CEPEDI – Desenvolvimento Tecnológico, Pesquisa, Desenvolvimento e Inovação

Discentes: Lilian Rocha dos Santos & Rogério Dos Santos Cerqueira.

Relatório de apresentação do projeto final da trilha de Ciência de Dados.

Feira de Santana, BA 2024.

SUMÁRIO

1. INTRODUÇÃO DO PROJETO

O projeto visa realizar uma análise exploratória e aplicar o algoritmo K-means para agrupar dados de atividades humanas, utilizando um conjunto de dados do repositório UCI Machine Learning. A análise foi realizada no Google Colab com bibliotecas como Scikit-learn, Pandas, NumPy e Matplotlib. O processo incluiu a leitura dos dados de treinamento e teste, análise descritiva com visualização das distribuições das variáveis por histogramas, análise estatística e detecção de outliers por boxplots. Além disso, foram avaliadas correlações entre variáveis e realizada a redução de dimensionalidade para facilitar a visualização e interpretação dos clusters. Essas etapas prepararam os dados para a aplicação do modelo K-means, que será abordado nas próximas fases do projeto.

PREPARAÇÃO DO DATASET

O objetivo deste trabalho é preparar e analisar um conjunto de dados de atividades humanas, aplicando o algoritmo K-means para agrupar as observações. O dataset, disponível no repositório UCI Machine Learning, contém medições de sensores de diversas atividades. A preparação envolveu a leitura e compreensão dos arquivos, análise exploratória detalhada e tratamento de variáveis e outliers. O processo foi realizado no Google Colab, utilizando as bibliotecas Python Scikit-learn, Pandas, NumPy e Matplotlib.

1. Definição e Preparação do Problema

1.1. Leitura dos Dados

A primeira etapa envolveu a leitura dos dados contidos nos arquivos fornecidos. O conjunto de dados foi composto por várias partes, incluindo características dos sensores (`X_train.txt`), rótulos de atividades (`y_train.txt`) e informações sobre os participantes (`subject_train.txt`), entre outros. Para importar esses dados, utilizamos a biblioteca Pandas, que oferece a função `read_csv()` para ler os arquivos e carregá-los em DataFrames.

O código utilizado para carregar os dados foi o seguinte:

```
import pandas as pd

X_train = pd.read_csv('train/X_train.txt', sep='\s+', header=None)
y_train = pd.read_csv('train/y_train.txt', sep='\s+', header=None)
subject_train = pd.read_csv('train/subject_train.txt', sep='\s+', header=None)
```

Figura 1. Importação dos datasets no Colab.

1.2. Entendimento do Dataset

Após a leitura dos arquivos, é importante entender a estrutura dos dados. A função `shape` foi utilizada para verificar o número de amostras e características presentes no dataset, enquanto a função `info()` forneceu informações adicionais, como o tipo de dados e a presença de valores nulos.

```
[4] print(X_train.shape)
    print(X_train.info())

(7352, 561)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7352 entries, 0 to 7351
Columns: 561 entries, 0 to 560
dtypes: float64(561)
memory usage: 31.5 MB
None
```

Figura 2. Verificação dos números de amostras e características.

Além disso, utilizamos o método `describe()` do Pandas para gerar um resumo estatístico das características, permitindo uma análise inicial da distribuição dos dados.

```
print(X_train.describe())
```

```
                0            1            2            3            4    \
count  7352.000000  7352.000000  7352.000000  7352.000000  7352.000000
mean      0.274488    -0.017695    -0.109141    -0.605438    -0.510938
std       0.070261     0.040811     0.056635     0.448734     0.502645
min      -1.000000    -1.000000    -1.000000    -1.000000    -0.999873
25%       0.262975    -0.024863    -0.120993    -0.992754    -0.978129
50%       0.277193    -0.017219    -0.108676    -0.946196    -0.851897
75%       0.288461    -0.010783    -0.097794    -0.242813    -0.034231
max       1.000000     1.000000     1.000000     1.000000     0.916238
```

Figura 3. Análise das amostras.

1.3. Análise Exploratória

Uma vez carregados os dados, iniciamos a análise exploratória para compreender melhor as distribuições das variáveis e identificar padrões ou anomalias.

1.3.1. Visualização das Distribuições das Características

Para avaliar as distribuições das variáveis de forma gráfica, utilizamos histogramas. A função `hist()` do Pandas foi aplicada para plotar os histogramas de cada coluna do dataset, ajudando a visualizar a distribuição e possíveis assimetrias.

```
import matplotlib.pyplot as plt

# Plota os histogramas para cada coluna de X_train
X_train.hist(bins=50, figsize=(20, 15))
plt.suptitle('Distribuição das Características', fontsize=16)
plt.show()
```

Figura 4. Plotando os histogramas.

A escolha entre exibir vários histogramas separados e que nos permitiu observar a distribuição de cada característica individualmente.Sendo útil quando as variações específicas de cada variável (por exemplo, se uma variável tem uma distribuição normal, se há assimetrias, outliers, etc.). Também fica mais fácil

identificar outliers em variáveis específicas, especialmente em conjuntos de dados com muitas variáveis de diferentes escalas.

