# EDA COM PCA

Um analista de dados geralmente começa identificando os ***objetivos da análise*** e os dados disponíveis. Com base nisso, ele pode escolher as técnicas de análise apropriadas.

## Análise exploratória de dados

**(EDA – Exploratory Data Analysis)** é a primeira etapa de qualquer análise de dados, onde o analista procura entender o conjunto de dados, identificando padrões, outliers, tendências e relações entre variáveis.

## **Estatísticas descritivas**

São usadas para resumir e descrever os dados, incluindo média, mediana, desvio padrão, etc.

Análise de regressão: é usada para entender como uma variável dependente é afetada por uma ou mais variáveis independentes.

## Análise de cluster

É usada para agrupar dados similares em grupos distintos.

## Análise de séries temporais

É usada para entender como uma variável muda ao longo do tempo.

## Aprendizado de máquina

É uma técnica avançada que é usada para prever ou classificar dados usando algoritmos de computador.

O analista deve avaliar ***as necessidades*** e recursos disponíveis para selecionar a técnica mais adequada para a análise. A *EDA* é a primeira etapa para todas as análises, pois permite entender a qualidade e a natureza dos dados.

### Códigos 1

import pandas as pd

import matplotlib.pyplot as plt

**# carregar o conjunto de dados**

df = pd.read\_csv("data.csv")

**# ver as primeiras linhas do conjunto de dados**

print(df.head())

**# ver as informações do conjunto de dados**

print(df.info())

**# ver a descrição estatística do conjunto de dados**

print(df.describe())

**# plotar um gráfico de histograma para uma variável**

df['column\_name'].hist()

plt.show()

import seaborn as sns

**# verificar se há valores ausentes**

print(df.isnull().sum())

**# plotar um gráfico de histograma para cada variável numérica**

df.hist(bins=10, figsize=(20,15))

plt.tight\_layout()

plt.show()

**# plotar um gráfico de densidade para cada variável numérica**

df.plot(kind='density', subplots=True, layout=(3,3), sharex=False, legend=False, fontsize=1)

plt.show()

**# plotar um gráfico de boxplot para cada variável numérica**

df.plot(kind='box', subplots=True, layout=(3,3), sharex=False, sharey=False, fontsize=8)

plt.show()

**# plotar um gráfico de scatter matrix para as variáveis numéricas**

sns.pairplot(df,diag\_kind='kde')

plt.show()

**#plotar um gráfico de barras para as variáveis categóricas**

for column in df.select\_dtypes(include=['object']).columns: df[column].value\_counts().plot(kind='bar', figsize=(10, 5)) plt.title(column) plt.ylabel('Frequency')

plt.show()

**#plotar um gráfico de heatmap para ver a correlação entre as variáveis numéricas**

corr = df.corr() sns.heatmap(corr, annot=True)

plt.show()

Este exemplo mostra várias técnicas de *EDA* adicionais, como verificação de valores ausentes, plotagem de gráficos de *histograma, densidade, boxplot, scatter matrix e heatmap*. Ele também mostra como explorar variáveis categóricas usando gráficos de barras. Ele usa as bibliotecas *pandas, seaborn e matplotlib* para carregar e manipular os dados, e para plotar gráficos. Isso ajudaria a explorar os dados melhor e obter uma compreensão geral sobre o conjunto de dados.

Além das técnicas mencionadas anteriormente, existem outras técnicas avançadas de *EDA* que podem ser usadas dependendo do objetivo da análise e do conjunto de dados em questão.

Alguns exemplos incluem:

## Análise de componentes principais (PCA)

Essa técnica é usada para identificar os principais componentes de variação nos dados e pode ser útil para reduzir a dimensionalidade dos dados.

## Análise de cluster

Essa técnica é usada para agrupar dados similares em grupos (ou clusters) e pode ser útil para identificar padrões ocultos nos dados.

## Análise de séries temporais

Essa técnica é usada para analisar dados que variam ao longo do tempo e pode ser útil para identificar tendências e prever eventos futuros.

## Análise de associação

Essa técnica é usada para identificar relações entre variáveis categóricas e pode ser útil para identificar regras de associação interessantes nos dados.

## Deep Learning (DL)

Quando se tem uma grande quantidade de dados, o uso de redes neurais como *CNN, RNN* e outras são uma ótima opção para realizar análise de dados e extrair insights importantes.

Essas são apenas algumas das técnicas avançadas de *EDA* que podem ser usadas. Cada técnica tem suas próprias vantagens e desvantagens e deve ser escolhida de acordo com o objetivo da análise e o conjunto de dados em questão.

