação entre variáveis é linear. Este módulo apresenta o Random Forest em sua versão de regressão, aplicado a um problema financeiro de alto impacto: estimar, em reais, o valor esperado da perda por inadimplência de cada cliente.

PRINCIPAIS TÓPICOS

- Random Forest — regressão
- Estimativa de valor esperado da perda
- Importância de variáveis
- Comparação com Regressão Linear

NA PRÁTICA

Você vai treinar um Random Forest para estimar o valor esperado da perda por inadimplência de uma carteira de clientes, comparando o erro (RMSE) desse modelo com o de uma Regressão Linear simples.

DESAFIO DO MÓDULO

Treine um Random Forest para estimar a perda esperada por inadimplência de uma base de clientes e compare o RMSE com o de uma Regressão Linear — recomende qual modelo usar e por quê.

ERRO COMUM A EVITAR

Escolher um algoritmo mais complexo só porque é mais 'moderno', sem comparar sua performance com modelos mais simples, é um erro comum — às vezes uma Regressão Linear bem construída já é suficiente.

DICA DO AUTOR

A importância de variáveis do Random Forest é uma ótima forma de validar, com dados, se as regras de crédito atuais da empresa fazem sentido estatístico.

O QUE VOCÊ VAI CONQUISTAR

Você será capaz de aplicar o Random Forest para estimar valores financeiros de risco, como a perda esperada por inadimplência.

- ♦ Modelagem com Random Forest (regressão)
- ♦ Leitura de importância de variáveis
- ♦ Estimativa de valor esperado da perda

09 Regressão → 10 RF Inadimplência → 11 RF Diabetes

FERRAMENTAS DO MÓDULO

[scikit-learn \(RandomForestRegressor\)](#)

11

RandomForest — Classificação de Diabetes com Precisão

Classifique risco de diabetes priorizando a métrica certa para saúde.

CONTEXTO

O Random Forest também se adapta a problemas de classificação. Este módulo aplica o mesmo algoritmo à predição de diagnóstico de diabetes a partir de variáveis clínicas, com foco especial em precisão e nas métricas certas para decisões de saúde.

PRINCIPAIS TÓPICOS

- Random Forest — classificação
- Precisão, recall e F1-score
- Importância de variáveis clínicas
- Trade-off entre falsos positivos e negativos

NA PRÁTICA

Você vai treinar um Random Forest para classificar o risco de diabetes de um paciente a partir de variáveis clínicas, e vai decidir qual métrica priorizar diante do custo de cada tipo de erro.

DESAFIO DO MÓDULO

Treine um Random Forest de classificação em uma base de diabetes e monte uma tabela com precisão, recall e F1-score, justificando qual métrica deveria orientar a decisão clínica.

O QUE VOCÊ VAI CONQUISTAR

Você será capaz de aplicar o Random Forest a problemas de classificação de risco em saúde, escolhendo a métrica certa para cada decisão.

- ♦ Classificação com Random Forest
- ♦ Escolha de métricas por contexto de negócio
- ♦ Avaliação por matriz de confusão

10 RF Inadimplência → 11 RF Diabetes → 12 XGB Energético

FERRAMENTAS DO MÓDULO

[scikit-learn \(RandomForestClassifier\)](#)

ERRO COMUM A EVITAR

Avaliar um classificador de saúde apenas pela acurácia, especialmente em bases desbalanceadas, é um erro grave — recall e precisão contam uma história muito mais honesta.

DICA DO AUTOR

Em problemas de saúde, o custo de um falso negativo costuma ser muito maior que o de um falso positivo — deixe isso guiar sua escolha de métrica e de ponto de corte.

12

XGBoosting — Previsão de Consumo Energético

Preveja consumo de energia com um dos algoritmos de ensemble mais usados no mercado.