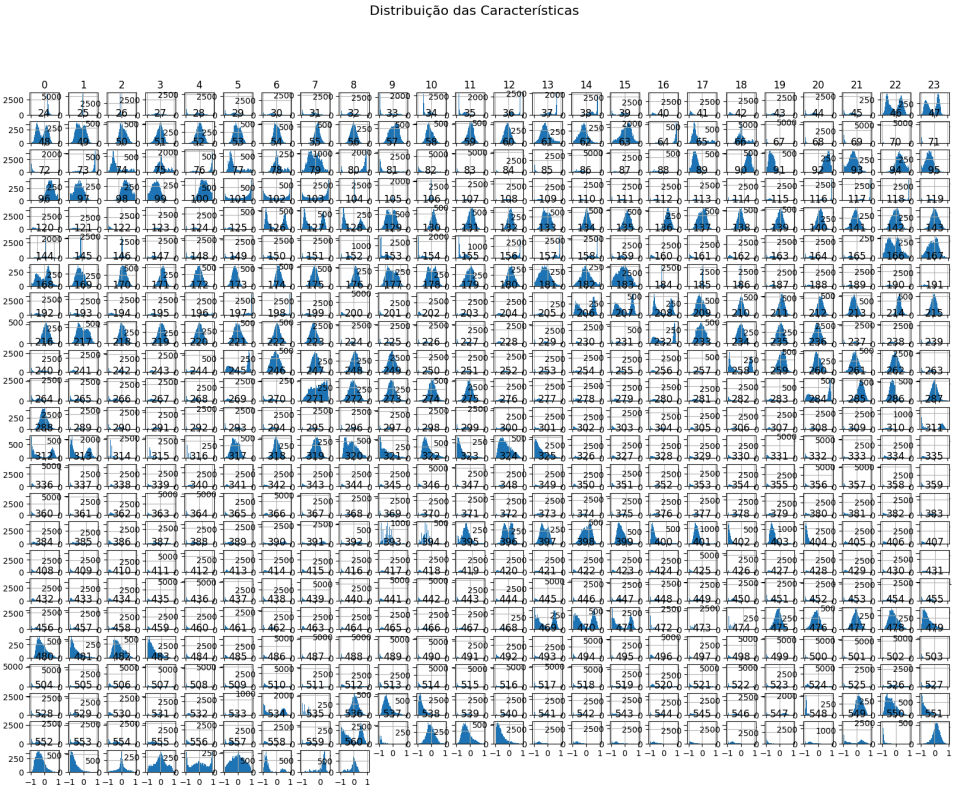

Figura 5. Histogramas separado.

1.3.2. Análise Descritiva das Características

Com a função `describe()`, obtemos um resumo estatístico das variáveis, que inclui informações como média, desvio padrão, valores mínimos, máximos e quartis. Esses dados são essenciais para entender a variabilidade e a dispersão das variáveis.

```
# Resumo estatístico de todas as características
print(X_train.describe())
```

```
                0            1            2            3            4    \
count  7352.000000  7352.000000  7352.000000  7352.000000  7352.000000
mean      0.274488    -0.017695    -0.109141    -0.605438    -0.510938
std       0.070261     0.040811     0.056635     0.448734     0.502645
min      -1.000000    -1.000000    -1.000000    -1.000000    -0.999873
25%       0.262975    -0.024863    -0.120993    -0.992754    -0.978129
50%       0.277193    -0.017219    -0.108676    -0.946196    -0.851897
75%       0.288461    -0.010783    -0.097794    -0.242813    -0.034231
max       1.000000     1.000000     1.000000     1.000000     0.916238

                5            6            7            8            9    ... \
count  7352.000000  7352.000000  7352.000000  7352.000000  7352.000000  ...
mean     -0.604754    -0.630512    -0.526907    -0.606150    -0.468604  ...
std       0.418687     0.424073     0.485942     0.414122     0.544547  ...
min      -1.000000    -1.000000    -1.000000    -1.000000    -1.000000  ...
25%      -0.980233    -0.993591    -0.978162    -0.980251    -0.936219  ...
50%      -0.859365    -0.950709    -0.857328    -0.857143    -0.881637  ...
75%      -0.262415    -0.292680    -0.066701    -0.265671    -0.017129  ...
max       1.000000     1.000000     0.967664     1.000000     1.000000  ...
```

Figura 6. Resumo estatístico.

2. Identificação de Outliers

A presença de outliers pode afetar a performance de modelos como o K-means, que é sensível a distâncias. Para identificar possíveis outliers, foi utilizado o boxplot, que permite visualizar os valores fora do intervalo interquartil (valores considerados outliers).

