

Álgebra Linear - PCA

Fernanda Rafaela

Novembro, 2024

1 Dataset

O dataset é sobre preferencia de destino de férias, Praia ou Montanha, tendo diversas informações sobre estilo de vida das pessoas.

Link: <https://www.kaggle.com/datasets/jahnavipaliwal/mountains-vs-beaches-preference?resource=download>

2 Desenvolvimento

Para realizar a normalização dos dados e aplicar o algoritmo de PCA, serão usadas as seguintes bibliotecas no Python:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
```

Figure 1: Bibliotecas utilizadas para análise

Primeiro, importamos o arquivo CSV, transformamos as colunas de texto em valor numérico e normalizamos o dataset, calculando a média de cada coluna e subtraindo do total:

```
df = pd.read_csv('mountains_vs_beaches_preferences.csv')
df['Preference'] = df.iloc[:, -1].map({0: 'Praia', 1: 'Montanha'})

text_columns = df.select_dtypes(include=['object']).columns
labels = df[text_columns[-1]] if len(text_columns) > 0 else None

label_encoder = LabelEncoder()
for col in text_columns:
    df[col] = label_encoder.fit_transform(df[col])

df_scaled = (df - df.mean()) / df.std()
```

Figure 2: Dataset

Em seguida, calculamos a matriz de covariância, transpondo-a, calculamos os autovetores e autovalores ordenando do maior para o menor, e calculamos a variancia explicada:

```
19 | cov = np.cov(df_scaled.T)
20 | eigvalues, eigvectors = np.linalg.eig(cov)
21 | order = np.argsort(eigvalues)[::-1]
22 | eigvalues = eigvalues[order]
23 | eigvectors = eigvectors[:, order]
24 |
25 | explained_variance = eigvalues / np.sum(eigvalues)
```

PROBLEMAS SAÍDA CONSOLE DE DEPURACÃO TERMINAL Filtrar (por exemplo, text, \exclude, \escape)

```
eigvalues[:2]
array([1.70390488, 1.02200723])
eigvectors[:2]
array([[ 7.13994508e-04,  3.21447202e-01,  2.87013785e-01,
        -2.63885629e-01, -3.81351335e-01,  3.15745989e-01,
         1.87921595e-01, -3.22901378e-02, -2.42213110e-01,
        -7.98591568e-02, -5.89661496e-01, -2.04976084e-01,
        -9.17061538e-02, -2.55541356e-03],
       [ 5.62301772e-04,  5.02878434e-01,  2.61268033e-01,
         1.93927987e-01,  6.37866395e-02, -3.54960625e-01,
        -2.88966975e-02, -3.13789638e-01,  1.73176208e-01,
         7.52134872e-02,  8.41147440e-02, -4.12996115e-01,
         4.45634133e-01, -1.11797634e-03]])
explained_variance
array([0.12170749, 0.07300052, 0.07273433, 0.07240778, 0.07213513,
       0.07181422, 0.07158674, 0.07124727, 0.07084369, 0.07060798,
       0.07034303, 0.07013068, 0.06988799, 0.02155315])
```

Figure 3: Algoritmo

Usando os 2 maiores autovalores calculamos o pca e colocamos a label que definimos durante o tratamento inicial do dataset:

```
k = 2
pca = np.matmul(df_scaled, eigvectors[:, :k])

if labels is not None:
    pca['Labels'] = labels
```

Figure 4: PCA Dataset

Por fim, o gráfico final usando os valores do PCA:

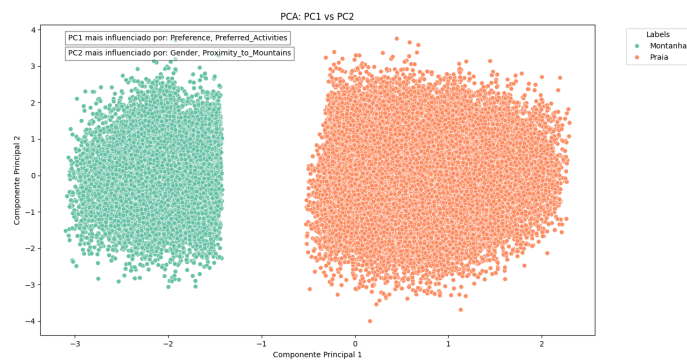


Figure 5: Gráfico PCA 2 dimensões

repo: <https://github.com/xfehrxx/pca>