CONTEXTO

Este módulo aplica o XGBoosting, um dos algoritmos de ensemble mais usados no mercado, a um problema de operações: prever o consumo de energia a partir de variáveis históricas e operacionais — informação crítica para planejamento de capacidade e custos.

PRINCIPAIS TÓPICOS

- XGBoosting — regressão
- Importância de variáveis
- Principais hiperparâmetros
- Comparação com outros algoritmos de regressão

NA PRÁTICA

Você vai treinar um XGBoosting para prever o consumo de energia a partir de variáveis operacionais e sazonais, e vai comparar sua performance com a de um Random Forest treinado anteriormente.

DESAFIO DO MÓDULO

Treine um XGBoosting para prever o consumo de energia de uma base operacional e compare o RMSE com o do Random Forest do módulo anterior.

O QUE VOCÊ VAI CONQUISTAR

Você será capaz de aplicar o XGBoosting a problemas de previsão numérica em contextos operacionais e de energia.

- ♦ Modelagem com XGBoosting
- ♦ Ajuste de hiperparâmetros
- ♦ Aplicação em planejamento operacional

11 RF Diabetes → 12 XGB Energético → 13 XGB Fraude

FERRAMENTAS DO MÓDULO

xgboost (Regressor)

ERRO COMUM A EVITAR

Usar o XGBoosting com hiperparâmetros padrão, sem ajustar taxa de aprendizado e profundidade das árvores, é um erro comum que pode limitar bastante a performance do modelo.

DICA DO AUTOR

Em problemas de planejamento operacional, um erro de previsão tem custo direto em capacidade — sempre valide o modelo com dados de teste antes de usá-lo em um planejamento real.

13

XGBoosting — Fraudes em Cartões de Crédito

Detecte transações fraudulentas em bases fortemente desbalanceadas.

CONTEXTO

Detecção de fraude é um dos problemas mais desafiadores de Machine Learning por conta do desbalanceamento extremo entre classes. Este módulo aplica o XGBoosting à classificação de transações fraudulentas em cartões de crédito, tratando diretamente esse desbalanceamento.

PRINCIPAIS TÓPICOS

- XGBoosting — classificação
- Tratamento de bases desbalanceadas
- Curva ROC e AUC
- Ponto de corte (threshold) para fraude

NA PRÁTICA

Você vai treinar um XGBoosting para classificar transações como fraudulentas ou legítimas, tratando o desbalanceamento entre classes e definindo um ponto de corte que equilibre detecção de fraude e experiência do cliente.

DESAFIO DO MÓDULO

Treine um XGBoosting em uma base de fraude de cartões e defina, com base na curva ROC, um ponto de corte que você recomendaria para produção — justificando o trade-off escolhido.

ERRO COMUM A EVITAR

Avaliar um modelo de fraude pela acurácia geral é um erro grave e recorrente — em bases com 1% de fraude, um modelo que nunca prevê fraude já teria 99% de acurácia e seria inútil.

DICA DO AUTOR

Em fraude, a curva ROC e a AUC contam a história real do modelo — a acurácia isolada, quase nunca.

O QUE VOCÊ VAI CONQUISTAR

Você será capaz de aplicar o XGBoosting a problemas de detecção de fraude, tratando corretamente bases fortemente desbalanceadas.

- ♦ Classificação com XGBoosting
- ♦ Tratamento de desbalanceamento de classes
- ♦ Definição de ponto de corte para produção

12 XGB Energético → 13 XGB Fraude → 14 SVR Cliente

FERRAMENTAS DO MÓDULO

xgboost (Classifier)

14

SVR — Análise Estratégica do Comportamento do Cliente

Estime métricas contínuas de comportamento do cliente com Support Vector Regression.

CONTEXTO

Este módulo apresenta o SVR (Support Vector Regression), aplicado à estimativa de métricas contínuas de comportamento do cliente — como valor de vida útil (LTV) ou propensão de gasto — a partir de variáveis de relacionamento e histórico de compra.