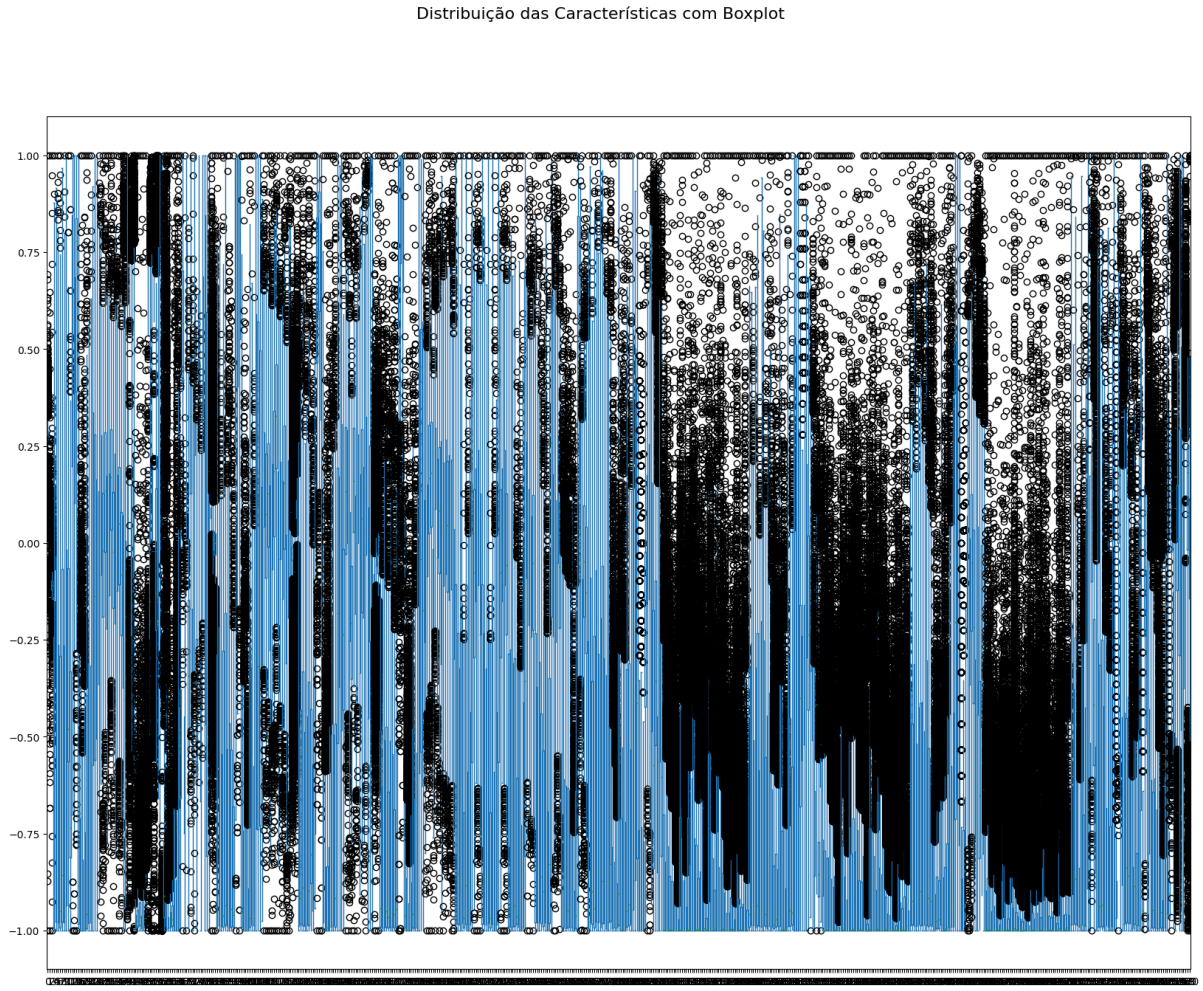

Figura 7. Outliers da base.

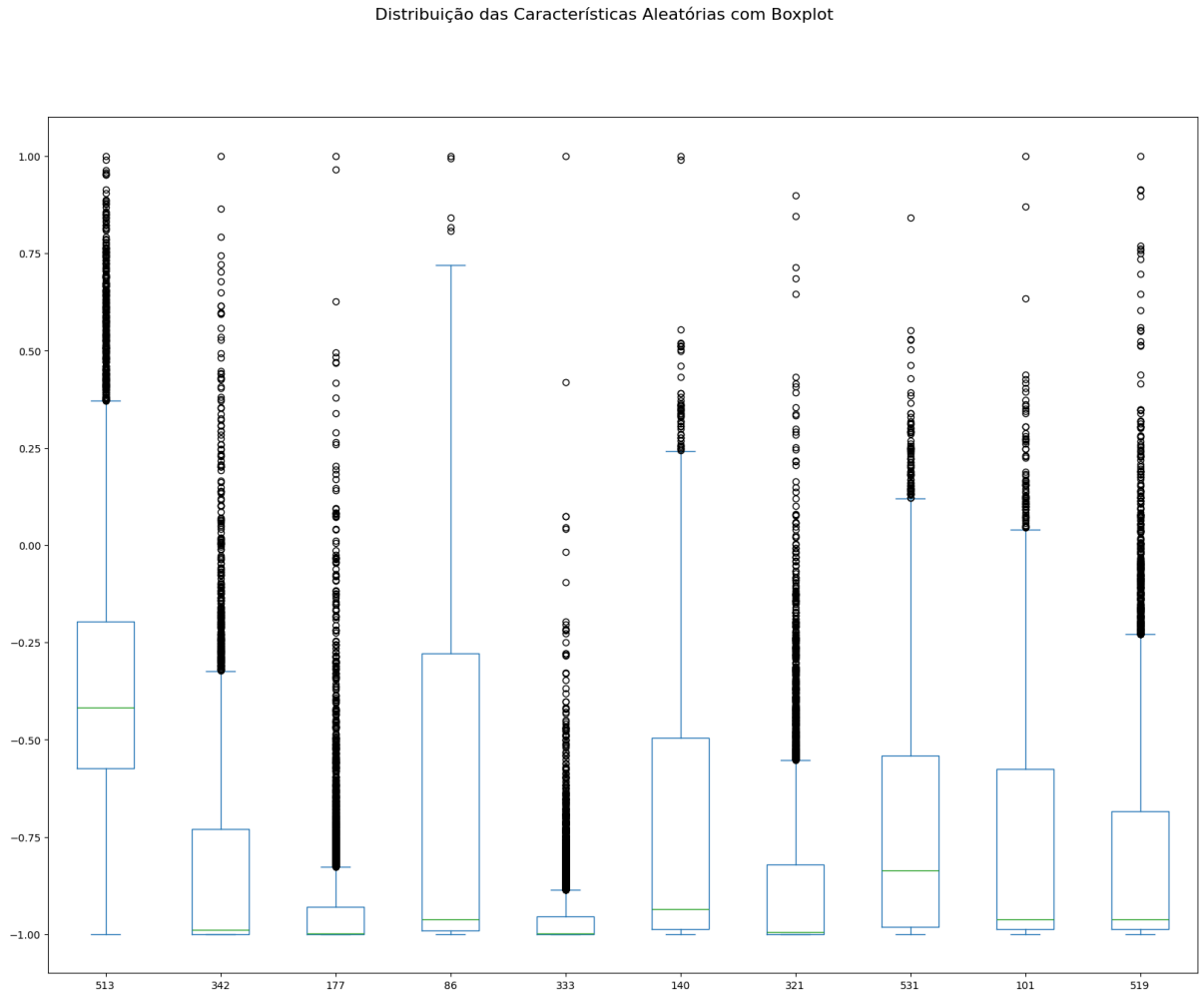

Figura 8. Outliers do subconjunto dos dados

O boxplot forneceu uma visão clara dos valores extremos para cada variável, ajudando a identificar as variáveis que apresentaram outliers. Em seguida, discutimos a necessidade de tratá-los antes de prosseguir com a modelagem, já que esses valores podem distorcer a definição dos clusters.

