ANA RAQUEL

PÓS TECH

DATA ANALYTICS

DEPLOY DE APLICAÇÕES

AULA 01

SUMÁRIO

O QUE VEM POR AÍ?

Você aprendeu, ao longo da jornada de aprendizado de máquina, as técnicas e algoritmos necessários para construir um modelo de aprendizado de máquina, mas como será que podemos colocar esses modelos em produção? Como esse modelo consegue ser utilizado na ponta final? Chegou o momento de aprender uma das etapas primordiais do pipeline de algoritmos de Machine Learning, o tratamento de dados!

HANDS ON

Para introduzir o assunto, nas aulas de deploy de aplicações utilizaremos como exemplo um case sobre **análise de crédito**, especificamente sobre a etapa de concessão. Basicamente, essa etapa concede recursos que possibilitam ao indivíduo ter capital para realizar a aquisição de bens ou serviços. Para isso, é realizada a **análise de alguns indicadores e informações pessoais** e, então, é concedido o crédito.

Aqui, você já deve estar imaginando, com base nas fases anteriores e tudo o que aprendeu até agora, é possível realizar um modelo de Machine Learning para aprovar a concessão de crédito, certo?

Nas aulas dessa disciplina, passaremos por todos os passos necessários para realizar o pipeline e colocar o modelo em produção. O nosso objetivo aqui será construir um **modelo para prever se um cliente é "bom pagador" ou "mau pagador", de acordo com as informações fornecidas**. A ponta final do nosso projeto será uma aplicação que irá receber os dados desses clientes e, então, tomar a decisão de aprovação de crédito.

Para essa aula, temos um notebook disponível. Acesse-o no Github da nossa disciplina!

SAIBA MAIS

Um dos primeiros passos quando vamos construir um modelo de Machine Learning é realizar algumas etapas primordiais, dentro do conjunto de etapas que podemos definir como "**Data Clean"**. Na etapa de limpeza e tratamento dos dados, extrairemos os dados de alguma origem e começaremos a **conhecer os dados**. Ao conhecê-los, começamos a **encontrar algumas sujeiras e discrepâncias**. Vamos detalhar um pouco mais o que são essas inconsistências?

PREPARE OS DADOS PARA ALGORITMOS DO APRENDIZADO DE MÁQUINA

É hora de preparar os dados para os seus algoritmos de aprendizado de máquina! Em vez de fazer esse trabalho manualmente, eu te convido a pensar em escrever funções, pelos seguintes motivos:

- Reproduzir as transformações facilmente em qualquer conjunto de dados.
- Utilizar as funções em sistemas ao vivo para transformar os novos dados, antes de fornecê-los aos seus algoritmos.

OS DADOS DO MUNDO TRANSACIONAL

Sabemos que os sistemas e softwares são alimentados por dados, e esses dados precisam ser armazenados em algum repositório, que normalmente são bancos de dados. Além dos sistemas serem alimentados por dados, eles também **geram dados**. Como os dados oriundos de sistemas são gerados por humanos, podemos encontrar algumas falhas. Por exemplo, pode acontecer de, em um sistema de cadastro, a pessoa registrar a data de nascimento errada, gerando assim um registro errado para o banco de dados. Sendo assim, podem ocorrer as chamadas **inconsistências nos dados**, que podem ser por **erros humanos ou sistêmicos**.

Nós, profissionais de dados, acabamos sendo impactados(as) por esses tipos de situações, pois analisamos e trabalhamos com dados que são oriundos do mundo transacional. É por isso que a etapa de data clean é super importante em todo trabalho que envolve dados. Caso os dados não sejam tratados, garantindo sua consistência,

o algoritmo pode tomar **decisões enviesadas**, perdendo assim seu potencial preditivo.

INICIANDO A ESTRATÉGIA DA CONSTRUÇÃO DO PIPELINE

Conheceremos agora quais são os passos primordiais para realizar o deploy de um modelo de aprendizado de máquina.