PRINCIPAIS TÓPICOS

- SVR — Support Vector Regression
- Padronização de variáveis para SVR
- Kernels e hiperparâmetros
- Comparação entre algoritmos de regressão

NA PRÁTICA

Você vai treinar um SVR para estimar uma métrica contínua de comportamento do cliente, fechando a comparação entre os algoritmos de regressão estudados na formação — Linear, Random Forest, XGBoosting e SVR.

DESAFIO DO MÓDULO

Treine um SVR para estimar uma métrica de comportamento do cliente e monte uma tabela comparativa de RMSE com os demais algoritmos de regressão estudados.

ERRO COMUM A EVITAR

Usar o SVR com hiperparâmetros padrão, sem testar diferentes kernels, é um erro comum que pode limitar bastante a performance do modelo.

DICA DO AUTOR

SVR tende a funcionar melhor em bases com variáveis bem padronizadas — revise a engenharia de variáveis do Módulo 5 antes de treinar o modelo.

O QUE VOCÊ VAI CONQUISTAR

Você será capaz de aplicar o SVR a problemas estratégicos de comportamento do cliente e comparar criticamente os principais algoritmos de regressão.

- ♦ Modelagem com SVR
- ♦ Comparação entre quatro algoritmos de regressão
- ♦ Análise estratégica de comportamento do cliente

13 XGB Fraude → 14 SVR Cliente → 15 SVM Churn

FERRAMENTAS DO MÓDULO

[scikit-learn \(SVR\)](#)

15

SVM — Predict Customer Churn

Preveja cancelamento de clientes com margens de separação bem definidas.

CONTEXTO

Este módulo aplica o SVM (Support Vector Machine) à predição de churn — um dos problemas de classificação mais comuns em negócios com receita recorrente — em que a margem de separação entre clientes que ficam e clientes que cancelam é central para a qualidade do modelo.

PRINCIPAIS TÓPICOS

- SVM — classificação
- Aplicação em predição de churn
- Kernels e margem de separação
- Comparação com outros classificadores

NA PRÁTICA

Você vai treinar um SVM para prever o cancelamento de clientes a partir de variáveis de uso e relacionamento, e vai comparar sua performance com a de outros classificadores estudados na formação.

DESAFIO DO MÓDULO

Treine um SVM em uma base de churn e monte uma tabela comparativa de acurácia, sensibilidade e especificidade com os demais classificadores estudados até aqui.

ERRO COMUM A EVITAR

Escolher o kernel do SVM sem testar alternativas é um erro comum — o kernel linear pode não capturar a separação real entre clientes que ficam e os que cancelam.

DICA DO AUTOR

Em churn, um falso negativo (não identificar um cliente que vai cancelar) tem custo direto em receita perdida — priorize a sensibilidade do modelo nessa decisão.

O QUE VOCÊ VAI CONQUISTAR

Você será capaz de aplicar o SVM a problemas de classificação de churn e de comparar criticamente os principais algoritmos de classificação.

- ♦ Classificação com SVM
- ♦ Leitura de margem de separação
- ♦ Predição de cancelamento de clientes

14 SVR Cliente → 15 SVM Churn → 16 Decision Tree

FERRAMENTAS DO MÓDULO

[scikit-learn \(SVC\)](#)

16

Decision Tree: Satisfação do Cliente — Classificação

Classifique satisfação do cliente com o modelo mais interpretável de todos.

CONTEXTO

Este módulo apresenta a Árvore de Decisão aplicada à predição de satisfação do cliente — um dos modelos mais interpretáveis de Machine Learning, capaz de ser lido diretamente como um conjunto de regras de negócio pela equipe de atendimento.

PRINCIPAIS TÓPICOS

- Árvore de Decisão — classificação
- Poda da árvore (pruning)
- Leitura da árvore e regras de decisão
- Aplicação em satisfação do cliente

NA PRÁTICA

Você vai treinar uma Árvore de Decisão para classificar o nível de satisfação de um cliente a partir de variáveis de atendimento e uso, e vai ler a árvore resultante como regras explicáveis para a equipe de CX.

