import pandas as pd

**# Criação do dataset por ChatGPT**

automoveis = {'modelo': ['Ferrari 488', 'Lamborghini Huracan', 'Aston Martin Vantage', 'McLaren 720S', 'Porsche 911 Turbo', 'Audi R8', 'BMW i8', 'Mercedes AMG GT', 'Nissan GTR', 'Ford GT'],

'km/litro': [8, 9, 10, 11, 12, 13, 14, 15, 16, 17],

'cilindros': [8, 10, 12, 8, 6, 8, 3, 8, 6, 8],

'peso/kg': [1400, 1350, 1250, 1500, 1700, 1600, 1700, 1800, 2000, 1800],

'velocidade\_maxima': [320, 340, 350, 360, 370, 380, 390, 400, 410, 420],

'cavalos': [700, 750, 800, 850, 900, 950, 1000, 1050, 1100, 1150],

'tipo\_de\_transmissao': ['Manual', 'Automático', 'Automático', 'Manual', 'Automático', 'Manual', 'Automático', 'Automático', 'Manual', 'Automático'],

'valor': [300000, 280000, 260000, 240000, 220000, 200000, 180000, 160000, 140000, 120000],

'categoria': ['Importado', 'Importado', 'Importado', 'Importado', 'Importado', 'Importado', 'Importado', 'Importado', 'Importado', 'Nacional']}

df = pd.DataFrame(automoveis)

**# Exibição do dataset**

print(df)

Para fazer uma EDA (análise exploratória de dados) sobre esse dataset, você pode usar as funções e métodos do pandas, como:

df.describe(): para obter estatísticas básicas sobre as variáveis numéricas

df.info(): para obter informações gerais sobre o dataset

df.corr(): para calcular a correlação entre as variáveis numéricas

df.groupby(): para agrupar os dados por uma determinada coluna e calcular estatísticas

df.plot.scatter(): para plotar gráficos de dispersão entre as variáveis numéricas

df.plot.hist(): para plotar histogramas das variáveis numéricas

df.plot.bar(): para plotar gráficos de barras das variáveis categóricas

Para fazer uma PCA (análise de componentes principais) sobre esse dataset, você pode usar a biblioteca scikit-learn:

from sklearn.decomposition import PCA

**# Selecionando as colunas numéricas para a PCA**

colunas\_numericas = ['km/litro', 'cilindros', 'peso/kg', 'velocidade\_maxima', 'cavalos', 'valor'] df\_pca = df[colunas\_numericas]

**#Instanciando o objeto PCA**

pca = PCA(n\_components=2)

**#Ajustando o modelo PCA aos dados**

pca.fit(df\_pca)

**#Transformando os dados para as novas dimensões**

df\_pca\_transformado = pca.transform(df\_pca)

**#Plotando os dados transformados em um gráfico de dispersão**

import matplotlib.pyplot as plt plt.scatter(df\_pca\_transformado[:, 0], df\_pca\_transformado[:, 1])

plt.xlabel('Componente Principal 1')

plt.ylabel('Componente Principal 2')

plt.show()

Lembrando que a PCA é uma técnica não supervisionada, então é importante verificar se os componentes principais estão realmente capturando a variação dos dados e se fazem sentido para o problema em questão. E é importante também verificar se os componentes principais são linearmente independentes

Existem várias maneiras de verificar se os componentes principais são linearmente independentes. Uma delas é observar a matriz de componentes principais (também conhecida como matriz de rotação), que é a matriz que relaciona as variáveis originais com os componentes principais. Se as colunas dessa matriz são ortogonais (ou seja, se são vetores normais que são perpendiculares entre si), então os componentes principais são linearmente independentes.

Outra maneira é observar os autovalores e os autovetores da matriz de covariância dos dados, que são os valores e os vetores que maximizam o critério de variação explicada pelo PCA. Se os autovalores são distintos, então os componentes principais são linearmente independentes.

Além disso, podemos verificar se os componentes principais estão relacionado com as variaveis originais usando o coeficiente de correlação de Pearson, se todos os coeficientes são próximos de 0 (ou 1) então os componentes principais são linearmente independentes.

**# Verificando a matriz de componentes principais**

print(pca.components\_)

**# Verificando os autovalores e os autovetores**

print(pca.explained\_variance\_)

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

**# Verificando o coeficiente de correlação de Pearson**

df\_pca\_transformado\_df = pd.DataFrame(df\_pca\_transformado, columns=["PC1", "PC2"])

df\_pca\_transformado\_df.corr()

Lembrando que a independência linear dos componentes principais garante que cada componente principal capture uma dimensão diferente da variação dos dados, o que é importante para a interpretação dos componentes principais e para a seleção de quais componentes utilizar na análise.