3. Preparação dos Dados para Modelagem.

Com base nas análises realizadas nas etapas anteriores, tomamos decisões importantes sobre como tratar os dados para otimizar a aplicação do algoritmo K-means.

3.1. Normalização dos Dados

Uma das decisões mais importantes foi a normalização dos dados. Como as variáveis possuem escalas diferentes (algumas com valores muito pequenos e

outras com valores muito grandes), foi decidido aplicar técnicas de normalização ou padronização para garantir que todas as variáveis contribuem de forma equilibrada para a definição dos clusters. Embora a normalização não tenha sido implementada nesta parte, ela será um passo crucial antes de aplicar o K-means.

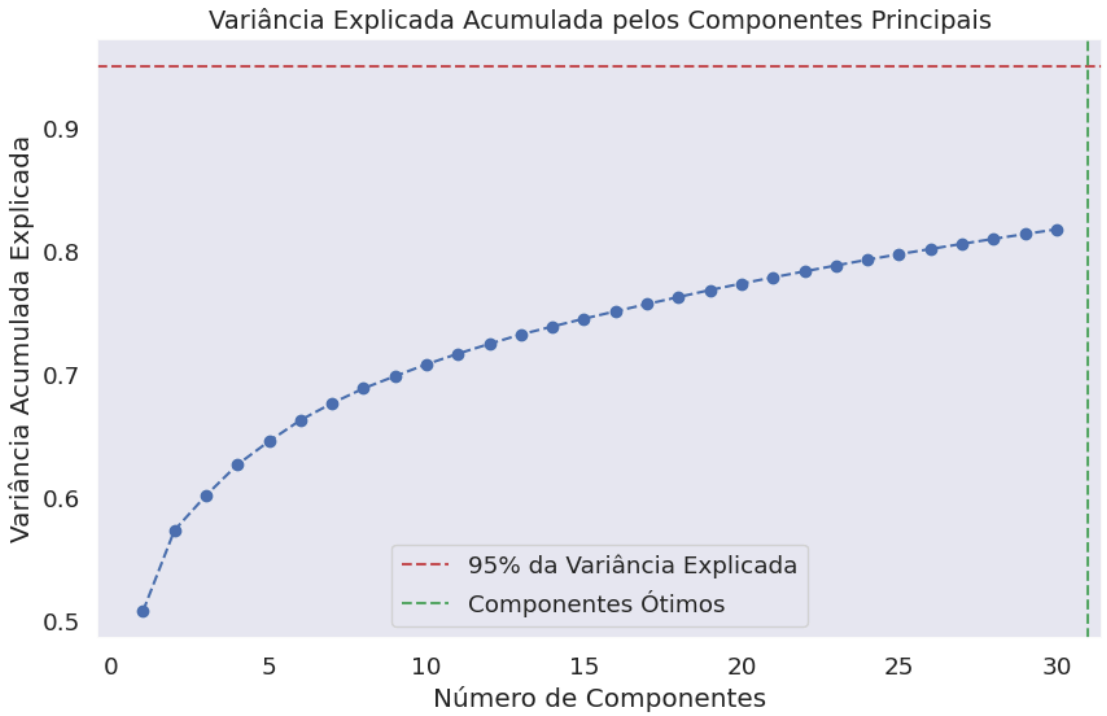

Figura 9.Normalização dos dados.

A normalização foi realizada com o StandardScaler, ajustando as variáveis para média zero e desvio padrão igual a 1, garantindo escalas uniformes e evitando distorções nos modelos. A alta dimensionalidade foi tratada com a redução de dimensionalidade, como o PCA..

3.2 Análise das Características em Função das Atividades

Esse passo é fundamental, pois cria uma associação explícita entre as características e suas classes (atividades), o que é necessário para fazer análises mais profundas, como a visualização das diferentes distribuições de atividades.