DESAFIO DO MÓDULO

Treine uma Árvore de Decisão em uma base de satisfação do cliente e escreva, em linguagem de negócio, as duas regras de decisão mais relevantes encontradas pela árvore.

O QUE VOCÊ VAI CONQUISTAR

Você será capaz de aplicar a Árvore de Decisão para prever satisfação do cliente e de traduzir o modelo em regras de negócio explicáveis.

- ♦ Classificação com Árvore de Decisão
- ♦ Leitura de regras de decisão
- ♦ Avaliação por matriz de confusão

15 SVM Churn → 16 Decision Tree → 17 Apriori

FERRAMENTAS DO MÓDULO

scikit-learn (DecisionTreeClassifier)

ERRO COMUM A EVITAR

Deixar a árvore crescer sem limite de profundidade é um erro comum que causa overfitting e produz árvores tão complexas que perdem a principal vantagem do modelo: a interpretabilidade.

DICA DO AUTOR

Uma árvore de decisão só é útil para a equipe de atendimento se puder ser lida como um fluxograma de regras — evite árvores tão profundas que percam essa clareza.

17

Apriori — Marketing Digital Orientado por Dados

Descubra regras de associação entre produtos para estratégias de cross-sell.

CONTEXTO

Este módulo introduz uma família diferente de algoritmos: sistemas de recomendação baseados em regras de associação. O Algoritmo Apriori identifica padrões do tipo 'quem compra A também compra B', a lógica por trás de recomendações e campanhas de marketing digital orientadas por dados.

PRINCIPAIS TÓPICOS

- Algoritmo Apriori
- Regras de associação
- Suporte, confiança e lift
- Aplicação em campanhas de cross-sell

NA PRÁTICA

Você vai aplicar o Apriori a uma base de transações para descobrir regras do tipo 'clientes que compram o produto A também compram o produto B com frequência X' — a lógica por trás de campanhas de cross-sell.

DESAFIO DO MÓDULO

Aplique o Apriori a uma base de transações e identifique a regra de associação com maior lift — explique o que essa regra significaria para uma campanha de marketing.

O QUE VOCÊ VAI CONQUISTAR

Você será capaz de aplicar o Algoritmo Apriori para identificar regras de associação e apoiar campanhas de marketing digital.

- ♦ Regras de associação (Apriori)
- ♦ Leitura de suporte, confiança e lift
- ♦ Estratégias de cross-sell orientadas por dados

16 Decision Tree → 17 **Apriori** → 18 KNN

FERRAMENTAS DO MÓDULO

[mlxtend \(apriori\)](#)

ERRO COMUM A EVITAR

Interpretar uma regra com alto suporte, mas baixo lift, como uma descoberta relevante é um erro comum — é o lift que indica se a associação é realmente mais forte que o esperado pelo acaso.

DICA DO AUTOR

Regras de associação geram muitas combinações — filtre por lift e valide se a regra faz sentido de negócio antes de transformá-la em campanha.

18

KNN — Recomendações Inteligentes

Recomende produtos ou conteúdos com base na proximidade entre perfis semelhantes.

CONTEXTO

Este módulo apresenta o KNN (K-Nearest Neighbors) aplicado a recomendações inteligentes: identificar o que sugerir a um cliente com base em outros clientes com perfil semelhante. Por depender diretamente da distância entre pontos, o KNN é um dos algoritmos mais sensíveis à escala das variáveis.

PRINCIPAIS TÓPICOS

- KNN — K-Nearest Neighbors
- Sensibilidade à escala das variáveis
- Escolha do valor de K
- Aplicação em sistemas de recomendação

NA PRÁTICA

Você vai treinar um KNN para recomendar produtos a um cliente com base em clientes de perfil semelhante, testando diferentes valores de K e comparando a qualidade das recomendações.