## Outros exemplos

Outros exemplos mais avançados de análise de componentes principais *(PCA)* usando ***Python*** e a biblioteca *scikit-learn:*

### Códigos 2

import pandas as pd

from sklearn.decomposition import PCA

from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt

**# carregar o conjunto de dados**

df = pd.read\_csv("data.csv")

**# separar as variáveis numéricas do conjunto de dados**

X = df.drop(['target'], axis=1)

**# normalizar as variáveis numéricas**

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

**# inicializar o objeto PCA**

pca = PCA(n\_components=2)

**# ajustar o PCA aos dados normalizados**

pca.fit(X\_scaled)

**# transformar os dados para o espaço de componentes principais**

X\_pca = pca.transform(X\_scaled)

**# plotar os dados transformados**

plt.scatter(X\_pca[:, 0], X\_pca[:, 1])

plt.xlabel('Primeiro componente principal')

plt.ylabel('Segundo componente principal')

plt.show()

**# ver a variação explicada pelos componentes principais**

print(pca.explained\_variance\_ratio\_)

**#plotar o gráfico de variação explicada**

plt.plot(range(1, len(pca.explained\_variance\_ratio\_) + 1), pca.explained\_variance\_ratio\_)

plt.xlabel('Número de componentes principais')

plt.ylabel('Variação explicada')

plt.show()

## Explicações

**ChatGPT**

Neste exemplo, primeiro carregamos o conjunto de dados usando a biblioteca *pandas* e, em seguida, separamos as variáveis numéricas do conjunto de dados. Em seguida, usamos a classe *StandardScaler* do *scikit-learn* para normalizar as variáveis numéricas. Em seguida, inicializamos o objeto *PCA* com *n\_components = 2*, que significa que queremos manter apenas os dois primeiros componentes principais.

Em seguida, usamos o método *fit* do *PCA* para ajustar o *PCA* aos dados normalizados e usamos o método *transform* para transformar os dados para o espaço de componentes principais. Finalmente, usamos o método *scatter* do *matplotlib* para plotar os dados transformados no espaço de componentes principais e podemos ver a variação explicada pelos componentes principais.

Este exemplo mostra como usar a *PCA* para reduzir a dimensionalidade dos dados e visualizar os dados em um espaço de dimensão menor. Ele também mostra como verificar a variação explicada pelos componentes principais. A *PCA* é uma técnica poderosa para explorar e visualizar dados de alta dimensionalidade, como imagens, texto e outros tipos de dados.

Agora que compreendemos melhor por onde podemos começar, vale a pena pesquisar os sites com Datasets prontos, indicados abaixo pelo ChatGPT. É de suma importância que se pratique o que foi aprendido, para fixar bem as técnicas e para esclarecer os conceitos aplicados. Da minha parte eu indico alguns canais do Youtube e algumas páginas da Web, lembrando que não existe uma fórmula mágica para analisar dados, tudo depende do esforço pessoal, do tempo e da experiência adquirida com ele.

## Indicações

### Pelo ChatGPT:

***UCI Machine Learning Repository:*** esse repositório contém mais de 500 conjuntos de dados de diversas áreas, incluindo ciência da computação, engenharia, ciências sociais e saúde.

***Kaggle:*** esse site contém milhares de conjuntos de dados de diversas áreas, incluindo negócios, saúde, ciências e tecnologia. Ele também contém desafios de competição de aprendizado de máquina que você pode participar.

***Data.gov:*** esse site contém conjuntos de dados governamentais de vários países, incluindo os Estados Unidos. Ele também contém conjuntos de dados de várias áreas, incluindo saúde, meio ambiente, economia e outras.

**Data.world:** Esse site contém uma variedade de conjuntos de dados abertos e é uma ótima opção para encontrar conjuntos de dados para projetos de negócios e ciência de dados.

***GitHub:*** muitos desenvolvedores compartilham seus conjuntos de dados e projetos de aprendizado de máquina no GitHub. Você pode encontrar conjuntos de dados interessantes procurando por repositórios relacionados ao seu campo de interesse.

### Pelo autor:

*EDUARDO – CIÊNCIA DOS DADOS :*

<https://youtu.be/wCXeHzgjU7A> - Fala sobre o EDA

[(837) Eduardo | Ciência dos Dados - YouTube](https://www.youtube.com/@CienciadosDados) – Página no YouTube

[Ciência dos Dados (cienciadosdados.com)](https://www.cienciadosdados.com/) – Página na Web

*MEIGARON – SEJA UM DATA SCIENTIST;*

[(837) Seja Um Data Scientist - YouTube](https://www.youtube.com/@SejaUmDataScientist) – Página no YouTube

<https://youtu.be/Q4dtsGp_fbY> - Vídeos sobre a EDA

*ZURUBABEL*

<https://youtu.be/4SetLMXelUY> - Página no YouTube com EDA completa