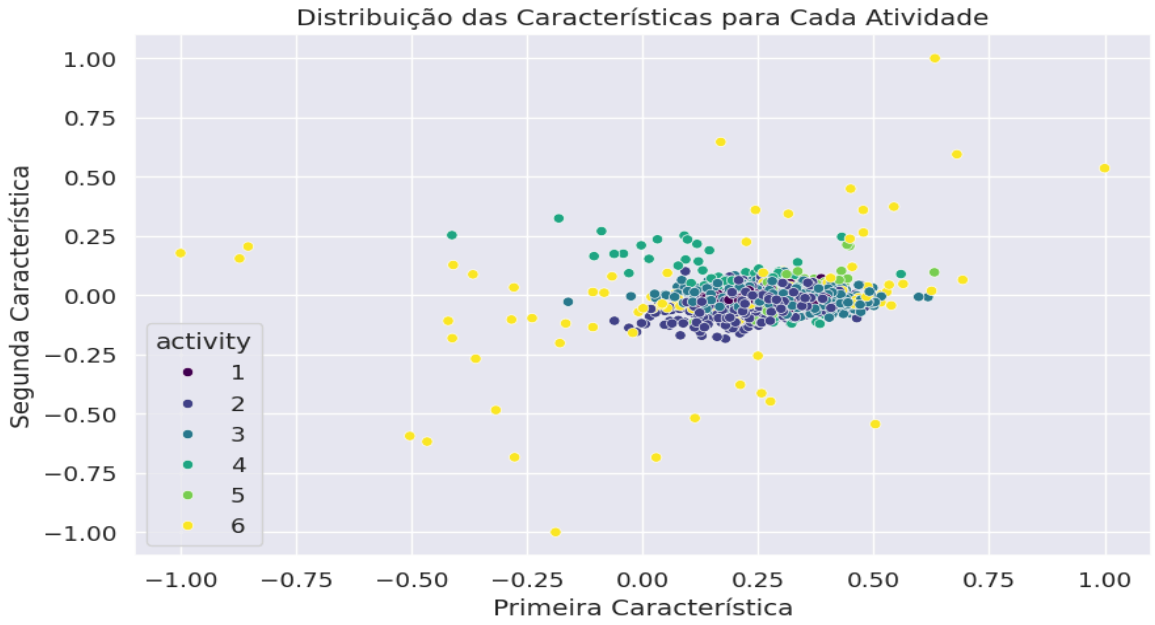

Figura 10. Distribuição das características para cada atividade.

1. Foi gerado um gráfico de dispersão com Seaborn para visualizar a distribuição das atividades em relação às duas primeiras características de X_train. A análise mostrou três possibilidades: atividades bem separadas indicam características relevantes para classificação, atividades sobrepostas sugerem características insuficientes, e a presença de outliers pode exigir tratamento.

3.5. Análise Temporal (Opcional).

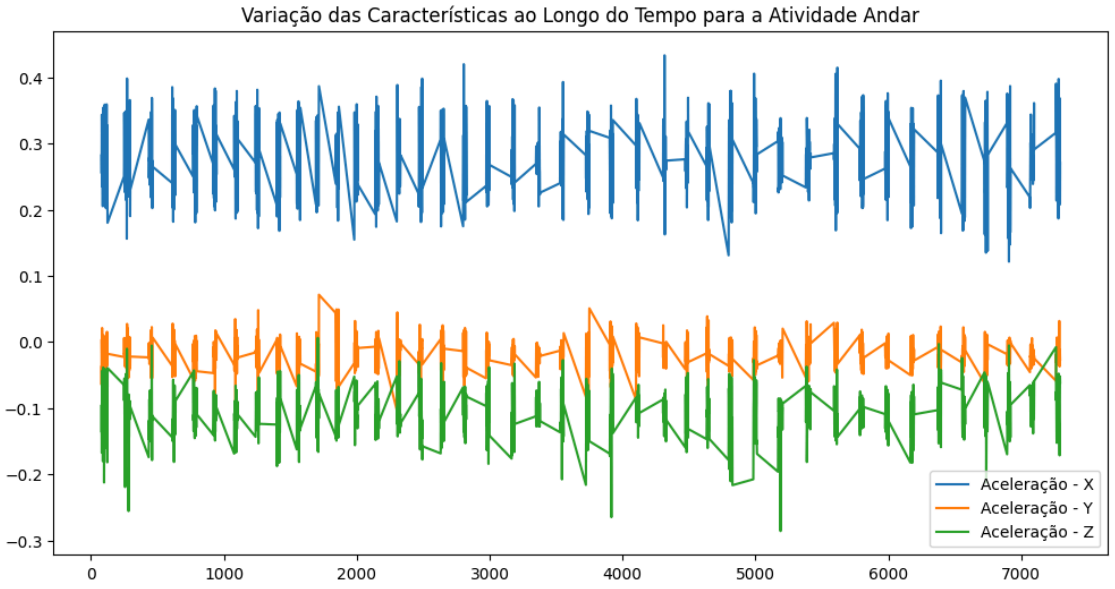

Figura 11. Análise Temporal das características.

Padrões Regulares: Como as linhas do gráfico mostram padrões consistentes e regulares (como uma onda sinusoidal), isso indica que a atividade tem movimentos repetitivos bem definidos.