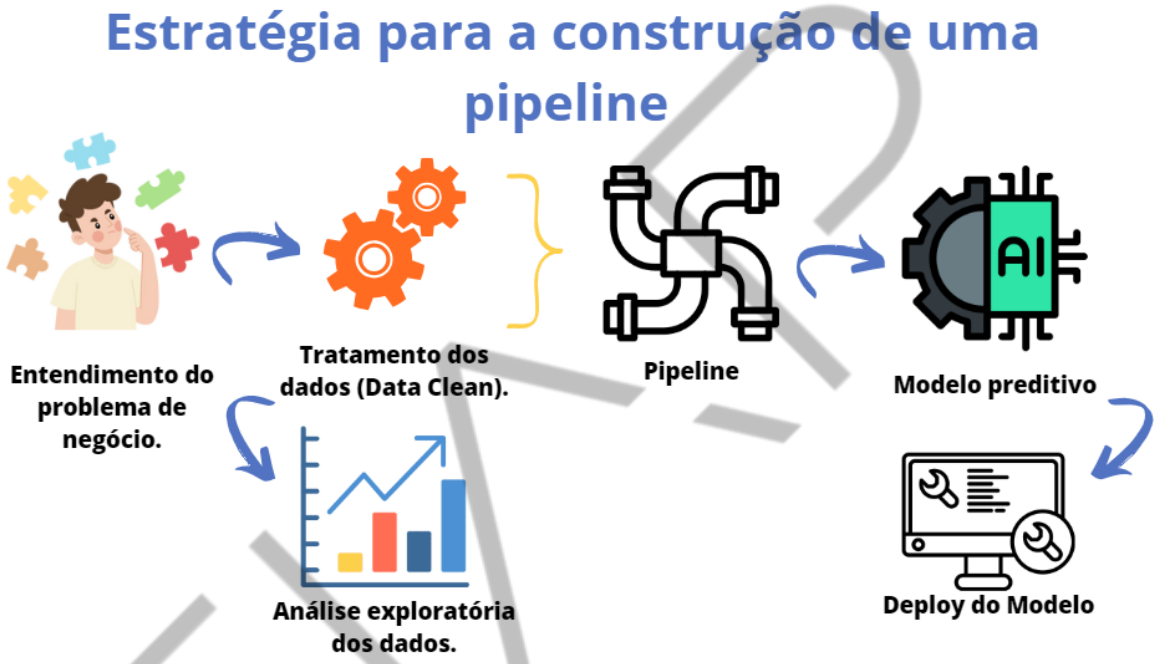

Figura 1 – Estratégia para a construção de uma pipeline para aprendizado de máquina
Fonte: Elaborado pela autora (2023)

ENTENDIMENTO DO PROBLEMA DE NEGÓCIO

Antes de detalharmos quais são as etapas de limpeza de dados necessárias para garantir um bom desempenho do algoritmo, é muito importante ressaltar essa fase.

Ao criar um algoritmo de Machine Learning, estamos criando uma **solução para um problema real de negócio**. É muito importante estudar os dados e conhecer as regras de negócio antes de iniciar a construção do modelo. No nosso case de exemplo, estamos construindo um pipeline para classificar quando um cliente é um “bom pagador” ou “mau pagador”, dada algumas características sobre análise de crédito. Nesse caso, é necessário entender como funciona o mundo de análise de

crédito e estudar algumas características, como análise de bens da pessoa, tipo de moradia, rendimento anual, grau de escolaridade, entre outros fatores que podem influenciar a tomada de decisão de concessão de crédito. Nessa etapa, é muito comum você trocar ideias e obter conhecimento com a área de negócio. É a etapa em que temos várias reuniões de alinhamento e entendimento do projeto.