DESAFIO DO MÓDULO

Treine um recomendador KNN em uma base de clientes, testando pelo menos três valores diferentes de K, e avalie qual configuração gera recomendações mais consistentes.

O QUE VOCÊ VAI CONQUISTAR

Você será capaz de aplicar o KNN a sistemas de recomendação e de reconhecer quando a escala das variáveis compromete o resultado do modelo.

- ♦ Recomendação com KNN
- ♦ Escolha do valor de K
- ♦ Reforço da importância da padronização

17 Apriori → 18 KNN → 19 Séries Temporais

FERRAMENTAS DO MÓDULO

[scikit-learn \(KNeighbors\)](#)

ERRO COMUM A EVITAR

Treinar um KNN sem padronizar as variáveis é um erro comum e especialmente grave — como o algoritmo se baseia em distância, uma variável em escala maior domina artificialmente o resultado.

DICA DO AUTOR

Não existe um K 'certo' universal — teste alguns valores e escolha aquele que equilibra qualidade de recomendação e capacidade de generalização.

19

Séries Temporais — Análise da Depreciação de Produtos, Máquinas e Veículos

Preveja a depreciação de ativos ao longo do tempo com modelos de séries temporais.

CONTEXTO

Este módulo encerra a formação com séries temporais, aplicadas a um problema recorrente em operações e finanças: prever a depreciação de produtos, máquinas e veículos ao longo do tempo — informação essencial para decisões de reposição, seguro e revenda de ativos.

PRINCIPAIS TÓPICOS

- Componentes de uma série temporal (tendência, sazonalidade)
- Modelos ARIMA/Prophet
- Estacionariedade e testes
- Avaliação de previsões de séries temporais

NA PRÁTICA

Você vai modelar a curva de depreciação de um ativo (produto, máquina ou veículo) ao longo do tempo, decompondo tendência e sazonalidade, e vai gerar uma previsão de valor futuro com intervalo de confiança.

DESAFIO DO MÓDULO

Modele a série de depreciação de um ativo e gere uma previsão de valor para os próximos períodos, indicando o intervalo de confiança da previsão.

O QUE VOCÊ VAI CONQUISTAR

Você encerra a formação sendo capaz de modelar séries temporais e prever a depreciação de ativos com rigor estatístico.

- ♦ Modelagem de séries temporais
- ♦ Decomposição de tendência e sazonalidade
- ♦ Previsão de depreciação de ativos

18 KNN → 19 Séries Temporais → 20 Projeto Final

FERRAMENTAS DO MÓDULO

statsmodels

Prophet

ERRO COMUM A EVITAR

Aplicar um modelo de série temporal sem verificar estacionariedade é um erro comum que compromete a validade estatística de toda a previsão gerada.

DICA DO AUTOR

Sempre visualize a série antes de modelar — tendência e sazonalidade costumam ser visíveis a olho nu e orientam a escolha do modelo certo.

CONTEXTO

Este é o projeto final da formação — o momento em que você integra estatística descritiva, engenharia de variáveis e algoritmos de regressão e classificação para resolver um dos problemas mais valiosos de Machine Learning aplicado a negócios: onde e quanto uma empresa pode economizar sem comprometer resultado ou qualidade. Redução de custo é um mandato constante em operações, logística, energia e atendimento — e é aqui que você une Regressão Linear, Random Forest e XGBoosting (Módulos 9, 10 e 12) para construir, do zero, um modelo defensável perante uma diretoria de operações.