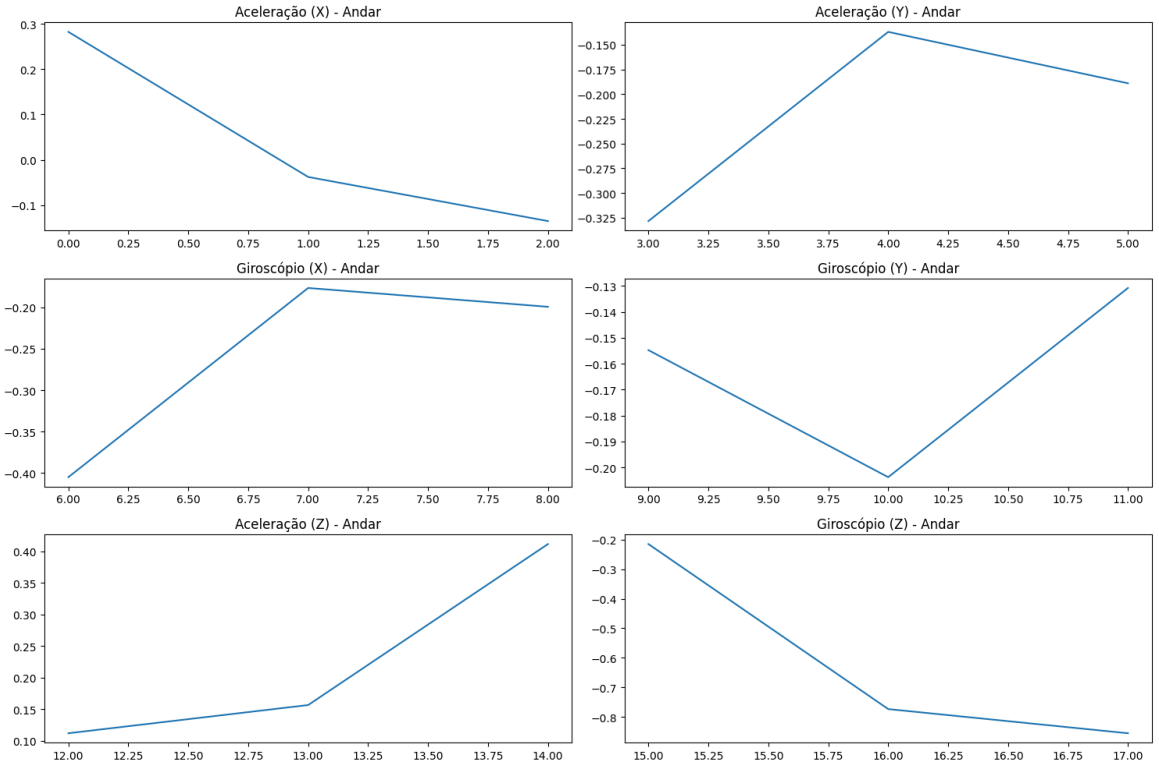

Figura 12. ***Filtragem de uma atividade específica (por exemplo, "andar")***

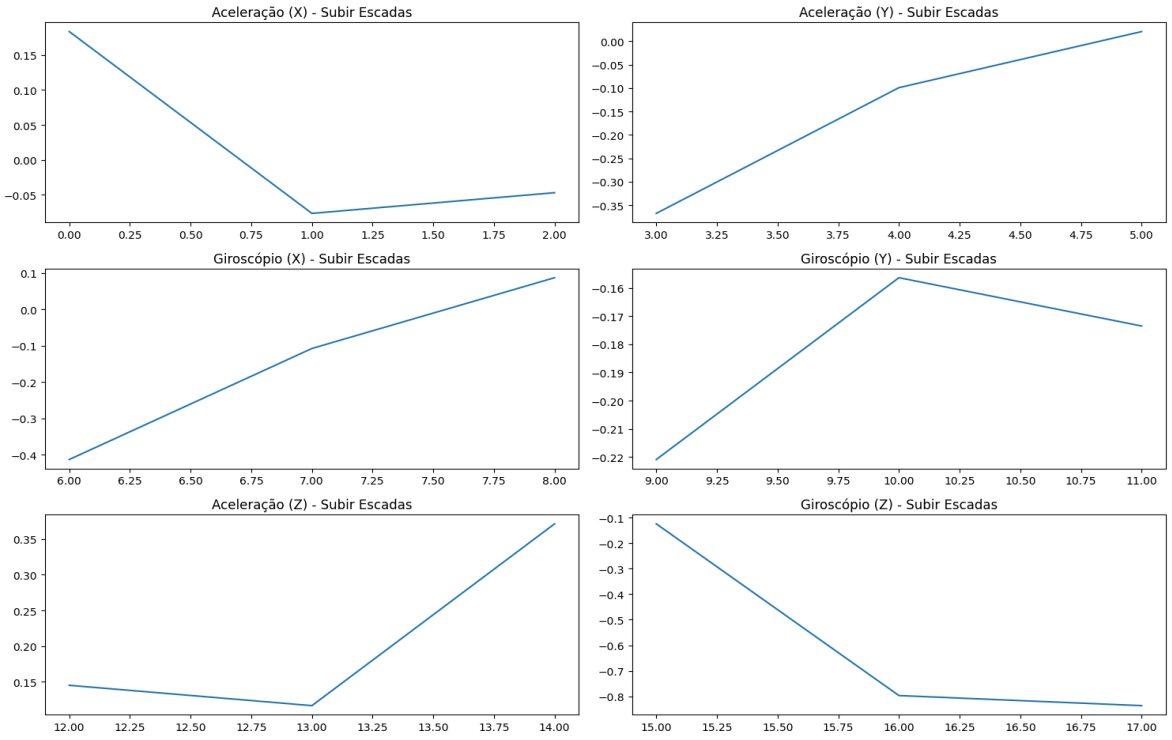

Figura 13.

3.6 Distribuição das atividades

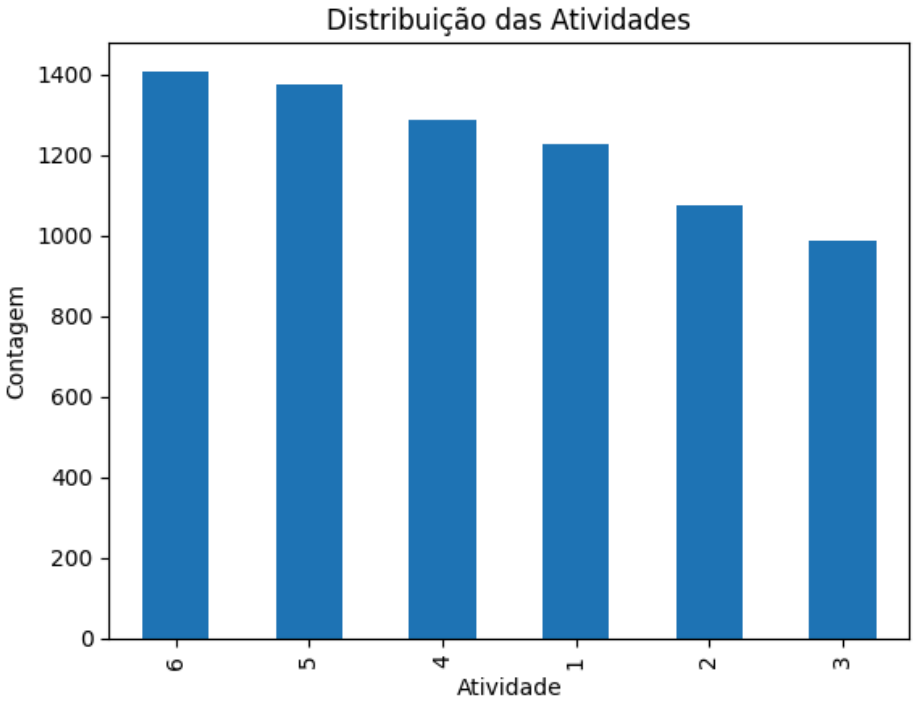

Figura 14. Distribuição das Atividades.