TRATAMENTO DOS DADOS

Observe a figura 1, onde temos a figuração da estratégia de construção de uma pipeline de modelo de Machine Learning. O conjunto de etapas que definimos como **“Data Clean”** podem ser encaixadas dentro das etapas de **ETL (Tratamento de dados)**. Basicamente, o ETL (Extract, Transform and Load) consiste em realizar as etapas de extração, transformação e carregamento dos dados. Dentro das etapas de tratamento de dados, podemos **identificar os tipos de data types das variáveis** da base de dados, para direcionar a estratégia de limpeza dos dados. Então, nesse caso, conseguimos tratar as **variáveis qualitativas e variáveis quantitativas**.

ENTENDENDO OS TIPOS DE VARIÁVEIS

Figura 2 - Tipos de variáveis quantitativas e qualitativas.
Fonte: Elaborado pela autora (2023)

Qualitativas: podem ser chamadas de categorias. Definidas por categorias, podem representar classificações.

- **Qualitativa nominal:** não existe ordenação entre as categorias.
- **Qualitativa ordinal:** existe uma ordenação entre as categorias

Quantitativas: podem ser descritas por números.

- **Quantitativa discreta:** números que são resultados de contagens, obtendo assim números inteiros.
- **Quantitativa contínua:** números que são resultados de medições, obtendo assim números decimais.

É importante reconhecer os tipos de variáveis, pois cada tipo possui uma estratégia de tratamento diferente. Quando falamos de variáveis qualitativas, podemos, por exemplo, utilizar alguns comandos muito úteis no Python para nos auxiliar a entender quais são os tipos de categorias que existem dentro de uma coluna específica na base de dados, como o **unique()** ou o **set()**.

```
df_cadastrados_limpo['Ocupacao'].unique()
set(df_cadastrados_limpo['Ocupacao'])
```

Que tal experimentar esses comandos no Python? Eles são muito úteis quando precisamos analisar os itens únicos dentro de uma coluna de categoria.

Também para facilitar a análise, podemos utilizar comandos como, por exemplo, o **value_counts(),** para contabilizar quantas instâncias temos dentro de uma certa variável qualitativa.

```
df_cadastrados_limpo['Anos_empregado'].value_counts()
```

Quando realizamos análises para as variáveis do tipo quantitativas, podemos utilizar a nosso favor os gráficos, como **histogramas** ou **box plots,** para nos auxiliar na visualização das distribuições dos dados.

```
sns.histplot(data=df_cadastrados_limpo,
x='Rendimento_anual', bins=50)
plt.xticks(rotation=45)
plt.show()
```