PRINCIPAIS TÓPICOS

- Definição do problema de redução de custo
- Regressão e ensemble aplicados a custo (Linear, Random Forest, XGBoosting)
- Simulação de cenários e impacto financeiro
- Engenharia de variáveis de custo e operação
- Identificação de variáveis com maior potencial de economia
- Apresentação executiva do modelo

NA PRÁTICA

Usando uma base prática de custos operacionais, você vai construir e comparar modelos de Regressão Linear, Random Forest e XGBoosting para prever um custo-alvo, identificar as variáveis com maior peso na formação desse custo, simular o impacto financeiro de agir sobre cada uma delas, e encerrar com uma apresentação executiva das oportunidades de economia encontradas.

DESAFIO DO MÓDULO

Compare Regressão Linear com pelo menos um algoritmo de ensemble (Random Forest ou XGBoosting) no mesmo problema de custo, aponte as duas variáveis com maior potencial de redução e estime, em reais ou percentual, o impacto financeiro de agir sobre elas.

ERRO COMUM A EVITAR

Apontar uma variável como 'oportunidade de redução de custo' apenas porque tem alta importância no modelo, sem avaliar se ela é de fato controlável pela operação, é um erro estratégico grave — nem toda variável importante é uma palavra acionável.

DICA DO AUTOR

Um modelo de redução de custo só é útil se termina em uma recomendação que alguém pode executar. Sempre traduza a saída do modelo em uma ação concreta e seu impacto financeiro estimado antes de apresentá-la à diretoria.

O QUE VOCÊ VAI CONQUISTAR

Você vai entregar um projeto prático completo de redução de custo, do dado bruto à recomendação executiva — aplicando de forma integrada engenharia de variáveis, regressão e os algoritmos de ensemble da formação.

- Projeto de portfólio pronto para currículo
- Modelo de redução de custo completo
- Simulação de impacto financeiro por cenário

19 Séries Temporais → 20 Projeto Final → Fim da formação

FERRAMENTAS DO MÓDULO

[pandas](#) [scikit-learn](#) [xgboost](#)

POR QUE ESTE É O ENCERRAMENTO IDEAL

Redução de Custo integra praticamente todos os blocos da formação — estatística descritiva, engenharia de variáveis, regressão e ensemble — em um problema de negócio prático e mensurável financeiramente. O encerramento perfeito para comprovar domínio de ponta a ponta em Machine Learning com Python.

O QUE VOCÊ VAI DOMINAR

Ao concluir a formação, você terá saído do zero à construção de modelos de Machine Learning

Ao concluir a formação em **Ciência de Dados, Estatística e Modelagem com Python**, você terá percorrido um caminho completo — dos fundamentos da linguagem ao projeto final de redução de custo. Mais do que dominar doze algoritmos isoladamente, você terá desenvolvido a capacidade de escolher a técnica certa para cada tipo de problema, validar sua performance com rigor e traduzir a saída de um modelo em uma decisão de negócio defensável.

SUA JORNADA COMPLETA



- Programação em Python, Pandas e Numpy para análise de dados
- Processamento de alta performance com Datatable e Polars
- Engenharia de variáveis e tratamento de dados faltantes
- Estatística descritiva e testes de hipótese
- Redução de dimensionalidade com PCA
- Segmentação de clientes com K-means
- Regressão aplicada a crédito, energia e comportamento do cliente (Linear, Random Forest, XGBoosting, SVR)
- Classificação aplicada a saúde, fraude e churn (Árvore, Random Forest, XGBoosting, SVM)
- Sistemas de recomendação com Apriori e KNN
- Modelagem de séries temporais para previsão de depreciação
- Projeto final integrador: diagnóstico e redução de custo

PARA ONDE ESSA FORMAÇÃO PODE TE LEVAR

O conjunto de competências construído nesta formação é a base analítica exigida em posições como:

Cientista de Dados Analista de Machine Learning Analista de Risco / Crédito Analista de CRM / Marketing Analytics

Consultor(a) de Data Science

Engenheiro(a) de Modelos Preditivos

Pronto para construir modelos que decidem com você?

Comece agora sua formação em Ciência de Dados, Estatística e Modelagem com Python.