3.7 Visualização das atividades em relação a variáveis

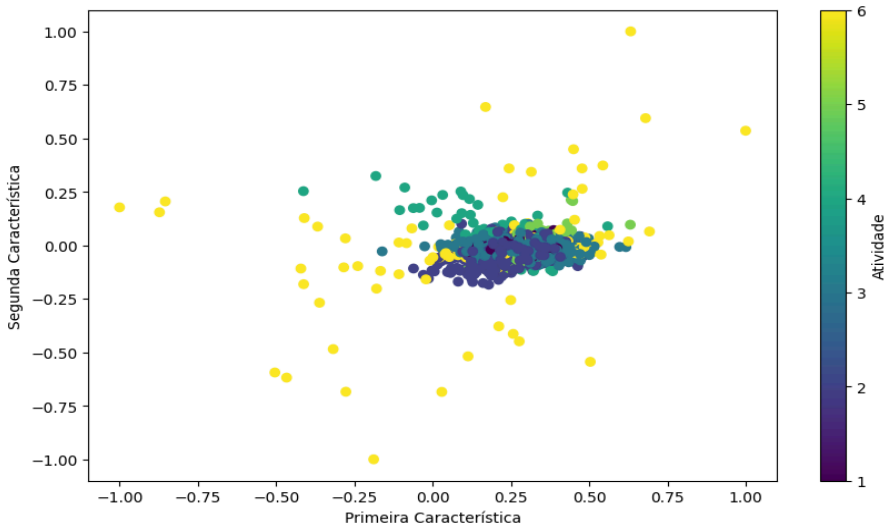

Figura 15. Visualização das atividades em relação a variáveis.

O objetivo desta etapa é criar uma visualização que relacione as duas primeiras características do conjunto de dados com as diferentes atividades (classes) representadas no rótulo `y_train`*.*

Como os pontos associados a diferentes atividades (cores distintas) formam agrupamentos bem definidos, isso indica que as duas características têm um bom poder discriminativo para diferenciar as atividades.

4. Conclusão da Preparação do Dataset

A preparação do dataset envolveu diversas etapas essenciais para garantir que os dados estivessem prontos para a aplicação do modelo K-means. A leitura e compreensão dos dados foram realizadas com o auxílio do Pandas, enquanto a análise exploratória foi fundamental para entender as distribuições das variáveis, detectar outliers e avaliar a necessidade de normalização.

Os passos descritos neste relatório são cruciais para o sucesso da modelagem, pois permitem ajustar os dados de forma que o algoritmo de agrupamento seja capaz de gerar resultados significativos e precisos.

Agora que os dados estão preparados, as próximas etapas do projeto incluem a implementação do modelo K-means e a análise dos resultados obtidos.

Este relatório foi desenvolvido para documentar o processo completo de preparação do dataset, sendo uma base sólida para a próxima fase do projeto, que envolve a modelagem e avaliação dos resultados.

IMPLEMENTAÇÃO DO K-MEANS

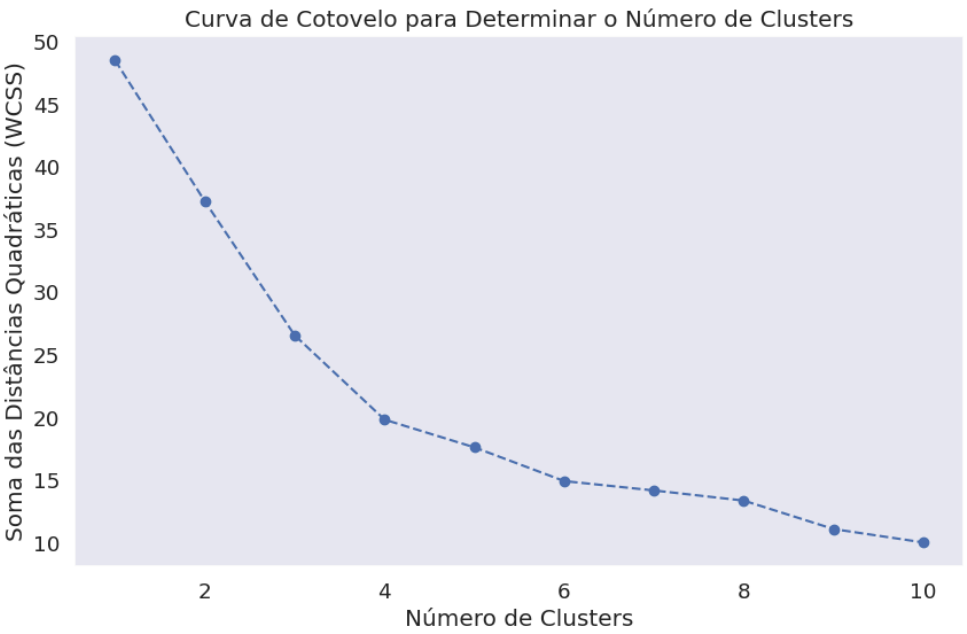

Figura 16. Queda em WCSS.