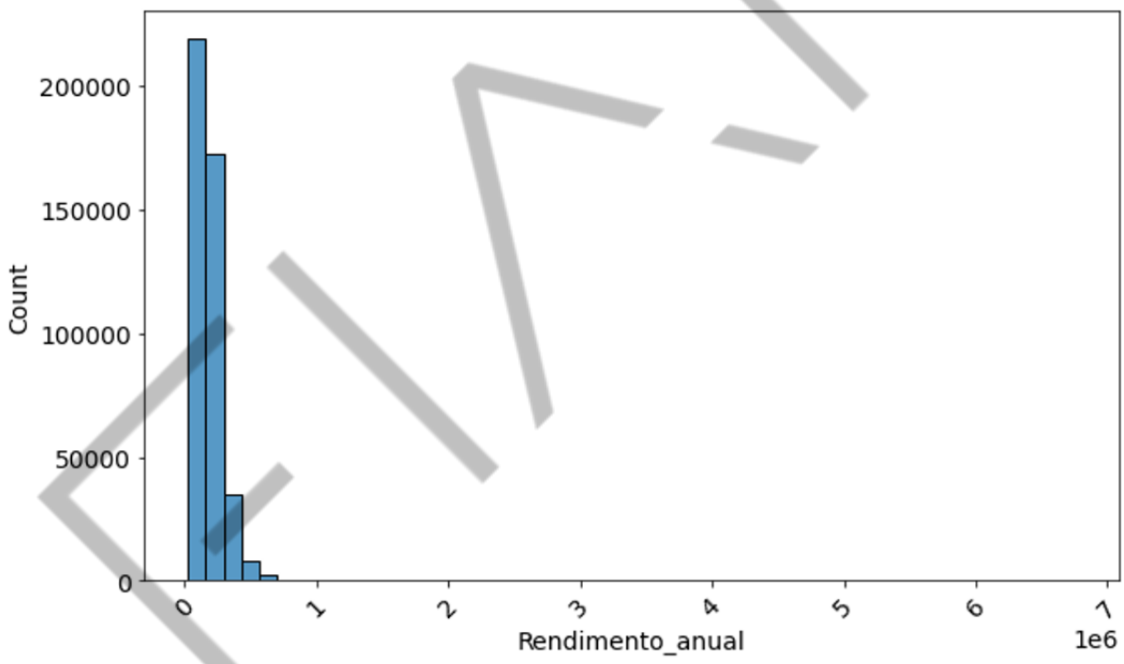

Figura 3 - Histograma para análise de variáveis quantitativas.
Fonte: Elaborado pela autora (2023)

COMO TRATAR DADOS?

Depois de identificar as inconsistências dos dados, podemos realizar algumas estratégias para limpar os dados, tais como os comandos **dropna(), fillna(), drop_duplicates()** e **drop().**

Remover linhas com valores ausentes:

df = df.dropna()

Preencher valores ausentes com um valor zero:

df = df.fillna(0)

Remover uma única coluna:

df = df.drop('coluna', axis=1)

Remover várias colunas:

df = df.drop(['coluna1', 'coluna2'], axis=1)

Preencher valores ausentes com a média dos valores não nulos:

df = df.fillna(df.mean())

O tratamento não consiste apenas em limpar os dados. Na sua jornada, você pode encontrar alguns desafios, como **realizar transformações nos dados** para **criar novas colunas,** ou **organizar os dados para um melhor entendimento** do negócio.

PADRONIZAÇÃO DOS DADOS QUALITATIVOS

Falando de **colunas qualitativas**, é preciso transformá-las em **representação binária ou numérica,** para o algoritmo conseguir processar esses dados. Aqui podemos pensar em técnicas como, por exemplo, **Label Encoding**. A Label Encoding é uma técnica que consiste em criar uma **representação numérica** para os dados no formato de categoria.

Também podemos pensar aqui em **One Hot Encoding**, técnica que cria uma matriz binária para representar a informação contida dentro da categoria, deixando 1, quando a informação é ativa, e 0, quando não é ativa para a linha observada.

Aqui embaixo estão os links das bibliotecas dentro do Sklearn que podem fazer essas transformações nos dados. Que tal dar uma olhada?

Label Encoder

One Hot Encoder

Caso queira revisar os conceitos de Label Encoding e One Hot Encoding, volte na aula 1 de Machine Learning Avançado para mais detalhes.

PADRONIZAÇÃO DE VARIÁVEIS QUANTITATIVAS

Além de padronizar as variáveis do tipo categórica, também é importante, no pipeline, **avaliar a necessidade da normalização ou padronização dos dados quantitativos**. Nesse caso, **depende muito do tipo de modelo a ser utilizado e de como estão as escalas dos dados**.

Por exemplo, comparar salário com metros não seria muito sensato em escala original dos dados, pois dinheiro e centímetros não estão na mesma unidade de medida. Esse tipo de escala diferente pode comprometer o comportamento de modelos, como os de classificação, que podem se confundir ao analisar dados com escalas diferentes. Já em modelos de regressão linear, por exemplo, a escala dos dados não afeta a performance do modelo, pois o objetivo da regressão é encontrar uma reta linear sobre a influência das variáveis independentes na variável dependente. Você pode realizar as transformações e comparar a performance do modelo antes e depois da aplicação de feature scaling. Para mais detalhes, também consultar a aula 1 de Machine Learning Avançado.