A curva mostrar uma grande queda em WCSS até um determinado número de clusters e, depois, uma queda mais suave, o número de clusters ideal será provavelmente o ponto onde a queda se estabiliza.

A curva continuar caindo acentuadamente até o final, talvez seja necessário testar um valor de K maior para ver se o comportamento se estabiliza.

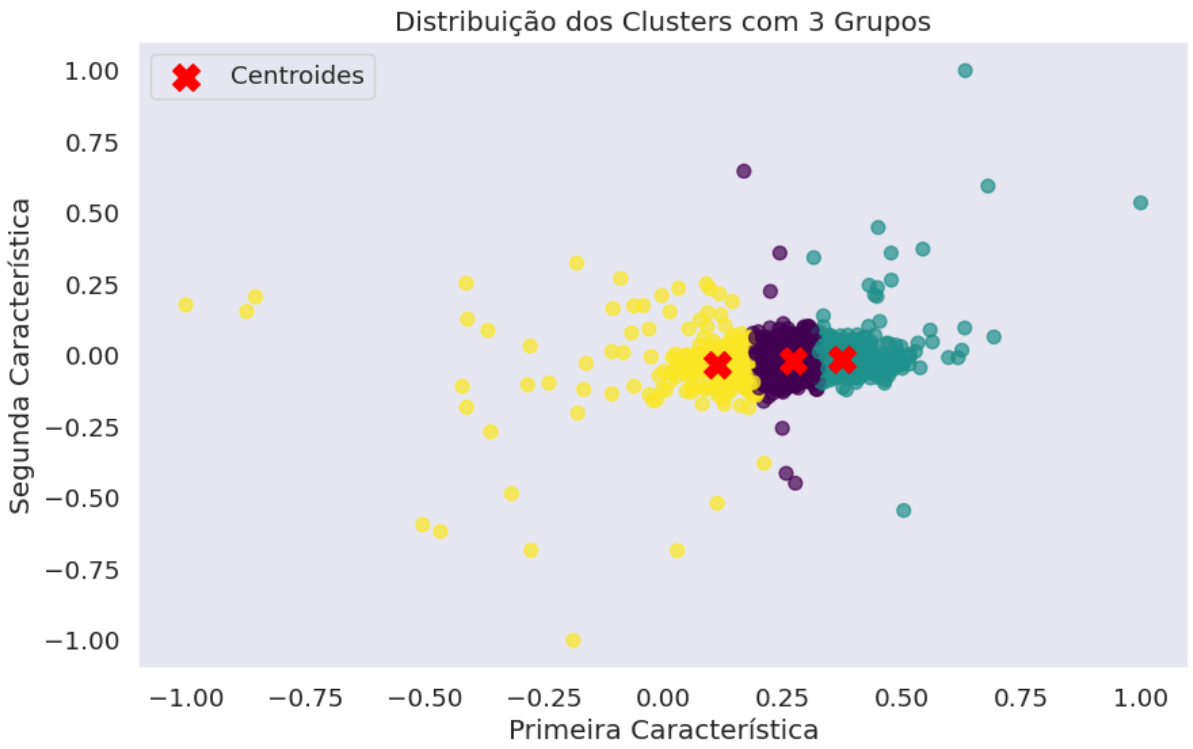

Figura 17. Centroides.

CONCLUSÃO

Uma pontuação Silhouette de 0.60 é um bom indicativo de que os clusters estão bem formados. Em geral:

1. Valores próximos de 1 indicam que os clusters estão bem separados e bem coesos;
2. Valores próximos de 0 indicam que os clusters estão sobrepostos ou não bem definidos;
3. Valores negativos sugerem que os dados podem ter sido agrupados de forma errada, com pontos mais próximos de clusters diferentes do que de seu próprio grupo.

REFERÊNCIAS

1. Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
    - Livro clássico que aborda algoritmos de aprendizado de máquina, incluindo K-means, e a importância da normalização e análise de dados.

2. UCI Machine Learning Repository. (n.d.). *Human Activity Recognition Using Smartphones Data Set*. Disponível em: https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones
    - Fonte oficial do dataset utilizado no projeto.
3. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., et al. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.
    - Artigo técnico sobre a biblioteca Scikit-learn, utilizada para implementar o K-means e normalizar os dados.
4. McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. Proceedings of the 9th Python in Science Conference.
    - Artigo técnico sobre a biblioteca Pandas, utilizada para leitura e análise exploratória dos dados.
5. Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., et al. (2020). *Array programming with NumPy*. Nature, 585(7825), 357–362.
    - Artigo técnico sobre a biblioteca NumPy, utilizada para manipulação de dados e operações matemáticas.
6. Hunter, J. D. (2007). *Matplotlib: A 2D Graphics Environment*. Computing in Science & Engineering, 9(3), 90–95.
    - Artigo técnico sobre a biblioteca Matplotlib, utilizada para gerar gráficos no projeto.
7. Tan, P.-N., Steinbach, M., Karpatne, A., & Kumar, V. (2019). *Introduction to Data Mining* (2ª ed.). Pearson.
    - Livro abrangente sobre mineração de dados, incluindo a análise exploratória e o algoritmo K-means.
8. Seaborn Documentation. (n.d.). *Seaborn: Statistical Data Visualization*. Disponível em: https://seaborn.pydata.org/
    - Documentação oficial da biblioteca Seaborn, utilizada para criar visualizações estatísticas no projeto.
9. Abadi, M., Barham, P., Chen, J., et al. (2016). *TensorFlow: A system for large-scale machine learning*. OSDI.

    - Embora TensorFlow não tenha sido usado diretamente, pode ser citado caso futuras implementações explorem redes neurais baseadas em clustering.
10. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2ª ed.). Springer.
- Livro técnico que detalha técnicas de aprendizado de máquina, incluindo algoritmos de clustering como o K-means.