Para mais detalhes, acesse o Sklearn:

ANÁLISE EXPLORATÓRIA DOS DADOS

Perceba que, ao citar estratégias para analisar as variáveis quantitativas, dei a dica do uso de gráficos. Isso explica a importância da análise exploratória, que basicamente consiste em **realizar gráficos para entender o comportamento dos dados**, tais como padrões, tendências, correlações e identificar algumas **inconsistências**. Por exemplo, ao analisar uma variável quantitativa discreta sobre o número de filhos com um histograma, você pode, talvez, identificar uma simetria à esquerda, mostrando que sua distribuição pode ter alguma inconsistência.

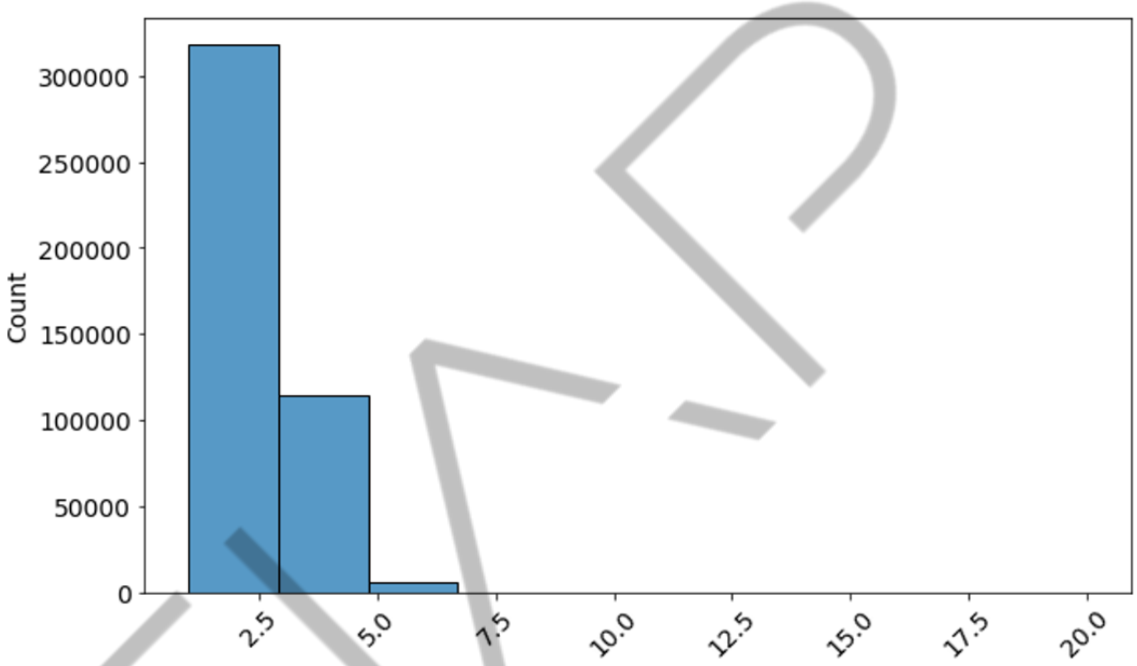

Figura 4 - Histograma para número de filhos
Fonte: Elaborado pela autora (2023)

O QUE VOCÊ VIU NESTA AULA?

Nessa aula, você revisou alguns conceitos fundamentais, já ensinados no começo da sua jornada, mas que fazem total diferença quando estamos criando um projeto de aprendizado de máquina. Limpar e tratar os dados é uma etapa muito importante e necessária em todo pipeline.

Tem alguma dúvida ou quer conversar sobre o tema desta aula? Entre em contato conosco pela comunidade do Discord! Lá você pode fazer networking, receber avisos, tirar dúvidas e muito mais.

REFERÊNCIAS

BRUCE, A. Estatística Prática para Cientistas de Dados. Sebastopol: O'Reilly Media, 2019.

Documentação SCIKIT-LEARN. **Scikit-learn**. Disponível em: <https://scikit-learn.org/stable/>. Acesso em: 03 ago 2023.

GÉRON, A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. Sebastapol: O'Reilly Media, 2019.

PALAVRAS-CHAVE

Tratamento de dados, variáveis quantitativas e qualitativas.